PURDUE UNIVERSITY

NCN

NEMO5

User Manual

September 2, 2015

Author:
Sebastian Steiger

Lead:
Prof Gerhard Klimeck

NEMO5 core developers:
Michael Povolotskyi
Tillmann Kubis
Hong-Hyun Park
Sebastian Steiger

Network for Computational Nanotechnology
Purdue University
West Lafayette IN, USA

Contents

1 Introduction **7**
- 1.1 The purpose of NEMO5 . . . 7
- 1.2 Key elements that distinguish NEMO5 from other simulators . . . 8
- 1.3 Short history of NEMO5 . . . 8
- 1.4 To whom this manual is addressed . . . 9
- 1.5 License . . . 9
- 1.6 Additional sources of information . . . 10
- 1.7 nanoHUB tools employing NEMO5 . . . 10

2 Installation **11**
- 2.1 Obtaining pre-compiled versions . . . 11
- 2.2 Installation from source . . . 12
- 2.3 Directories . . . 12

3 Input and Output **13**
- 3.1 Input . . . 13
  - 3.1.1 The material file . . . 13
  - 3.1.2 Specifying material parameters in the input deck . . . 14
- 3.2 Output . . . 14
  - 3.2.1 Visualization tools . . . 15

4 The Input Deck **16**
- 4.1 Example . . . 16
- 4.2 `Structure` section . . . 20
  - 4.2.1 `Material` sections . . . 20
  - 4.2.2 `Geometry` section . . . 23
    - 4.2.2.1 `Region` sections . . . 23

4.2.2.2 `Boundary_region` sections . . . 23
4.2.3 `Domain` sections . . . 23
4.2.4 `Partitioning` section . . . 26
4.3 `Solvers` section . . . 27
4.4 `Global` section . . . 27

5 Solvers and Options **29**
5.1 Options common to all solvers . . . 29
5.2 `AdaptiveGrid` . . . 32
5.3 `ApplyModeSpace` . . . 37
5.4 `BackwardNonlocalRGFModule` . . . 39
5.5 `BackwardRGFModule` . . . 40
5.6 `BellEqNeq` . . . 43
5.7 `BellRGF` . . . 46
5.8 `BellRGF2` . . . 49
5.9 `BellRGFModeSpace` . . . 52
5.10 `BellRGFModeSpace2` . . . 55
5.11 `BellRGFTrans` . . . 58
5.12 `BlockDiagonalRGF` . . . 61
5.13 `BlockSchroedingerModule` . . . 65
5.14 `BosonHGLesserGreensolver` . . . 66
5.15 `BosonInverseGreensolver` . . . 76
5.16 `BosonLesserGreensolver` . . . 85
5.17 `BosonRetardedContactSelfenergy` . . . 86
5.18 `BosonRetardedGreensolver` . . . 99
5.19 `BosonSpectralGreensolver` . . . 108
5.20 `Brillouin` . . . 118
5.21 `Cluster` . . . 119
5.22 `Cluster_fitting` . . . 121
5.23 `ComplexContourEnergyGrid` . . . 122
5.24 `ConcatenateDomains` . . . 124
5.25 `CorrelationGreenModule` . . . 125
5.26 `DOFmapMultisizedTest` . . . 127
5.27 `DOFmapping` . . . 128
5.28 `DensityWrapper` . . . 130
5.29 `DomainLeadOrganizer` . . . 132
5.30 `DomainRandomizer` . . . 133
5.31 `DomainRedistribution` . . . 134
5.32 `DriftDiffusion` . . . 135
5.33 `Dummy1` . . . 137
5.34 `Dummy2` . . . 138

5.35 EHT3_unit_spin_test . . . 139
5.36 EHT3_unit_test . . . 140
5.37 EMSchrodingerFEM . . . 142
5.38 EqNeqRGFModule . . . 142
5.39 ExtHuckel3 . . . 144
5.40 Extrapolator . . . 160
5.41 FermionHGLesserGreensolver . . . 161
5.42 FermionInverseGreensolver . . . 171
5.43 FermionLesserContactSelfenergy . . . 180
5.44 FermionLesserGreensolver . . . 192
5.45 FermionLesserScatteringSelfenergy . . . 202
5.46 FermionRetardedConstantScatteringSelfenergy . . . 214
5.47 FermionRetardedContactSelfenergy . . . 226
5.48 FermionRetardedGreensolver . . . 238
5.49 FermionRetardedScatteringSelfenergy . . . 247
5.50 FermionSpectralGreensolver . . . 259
5.51 ForwardNonlocalRGFModule . . . 269
5.52 ForwardRGFModule . . . 270
5.53 ForwardRGFSolver . . . 272
5.54 ForwardRGFWrapperModule . . . 282
5.55 GeneralLeadGreenModule . . . 283
5.56 GeometryVIS . . . 286
5.57 GreenModule . . . 288
5.58 Green_Propagation . . . 290
5.59 HamiltonianUTBgenerator . . . 299
5.60 Huckel_RGF . . . 301
5.61 IV . . . 303
5.62 Interpolator . . . 306
5.63 KeatingStrain . . . 307
5.64 LRAModule . . . 311
5.65 LRARGFModule . . . 311
5.66 Lanczos . . . 313
5.67 LocalBandstructureModule . . . 315
5.68 MCBoltzmann . . . 317
5.69 MCbulk . . . 327
5.70 MatrixElements . . . 330
5.71 MatrixTest1 . . . 333
5.72 Meta55 . . . 334
5.73 Meta5Schottky . . . 346
5.74 Meta5_schottky . . . 360
5.75 MetaBellQTBM . . . 374

5.76 MetaBellQTBM2 . . . 383
5.77 MetaDummy . . . 393
5.78 MetaLinearDropQTBM . . . 402
5.79 MetaOmenQTBM . . . 412
5.80 MetaOmenTransport . . . 421
5.81 MetaOxide . . . 430
5.82 MetaPoissonQTBM2 . . . 445
5.83 MetaPoissonQTBM3 . . . 459
5.84 MetaPoissonQTBM4 . . . 473
5.85 MetaPoissonQTBM4UTB . . . 487
5.86 MetaPoissonQTBM5 . . . 501
5.87 MetaPoissonQTBM5_2top_gates . . . 516
5.88 MetaPoissonQTBM5_EC . . . 530
5.89 MetaRGF . . . 540
5.90 MetaSolver . . . 550
5.91 MetafullRGF . . . 551
5.92 Metatemp5 . . . 561
5.93 ModuleDummy . . . 571
5.94 ModuleDummyConstruct . . . 573
5.95 ModuleDummyNested . . . 574
5.96 MultiEqNeq . . . 575
5.97 NEGF . . . 578
5.98 NEGFG . . . 580
5.99 NemoMatrixDummy . . . 581
5.100NonlinearPoisson . . . 583
5.101NonlinearPoissonFEM . . . 591
5.102NonlocalRGFModule . . . 600
5.103NonorthoRGFModule . . . 602
5.104PartitionRetardedSelfModule . . . 606
5.105PartitionRetardedSurfaceGreenModule . . . 608
5.106Phonons . . . 610
5.107Poisson . . . 613
5.108PoolObjectTest1 . . . 616
5.109Propagation . . . 618
5.110PythonSolver . . . 627
5.111QTBMModule . . . 628
5.112QTBMTaskManagerModule . . . 630
5.113QTBM_Propagation . . . 631
5.114RGFModule . . . 644
5.115Ramper . . . 649
5.116ReadInCharge . . . 652

5.117ReadInCharge3D 653
5.118ReadInPotential 655
5.119ReadInPotentialWithAtoms 656
5.120ReadinHamiltonianModule 656
5.121Reinit 657
5.122Repartition 659
5.123ResonanceFinder 662
5.124ResonanceFinderModule 665
5.125ResonanceMesh 666
5.126RetardedSelfModule 668
5.127Schroedinger 670
5.128SchroedingerDensity 687
5.129Schroedinger_huckel 689
5.130Schroedinger_intra_atomic 705
5.131Self_Propagation 720
5.132Selfconsistent 732
5.133SelfconsistentBornModule 733
5.134Semiclassical 739
5.135SemiclassicalFEM 743
5.136SeparateDomains 743
5.137SimpleExample 744
5.138SimulationModules 745
5.139Step 748
5.140StrainDomain 749
5.141StrainVFF 751
5.142Structure 754
5.143TDSManager 756
5.144TDSchroedinger 758
5.145TaskOffload 761
5.146TopOfBarrier 763
5.147Transformation 767
5.148TranslateSelfEnergy 771
5.149UnderBarrier 772
5.150Unfold 775
5.151VFFStrain 776
5.152ValleyGrid 780
5.153WF 781

6 Material Parameter Names **783**
6.1 `Structure`, `Brillouin` 783
6.2 `VFFStrain`, `Phonons` 783
6.3 `Poisson`, `NonlinearPoisson` 783
6.4 `Schroedinger`, `WF`, `NEGF` 785
6.5 `Semiclassical` 786

Bibliography **787**

Chapter 1

Introduction

1.1 The purpose of NEMO5

NEMO5 [1] is intended to be an all-purpose nanoelectronics simulation toolbox. Its modular design permits developers to add and extend physical models using clean APIs. Currently NEMO5 is able to do the following:

Construct atomistic representations of nanostructure using various elements, crystal structures, crystal directions.

Calculate the Brillouin zone of the corresponding reciprocal lattice.

Calculate the relaxed atomic positions using various flavors of the valence force field (VFF) model.

Calculate phonon spectra using VFF models.

Solve the atomistic tight-binding Schrödinger equation.

Solve the Poisson equation and coupled Schrödinger -Poisson problems.

Use the full-band eigenstates as input to a Top-Of-Barrier model for ballistic transport in nanotransistors.

Calculate transport properties such as current, transmission spectra, through the open-boundary wave function formalism.

Calculate transport properties such as current, transmission spectra, through the NEGF formalism.

There are also first attempts on phonon transport.

1.2 Key elements that distinguish NEMO5 from other simulators

Advanced atomistic nanostructures of non-standard (i.e., non-zincblende) structures can be constructed using primitive unit cells. Currently implemented are simple-cubic, diamond, zincblende, wurtzite, and rhombohedral lattices.

NEMO5 is highly MPI-parallelized, making it suitable for large-scale computations on supercomputers.

The code is written in a modular way with clear and well-documented interfaces. This permits extensive testing of individual components, allows of code extensions and usage of selected modules in other codes.

The power of advanced numerical packages is leveraged: libmesh (for the FEM discretization of the Poisson equation), PETSc (for matrices, linear and nonlinear equation systems), SLEPc (for eigenvalue problems), qhull (for the Voronoi cell), boost (for parsing the input deck and a few filesystem operations), VTK (as an output format), Silo (as a parallel output format) and others.

1.3 Short history of NEMO5

NEMO5 was started by the 4 post-docs (in alphabetical order) Tillmann Kubis, Hong-Hyun Park, Michael Povolotskyi and Sebastian Steiger in the group of Prof Gerhard Klimeck at Purdue University. It draws upon extensive prior experience of the group in nanoelectronics simulation:

NEMO-1D [2] was historically the first simulation tool to correctly predict I-V curves of resonant tunneling diodes (RTDs), which is a prototypical nanoelectronic device. It was a large and successful project being co-led by Gerhard Klimeck in the mid-90s and featured tight-binding Schoedinger-Poisson and NEGF functionality.

NEMO-3D [3,4] focused on atomistic Schroedinger-Poisson simulations of large systems (such as quantum dots consisting out of millions of atoms). It was developed in the early 2000s under Gerhard Klimeck's supervision at JPL. It includes calculations of strain, polarization and optical matrix elements and is still being used. Its main achievement was the ability to carry out massively parallel calculations employing 1000's of CPUs at the same time.

OMEN [5] is the work of Mathieu Luisier and has been developed from 2005-2010 at ETH Zurich and Purdue. It is able to accurately compute transport in nanowires, HEMTs, MOSFETs, TFETs and other devices in a massivey parallel way (near-perfect scaling up to 200'000 cores was demonstrated).

NEMO-3D [?] has been extended from 2008-2010 by Sunhee Lee and Hoon Ryu to have a 3D space parallelization, Coulomb and exchange matrix element calculations, and more. Its main target has been the massively parallel computation of properties of phosphorus impurities in Si.

Given these four prior incarnations of NEMO (NanoElectronics MOdeling), the authors decided to call their tool NEMO5.
In addition to the knowledge of the group, the experience of the authors from their prior jobs was pivotal in the development of NEMO5. Each was involved in large-scale projects, amongst which are NextNano, TiberCAD, tdkp/AQUA, Simsn, and other projects.

1.4 To whom this manual is addressed

This manual might be beneficial to the following classes of people:

Researchers in the nanoelectronics area who are in need of a simulation and want to know if NEMO5 can help them.

NEMO5 Users who want to understand the full functionality of NEMO5 and how to set up input decks and correct simulation parameters.

NEMO5 Developers who are unclear about the meaning of certain classes or keywords. However, most of the documentation relevant to developers is provided through *doxygen* in the source code.

What you will find in this manual, and what you don't
This manual should provide the user with an accurate and complete description on the functionality of NEMO5 and on how to run simulations, i.e. setting up an input deck and obtaining results. However, it cannot explain all the output generated by the simulator, or all the possible error messages. It can also not explain the entire physics behind the simulation: if the reader does not know what the Schrödinger equation is of what the local density of states in NEGF means, then he or she has to consult other sources of information.

1.5 License

NEMO5 is distributed under the NEMO5 academic license. This means that it is free for academic use. Commercial use is not permitted under this license and requires a different license.

1.6 Additional sources of information

In case the reader is in need of additional information, he may think about doing one of the following:

Check the developer documentation that is provided with the source code in the `doc/` folder.

Supplementary material (such as Powerpoint slides) is available in the `supplements/` folder.

A scientific paper has been published [1].

Send Gerhard Klimeck, one of his group members or one of the developers an e-mail.

1.7 nanoHUB tools employing NEMO5

NEMO5 is the engine of the following tools on nanoHUB.org:

Quantum Dot Lab:

`https://nanohub.org/tools/qdot`

1D Heterostructure Tool:

`https://nanohub.org/tools/1dhetero`

Crystal Viewer:

`https://nanohub.org/tools/crystal_viewer`

Brillouin Zone Viewer:

`https://nanohub.org/tools/brillouin`

These tool mainly consist of Rappture GUIs and translating wrappers that create input decks for certain use cases out of user-supplied options and retrieve simulation output.
Incorporation of the following tools is in progress:

Band Structure Lab:

`https://nanohub.org/tools/bandstrlab`

Resonant Tunneling Diode Simulation with NEGF:

`https://nanohub.org/tools/rtdnegf`

Chapter 2

Installation

2.1 Obtaining pre-compiled versions

Ready-to-use versions can be downloaded from nanoHUB:

`https://nanohub.org/groups/nemo5distribution`

Currently only Linux operating systems are supported. The following platforms are available:

Kubuntu 10.04 64bit using the GCC compiler.

Red Hat (RHEL4) 64bit using GCC, as set up on `coates.rcac.purdue.edu`.

Debian 2.6.24 (Lenny) 64bit, as set up on `nanohub.org`.

Installation of such a distribution is straightforward:

1. Download the archive (`.tar.gz`) and save in an appropriate location.
2. The command `tar zxvf <file>.tar.gz` will extract NEMO 5.
3. A quick check whether it works can be done by running NEMO 5 on the supplied `.in`-file.

The distribution may require some packages which are not installed by default, such as *boost*, the GCC compiler, and so on. However, all these packages are free and easy to obtain.

2.2 Installation from source

NEMO 5 has a fairly automated build system, however, problems can arise from the large number of required 3rd-party libraries which need to be present for the configuration. The user may follow these steps:

1. Download the source code archive from
   `https://nanohub.org/groups/nemo5distribution`
   and save in an appropriate location.
2. Extract the archive.
3. Follow the instructions in the provided `README` file.

2.3 Directories

NEMO 5 has the following directories:

`bin/`	Location of executable (binary) `nemo`.
`contrib/`	Various software projects that use a part of NEMO 5 (e.g. the input parser or the MPI parallelization scheme) as a library. These projects are not considered to be part of NEMO 5.
`doc/`	*doxygen* developer documentation.
`examples/`	Example input decks that can be used as templates.
`include/`	C++ header files that can be used for the development of other codes when using NEMO 5 as a library.
`lib/`	The shared NEMO 5 libraries.
`manual/`	This user manual.
`materials/`	Material parameters. Currently all parameters are condensed in a single file `all.mat`.
`src/`	The source code. Users should not have to look at this directory.
`supplement/`	Loose pieces of additional information about implementation approaches, physical models, etc.
`tests/`	A suite of tests to test the functionality of individual modules.

Chapter 3

Input and Output

3.1 Input

The simulation is steered through a single file, the input deck, which is described in detail in chapter 4. A second type of input are the material parameters.

3.1.1 The material file

Currently all material parameters are provided through a single file. The user is free to use his/her own file through specification in the input deck, but the NEMO distribution comes with the file `materials/all.mat`. Inspection of the file reveals that it is more than a mere collection of numbers. NEMO parses and understands simple mathematical expressions entered for a parameter, something that comes in handy when e.g. a temperature dependence of the lattice constant or the band gap shall be included. A few simple rules govern the syntax of the file:

Parameters are **grouped** and **hierarchized** using the `group` keyword. The top group specifies the material.

Entries must have an **end of line** character `;`. Omission of `;` has the result that the next entry will not be recognized.

Possible **mathematical operators** include `+`, `-`, `*`, `/`, `^`, `sqrt` as well as round brackets `()`.

Parameters can be **cross-referenced**, but a parameter that is not in the same (sub-)group must be preceded with its full location with groups separated by a colon. Example: `GaAs:Lattice:a_lattice`.

Double quotation marks (") should be used for **strings**.

If the code requests a **numerical entry** for some parameter, NEMO will try to parse whatever expression is given and throw an error if the result is not convertable into a number.

The code is smart in that entries are only evaluated when they are requested. This means that a parameter can be reset and all parameters depending on it will have the modified value when they are being read the next time. This should however not be relevant to the user since it is up to the code to handle these things correctly.

The material database is a stand-alone module and was originally developed by Ben Hailey at Purdue.

3.1.2 Specifying material parameters in the input deck

In addition to modifying material file, a user is able to specify material parameters in the input deck. Such a specification will supercede whatever entry is present in the material file, **however** if other parameters in that file depend on the specified parameter, an update of these parameters is *not* automatic. There are some cases where NEMO does these updates by an additional call to the material database, but the user should be cautious and check that the correct parameters are used.
An input deck material parameter is placed in the corresponding `Material` section and needs to have the full tree except for the material name, e.g. `Lattice:a_lattice`.

3.2 Output

Output is provided in various formats depending on the type of output. The following table gives an overview of possible output formats:

Data type	**File format**
Atomistic structure	VTK, XYZ, PDB, Silo*
Atom-based fields	VTK, Silo*
Continuum (i.e. non-atomistic) meshes	VTK, PLT
Vertex- or element-based continuum data	VTK
1-D (XY) data	ASCII

*Silo is the only format that can be employed in simulations where the atomic grid is distributed onto several MPI processes. The atomic grid is saved in the form of a `DB_POINTMESH`, i.e. without the bonds, to reduce the file size and be able and generate molecule-type plots in *Visit*. Atom-based data is saved as `Ucdvar` using the Multimesh and Poor Man's Parallel

IO (PMPIO) concepts for parallel simulations. Currently there is only a single output file containing the data aggregated from all processes that store the distributed grid.
VTK also comprises different internal formats. Atomistic data is saved in the `POLYDATA` format, the structure consisting of `POINTS`, `VERTICES` and `LINES` (this permits a visualization as a *molecule* plot in Visit) and datasets being `POINT_DATA` fields. Continuum data is saved in the `UNSTRUCTURED` format, the mesh consisting of `POINTS` and `CELLS`.
The DX file format is implemented for the special case of fields on simple-cubic atomic lattices.

3.2.1 Visualization tools

A variety of visualization software is freely available. Matlab and Tecplot are non-free but widely used.

File format	**Visualization tool**
VTK	Visit (`https://wci.llnl.gov/codes/visit`) Paraview (`http://www.paraview.org`)
Silo	Visit, Paraview
XYZ	Jmol (`http://jmol.sourceforge.net`), Visit, Paraview
PLT	Tecplot
PDB	Jmol
ASCII	Xmgrace, Matlab

Chapter 4

The Input Deck

4.1 Example

Listing 4.1 gives an example of an input deck. The example sets up a GaAs cube consisting of 5x5x5 regular unit cells. The structure is first saved to a VTK file. Then a calculation of a couple of quantum states in the sp^3s^* d^5 tight-binding model is performed using closed, non-periodic boundary conditions, and the results are saved to file (atomic wavefunctions again in VTK format). Lastly, overlap matrix elements between wavefunctions are computed and saved to file. Although the various sections are discussed in detail in the later parts of this chapter, a brief orientation is given here:

The first section `Structure` contains information about structure and materials. Information about materials, its atomic composition, and nonstandard material parameters (i.e. additional or superceding the material parameter file) is contained in the `Materials` subsection. The `Geometry` subsection specifies the geometric shape of individual regions. Currently a structure needs to be composed in the input deck and cannot be read in from an external file. Lastly, the `Domain` subsections define which regions are aggregated to a domain (each *Simulation* takes place on a certain *Domain*, and there can be several *Simulations* carried out, possibly in a coupled way, using a single input deck – such as in the present example).

The section `Solvers` sets the `Simulation` types. Each simulation, given in a `solver` subsection, has a set of options specific to its task.

The section `Global` defines the location of the material parameters and which of the defined `Simulation` entities are executed one after the other on the top level.

In the present input deck, there is only a single material type – GaAs –, a single region – a cuboid – and a single domain that is being employed by all simulations. There are three

`Simulation` objects - one for the output of the domain to a VTK file, one for the Schrödinger solver, and one for the calculation of matrix elements between eigenfunctions of the Schrödinger solver.
The remainder of the chapter is devoted to a detailed descriptions of these sections and their options. Some general remarks:

Comments are done in C++-style: `//` marks the remainder of a line as comment and `/* ... */` allows for multi-line comments.

Only `{ ... }` are accepted as **brackets**. The opening bracket needs to be put on the same line as the section name and the closing bracket on a separate line. There is no **end of line** character (such as `;`).

Vectors are given as `(a,b,c)` or `(1,2,3)`.

Vectors of vectors are given as `[(a,b), (c,d)]`.

Misspelled parameters are simply ignored and do not create an error, unless the corrected parameter is mandatory in a simulation and missing in the input deck.

Spaces can be either spaces or tabstops, it does not matter.

Listing 4.1: Example of a NEMO 5 input deck.

```
Structure
{
    Material
    {
        name = GaAs
        tag = substrate
        regions = (1)
        crystal_structure = zincblende
        Bands:TB:sp3d5sstar_SO:param_set = param_Jancu
    }

    Domain
    {
        name = structure1
        type = pseudomorphic
        base_material = substrate
        dimension = (20,20,20)
        periodic = (false, false, false)

        crystal_direction1 = (1,0,0)
```

```
        crystal_direction2 = (0,1,0)
        crystal_direction3 = (0,0,1)
        space_orientation_dir1 = (1,0,0)
        space_orientation_dir2 = (0,1,0)

        regions = (1)
        geometry_description = simple_shapes
    }

    Geometry
    {
        Region
        {
            shape = cuboid
            region_number = 1
            priority = 1
            min = (0,0,0)
            max = (5,5,5)
            tag = quantumdot
        }
    }
}
Solvers
{
    solver
    {
        name = my_schroedi
        type = Schroedinger
        domain = structure1
        active_regions = (1)
        tb_basis = sp3d5sstar_SO

        job_list = (assemble_H, passivate_H, calculate_band_structure)
        output = (energies, eigenfunctions_VTK)

        charge_model = electron_hole
        automatic_threshold = true
        chem_pot = 0.0
        temperature = 300

        eigen_values_solver = krylovschur
```

```
        number_of_eigenvalues = 10
        shift = 0.5

        k_space_basis = cartesian
        k_points = [(0,0,0)]
    }
    solver
    {
        name = my_overlap
        type = MatrixElements
        domain = structure1
        active_regions = (1)

        operator = overlap
        wf_simulation = my_schroedi
        output_file = matrix_elements
    }
    solver
    {
        name = my_structure
        type = Structure
        domain = structure1
        active_atoms_only = true
        structure_file = structure.vtk
        unit_cell_file = unit_cell.vtk
        output_format = vtk
    }
}

Global
{
    solve = (my_structure,my_schroedi,my_overlap)
    database = all.mat
}
```

The following color code will be used henceforth:

blue parameter	Mandatory in all cases.
`black parameter`	Optional (defaults will be used otherwise) or mandatory only in some cases.

4.2 `Structure` section

4.2.1 `Material` sections

Each appearing physical material – Si, GaAs and so on – must be described by such a `Material` section. The aim of the section is to provide information about the crystallographic structure (this information is necessary as e.g. nitrides may appear both in zincblende and wurtzite phases) as well as material parameters that are not present in the parameter file or should be overwritten.
Currently the following crystals are supported:

`diamond`: FCC Bravais lattice with 2-atomic basis, where the 2 atoms have the same type. Atoms at $(0,0,0)$ and $a/4(1,1,1)$. Lattice constant given by the parameter `Lattice:a_lattice`.

`zincblende`: FCC Bravais lattice with 2-atomic basis, where the 2 atoms have different type (cation and anion). Atoms at $(0,0,0)$ and $a/4(1,1,1)$. Lattice constant given by the parameter `Lattice:a_lattice`.

`simplecubic`: Simple-cubic Bravais lattice with 1-atomic basis. Atom at $(0,0,0)$. Lattice constant (discretization) given by the parameter `Lattice:a_lattice`.

`wurtzite`: Hexagonal Bravais lattice with 4-atomic basis. Lattice constants given by the parameters `Lattice:a_wurtzite`, `Lattice:c_wurtzite` and, for the determination of the internal structure parameter u, `Lattice:a_lattice`.

`hexagonal`: Hexagonal Bravais lattice with 1-atomic basis. Lattice constants given by the parameters `Lattice:a_wurtzite` and `Lattice:c_wurtzite`.

`Bi2Te3`: Rhombohedral (trigonal) Bravais lattice with 5-atomic basis. All atoms are on the z-axis. Lattice constants given by the parameters `Lattice:a_lattice`, `Lattice:c_lattice`, `Lattice:u_lattice` and `Lattice:v_lattice`.

`graphene`: Hexagonal Bravais lattice with 2-atomic basis. The third direction (c-axis, [0001] direction) is meaningless. Lattice constants given by `Lattice:a_lattice`.

`CNT`: 1D-translational carbon nanotubes with a basis that depends on the user's choice of `nanotube_indices`. By default the nanotube will be extended along the x-direction. The lattice constant of the unrolled sheet is given by `Lattice:a_lattice`.

`buckyball`: C_{60} molecule, really only for the fun of it. Lattice constant given by `Lattice:a_lattice`.

`bcc`: BCC Bravais lattice with 1-atomic basis. Atom at $(0,0,0)$. Lattice constant given by `Lattice:a_lattice`.

`caesiumchloride`: Simple-cubic Bravais lattice with 2-atomic basis where the atoms have different types. Atoms at $(0,0,0)$ and $a/2(1,1,1)$. Lattice positions are the same as for BCC, but the atomic types are different. Lattice constant given by `Lattice:a_lattice`.

`sodiumchloride` (rocksalt, NaCl): FCC Bravais lattice with 2-atomic basis where the atoms have different type. Atoms at $(0,0,0)$ and $a/2(1,0,0)$. Lattice constant given by `Lattice:a_lattice`.

`tetragonal` Tetragonal primitive (P) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice` and `Lattice:c_lattice`.

`tetragonalI` Tetragonal body-centered (I) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice` and `Lattice:c_lattice`.

`orthorhombic` Orthorhombic primitive (P) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice` and `Lattice:c_lattice`.

`orthorhombicC` Orthorhombic base-centered (C) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice` and `Lattice:c_lattice`.

`orthorhombicF` Orthorhombic face-centered (F) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice` and `Lattice:c_lattice`.

`orthorhombicI` Orthorhombic body-centered (I) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice` and `Lattice:c_lattice`.

`monoclinic` Monoclinic primitive (P) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice`, `Lattice:c_lattice` and `Lattice:alpha_lattice`.

`monoclinicC` Monoclinic base-centered (C) Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice`, `Lattice:c_lattice` and `Lattice:alpha_lattice`.

`triclinic` Triclinic Bravais lattice with 1-atomic basis. Lattice constants given by `Lattice:a_lattice`, `Lattice:b_lattice`, `Lattice:c_lattice`, `Lattice:alpha_lattice`, `Lattice:beta_lattice` and `Lattice:gamma_lattice`.

Please note that for crystals with a 1-atomic base the name of that atom is given by `Lattice:element` (e.g., `"Si"`) whereas for 2-atomic basis the atom names are given by `Lattice:cation` and `Lattice:anion`.

Parameter	Description
name	Material name

Parameter	Description
tag	Additional description
regions	A list of integers defining the regions that have this material (definition takes place in the `Region` sections)
crystal_structure	Crystallographic structure: see text for choices.
`nanotube_indices`	Relevant only for carbon nanotubes (CNTs). This determines the type of nanotube.
`doping_type`	One of `N` or `P`.
`doping_density`	The doping density in cm ^{3}.
`charge_model`	Can be `electron_core` or `electron_hole`. In the former option there are no holes, only electrons, an the bands are populated starting from the very bottom.
`doping_ionization_model`	Can be `full_ionization` (default) or `thermal_ionization`.
`doping_temperature`	Related to incomplete ionization: population of doping level E_D with factor $1 + \exp(\frac{E_D - E_F}{kT_D})^{1}$.
`ionization_energy`	Related to incomplete ionization: population of doping level E_D with factor $1 + \exp(\frac{E_D - E_F}{kT_D})^{1}$.
`doping_degeneracy`	Related to incomplete ionization: population of doping level E_D with factor $1 + \exp(\frac{E_D - E_F}{kT_D})^{1}$.
?	Additional material parameters which replace the ones from the material file (see also section 3.1.2).
`disorder_type`	In the case `totally_random_dopant` atomistically random doping will be generated. In all other cases, doping is treated in the virtual crystal approximation (Jellium model). Can also be `totally_random_alloy`
`seed`	set to an integer to use the same seed for the random number generator. Useful in conjunction with `totally_random_alloy` if you want the same randomized structure each time you run a simulation.
`polarization`	This optional 3-vector specifies the spontaneous (pyroelectric) polarization of the material.
`strain_simulation`	This optional specification of a strain solver will have a Poisson simulation get the strain tensor at every atomic location and compute the piezo-electric polarization.

The doping should be a region property rather than a material property, but it is in here for the moment.

4.2.2 `Geometry` section

4.2.2.1 `Region` sections

These sections define the basic geometrical shapes that constitute the simulated structure. Each region has a unique material (and crystallographic structure) which is associated in the `Material` section.

Parameter	Description
shape	One out of `cuboid`, `spheroid`, `pyramid`, `cylinder` or `dome`.
tag	Additional description
region_number	A unique integer number associated with the region.
priority	If regions are overlapping, this integer number determines which region 'wins'. A higher number means higher priority (wins). In case of overlapping regions with equal priorities, the outcome is somewhat arbitrary and depends on the order of construction in the code.
min, **max**	Vectors that define corner points of the shape. The precise meaning depends on the type of shape: `cuboid` - two opposite corners. `spheroid` - two opposite corners or the enclosing cuboid. `pyramid` - two opposite corners or the enclosing cuboid (?). `cylinder` - two opposite corners or the enclosing cuboid (?). Or you can use radius(

The pyramid and the cylinder currently have their axis along z. Note that after the geometry construction a rigid rotation of the whole structure can be performed using the `space_orientation` options in the `Domain` section.

4.2.2.2 `Boundary_region` sections

These sections **determine the contacts** and can be the same 3D geometrical entities (with the same options) as in a `Region` section. They have no direct influence on the constructed atomic grid but can be used as special entities in certain solvers.

4.2.3 `Domain` sections

A `Domain` is a collegion of `Regions` and defines the structure on which a `Simulation` is solved.

Parameter	Description
name	The name of the domain.
type	One of `pseudomorphic`, `finite_elements`. `pseudomorphic` means a dislocation-free atomistic grid, `finite_elements` currently uses the atomic unit cells as finite elements for a continuum grid.

The remaining options depend on the type of domain.

`Pseudomorphic Domain` options

The construction of this type of domain is illustrated in Fig. 4.1. The final domain is the intersection of a *canvas*, which is constructed by translations of the unit cell in the three Bravais directions, with the regions specified in the domain. Depending on the size, either of these two entities can limit the size of the final object.

Note that the unit cell is specified with the `crystal_directionN` and `space_orientation_dirN`

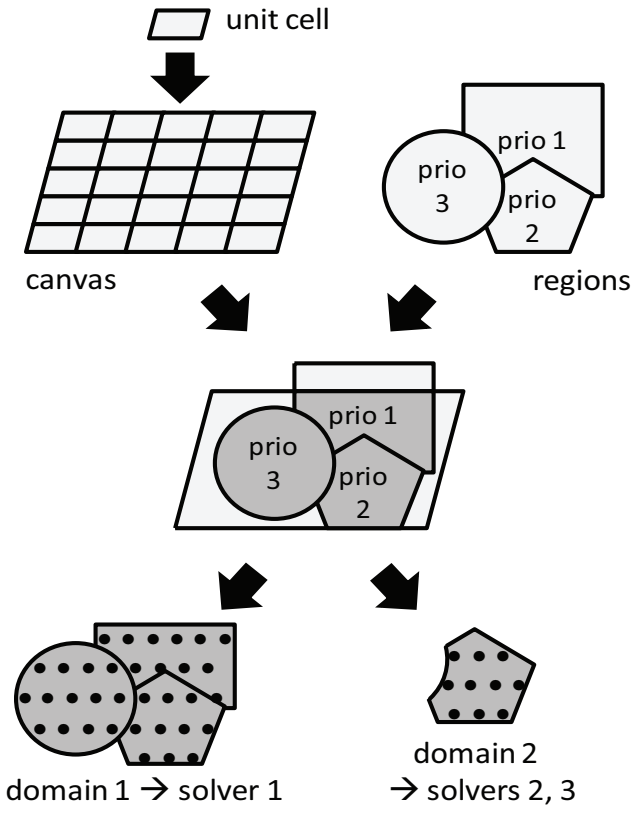

Figure 4.1: Schematic of the construction of a `PseudomorphicDomain`.

options. The former specifies the Bravais vectors in a standard notation connected to the

crystal. The latter provides the ability to perform a rotation of the constructed Bravais vectors.

Parameter	Description
regions	A list of integer numbers describing the regions that constitute the domain (as defined in the `Region` sections).
base_material	Must correspond to a tag in one of the defined `Material` entities.
dimension	This 3-vector gives the **maximum** extension, in **unit cells**, that the domain has. It defines a parallelepiped from which the regions will be cut out.
periodic	Entry X in this 3-vector defines whether the crystal is periodic in the direction `crystal_dimensionX`. It can be `true` or `false`.
crystal_directionN	(N=1,2,3) These 3-vectors give the (cartesian) coordinates of the vectors spanning the unit cell, in units of lattice constants.
crystal_direction_type	`plane` or `axis`. Specified whether crystal directions define crystal plane or crystal axis. Default is `axis`
space_orientation_dirN	(N=1,2) These vectors determine a rotation of the whole crystal, such that e.g. the first unit cell vector can in the end be pointing into the x-direction. The third vector is constructed out of the first two.
geometry_description	Can only be `simple_shapes` at the moment. In the future this parameter will determine whether the geometrical description of the structure is given by `Region` and `Boundary_region` sections in the input deck, or whether it is read in from an external file.
starting_cell_coordinate	A 3-vector that determines the starting point for the atomic grid construction.
origin	A 3-vector that is used for shifting everything a little bit for fine tuning.
`output`	A list that currently can be either empty or contain `xyz`. This option is deprecated since its functionality is replaced and extended by the `Structure` simulation.
`passivate`	Putting `false` will deactivate the attachment of H-atoms at non-periodic surfaces.

Parameter	Description
`passivation_regions`	This option matters for simulations where the tight-binding Schrödinger equation is solved only on a subset of regions (multiscale simulations). Atoms in regions with numbers contained in this list will be treated as Hydrogen-like in the tight-binding Schrödinger equation. Thus this list should be the complement of the list given in the `regions` option of the `Schroedinger` solver.
`conformal_partitioning`	Putting `false` will enable broken unit cells at the structure edges.
`FEM_1D`	A boolean that determines whether a 1D or a 3D FEM-Poisson equation will be set up. Setting this option to `true` speeds up 1D Schrödinger -Poisson simulations tremendously.

Table 4.2: `PseudomorphicDomain` parameters.

Note that for 2D-like graphene, 1D-like carbon nanotubes and 0D-like C_{60} buckyballs the Bravais vectors are meaningless. They are arbitrarily chosen to have some large value.

`Continuum Domain` options

Currently the FEM mesh is created out of the unit cells of the atomistic domain.

mesh_from_domain	The name of the atomistic domain from which the FEM mesh is built.

4.2.4 `Partitioning` section

This optional section determines the distribution of the grid(s) onto multiple CPUs in the case of MPI-parallelized simulations. Currently this distribution is done in a "3D-cuboid" way. A user has two options:
1. The user specifies the minimum and maximum extension in each direction plus the number of CPUs he/she wants for the distribution. In this case the options are:

{x,y,z}_extension	Three vectors with two entries (xmin,xmax) etc. that specify the extension of the structure. All atoms must be contained in the cuboid, and the cuboid should be fitted tightly to the actual structure.
`num_geom_CPUs`	The number of MPI process to employ in the grid distribution. When this parameter is omitted then the maximum number is employed. Note that this number must be formable as a product $n_x n_y n_z$ where $n_{x,y,z}$ 1.

The code will choose the combination (n_x, n_y, n_z) with a minimal surface between the partitions.
2. The user fine-tunes the partitioning by specifying the exact locations of the partition interfaces.

{x,y,z}_partition_nodes	Three lists with the location of the partition boundaries and interfaces. The number of processes employed in the grid distributed derived from $n_x n_y n_z$, where the lists have $n_{x,y,z}+1$ entries ($n_{x,y,z}$ 1).

4.3 `Solvers` section

This section consists of `solver` subsections. An example for an individual `solver` type would be a Schroedinger calculator of electronic eigenstates and energies. These entities can, and sometimes have to, interact with each other. A detailed description of all available solvers including their parameters is provided in chapter 5.

4.4 `Global` section

This section contains global simulation options:

Parameter	Description
solve	defines which of the defined `Simulation` types are actually solved. This set will sometimes *not* be the same as given in the `Solvers` section. One example could be a self-consistent Schroedinger-Poisson simulation, which logically consists of three `solver` types – `Schroedinger`, `Poisson` and the self-consistent coupler which utilizes the latter two – but only the `solve`-routine of the coupler is called. Why do it like this, you might ask? To preserve the modularity of individual tasks: Imagine e.g. `Schroedinger`, which really implements the atomistic tight-binding Hamiltonian, to be an interchangeable module for the density calculation, replacable by a **kp** simulator, and so on.
database	the location of the material parameter file
`messaging_level`	A number between 0 and 6 that specifies the amount of terminal output. 0 is near-silent and 6 has a lot of debug output. Usually 2-4 is a good choice. Levels 4-6 will invoke the memory managment reference counting output.
`logfile`	A file where terminal output gets saved into.
`output`	defines simulations that will perform output at the very end of NEMO5 execution.

Parameter	Description

`petsc_double_report`
 	
`petsc_complex_report`
	When set to `true`, then these options enable some terminal output at the end of the simulation that provides statistics about double and complex PETSc and SLEPc routines (SLEPc statistics are included in the PETSc option). These statistics include peak memory, flops, and spent time per PETSc/SLEPc routine and globally. The statistics do *not* include memory, flops and time used outside PETSc/SLEPc.

Chapter 5

Solvers and Options

This chapter describes the available solvers and their input deck parameters. A solver typically solves some set of equations, but there are also solvers which simply save something to file or do simple post-processing. Therefore a more general name is `Simulation`. These entities can interact and sometimes depend on other entities - the Poisson equation, which solves for the electrostatic potential, needs to obtain the excess carrier charge from another `Simulation`, for example. The color coding will be again as follows:

blue parameter	Mandatory in all cases.
`black parameter`	Optional (defaults will be used otherwise) or mandatory only in some cases.

5.1 Options common to all solvers

Each solver (`Simulation` object) has the following common parameters:

Parameter	Description
Parameter	Description
type	The simulation type: one of AdaptiveGrid, ApplyModeSpace, BackwardNonlocalRGFModule, BackwardRGFModule, BellEqNeq, BellRGF, BellRGF2, BellRGFModeSpace, BellRGFModeSpace2, BellRGFTrans, BlockDiagonalRGF, BlockSchroedingerModule, BosonHGLesserGreensolver, BosonInverseGreensolver, BosonLesserGreensolver, BosonRetardedContactSelfenergy, BosonRetardedGreensolver, BosonSpectralGreensolver, Brillouin, Cluster, Cluster_fitting, ComplexContourEnergyGrid, ConcatenateDomains, CorrelationGreenModule, DOFmapMultisizedTest, DOFmapping, DensityWrapper, DomainLeadOrganizer, DomainRandomizer, DomainRedistribution, DriftDiffusion, Dummy1, Dummy2, EHT3_unit_spin_test, EHT3_unit_test, EMSchrodingerFEM, EqNeqRGFModule, ExtHuckel3, Extrapolator, FermionHGLesserGreensolver, FermionInverseGreensolver, FermionLesserContactSelfenergy, FermionLesserGreensolver, FermionLesserScatteringSelfenergy, FermionRetardedConstantScatteringSelfenergy, FermionRetardedContactSelfenergy, FermionRetardedGreensolver, FermionRetardedScatteringSelfenergy, FermionSpectralGreensolver, ForwardNonlocalRGFModule, ForwardRGFModule, ForwardRGFSolver, ForwardRGFWrapperModule, GeneralLeadGreenModule, GeometryVIS, GreenModule, Green_Propagation, HamiltonianUTBgenerator, Huckel_RGF, IV, Interpolator, KeatingStrain, LRAModule, LRARGFModule, Lanczos, LocalBandstructureModule, MCBoltzmann, MCbulk, MatrixElements, MatrixTest1, Meta55, Meta5Schottky, Meta5_schottky, MetaBellQTBM, MetaBellQTBM2, MetaDummy, MetaLinearDropQTBM, MetaOmenQTBM, MetaOmenTransport, MetaOxide, MetaPoissonQTBM2, MetaPoissonQTBM3, MetaPoissonQTBM4, MetaPoissonQTBM4UTB, MetaPoissonQTBM5, MetaPoissonQTBM5_2top_gates, MetaPoissonQTBM5_EC, MetaRGF, MetaSolver, MetafullRGF, Metatemp5, ModuleDummy, ModuleDummyConstruct, ModuleDummyNested, MultiEqNeq, NEGF, NEGFG, NemoMatrixDummy, NonlinearPoisson, NonlinearPoissonFEM, NonlocalRGFModule, NonorthoRGFModule, PGAPackOptimization, PartitionRetardedSelfModule, PartitionRetardedSurfaceGreenModule, Phonons, Poisson, PoolObjectTest1, Propagation, PythonSolver, QTBMModule, QTBMTaskManagerModule, QTBM_Propagation, RGFModule, Ramper, ReadInCharge, ReadInCharge3D, ReadInPotential, ReadInPotentialWithAtoms, ReadinHamiltonianModule, Reinit, Repartition, ResonanceFinder, ResonanceFinderModule,

Parameter	Description
name	A unique name tag / identifier for the specific instance of the solver.
domain	The name of the domain on which the simulation is carried out, corresponding the `name` parameter in the `Domain` section.

Table 5.1: Parameters common to all simulations.

5.2 AdaptiveGrid

Provides various adaptive/inhomogeneous energy meshes for transport solvers. Currently operational: density of states based nanowire and ultra-thin body meshes. This solver allows using as few energy points as possible, but the important ones.

Parameter : Description	Default Value
Drain_chemical_potential : Drain Fermi level as defined in the transport solver.	(0.0 [eV]).
EOffset : Energy cutoff applied due to a drop in Fermi distribution for high energies	(25*k*300/q [eV]).
Elimit : One of the options that control how is the adaptive energy grid constructed given a band structure. Should not be changed. If it is necessary to change the density of energy points use scale_inhomogeneous_energy_steps option	(50.e-3 [eV]).
Emin_tail : One of the options that control how is the adaptive energy grid constructed given a band structure. Should not be changed. If it is necessary to change the density of energy points use scale_inhomogeneous_energy_steps option	(4.e-3 [eV]).
Parallelizer : Not used. If defined must be equal to Propagation_Parallelizer	().
Source_chemical_potential : Source Fermi level as defined in the transport solver.	(0.0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
bandstructure__smart__parallelization : Allows idle cores in band structure calculation, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
bandstructure__solver__drain : Name of the solver that gives the drain band structure	
bandstructure__solver__source : Name of the solver that gives the source band structure	
boundary__regions : required regions as boundary conditions	(()).
dE__f : One of the options that control how is the adaptive energy grid constructed given a band structure. Should not be changed. If it is necessary to change the density of energy points use scale_inhomogeneous_energy_steps option	(4.e-3 [eV]).

Parameter : Description	Default Value
dE_in : One of the options that control how is the adaptive energy grid constructed given a band structure. Should not be changed. If it is necessary to change the density of energy points use scale_inhomogeneous_energy_steps option	(1.e-3 [eV]).
dE_sep : One of the options that control how is the adaptive energy grid constructed given a band structure. Should not be changed. If it is necessary to change the density of energy points use scale_inhomogeneous_energy_steps option	(1.e-4 [eV]).
domain : Domain required for this simulation	().
energies_file_type : If energies_from_file is defined, it determines its file format. ['txt' for comma separated ,'omen' for space separated like in OMEN code]	(txt).
energies_from_file : Load energies from a txt file	().
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
external_output : Names of solvers external to AdaptiveGrid whose output is required.	().
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
name : A unique name tag / identifier for the specific instance of the solver.	
new_nanowire_simulation_flow : Allows new program flow for nanowire simulations, where AdaptiveGrid calls bandstructure solvers and not Poisson. May be important only for developers.	(false).
new_point_MPI_strategy : Not used	(cluster).

Parameter : Description	Default Value
non_rectangular : This option must be true for k-dependent energy grids, like in UTB devices, where there is transverse k integration	(false).
nonuniform_grid_type : Type of adaptive energy grid. Currently only the energy grid based on the lead DOS is implemented. The energy grid based on transmission resonances is implemented in a different solver	
output_file_suffix : suffix to be added on output files	().
scale_inhomogeneous_energy_steps : Used when strategy = omen. Makes the adaptive energy grid more or less dense, by multiplying the energy steps with a real number. Less than one means denser grid.	(1.0).
small_number_of_points :	(When strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).

Parameter : Description	Default Value
smart_adaptive_default_number_of_points : ONLY FOR 'strategy=smart_adaptive', this option defines the number of energy points to be used if energy range size is smaller that the defined threshold, see option smart_adaptive_energy_range_threshold.	(15).
smart_adaptive_energy_range_threshold : ONLY FOR 'strategy=smart_adaptive', this option defines the energy range size threshold.	(0.02).
smart_adaptive_initial_percent_of_points : ONLY FOR 'strategy=smart_adaptive', this option defines the Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid.	(0.8).
smart_adaptive_small_number_of_points : ONLY FOR 'strategy=smart_adaptive', this option defines the number of energy points to be used if energy range size is bigger that the defined threshold, see option smart_adaptive_energy_range_threshold.	(300).
strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).
surface_of_regions : surface regions required to calculate boundary conditions	().
temperature : Temperature to be used in this solver.	(Global temperature).
tic_toc_name : Prefix for tic-tocs	($(name)).
transmission_solver : Not used.	().

Parameter : Description	Default Value
transport_direction_cartesian : Vector of transport direction indices within the spatial cartesian coordinate system in the leads. 0 corresponds to x, 1 to y, and 2 to z. Each vector element is for one lead. Currently both leads must be equal. This is used to tell the adaptive grid solver the wave vector component that corresponds to the transport direction in the lead. It is necessary since the transport direction in the lead where the band structure is calculated is periodic and there can be multiple periodic directions, only one of which is transport direction.	((0,0)).
transport_direction_crystal : Vector of transport direction indices within the crystal coordinate system in the leads. Similar to transport_direction_cartesian, but this option specifies transport direction index within the crystal coordinate system. Currently both leads must be equal.	(transport_direction_cartesian).
transport_direction_k_index : Transport direction index within the spatial cartesian coordinate system in the leads. 0 corresponds to x, 1 to y, and 2 to z. This is used to tell the adaptive grid solver the wave vector component that corresponds to the transport direction in the lead. It is necessary since the transport direction in the lead where the band structure is calculated is periodic and there can be multiple periodic directions, only one of which is transport direction.	(0).

Table 5.2: Parameters of the AdaptiveGrid Solver.

5.3 ApplyModeSpace

This solver applies mode-space to input Hamiltonians and generate unitary matrix and reduced Hamiltonains.

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
d : Name of the simulation to be 'tested' (get_data(vec<double>)).	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.3: Parameters of the ApplyModeSpace Solver.

5.4 BackwardNonlocalRGFModule

[MODULE] BackwardRGF Module solver for arbitrary amount of offdiagonal blocks

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.4: Parameters of the BackwardNonlocalRGFModule Solver.

5.5 BackwardRGFModule

[MODULE] Forward Retarded Green's Function Module Solver

Parameter : Description	Default Value
$(lead_surface_green_$(lead_domain))_solver : This is the name of the solver which generates surface green's function in $(lead_domain)	().
$(momentum_names)_constructor : The name of the solver which creates the mesh or grid for the corresponding momentum_name	
Parallelizer : This is the name of the solver which creates the momentum (energy, k etc) grid	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : This is a list of integers defining the regions of the simulation domain where the solver is solved or contain active atoms	
boundary__regions : required regions as boundary conditions	(()).
chem__pot : defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
extra__domain__name__element : This string is used as a suffix to the name of each repartitioned subdomain	().
lead__domain : The name of the contact domain which gives the self-energy for the last subdomain (from where backward iteration starts)	
lead__surface__green__$(lead__domain) : This string refers to the name of the output Matrix or surface green's function generated by the lead self energy solver	().

Parameter : Description	Default Value
local_current_density : Option to turn on the calculation and output of the position resolved current density	(false).
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
potential_solver : The name of the solver (usually poisson solver) which generates the external potential profile to be added to the Hamiltonian Matrix.	().
result_propagator_name : This is the name of the propagator object which contains the final result of the solver, i.e. G< for all (E,k) values	
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The name of the Schroedinger solver which provides the Hamiltonian matrix for target_domain	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Name for advanced timing output	($(name)).
use_modified_correlation_equation : Used in RGFv2 for setting whether to assume g< is antihermitian or not. This assumption true in infinite precision but can cause stabilty issue with lambda*G scattering in finite precision.	(true).

Parameter : Description	Default Value

Table 5.5: Parameters of the BackwardRGFModule Solver.

5.6 BellEqNeq

BellEqNeqSolver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
EqNeqEq_regions : An array of 3 arrays. The first array contains a list of the regions where the device is in equilibrium, the second contains a list of regions where the device is non-equlibrium, and the third one lists the regions where the device is again in equilibrium, e.g. [(1,2),(3),(4,5)]	
Source_chemical_potential : Full chemical potential of start_lead [eV]	
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
automatic__energy__window : Energy windows based on conduction band and fermi levels	(false).
boundary__regions : required regions as boundary conditions	(()).
bsr__storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).
debug__output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e__points : Number of energy points to use	
eliminate__bad__points : Eliminate bad (E,k) points	(false).
energy__grid__kind : Specifies what kind of energy grid will be used.Homogeneous or Gauss. Gauss means energy grid from adaptive grid solver will be used.	
equilibrium__eta : Added to the Imaginary value for the energies in the equilibrium zones	
integration__order : Integration order over the k-mesh	(1).

Parameter : Description	Default Value
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
number_of_output_points : Number of points fot line segment for (E,k) resolved charge	().
output_ek_resolved_charge : If true, will output (E,k) resolved atomic charge	(false).
output_file_suffix : suffix to be added on output files	().
output_line_corners : Output line corners for (E,k) resolved charge	().
potential_solver : Solver that can provide electric potential on atoms	
sancho_rubio_eta : Added to the Imaginary part of energy used only for Sancho-Rubio, recommended vauel is 1e-9	
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	

Parameter : Description	Default Value
simulation_eta : Imaginary part of energy used both in self energy and RGF	
stop_if_negative_density : If true, will check if the density is probability is negative and stop if the happens	(false).
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.6: Parameters of the BellEqNeq Solver.

5.7 BellRGF

BellRGFSolver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
Source_chemical_potential : Full chemical potential of start_lead [eV]	
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
automatic__energy__window : Energy windows based on conduction band and fermi levels	(false).
boundary__regions : required regions as boundary conditions	(()).
bsr__storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).
debug__output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e__points : Number of energy points to use	
end__lead : List of the names of the leads domains that define the end for the RGF	().

Parameter : Description	Default Value
energy_window : Size of the energy window [eV] to sum to the conduction band in the case the automatic_energy_window option is used. By default it is calculated automatically according to source and draing temperature	(1).
integration_order : Integration order over the k-mesh	(1).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_G_corner : Do output corner block of G retarded	(false).
output_file_suffix : suffix to be added on output files	().
potential_solver : Solver that can provide electric potential on atoms	

Parameter : Description	Default Value
sancho_rubio_eta : Added to the Imaginary part of energy used only for Sancho-Rubio, recommended vauel is 1e-9	
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	
simulation_eta : Eta for Sancho-Rubio and RGF, recommended vauel is 1e-9	
start_lead : List of the names of the leads domains that create a start for the RGF	
stop_if_negative_density : If true, will check if the density is probability is negative and stop if the happens	(false).
strategy : Strategy for layer partitioning	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use potentially dangerous trick to speed-up the Hamiltonian construction	(false).

Table 5.7: Parameters of the BellRGF Solver.

5.8 BellRGF2

BellRGF2Solver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).

Parameter : Description	Default Value
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
Source_chemical_potential : Full chemical potential of start_lead [eV]	
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
bsr_storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).

Parameter : Description	Default Value
debug_output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e_points : Number of energy points to use	
end_lead : List of the names of the leads domains that define the end for the RGF	().
integration_order : Integration order over the k-mesh	(1).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().
potential_solver : Solver that can provide electric potential on atoms	
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	
start_lead : List of the names of the leads domains that create a start for the RGF	

Parameter : Description	Default Value
strategy : Strategy for layer partitioning	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.8: Parameters of the BellRGF2 Solver.

5.9 BellRGFModeSpace

BellRGFModeSpaceSolver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
Source_chemical_potential : Full chemical potential of start_lead [eV]	
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
automatic_energy_window : Energy windows based on conduction band and fermi levels	(false).
basis_generator : solver for generating the basis functions, usually the mode space solver	
boundary_regions : required regions as boundary conditions	(()).
bsr_storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).
debug_output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e_points : Number of energy points to use	
eletron_hole_model : if true, consider both electron and hole, otherwise always electron	(false).
end_lead : List of the names of the leads domains that define the end for the RGF	().
energy_grid_generator : solver for generating the energy grid, used when energy_grid_kind = adaptive	(homogeneous).

Parameter : Description	Default Value
energy_grid_kind : choose from: homogeneous, Gauss, adaptive	(homogeneous).
energy_window : Size of the energy window [eV] to sum to the conduction band in the case the automatic_energy_window option is used. By default it is calculated automatically according to source and draing temperature	(1).
integration_order : Integration order over the k-mesh	(1).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().
potential_solver : Solver that can provide electric potential on atoms	

Parameter : Description	Default Value
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	
simulation_eta : Eta for Sancho-Rubio and RGF, recommended vauel is 1e-9	
start_lead : List of the names of the leads domains that create a start for the RGF	
strategy : Strategy for layer partitioning. For mode space here, please use based_on_unit_cells	(bfs).
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use potentially dangerous trick to speed-up the Hamiltonian construction	(false).

Table 5.9: Parameters of the BellRGFModeSpace Solver.

5.10 BellRGFModeSpace2

BellRGFModeSpace2Solver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
Source_chemical_potential : Full chemical potential of start_lead [eV]	

Parameter : Description	Default Value
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
automatic_energy_window : Energy windows based on conduction band and fermi levels	(false).
basis_generator : solver for generating the basis functions, usually the mode space solver	
boundary_regions : required regions as boundary conditions	(()).
bsr_storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).
debug_output : Do debugging output?	(false).

Parameter : Description	Default Value
domain : Domain required for this simulation	().
e_points : Number of energy points to use	
eletron_hole_model : if true, consider both electron and hole, otherwise always electron	(false).
end_lead : List of the names of the leads domains that define the end for the RGF	().
energy_grid_generator : solver for generating the energy grid, used when energy_grid_kind = adaptive	(homogeneous).
energy_grid_kind : choose from: homogeneous, Gauss, adaptive	(homogeneous).
energy_window : Size of the energy window [eV] to sum to the conduction band in the case the automatic_energy_window option is used. By default it is calculated automatically according to source and draing temperature	(1).
integration_order : Integration order over the k-mesh	(1).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).

Parameter : Description	Default Value
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().
potential_solver : Solver that can provide electric potential on atoms	
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	
simulation_eta : Eta for Sancho-Rubio and RGF, recommended vauel is 1e-9	
start_lead : List of the names of the leads domains that create a start for the RGF	
strategy : Strategy for layer partitioning. For mode space here, please use based_on_unit_cells	(bfs).
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use potentially dangerous trick to speed-up the Hamiltonian construction	(false).

Table 5.10: Parameters of the BellRGFModeSpace2 Solver.

5.11 BellRGFTrans

BellRGFTransSolver

Parameter : Description	Default Value
BSR_block_size : Block size for the BSR format, this value must correspont with the tight binding size, if not the simulation is going to crash	(10).

Parameter : Description	Default Value
Drain_chemical_potential : Full chemical potential of end_lead [eV]	
Drain_temperature : Electron temperature in end_lead [K]	(300).
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
Source_chemical_potential : Full chemical potential of start_lead [eV]	
Source_temperature : Electron temperature in start_lead [K]	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
automatic_energy_window : Energy windows based on conduction band and fermi levels	(false).

Parameter : Description	Default Value
boundary_regions : required regions as boundary conditions	(()).
bsr_storage : True, if the hamiltonian matrices should be represent in BSR format. False otherwise	(1).
debug_output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e_points : Number of energy points to use	
end_lead : List of the names of the leads domains that define the end for the RGF	().
energy_window : Size of the energy window [eV] to sum to the conduction band in the case the automatic_energy_window option is used. By default it is calculated automatically according to source and draing temperature	(1).
integration_order : Integration order over the k-mesh	(1).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).

Parameter : Description	Default Value
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().
potential_solver : Solver that can provide electric potential on atoms	
sancho_rubio_eta : Added to the Imaginary part of energy used only for Sancho-Rubio, recommended vauel is 1e-9	
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	
simulation_eta : Eta for Sancho-Rubio and RGF, recommended vauel is 1e-9	
start_lead : List of the names of the leads domains that create a start for the RGF	
stop_if_negative_density : If true, will check if the density is probability is negative and stop if the happens	(false).
strategy : Strategy for layer partitioning	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use potentially dangerous trick to speed-up the Hamiltonian construction	(false).

Table 5.11: Parameters of the BellRGFTrans Solver.

5.12 BlockDiagonalRGF

BlockDiagonalRGFSolver

Parameter : Description	Default Value
$(momentum_names)_constructor : The solver that should provide the $(momentum_name). Some typical examples for transport calculations are the parallelizer for homogenous grid or adaptive grid solver for adaptive grid	
Hamilton_constructor : Name of the Hamiltonian constructor for this module.	().
Hamilton_constructor_$(lead_domain) : The simulation type that is to provide the Hamiltonian and DOFmap as required for the $(lead_domain)	
Hamilton_momenta : A list of extra momenta to be used by the Hamiltonian constructor.	().
Parallelizer : The simulation that is to provide the energy and possibly k parallelization. In most cases it is of type Green_Propagation and is done within the Propagation network	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
active_regions : The regions that are to be activated, if required, and used by the module that inherits PropagationModule.	
boundary_regions : required regions as boundary conditions	(().
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
destroy_auxiliary_solvers : If true, destroy auxiliary Schroedinger solvers after retarded self energy is obtained through RetardedSelfModule	(false).
do_output : Prints the propagator if true.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
domain_0 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_1 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_2 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_3 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
lead_domain : The domain that is connected to this PropagationModule. It can be used, for example, to get the coupling Hamiltonian between this domain and the $(lead_domain)	

Parameter : Description	Default Value
memory_thin_calculation : If true, the modules will try to deallocate results as soon as possible. If the result is queried again, it needs to be recalculated. If false, the results are kept in memory.	(true).
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
partitioned_lead_domain : Name of the lead domain which was partitioned using domain_[0-3] or repartition solver.	().
partitions_starting_from_device : Should the partitions start from the device?	(true).
potential_solver : Name of the potential solver	().
result_propagator_name : The name of the propagator that should be returned by the PropagationModule. It should correspond to the name in the NEGF object list for other Propagation solvers	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	($name_retarded_Green_Fermion_dra
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The simulation type that is to prove the Hamiltonian and DOF map of the target_domain.	
target_domain : Name of the domain that the PropagationModule will be used in	

Parameter : Description	Default Value
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature of device in Kelvin. Applied to Schroedinger solver	(300.0).
tic_toc_name : Custom user-defined name for use in timing/memory profiling	($(name)).

Table 5.12: Parameters of the BlockDiagonalRGF Solver.

5.13 BlockSchroedingerModule

This solver class provides a Hamiltonian in tri-blockdiagonal matrix shape

Parameter : Description	Default Value
OPTION_NAME : OPTION_DESCRIPTION	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
repartition_solver : If set, an external repartition solver will be used. If this is not given, this module will create a repartitioner.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.13: Parameters of the BlockSchroedingerModule Solver.

5.14 BosonHGLesserGreensolver

This solves the HG< for Bosons in the Block diagonal RGF

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).

Parameter : Description	Default Value
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(().
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(().
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(().
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).

Parameter : Description	Default Value
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).
calculate__Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).

Parameter : Description	Default Value
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).

Parameter : Description	Default Value
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().

Parameter : Description	Default Value
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(().

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).

Parameter : Description	Default Value
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).

Parameter : Description	Default Value
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).

Parameter : Description	Default Value
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.14: Parameters of the BosonHGLesserGreensolver Solver.

5.15 BosonInverseGreensolver

This is solving the inverse retarded Green's function for Bosons (typically something like (E*E-D-sigmaR))

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).

Parameter : Description	Default Value
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(().
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().

Parameter : Description	Default Value
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).

Parameter : Description	Default Value
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).

Parameter : Description	Default Value
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).

Parameter : Description	Default Value
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).

Parameter : Description	Default Value
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).

Parameter : Description	Default Value
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.15: Parameters of the BosonInverseGreensolver Solver.

5.16 BosonLesserGreensolver

Class to solve the lesser Green's function for Bosons

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.16: Parameters of the BosonLesserGreensolver Solver.

5.17 BosonRetardedContactSelfenergy

This solves the retarded contact self-energy for Bosons

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).

Parameter : Description	Default Value
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().

Parameter : Description	Default Value
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).

Parameter : Description	Default Value
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	().
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average_eps0 :	(12.0).
average_epsinf :	(10.0).
boundary_regions : required regions as boundary conditions	().
bp_analytical_momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).

Parameter : Description	Default Value
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).

Parameter : Description	Default Value
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).

Parameter : Description	Default Value
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).

Parameter : Description	Default Value
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().

Parameter : Description	Default Value
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).

Parameter : Description	Default Value
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.17: Parameters of the BosonRetardedContactSelfenergy Solver.

5.18 BosonRetardedGreensolver

This solves the retarded Green's function for Bosons

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).

Parameter : Description	Default Value
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).

Parameter : Description	Default Value
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	().
_output : list of solver to trigger output command	().
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	().

Parameter : Description	Default Value
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	

Parameter : Description	Default Value
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).

Parameter : Description	Default Value
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().

Parameter : Description	Default Value
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.18: Parameters of the BosonRetardedGreensolver Solver.

5.19 BosonSpectralGreensolver

This solves the spectral function for Bosons in the RGF algorithm

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().

Parameter : Description	Default Value
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).

Parameter : Description	Default Value
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).

Parameter : Description	Default Value
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).

Parameter : Description	Default Value
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	((()).

Parameter : Description	Default Value
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).

Parameter : Description	Default Value
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().

Parameter : Description	Default Value
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).

Parameter : Description	Default Value
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.19: Parameters of the BosonSpectralGreensolver Solver.

5.20 Brillouin

This simulation determines the Brillouin zone (BZ) of the given unit cell and saves it to file.
* Only the specification of the unit cell via cristal_directionN corresponding.
* BZs of supercells can be constructed by enlarging the cristal_directionN correspondingly.
* The saved structure always has its coordinates in Cartesian space.
* The result is currently always located in the file BZ.vtk

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).

Parameter : Description	Default Value
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file : VTK file where the 3D body will be saved. Please omit the .vtk-ending in this option	(./BZ).
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : name for the tic-toc output show up in the profiling file	($name).

Table 5.20: Parameters of the Brillouin Solver.

5.21 Cluster

ClusterSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.21: Parameters of the Cluster Solver.

5.22 Cluster_fitting

Cluster_fittingSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.22: Parameters of the Cluster_fitting Solver.

5.23 ComplexContourEnergyGrid

This solver creates an energy contour in the upper complex plane. This can be used to carry out an integration on a complex contour instead of the real energy axis using the Cauchy Integral Theorem

Parameter : Description	Default Value
Drain_chemical_Potential : Right contact fermi level	
Emin : Lower limit of Complex Energy Integration	
Source_chemical_Potential : Left contact fermi level	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).

Parameter : Description	Default Value
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
convergence_limit_w : Convergence limit in search of Gaussian quadrature weights	
convergence_limit_x : Convergence limit in search of Gaussian quadrature abscissas	
domain : Domain required for this simulation	().
llength : Length of the l part in the contour, it is the distance from right contact fermi level to the end point of l	
max_iteration : Maximum iteration number in search of Gaussian quadrature convergence	
name : A unique name tag / identifier for the specific instance of the solver.	
num_energy_points_on_l : Number of complex energy mesh points on l	
num_energy_points_on_sc : Number of complex energy mesh points on sc	
num_energy_points_on_v : Number of complex energy mesh points on v	
num_of_Efr_poles : Number of poles of the right contact fermi function included in the contour	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Parameter : Description	Default Value

Table 5.23: Parameters of the ComplexContourEnergyGrid Solver.

5.24 ConcatenateDomains

ConcatenateDomainsSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.24: Parameters of the ConcatenateDomains Solver.

5.25 CorrelationGreenModule

[MODULE] Correlation Green's function Module Solver

Parameter : Description	Default Value
$(momentum_names)_constructor : The name of the solver which creates the mesh or grid for the corresponding momentum_name	
$Hamilton_constructor_(neighbor_domains) : The name of the Schroedinger solver which provides the Hamiltonian for the neighbor_domain	
Parallelizer : This is the name of the solver which creates the momentum (energy, k etc) grid	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable_step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search_flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger_flag : nested execution of solvers,	(4).
active_regions : This is a list of integers defining the regions of the simulation domain where the solver is solved or contain active atoms	
boundary_regions : required regions as boundary conditions	(()).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
neighbor_domains : Name of the neighbor domains. A vector of strings.	(()).
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
potential_solver : The name of the solver which provides the external potential to be added to the diagonal terms of the Hamiltonian in the Schroedinger solver	().
result_propagator_name : This is the name of the propagator object which contains the final result of the solver, i.e. G<	
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature of the simulation.	(300.0).
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.25: Parameters of the CorrelationGreenModule Solver.

5.26 DOFmapMultisizedTest

DOFmapMultisizedTestSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().

Parameter : Description	Default Value
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.26: Parameters of the DOFmapMultisizedTest Solver.

5.27 DOFmapping

DOFmappingSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.27: Parameters of the DOFmapping Solver.

5.28 DensityWrapper

Propagation Density Wrapper

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__reset__set : If set to true, density solver is reset before solving	(false).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
approximate__derivative : Use approximate model for charge derivative	(false).
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
contact_self_lesser_solver : This is a set of solver names which calculates Sigma lesser in the contacts	().
conversion_factor : Density or jacobian value is multiplied by this factor to deliver the final result to potential solver	(1.0).
density_solver : name of the density solver	().
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
external_solver : This is a set of solver names which needs to be solved before solving density or jacobian like Propagation_Parallelizer etc.	().
external_solver_for_jacobian : This is a solver name which does some preparatory stuff for jacobian calculation.If not needed, keep this option empty	().
lambda : lambda parameter for the approximate jacobian	(1.0).
name : A unique name tag / identifier for the specific instance of the solver.	
output : This is a set of strings which defines the output to be generated. Currently, available options are charge and charge_derivative	().
output_file_suffix : suffix to be added on output files	().
output_solver :	(Call output for these external solvers.).
potential_solver : name of the potential solver	().
solve_before_density : This solver needs to be solved before density solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.28: Parameters of the DensityWrapper Solver.

5.29 DomainLeadOrganizer

DomainLeadOrganizerSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
domains : Domains that form a 1D chain	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.29: Parameters of the DomainLeadOrganizer Solver.

5.30 DomainRandomizer

DomainRandomizerSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.30: Parameters of the DomainRandomizer Solver.

5.31 DomainRedistribution

DomainRedistributionSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	().
_output : list of solver to trigger output command	().

Parameter : Description	Default Value
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.31: Parameters of the DomainRedistribution Solver.

5.32 DriftDiffusion

This simulation solves the stationary continuity equation $\nabla \quad \mathbf{J} = G \quad R$, where the current density **J** is expressed by the combination of drift and a diffusion term.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
poisson__solver : Name of the simulation that provides Poisson solver.	
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Name for advanced timing output	($(name)).

Table 5.32: Parameters of the DriftDiffusion Solver.

5.33 Dummy1

This simulation is a TEST simulation, it just triggers other simulation.

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
d : Name of the simulation to be 'tested' (get_data(vec<double>)).	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.33: Parameters of the Dummy1 Solver.

5.34 Dummy2

This simulation is a TEST simulation, it just print on screen values of each input option.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
a : option 'a'.	(not_defined).
active_regions : required regions to be activated for this solver	(()).
b : option 'b'.	(not_defined).
boundary_regions : required regions as boundary conditions	(()).
c : option 'c'.	(not_defined).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.34: Parameters of the Dummy2 Solver.

5.35 EHT3_unit_spin_test

EHT3_unit_spin_testSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).

Parameter : Description	Default Value
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.35: Parameters of the EHT3_unit_spin_test Solver.

5.36 EHT3_unit_test

EHT3_unit_testSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.36: Parameters of the EHT3_unit_test Solver.

5.37 EMSchrodingerFEM

EMSchrodingerFEMSolver

Parameter : Description	Default Value

Table 5.37: Parameters of the EMSchrodingerFEM Solver.

5.38 EqNeqRGFModule

This is the module for the transport calculation of a device using RGF equations in both equilibrium and non-equilibrium regions. An extra feature is that it incorporates the coupling of emitter quasi bound states with lead continuum states by adding an imaginary part to the on-site energies of equilibrium regions.

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(().
automatic_domain_repartition : Set to true, when EqNeq solver is defined on one large domain, this option enables partitioning of that large domain into 3 subdomains- equilibrium1, non-equilibrium and equilivrium2	(false).
boundary_regions : required regions as boundary conditions	(().
chemical_potential_for_$(end_leads) : Chemical potential of end lead	(0.0).
chemical_potential_for_$(start_leads) : Chemical potential of start lead	(0.0).
current : If set to true, generates current output files	(false).
debug_output : If set to true, produces outputs from Repartition solvers in Leads	(false).
density_Jacobian_output : If set to true, generates density jacobian output files	(false).
density_output : If set to true, generates density output files	(false).
domain : Domain required for this simulation	().
end_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green's functionin drain contact	().
name : A unique name tag / identifier for the specific instance of the solver.	
no_integration_for_transmission : Turns ON or OFF k-space integration in the periodic directions	(false).
output_file_suffix : suffix to be added on output files	().
potential_solver : The external potential output from this solver is added to the Hamiltonian	().

Parameter : Description	Default Value
start_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green'sfunction in the source contact	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
temperature_for_$(end_leads) : Temperature used for Fermi distribution and other temperature-dependent calculation of end_lead	(300.0).
temperature_for_$(start_leads) : Temperature used for Fermi distribution and other temperature-dependent calculation of start_lead	(300.0).
tic_toc_name : Prefix for tic-tocs	($(name)).
transmission : If set to true, produces transmission outputs	(false).

Table 5.38: Parameters of the EqNeqRGFModule Solver.

5.39 ExtHuckel3

This simulation solves the tight-binding schrodinger equation. It calculates eigenenergies and eigenstates as well as integrates the results to obtain electron and hole densities (if requested).

Parameter : Description	Default Value
AP_vca_model : strained SiGe VCA model as implemented by A. Paul, IEEE EDL v31 no4 2010	(false).
BSR_block_size : Size of matrix blocks when using block sparse row (BSR) format with sparse_storage_format option; ignored if format is compressed sparse row (CSR)	(10).
Center_Position : related to region: (min+max)/2 for magnetic_vector_potential A	((0,0,0)).
DOS_broadening : (for DOS calculation) Broadening strength in eV.	(dE).

Parameter : Description	Default Value
DOS_broadening_model : (for DOS calculation, default 2) 0: Lorentzian broadening, 1: Exponential broadening, 2: cosh broadening.	(2).
DOS_energy_grid : (for DOS calculation) Determines homogeneous energy grid in the form (Emin,Emax,dE).	(energy_input).
DOS_max_energy : (for DOS calculation) alternative specification of Emax.	(20.0).
DOS_min_energy : (for DOS calculation) alternative specification of Emin.	(-20.0).
DOS_points : (for DOS calculation) alternative specification of the number of energy points.	(100).
DOS_spin_factor : (for DOS calculation, default 1) optional multiplicative factor for DOS.	(1.0).
Dirac_gaussian_width : Width of Dirac delta function used for calculating scattering rate.	(1e-3).
Magnetic_Field : Magnetic field value in the form of (Bx, By, Bz)	().
Magnetic_Gauge : options: Symmetric (default) , Asymmetric_x , Asymmetric_y , Asymmetric_z.	(Symmetric).
Neumann_BC : Set Neumann BC for hamiltonian, which can generate a good basis for EM LRA, implemented by Lang Zeng	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Peierls_Phase : true/false for applying peierls phase for nonzero magnetic field..	(true).
State_solver : Gets the external state solver	().
Zeeman_Splitting : true/false for applying zeeman splitting for nonzero magnetic field.	(true).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
allow__reinit__k__points : When Schroedinger solver is reinitialized, reinitialize the k (wave vector) space object in order to be able to change the values of k points. Implemented for use with band solvers associated with adaptive grid.	(false).
atom__order__from : Determines ordering of atoms in Hamiltonian when repartitioner solver is attached to DOFmap. Most likely, set this to the name of the repartition solver	().

Parameter : Description	Default Value
automatic_threshold : If true, then the code automatically determines the energy threshold that separates electrons and holes (i.e. conduction and valence bands) by looking at the bulk band edges and the electrostatic potential. When iterative band structure solvers (like Krylov-Schur) are used this option causes the shift value to be equal to the threshold mentioned above. The actual value of the threshold depends on other options. For more details search for _automatic_threshold variable in the code above. This option is also used for density calculations in Schroedinger solver and not just band structure calculations. Possible values: true/false.	(false).
basis_generator : The (optional) name of the simulation object where the basis set for transforming the Hamiltonian is drawn from.	().
beta_degradation : Numerical parameter if eigensolver uses lanczos	(0.).
boundary_regions : required regions as boundary conditions	(().
brillouin_zone_meshsize : If calculate_brillouin_zone is true, this determines the mesh size. The spacing in every direction, in pi/a	(0.1).
calculate_automatic_shift_subbandbased : implemented by R. Kotlyar. When true, the convergence scheme based on determining automatic shifts from subband energies is specified.	(false).
calculate_brillouin_zone : When set to true, the entire Brillouin zone will be meshed and computed.	(0.1).
calculate_elastic_overlap : calculates scattering-related form factors. deprecated	(false).

Parameter : Description	Default Value
calculate_integrated_electron_density : implemented by R. Kotlyar. These options were added to fix the shift of the eigenvalue solver based on monitoring the subband energies from previ- ous iterations. Usually these shifts are determined by conduc- tion or valence band minima/maxima. Roza found that this convergence scheme works the best for high electric fields in the presence of large local band bendings.	(false).
calculate_integrated_hole_density : implemented by R. Kotlyar. If option is specified and spatially varying threshold = false, the integrated the total carrier density in cm-dim is computed by integration over the confinement dimensions (e.g. for wires this yields 1/cm). As similar option exists for electrons.	(false).
charge_model : In electron_hole you have negatively charged electrons and positively holes. In electron_core you have negatively charged electrons and positively charged ions. Can be electron_core or electron_hole. When set to electron hole (most cases), then an energy threshold separates between electrons and holes. When set to electron core model all states are assumed to be electrons	(electron_hole).
check_every_steps_number : Numerical parameter if eigensolver uses lanczos. Try a value between 20 and 500	(0).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
compute_edges_masses : If set to true, some band edges and masses will be computed and written to screen. This option was tested with bulk only and fails to compute correct transverse masses for X- and L- valleys in the presence of spin-orbit interaction.	(0.001).

Parameter : Description	Default Value
compute_edges_masses_delta : delta K to calculate numerical derivative of the band dispersion. in pi/a. stepsize for the finite difference scheme employed in compute edges masses,	(0.001).
convergence_limit : Convergence limit for iterative solution methods (accuracy of eigenvalues).	(1e-8).
convergence_method : Numerical parameter if eigensolver uses lanczos. Can be full_convergence or partial_convergence	().
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
cutoff_distance_to_bandedge : Energy cutoff which is added to bandedge for holes and subtracted for electrons. Setting this to some value Delta (in eV) leads to electron thresholds at Ec - ePhi(x) - Alpha and hole thresholds at Ev - e Phi(x) + Delta.	(0.2*(Ec_min-Ev_max)).

Parameter : Description	Default Value
dE_for_elastic_overlap :	(0.0,Warning: option calculate_elastic_overlap is deprecated. Energy threshold for calculating overlap for elastic scattering. Overlap is calculated if difference between energy_for_elastic_overlap option and energy eigenvalue is smaller than dE. Only takes effect if take_all_states_and_no_overlap option is false.).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
direct_order_of_partitions_$neighbor_domain_name$: Orders neighbor domain partitions directly if true, in reverse if false. Considered only with neighbor domain repartitioner is provided.	true).
domain : Domain required for this simulation	().
doping_density : Doping density of material. Only used if calculate_Fermi_level is included in job list. Doping type (i,n,p) set by option doping_type.	(0.0).
doping_type : Type of doping	(i).
e_xx : When giving strain as input, put epsilon_xx of strain tensor here(for tb_basis = kp only)	(0.01).

Parameter : Description	Default Value
e_xy : When giving strain as input, put epsilon_xy of strain tensor here(for tb_basis = kp only)	(0.00).
e_yy : When giving strain as input, put epsilon_yy of strain tensor here(for tb_basis = kp only)	(0.01).
e_yz : When giving strain as input, put epsilon_yz of strain tensor here(for tb_basis = kp only)	(0.00).
e_zx : When giving strain as input, put epsilon_zx of strain tensor here(for tb_basis = kp only)	(0.00).
e_zz : When giving strain as input, put epsilon_zz of strain tensor here(for tb_basis = kp only)	(0.01).
eff_mass_dump : true: for effective mass calculation only when tb_basis=kp/ false: for disabling it only when tb_basis=kp	(false).
eigen_values_solver : Which eigenvalue solver to use. Setting lapack always com- putes all eigenvalues and is feasible only for very small sys- tems. Recommended choices are krylovschur and arpack. Other choices are arnoldi, jd, gd.	().
electron_chem_pot : Chemical potential used to calculate density of states times Fermi derivative. Has units of eV.	(0.0).
electron_hole_heuristics : Given the atomically resolved energy threshold between electrons and holes, this option tells how to actually determine whether a particle is an electron or hole. Two options available: sharp (sharp transition at the threshold) and omen_smooth (smooth transition around the threshold, like in OMEN code)	(sharp).
electron_temperature : Temperature of the system	(300).
energy_for_elastic_overlap : Energy used for calculating matrix elements for wave functions	(0.0).

Parameter : Description	Default Value
energy_maximum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_minimum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_threshold_for_shift : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enthresh_foresolvershift).
epsilon_matrix : Constant strain tensor matrix, given in laboratory system co- ordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the laboratory system where crystal axis might have been rotated.	(strain_matrix).
epsilon_matrix_crystal : Constant strain tensor matrix, given in crystal system coordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the crystallographic system.	(strain_matrix).
extra_points : Extra K points set by user	(NULL).
fixed_shift : Sets the shift to be used by eigenvalue solvers (except when electron_core model: check option shift).	(0.0).
gaussian_quadrature_order_automatic : Affects k-space integration for periodic directions. If option integration_order is not defined and if gaussian_quadrature_order_automatic = true, then for number_of_k_points <= 22 use Gaussian quadrature with single partition, otherwise use simple rectangular with number of elements equal to number_of_k_points. In both cases the actual number of k points equals number_of_k_points.	(false).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
init_check_steps_number : lanczos option, for checking convergence after how many iterations	(20).
input_shift_subbandbased_for_restart : When true, the shift is read in from file and is used as initial shift to eigensolver. R. Kotylar	(false).
input_shiftfilename_for_restart : Filename for input_shift_subbandbased_for_restart. R. Kotylar	(string()).
input_stress_or_strain : stress: for giving stress as input/ strain: for giving strain as input directly(for tb_basis = kp only)	(strain).
integration_order : Sets the order of integration for Gaussian quadrature method	(1).
is_kp_strain_on : true: for enabling strain calculation with k.p/ false: for disabling strain calculation with k.p(for tb_basis = kp only)	(false).
is_piezoelectric_effect_on : true/false; for enabling piezoelectric calculation with k.p;(for k.p simulation only)	(false).
is_spatially_parallelized : true/false; Whether the device is space parallelized;(for k.p simulation only)	(false).
is_vff_strain_on : true/false; for enabling VFF strain calculation with k.p;(for k.p simulation only)	(false).
job_list : A list that deterimes what is done. Choose from assemble_H, passivate_H, include_strain_H, include_shear_strain_H, calculate_band_structure, electron_density, derivative_electron_density_over_potential, hole_density, derivative_hole_density_over_potential, spin, DOS. assemble_H is activated by any other option automatically.	(job_list).
k0 : k0 vector to solve eigenvalues at this particular k-point	([0 0 0]).

Parameter : Description	Default Value
k_degeneracy : The computed density is multiplied by this number to account for k-space degeneracy. E.g. when kxmax and kymax are set in a simulation with 2D k-space, k degeneracy should be 4.	(1.0).
k_points : Only relevant for calculate band structure. This parame- ter is a list of points in k-space along which the band structure is calculated.	(k_pointsV).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_basis : (Default: cartesian) Sets the basis in which k points is specified. When set to reciprocal the coordinates given in k points are assumed to be w.r.t. the reciprocal lattice vectors b1 =a2xa3 / a1.(a2xa3) etc.	(reciprocal).
kp_band_number : 8 band for direct band gap material / 6 for indirect band gap material / 4 for rhombohedral crystals of TI(for tb_basis = kp only)	(8).
kp_strain_dir1 : If stress is perpendicular to [x y z] plane then input x here(for tb_basis = kp only)	(1).
kp_strain_dir2 : If stress is perpendicular to [x y z] plane then input y here(for tb_basis = kp only)	(0).
kp_strain_dir3 : If stress is perpendicular to [x y z] plane then input z here(for tb_basis = kp only)	(0).
kp_stress_mag : When giving stress as input, put stress in GPa here(for tb_basis = kp only)	(1).
kpoint_to_monitor_subbands_for_shift : The k-point where to look for shift. default is 0. R. Kotylar	(kmagref_foresolvershift).
kpoint_tolerance_to_monitor_subbands_for_shift : The k-tolerance when looking for a k-point on k-grid. Default is 0.001. R. Kotylar	(kmagreftol_foresolvershift).

Parameter : Description	Default Value
kxmax : Upper boundary in 2 pi/a of the simulated kx space. kx ranges from 0 to this number.	(1.0).
kxmin : Lower boundary in 2pi/a range of the simulated kx space. kx begins from zero by default.	(0.0).
kymax : Upper boundary in 2 pi/a of the simulated kx space. ky ranges from 0 to this number.	(1.0).
kymin :	(0.0,Lower boundary in 2pi/a range of the simulated ky space. ky begins from zero by default.).
kzmax : Upper boundary in 2 pi/a of the simulated kz space. kz ranges from 0 to this number.	(1.0).
kzmin :	(0.0,Lower boundary in 2pi/a range of the simulated kz space. kz begins from zero by default.).
lanczos_iseed : Numerical parameter if eigensolver uses lanczos	(0.).
linear_solver : Linear solver employed in the shift-and-invert operation. This should be preferable a direct linear solver since the LU fac- torization can be reused during the Krylov iterations.	().
max_number_iterations : Maximum number of Krylov iterations (irrelevant for lapack).	(0).
maximum_delta_k : Upper bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(1e80).

Parameter : Description	Default Value
maximum_scattering_angle : Option connected to computation of scattering matrix elements	(180.0).
minimum_delta_k : Lower bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(0.0).
minimum_scattering_angle : Option connected to computation of scattering matrix elements	(0.0).
modes_for_current : Relevant for current calculation. Define the number of modes to be extracted from E-k while calc. ballistic TOB current.	().
monitor_convergence : When set to true, terminal output related to the Krylov iteration is generated.	(false).
mpd : Specialized numerical option. Refer to the SLEPc user manual for the meaning.	(0).
mumps_ordering : Relevant only for eigen values solver=mumps. Choose between pord and metis.	().
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : Specialized numerical option (size of the Krylov subspace).	(0).
no_of_spatial_division : No of cores used for parallelizing the device(for tb_basis = kp only)	(256).
noise_potential : noise potential used in qtbm to break degenerate modes.	(0.0).
number_of_eigenvalues : Number of eigenvalues to compute (irrelevant for lapack).	(0).
number_of_eigenvalues_to_use : Number of requested eigenvalues	(0).
number_of_k_points : This list gives the uniform discretization of each segment set by the k points parameter (note that specifying N points means N - 1 segments). For density calculations, setting this parameter to 0 leads to computation of k = 0 only and application of an analytical formula that assumes parabolic subbands.	(10).

Parameter : Description	Default Value
number_of_nodes : number of nodes for each k-segment	(()).
output : Choose from Hamiltonian, energies, k-points, DOS, electron density, electron density VTK, hole density, hole density VTK, ion density, eigenfunctions, eigenfunctions k0, eigenfunctions VTK, eigenfunctions VTK k0, eigenfunctions Silo, eigenfunctions Silo k0, spin.	(output_list_v).
output_eigenstate_energy : Energy around which eigenpair has to be output.	(0.0).
output_file_suffix : suffix to be added on output files	().
output_k_point : Value of k-point dumped as output	(k0).
output_precision : Accuracy of saved eigenvalues within output file.	(12).
output_shift_subbandbased_for_restart : When true, at the end of simulation the file can be output with the last eigensolver shift information. Using this shift and the last potential, the next simulation can be restarted from the last state of simulation. R. Kotylar.	(false).
p_parameter : It's the block size for block Lanczos. If the degeneracy of the eigenvalues is known, then set it to that value, otherwise start with 4.	(0).
parallelize_here : When set to false, some other simulation (like Propagator) can determine the parallelization.	(true).
potential_solver : The (optional) name of the simulation object where the electrostatic potential is drawn from.	().
preconditioner : This option controls how the eigenvalue problem is transformed before solution Possible options are: "lu" "mumps"...work ongoing . (default: lu) Preconditioner employed in the shift-and-invert operation.	(lu).

Parameter : Description	Default Value
project_DOS_ : This option allows to calculate the project density of states	(false).
rate_of_shift_changes_between_iterations : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enrate_foresolvershift).
reinit_parallelize : When Schroedinger solver is reinitialized, reinitialize the MPI parallelization as well.	(false).
resolution : The resolution to be used for the eigensolver	(0.0).
reuse_hamiltonian : if this option is false, hamiltonian would be recalculated every iteration, for different k always it is recalculated	(true).
seed : seed for noise potential.	(-1).
set_valley : set valley name to get Material properties	(DEFAULT).
shift : The eigensolver will search for eigenvalues around the shift value. This seems to only be used for electron_core model. fixed_shift is used otherwise.	(0.001).
solve_on_single_replica : Gather density to the single rank only	(false).
solver_transformation_type : (default: sinvert) Use sinvert for shift-and-invert, shift for shift.	(sinvert).
sort_like_electrons : This option controls the way the eigenenergies are sorted. When true (electrons) then ascending and when false (holes) then descending. Possible values: true/false.	(true if job list has electron density or band structure or electron core model. false if job list has hole density.).
sparse_storage_format : Sparse storage format in which PETSc will allocate sparse matrices; options available are "csr" (compressed sparse row) and "bsr" (block sparse row)	(csr).

Parameter : Description	Default Value
spatially_varying_threshold : If true, then the threshold energy discriminating between electrons and holes will vary spatially according to the elec- trostatic potential. Needed in Schroedinger-Poisson simulations.	(false).
spin_resolved : if this option is enabled all parameters will be spin dependant Up/Down	(false).
strain_calc_option : hydrostatic: for calculating the effect of hydrostatic strain/ biaxial: for calculating the effect of biaxial strain/ uniaxial: for calculating the effect of uniaxial strain (use this option only when giving strain as input)(for tb_basis = kp only)	(uniaxial).
strict_threshold : Solve for either conduction (electron density) or valence (hole density) bands depending on option sort_like_electrons. Possible values: true/false.	(true if job list has electron density or hole density. false if job list has band structure or if electron_core charge model is used.).
surface_of_regions : surface regions required to calculate boundary conditions	().
take_all_states_and_no_overlap : This needs the option "calculate_elastic_overlap" to be true. Allows to switch the Schroedinger from solving the scattering overlap for a subset of states to collect all states and make them ready for other solvers to get via get_data. Primary purpose of this is to support the fitting of TB to DFT including the TB wavefunctions	(false).
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	(sp3).
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Tic toc name to be used for saving timing results for this simulation.	().

Parameter : Description	Default Value
tolerance : The judgement of the convergence. Once the target reaches the tolerance then end the loop.	(0.0).
use_monomials : calculate monomials table based on the orbitals	(true).
use_neighbour_repartitioner_$neighbor_domain_name$: Repartitioner solver name for neighbor domain.	()

Table 5.39: Parameters of the ExtHuckel3 Solver.

5.40 Extrapolator

This solver extrapolates an input matrix given on a input domain to a target matrix on a target domain

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_subdomains : Number of subdomains to be created by device repartitioner	(1).
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.40: Parameters of the Extrapolator Solver.

5.41 FermionHGLesserGreensolver

This solves the HG< for Fermions in the Block diagonal RGF

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().

Parameter : Description	Default Value
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).

Parameter : Description	Default Value
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).
calculate__Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).

Parameter : Description	Default Value
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).

Parameter : Description	Default Value
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().

Parameter : Description	Default Value
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(().

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).

Parameter : Description	Default Value
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).

Parameter : Description	Default Value
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).

Parameter : Description	Default Value
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.41: Parameters of the FermionHGLesserGreensolver Solver.

5.42 FermionInverseGreensolver

This is solving the inverse retarded Green's function for Fermions (typically something like (E-H_sigmaR))

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).

Parameter : Description	Default Value
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().

Parameter : Description	Default Value
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).

Parameter : Description	Default Value
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).

Parameter : Description	Default Value
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).

Parameter : Description	Default Value
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).

Parameter : Description	Default Value
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).

Parameter : Description	Default Value
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.42: Parameters of the FermionInverseGreensolver Solver.

5.43 FermionLesserContactSelfenergy

This solves the lesser contact self-energy for Fermions

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().

Parameter : Description	Default Value
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).

Parameter : Description	Default Value
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().

Parameter : Description	Default Value
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average_eps0 :	(12.0).
average_epsinf :	(10.0).
boundary_regions : required regions as boundary conditions	(()).
bp_analytical_momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).

Parameter : Description	Default Value
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).

Parameter : Description	Default Value
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().

Parameter : Description	Default Value
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).

Parameter : Description	Default Value
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).

Parameter : Description	Default Value
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().

Parameter : Description	Default Value
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().

Parameter : Description	Default Value
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).

Parameter : Description	Default Value
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.43: Parameters of the FermionLesserContactSelfenergy Solver.

5.44 FermionLesserGreensolver

This solves the lesser Green's function for Fermions

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).

Parameter : Description	Default Value
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).

Parameter : Description	Default Value
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).
calculate__Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).

Parameter : Description	Default Value
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).

Parameter : Description	Default Value
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().

Parameter : Description	Default Value
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).

Parameter : Description	Default Value
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).

Parameter : Description	Default Value
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).

Parameter : Description	Default Value
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.44: Parameters of the FermionLesserGreensolver Solver.

5.45 FermionLesserScatteringSelfenergy

This solves the lesser scattering self-energy for Fermions in the case of self-consistent Born models

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).

Parameter : Description	Default Value
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(().
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().

Parameter : Description	Default Value
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average__eps0 :	(12.0).
average__epsinf :	(10.0).
boundary__regions : required regions as boundary conditions	(()).
bp__analytical__momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().

Parameter : Description	Default Value
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).

Parameter : Description	Default Value
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).

Parameter : Description	Default Value
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	((()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().

Parameter : Description	Default Value
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).

Parameter : Description	Default Value
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_funct we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).

Parameter : Description	Default Value
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().

Parameter : Description	Default Value
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.45: Parameters of the FermionLesserScatteringSelfenergy Solver.

5.46 FermionRetardedConstantScatteringSelfenergy

This solves the retarded scattering self-energy for Fermions in the case of Buettiker probes, constant eta or the lambda*G self-consistent Born model

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().

Parameter : Description	Default Value
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().

Parameter : Description	Default Value
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().

Parameter : Description	Default Value
_output : list of solver to trigger output command	().
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	().
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average_eps0 :	(12.0).
average_epsinf :	(10.0).
boundary_regions : required regions as boundary conditions	().
bp_analytical_momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().

Parameter : Description	Default Value
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).

Parameter : Description	Default Value
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).

Parameter : Description	Default Value
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().

Parameter : Description	Default Value
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).

Parameter : Description	Default Value
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.46: Parameters of the FermionRetardedConstantScatteringSelfenergy Solver.

5.47 FermionRetardedContactSelfenergy

This solves the retarded contact self-energy for Fermions

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).

Parameter : Description	Default Value
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().

Parameter : Description	Default Value
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().

Parameter : Description	Default Value
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average__eps0 :	(12.0).
average__epsinf :	(10.0).
boundary__regions : required regions as boundary conditions	(()).
bp__analytical__momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp__chem__potential__direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp__chem__potential__linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp__coupling : setup the BP coupling value, include real and imaginary part	().
bp__current__threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp__full__transmission__calculation : output the type for the transmission calculation	(false).

Parameter : Description	Default Value
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).

Parameter : Description	Default Value
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).

Parameter : Description	Default Value
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().

Parameter : Description	Default Value
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).

Parameter : Description	Default Value
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.47: Parameters of the FermionRetardedContactSelfenergy Solver.

5.48 FermionRetardedGreensolver

This solves the retarded Green's function for Fermions

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(().
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(().
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).

Parameter : Description	Default Value
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).

Parameter : Description	Default Value
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	().
_output : list of solver to trigger output command	().
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	().

Parameter : Description	Default Value
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	

Parameter : Description	Default Value
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).

Parameter : Description	Default Value
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().

Parameter : Description	Default Value
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.48: Parameters of the FermionRetardedGreensolver Solver.

5.49 FermionRetardedScatteringSelfenergy

This solves the retarded scattering self-energy for Fermions in the case of self-consistent Born models

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().

Parameter : Description	Default Value
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).

Parameter : Description	Default Value
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().

Parameter : Description	Default Value
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average_eps0 :	(12.0).
average_epsinf :	(10.0).
boundary_regions : required regions as boundary conditions	(()).
bp_analytical_momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).

Parameter : Description	Default Value
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).

Parameter : Description	Default Value
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().

Parameter : Description	Default Value
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).

Parameter : Description	Default Value
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).

Parameter : Description	Default Value
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().

Parameter : Description	Default Value
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().

Parameter : Description	Default Value
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).

Parameter : Description	Default Value
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.49: Parameters of the FermionRetardedScatteringSelfenergy Solver.

5.50 FermionSpectralGreensolver

This solves the spectral function for Fermions in the RGF algorithm

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).

Parameter : Description	Default Value
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).

Parameter : Description	Default Value
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).
calculate__Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).

Parameter : Description	Default Value
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).

Parameter : Description	Default Value
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(().
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().

Parameter : Description	Default Value
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(().

Parameter : Description	Default Value
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).

Parameter : Description	Default Value
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).

Parameter : Description	Default Value
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).

Parameter : Description	Default Value
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.50: Parameters of the FermionSpectralGreensolver Solver.

5.51 ForwardNonlocalRGFModule

[MODULE] ForwardRGF Module solver for arbitrary amount of offdiagonal blocks

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.51: Parameters of the ForwardNonlocalRGFModule Solver.

5.52 ForwardRGFModule

[MODULE] Forward Green's Function Module Solver

Parameter : Description	Default Value
$(momentum_names)_constructor : The name of the solver which creates the mesh or grid for the corresponding momentum_name	
Parallelizer : Name of the Propagation parallelizer solver	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).

Parameter : Description	Default Value
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Sets the active regions in this simulation domain.	
boundary_regions : required regions as boundary conditions	(()).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
extra_domain_name_element : Suffix to the name of newly created subdomains	().
lead_domain : Name of lead domain for which a surface Green solver is defined. This name will be used to obtain the surface Green's function for the lead.	
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_skipped_domains_at_the_end : This option sets how many subdomains the Forward RGF is skipping at the end of the iterations.	(0).

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
potential_solver : This solver (usually a poisson solver) generates the external potential to be added to the Hamiltonian. Required when possion solver is needed in the lead.	().
result_propagator_name : This is the name of the propagator object which contains G<	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : Name of the Hamiltonian constructor for the target domain.	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.52: Parameters of the ForwardRGFModule Solver.

5.53 ForwardRGFSolver

This solver is the library-kind version of the Forward RGF

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).

Parameter : Description	Default Value
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false)
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().

Parameter : Description	Default Value
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).

Parameter : Description	Default Value
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	((()).

Parameter : Description	Default Value
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(().
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).

Parameter : Description	Default Value
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().

Parameter : Description	Default Value
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).

Parameter : Description	Default Value
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.53: Parameters of the ForwardRGFSolver Solver.

5.54 ForwardRGFWrapperModule

This class wraps the MKL optimized forwardRGF to meet common interfaces

Parameter : Description	Default Value
OPTION_NAME : OPTION_DESCRIPTION	(DEFAULT_VALUE).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).

Parameter : Description	Default Value
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.54: Parameters of the ForwardRGFWrapperModule Solver.

5.55 GeneralLeadGreenModule

[MODULE] Forward Retarded Green's Function Module Solver

Parameter : Description	Default Value
$(momentum_names)_constructor : The solver that should provide the $(momentum_name). Some typical examples for transport calculations are the parallelizer for homogenous grid or adaptive grid solver for adaptive grid	
Parallelizer : The simulation that is to provide the energy and possibly k parallelization. In most cases it is of type Green_Propagation and is done within the Propagation network	
_disable_init : flag command to disable initialization	(false).

Parameter : Description	Default Value
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : The regions are to be activated for this simulation. This is a list of region numbers defined in the geometry section.	
boundary__regions : required regions as boundary conditions	(()).
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
extra__domain__name__element : A string appended to the name of the repartitioned domains e.g. test would give test_0, test_1 appended to new domain name	().
momentum__names : A vector of strings for 'momenta' , i.e. k,E, etc.	

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
potential_solver : This solver (usually a poisson solver) generates the external potential to be added to the Hamiltonian.	().
start_leads : Name of the domain on which the lead of the domain is defined. This is not the device domain but usually source_source_contact or drain_drain_drain_contact	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The simulation that provides the Hamilton constructor corresponding to the domain specified as target_domain	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis :	(The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).).
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.55: Parameters of the GeneralLeadGreenModule Solver.

5.56 GeometryVIS

This simulation outputs surfaces or volumes for all regions in a vtk file

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : filter regions to be visualized	().
boundary__disable__overlap : Disable volumes overlapping to speed up calculations	(false).
boundary__regions : required regions as boundary conditions	(()).
boundary__surfaces__filename : Filename to use for boundary-region surface	(geometry_boundary_surface.vtk).
boundary__surfaces__output : Generate vtk iso surfaces for each boundary region	(false).

Parameter : Description	Default Value
boundary_surfaces_resolution : surfaces mesh resolution in all 3 directions (n,n,n)	(10).
boundary_surfaces_resolution_vector : surface mesh-resolution for each direction (x,y,z)	((10,10,10)).
boundary_volumes_filename : Filename to use for boundary-region volume	(geometry_boundary_volume.vtk).
boundary_volumes_output : Generate vtk volume for each boundary region	(false).
boundary_volumes_resolution : volume mesh resolution in all 3 directions (n,n,n)	(10).
boundary_volumes_resolution_vector : volume mesh-resolution for each direction (x,y,z)	((10,10,10)).
domain : Domain required for this simulation	().
geometry_disable_overlap : Disable regions overlapping to speed up calculations	(false).
geometry_surfaces_filename : Filename to use for region surfaces	(geometry_surface.vtk).
geometry_surfaces_output : Generate vtk iso surfaces for each region	(false).
geometry_surfaces_resolution : surfaces mesh resolution in all 3 directions (n,n,n)	(10).
geometry_surfaces_resolution_vector : surface mesh-resolution for each direction (x,y,z)	((10,10,10)).
geometry_volumes_filename : Filename to use for region volumes	(geometry_volume.vtk).
geometry_volumes_output : Generate vtk volume for each region	(false).
geometry_volumes_resolution : volume mesh resolution in all 3 directions (n,n,n)	(10).
geometry_volumes_resolution_vector : volume mesh-resolution for each direction (x,y,z)	((10,10,10)).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.56: Parameters of the GeometryVIS Solver.

5.57 GreenModule

[MODULE] Retarded Self Consistent Module Solver

Parameter : Description	Default Value
$(momentum_names)_constructor : The name of the solver which creates the mesh or grid for the corresponding momentum_name	
$Hamilton_constructor_(neighbor_domains) : The name of the Schroedinger solver which provides the Hamiltonian for the neighbor_domain	
Parallelizer : This is the name of the solver which creates the momentum (energy, k etc) grid	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().

Parameter : Description	Default Value
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : This is a list of integers defining the regions of the simulation domain where the solver is solved or contain active atoms	
boundary_regions : required regions as boundary conditions	(()).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
neighbor_domains : Names of the neighbor domains. A vector of strings.	(()).
output_file_suffix : suffix to be added on output files	().
potential_solver : This solver (usually a poisson solver) generates the external potential to be added to the Hamiltonian. Required when possion solver is needed in the lead.	().
result_propagator_name : This is the name of the propagator object which contains the final result of the solver, i.e. G<	
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().

Parameter : Description	Default Value
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.57: Parameters of the GreenModule Solver.

5.58 Green_Propagation

Solver to calculate phonon modes, spectra, and related quantities

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(().
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(().
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).

Parameter : Description	Default Value
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().

Parameter : Description	Default Value
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(().

Parameter : Description	Default Value
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).

Parameter : Description	Default Value
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).

Parameter : Description	Default Value
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).

Parameter : Description	Default Value
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).

Parameter : Description	Default Value
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).

Parameter : Description	Default Value
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.58: Parameters of the Green_Propagation Solver.

5.59 HamiltonianUTBgenerator

HamiltonianUTBgeneratorSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
minustransport__transversal__lead : Lead domain along minus transport direction and shifted in transversal direction	
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
periodic__domain : periodic domain for test	
surface__of__regions : surface regions required to calculate boundary conditions	().
tb__basis : tight bindin basis	
tic__toc__name : Prefix for tic-tocs	($(name)).
transport__lead : Lead domain along transport direction	

Parameter : Description	Default Value
transport_transversal_lead : Lead domain along transport direction and shifted in transversal direction	
transversal_lead : Lead domain along transversal direction	

Table 5.59: Parameters of the HamiltonianUTBgenerator Solver.

5.60 Huckel_RGF

Module to compute the Transmission and density base on extended Huckel method

Parameter : Description	Default Value
Emax : Maximum value for the energy grid	(1.0).
Emin : Minimum value for the energy grid	(0.0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).

Parameter : Description	Default Value
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
automatic_energy_window : Energy windows based on conduction band and fermi levels	(false).
boundary_regions : required regions as boundary conditions	(()).
coupling_based_on :	(In Huckel method the coupling is based on interaction).
debug_output : Do debugging output?	(false).
domain : Domain required for this simulation	().
e_points : Number of energy points to use	(1).
end_lead : List of the names of the leads domains that define the end for the RGF	(tmp).
huckel_parameter : parameters for Extended Huckel method	
integration_order : Integration order over the k-mesh	(1).
k_space_integration_factor : Ratio between the full Brillouin zone size and considered part of it. Positive integer number	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).

Parameter : Description	Default Value
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().
sancho_rubio_iterations : Number of Sancho-Rubio algorithm iterations to calculate leads' selfenergies	(50.0).
simulation_eta : Eta for Sancho-Rubio and RGF, recommended vauel is 1e-9	(1e-9).
start_lead : List of the names of the leads domains that create a start for the RGF	
strategy : Strategy for layer partitioning	(bfs).
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis :	(The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).).
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.60: Parameters of the Huckel_RGF Solver.

5.61 IV

Propagation IV

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).

Parameter : Description	Default Value
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
contact : Name of boundary condition (contact) that IV ramper will iterate over	
contact1 : First extra contact to iterate over, other than the one defined by "contact" option	().
contact2 : Second extra contact to iterate over, other than the one defined by "contact" option	().
current__conversion__factor : Current will be multiplied by this factor. Currently used to account for spin degeneracy.	(1.0).
current__solver : Solver that solves the system's current; may be a transport solver	

Parameter : Description	Default Value
device_type : Device type. Can be one of wire, UTB, quasi_1D. Currently needed in order to print the correct units of current in output files.	(wire).
domain : Domain required for this simulation	().
external_output : Vector of solvers that must output. Must be defined if called from IV ramper	(()).
external_solve : Simulations which IV ramper must trigger	(()).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
poisson_solver : Solver that solves the Poisson equation	
prediction : If true this solver will extrapolate the next potential guess, based on previous solutions, and pass it to the solver that solves the nonlinear Poisson equation.	(false).
re_solve_for_current : true	(Do an additional (to Poisson do_solve) call to current_solver's do_solve before calling the get_data for current. This is a hack for Gordon Bell Prize 2014 with optimized RGF).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
voltage : Vector of voltages to iterate over. Voltages are applied to contact defined by "contact" option	

Parameter : Description	Default Value
voltage1 : Vector of voltages applied to "contact1"	().
voltage2 : Vector of voltages applied to "contact2"	().

Table 5.61: Parameters of the IV Solver.

5.62 Interpolator

This solver interpolates Propagator matrices from an old mesh onto the new mesh.

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
interpolated_propagation_solver : This is the Propagation that has the new Propagator map and the output job list	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
source_propagation_solver : This is the Propagation that has the old Propagator map and the output job list	
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.62: Parameters of the Interpolator Solver.

5.63 KeatingStrain

KeatingStrainSolver

Parameter : Description	Default Value
: If true then the strain tensor components are calculated according to the method of Pryor and Zunger as a post-processing step.	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
absolute__tol : Absolute tolerance convergence criterion of the Newton iteration.	().
active__regions : required regions to be activated for this solver	(()).
anharm__cutoff : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
anharm__smoothing : (anharmonic Areshkin only) dimensionless numerical smoothing parameter.	().
anharm__theta__limit : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
boundary__regions : required regions as boundary conditions	(()).
calculate__epsilon : bool, if true, calculate epsilon.	(false).
coulomb__cutoff__radius : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interaction is neglected. This parameter influences thesparsity of thematrix and memory consumption.	().

Parameter : Description	Default Value
coulomb_ewald_Kcut : (coulomb only) (3D-periodic (bulk) simulations only) integer cutoff outside which the long-range contribution is neglected,in units of reciprocal lattice vectors.	().
coulomb_ewald_Rcut : (coulomb only) (3D-periodic (bulk) simulations only) cutoff in real-space	().
domain : Domain required for this simulation	().
ewald_sigma : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interactionis neglected. This parameter influences the sparsity of thematrix and memory consumption.	().
hessian_info : If true then some numerical debug terminal output is created.	().
linear_solver : Which linear solver is employed in the Newton iteration. See the PETSc manual for possible choices - gmres is preferred.	().
linsolver_monitor : If true then the convergence of the iterative linear solver is monitored as terminal output.	().
linsolvers_max_iters : Maximum number of linear solver iterations (in the case of an iterative solver) for each Newton step.	().
max_num_iters : Maximum number of Newton iterations.	().
models : A list that may contain the following: harmonic or anharmonic Lazarenkova or anharmonic Areshkin or anharmonic Sui; stretch-bend; cross-stretch; coplanar-2ndNN; coulomb.	().
modify_ideal_length : (coulomb only) If set to true, then the parameter d which in the absence of Coulomb interaction has the meaning of an ideal bond length r0 is modified such that the enrgy functional retains its minimum at r0 .	().

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
num_harmonic_iters : Taking the harmonic model while the Newton iteration is far away from the solution improves convergence and speeds up the simulation. This number specifies for how many iterations this is done.	().
output_file_suffix : suffix to be added on output files	().
phonon_solver : The name of the Phonons simulation object, if present. Existence of this option enables the set-up of the dynamical matrix.	().
preconditioner : Which preconditioner is employed in the Newton iteration. See the PETSc manual for possible choices - asm is preferred.lu does not work for simulations with grid-parallelization.	().
relative_tol : Relative tolerance convergence criterion of the Newton iteration.	().
silo_file : When set then the atomic displacement and possibly the strain tensor are saved to a Silo file.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
test_mat_vec_mult : Debug routine to test matrix vector multiplication using the Hessian of the strain simulation.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
vtk_file : When set then the atomic displacement and possibly the strain tensor are saved to a VTK file. Works only for simulations without grid parallelization.	().
vtk_storage : Choose from ascii or binary.	().

Table 5.63: Parameters of the KeatingStrain Solver.

5.64 LRAModule

LRAModuleSolver

Parameter : Description	Default Value
ncv : This is used by the Krylov iterative solvers. It defines the size of the Krylov subspace	().
precond_type : This option controls how the eigenvalue problem is tranformed before solution by the solver_type. Possible options are 'lu', 'mumps'. Other options exist and an exhaustive list can be found in SLEPc documentation	().
ratio_of_eigenvalues_to_be_solved : This is used to reduce the source matrix by creating a Transformation matrix based on this value. The solver will only solve the fraction of the total eigenvalues based on this option. A value of 1 will solve all eigenvalues of the source matrix.	().
solver_type : Used to choose which eigensolver type to use. For an exhaustive list of options see SLEPc documentation.	(lapack).
transform_type : Main choices are none, shift or sinvert. These transforms are used by the eigensolver to shift the eigenvalue problem to a zero point near the value given by shift.	(none).

Table 5.64: Parameters of the LRAModule Solver.

5.65 LRARGFModule

[MODULE] Implements Recursive Green's Function algorithm with low rank approximation

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).

Parameter : Description	Default Value
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
debug__output : If set to true, produces outputs from Repartition solvers in device and leads	(false).
domain : Domain required for this simulation	().
drain__contact : Name of drain contact of simulated device	
end__leads : name of the lead that will be coupled to the last partition of device	(end_leads).
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
source_contact : Name of source contact of simulated device	
start_leads : name of the lead that will be coupled to partition 0 of device	(start_leads).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.65: Parameters of the LRARGFModule Solver.

5.66 Lanczos

Lanczos Solver.
Lanczos Solver. Don't use this if passivation is off.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).

Parameter : Description	Default Value
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
check_every_steps_number : calculates eigenvalues at every nth iteration	(0).
convergence_method : Currently full is the only option. Full checks whether the eigenvalue is converged with respect to absolute tolerance. The option partial is under development.	().
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
energy_maximum : Lanczos maximum value for eigenvalues	(0).
energy_minimum : Lanczos minimum value for eigenvalues	(0).
init_check_steps_number : If this is set, this value is used to tell Lanczos solver when to first check eigenvalues of the tridiagonal matrix. Its value should be greater than max_num_iterations and its default value is check_steps_every_number.	().
lanczos_iseed : Initial seed for random vector generation for Lanczos algorithm	(0.0).
max_number_iterations : Lanczos maximum number of iterations to be performed	(0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_eigenvalues : Lanczos number of eigenvalues to be calculated	(0).
output_file_suffix : suffix to be added on output files	().
resolution : separation of energies before considered degenerate	(0.0).

Parameter : Description	Default Value
save_wavefunctions : yes or no	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
tic_toc_name : Prefix for tic-tocs	($(name)).
tolerance : used by convergence_method as value to check for convergence	(0.0).
trace_output : If this is set to true then the trace output which in this case is the eigenvalues as a function of iteration number will be outputted to a data file each time the tridiagonal matrix is constructed. The tridiagonal matrix is constructed at every integer multiple of check_every_steps_number.	(false).
tridiag_output : If this is set to true then the tridiagonal output will be printed in Sparse Matrix Matlab format each time the tridiagonal matrix is constructed. The tridiagonalmatrix is constructed at every integer multiple of check_every_steps_number.	(false).
wavefunctions_output_format : like vtk or point3d	().

Table 5.66: Parameters of the Lanczos Solver.

5.67 LocalBandstructureModule

Band Structure of a line that goes through a device. This module considers only atomistic domains

Parameter : Description	Default Value
Dump_Strain : This is a flag with a default false value, if set true, it will dump the files with strainxx, strainyy, and strainzz along each line	

Parameter : Description	Default Value
Dump_unstrained_bandstructure : This is a flag with a default false value, if set true, it will dump the untrained bandedges	
Testing_mode : A flag that is by default is false = false, used by developers for testing purposes	
artificial_domain_init_coupling : A flag that is by default is true, used by developers for debuging a bug related to the artificial domain	
deformation_potential : This is a flag that enables a calculation of the Local band structure by the theory of deformation potentials, the default value of this flag is false, if you set it to true, then you need to make sure that all the deformation potential constants are entered for the material.	
p1 : A list of the first points along the lines over which the local band structures are calculated	
p2 : A list of the last points along the lines over which the local band structures are calculated, number of points in p2 should be equal to the number of points in p1	
periodic_unit_cell : This is a flag that enables the default way of calculations, using sample unit cells along the line with periodic boundary conditions, the default value of this flag is true	
r : The raduis of the cylinder surrounding the line, any atom inside this cyliner is considered along the line, has a default value of 0.18 nm	
small_schrodinger : Schrodinger solver name used for internal local bandstructure calculation	(small_schrodinger).
strain_sim : Name of the strain simulation, important only if deformation_potential = true	

Table 5.67: Parameters of the LocalBandstructureModule Solver.

5.68 MCBoltzmann

Monte Carlo transport (algorithm is taken in part from Tomizawa).

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
atoms__rate : a values between 0 and 1 that represent atoms rate into the system	(0.0).
boundary__regions : required regions as boundary conditions	(()).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	

Parameter : Description	Default Value
load_mesh : File you want to provide the mesh information from	
load_tetrahedra_neighbors_table : Name of the tetrahedra neighbor table if it is not named as tetrahedra_neighbors_table.dat	().
n_doping_density_0 : read doping concentration in 1/cm3	().
n_doping_density_1 : read doping concentration in 1/cm3	().
n_doping_density_10 : read doping concentration in 1/cm3	().
n_doping_density_11 : read doping concentration in 1/cm3	().
n_doping_density_12 : read doping concentration in 1/cm3	().
n_doping_density_13 : read doping concentration in 1/cm3	().
n_doping_density_14 : read doping concentration in 1/cm3	().
n_doping_density_15 : read doping concentration in 1/cm3	().
n_doping_density_16 : read doping concentration in 1/cm3	().
n_doping_density_17 : read doping concentration in 1/cm3	().
n_doping_density_18 : read doping concentration in 1/cm3	().
n_doping_density_19 : read doping concentration in 1/cm3	().
n_doping_density_2 : read doping concentration in 1/cm3	().
n_doping_density_20 : read doping concentration in 1/cm3	().
n_doping_density_21 : read doping concentration in 1/cm3	().

Parameter : Description	Default Value
n_doping_density_22 : read doping concentration in 1/cm3	().
n_doping_density_23 : read doping concentration in 1/cm3	().
n_doping_density_24 : read doping concentration in 1/cm3	().
n_doping_density_25 : read doping concentration in 1/cm3	().
n_doping_density_26 : read doping concentration in 1/cm3	().
n_doping_density_27 : read doping concentration in 1/cm3	().
n_doping_density_28 : read doping concentration in 1/cm3	().
n_doping_density_29 : read doping concentration in 1/cm3	().
n_doping_density_3 : read doping concentration in 1/cm3	().
n_doping_density_30 : read doping concentration in 1/cm3	().
n_doping_density_31 : read doping concentration in 1/cm3	().
n_doping_density_32 : read doping concentration in 1/cm3	().
n_doping_density_33 : read doping concentration in 1/cm3	().
n_doping_density_34 : read doping concentration in 1/cm3	().
n_doping_density_35 : read doping concentration in 1/cm3	().
n_doping_density_36 : read doping concentration in 1/cm3	().
n_doping_density_37 : read doping concentration in 1/cm3	().
n_doping_density_38 : read doping concentration in 1/cm3	().

Parameter : Description	Default Value
n_doping_density_39 : read doping concentration in 1/cm3	().
n_doping_density_4 : read doping concentration in 1/cm3	().
n_doping_density_40 : read doping concentration in 1/cm3	().
n_doping_density_41 : read doping concentration in 1/cm3	().
n_doping_density_42 : read doping concentration in 1/cm3	().
n_doping_density_43 : read doping concentration in 1/cm3	().
n_doping_density_44 : read doping concentration in 1/cm3	().
n_doping_density_45 : read doping concentration in 1/cm3	().
n_doping_density_46 : read doping concentration in 1/cm3	().
n_doping_density_47 : read doping concentration in 1/cm3	().
n_doping_density_48 : read doping concentration in 1/cm3	().
n_doping_density_49 : read doping concentration in 1/cm3	().
n_doping_density_5 : read doping concentration in 1/cm3	().
n_doping_density_50 : read doping concentration in 1/cm3	().
n_doping_density_51 : read doping concentration in 1/cm3	().
n_doping_density_52 : read doping concentration in 1/cm3	().
n_doping_density_53 : read doping concentration in 1/cm3	().
n_doping_density_54 : read doping concentration in 1/cm3	().

Parameter : Description	Default Value
n_doping_density_55 : read doping concentration in 1/cm3	().
n_doping_density_56 : read doping concentration in 1/cm3	().
n_doping_density_57 : read doping concentration in 1/cm3	().
n_doping_density_58 : read doping concentration in 1/cm3	().
n_doping_density_59 : read doping concentration in 1/cm3	().
n_doping_density_6 : read doping concentration in 1/cm3	().
n_doping_density_60 : read doping concentration in 1/cm3	().
n_doping_density_61 : read doping concentration in 1/cm3	().
n_doping_density_62 : read doping concentration in 1/cm3	().
n_doping_density_63 : read doping concentration in 1/cm3	().
n_doping_density_7 : read doping concentration in 1/cm3	().
n_doping_density_8 : read doping concentration in 1/cm3	().
n_doping_density_9 : read doping concentration in 1/cm3	().
n_doping_max_0 : read the maximum vertex	().
n_doping_max_1 : read the maximum vertex	().
n_doping_max_10 : read the maximum vertex	().
n_doping_max_11 : read the maximum vertex	().
n_doping_max_12 : read the maximum vertex	().
n_doping_max_13 : read the maximum vertex	().
n_doping_max_14 : read the maximum vertex	().
n_doping_max_15 : read the maximum vertex	().
n_doping_max_16 : read the maximum vertex	().

Parameter : Description	Default Value
n_doping_max_17 : read the maximum vertex	().
n_doping_max_18 : read the maximum vertex	().
n_doping_max_19 : read the maximum vertex	().
n_doping_max_2 : read the maximum vertex	().
n_doping_max_20 : read the maximum vertex	().
n_doping_max_21 : read the maximum vertex	().
n_doping_max_22 : read the maximum vertex	().
n_doping_max_23 : read the maximum vertex	().
n_doping_max_24 : read the maximum vertex	().
n_doping_max_25 : read the maximum vertex	().
n_doping_max_26 : read the maximum vertex	().
n_doping_max_27 : read the maximum vertex	().
n_doping_max_28 : read the maximum vertex	().
n_doping_max_29 : read the maximum vertex	().
n_doping_max_3 : read the maximum vertex	().
n_doping_max_30 : read the maximum vertex	().
n_doping_max_31 : read the maximum vertex	().
n_doping_max_32 : read the maximum vertex	().
n_doping_max_33 : read the maximum vertex	().
n_doping_max_34 : read the maximum vertex	().
n_doping_max_35 : read the maximum vertex	().
n_doping_max_36 : read the maximum vertex	().
n_doping_max_37 : read the maximum vertex	().
n_doping_max_38 : read the maximum vertex	().
n_doping_max_39 : read the maximum vertex	().
n_doping_max_4 : read the maximum vertex	().
n_doping_max_40 : read the maximum vertex	().
n_doping_max_41 : read the maximum vertex	().
n_doping_max_42 : read the maximum vertex	().
n_doping_max_43 : read the maximum vertex	().
n_doping_max_44 : read the maximum vertex	().
n_doping_max_45 : read the maximum vertex	().
n_doping_max_46 : read the maximum vertex	().

Parameter : Description	Default Value
n_doping_max_47 : read the maximum vertex	().
n_doping_max_48 : read the maximum vertex	().
n_doping_max_49 : read the maximum vertex	().
n_doping_max_5 : read the maximum vertex	().
n_doping_max_50 : read the maximum vertex	().
n_doping_max_51 : read the maximum vertex	().
n_doping_max_52 : read the maximum vertex	().
n_doping_max_53 : read the maximum vertex	().
n_doping_max_54 : read the maximum vertex	().
n_doping_max_55 : read the maximum vertex	().
n_doping_max_56 : read the maximum vertex	().
n_doping_max_57 : read the maximum vertex	().
n_doping_max_58 : read the maximum vertex	().
n_doping_max_59 : read the maximum vertex	().
n_doping_max_6 : read the maximum vertex	().
n_doping_max_60 : read the maximum vertex	().
n_doping_max_61 : read the maximum vertex	().
n_doping_max_62 : read the maximum vertex	().
n_doping_max_63 : read the maximum vertex	().
n_doping_max_7 : read the maximum vertex	().
n_doping_max_8 : read the maximum vertex	().
n_doping_max_9 : read the maximum vertex	().
n_doping_min_0 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_1 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_10 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_11 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_12 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_13 : parse the minimum vertex and put the values into min_coord	().

Parameter : Description	Default Value
n_doping_min_14 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_15 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_16 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_17 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_18 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_19 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_2 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_20 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_21 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_22 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_23 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_24 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_25 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_26 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_27 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_28 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_29 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_3 : parse the minimum vertex and put the values into min_coord	().

Parameter : Description	Default Value
n_doping_min_30 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_31 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_32 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_33 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_34 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_35 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_36 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_37 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_38 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_39 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_4 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_40 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_41 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_42 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_43 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_44 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_45 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_46 : parse the minimum vertex and put the values into min_coord	().

Parameter : Description	Default Value
n_doping_min_47 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_48 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_49 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_5 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_50 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_51 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_52 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_53 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_54 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_55 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_56 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_57 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_58 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_59 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_6 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_60 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_61 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_62 : parse the minimum vertex and put the values into min_coord	().

Parameter : Description	Default Value
n_doping_min_63 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_7 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_8 : parse the minimum vertex and put the values into min_coord	().
n_doping_min_9 : parse the minimum vertex and put the values into min_coord	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
poisson_solver : Name of the simulation that provides Poisson solver.	
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.68: Parameters of the MCBoltzmann Solver.

5.69 MCbulk

This simulation solves the stationary continuity equation $\nabla \quad \mathbf{J} = G \quad R$, where the current density **J** is expressed by the combination of drift and a diffusion term.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
acoustic__phonons__scattering : options are: yes or no	
active__regions : required regions to be activated for this solver	(()).
applied__electric__field__iterations : Valid only when initial_applied_electric_field, final_applied_electric_field and, applied_electric_field_iterations are all set values. The solver will calculate bulk MC at each Efield = iEfield : step : fEfield, Unit is Volt/meter	(0.0).
boundary__regions : required regions as boundary conditions	(()).
conduction__band__type : options are: parabolic, kane or full; please note that, if for parabolic & kane, number_of_conduction_bands must be 1	
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
electron__energy__step : electron energy step	
energy__file : input file for energy, it should be a directory string.	

Parameter : Description	Default Value
final_applied_electric_field : Valid only when initial_applied_electric_field, final_applied_electric_field and, applied_electric_field_iterations are all set values. The solver will calculate bulk MC at each Efield = iEfield : step : fEfield, Unit is Volt/meter	
initial_applied_electric_field : Valid only when initial_applied_electric_field, final_applied_electric_field and, applied_electric_field_iterations are all set values. The solver will calculate bulk MC at each Efield = iEfield : step : fEfield, Unit is Volt/meter	
k_points_file : input file for k-points, it should be a directory string.	
lattice_temperature : default is 0.	
material : Only Si is implemented for MCbulk, and Only Si in all.mat has group MC parameters	().
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_conduction_bands : for conduction_band_type=parabolic or kane, then it must be 1. Note: it can't be greater than default MAX_NUMBER_OF_CONDUCTION_BANDS=16.	
number_of_energy_steps : Note: it can't be greater than default MAX_NUMBER_OF_ENERGY_STEPS=4096	
number_of_iterations : The number of iterations for the loop containing drift(), scattering() and average()	
optical_phonons_scattering : options are: yes or no	
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
reorder_conduction_bands : default is false, when it is true, it will reorder the conduction bands according to the slopes of the curves. Step1: from bottom of valley to the right; Step2: from bottom of valley to the left.	
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.69: Parameters of the MCbulk Solver.

5.70 MatrixElements

This simulation calculates matrix elements between quantum states as a post-processing step. It relies on another simulation which calculates the wavefunctions and is specified using the parameter wf_simulation

The general idea is that given two states $|\ _1\rangle$ and $|\ _2\rangle$, the calculation of

$$|\langle\ _2|\hat{A}|\ _1\rangle|^2 \qquad |\langle\ _2||\tilde{\ }\ \rangle|^2 \tag{5.1}$$

looks similar for all operators $\hat{A}$. All that has to be implemented is the calculation of $|\tilde{\ }\ \rangle\ \ \hat{A}|\ _1\rangle$.

Parameter : Description	Default Value
Append_Solver_Name_to_files : A flag when set to true, it will append the solver name to the file names, which is useful if you want to have more than matrix element solver in the same simulation	(false).
Dont_add_Emin_to_Ef : Without setting this flag true, the minimum energy obtained by shrodinger will be be considered the reference for Ef and will be added to it	(false).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
absorption : If true, an absortion plot alpha(hw) will be generated	(false).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
ef : The first entry determines the Fermilevel; subsequent values are the sweep values if sweep=ef	(()).
egrid__size : The number of plot energy (hw) grid points	(100).
gamma : Linewidth of the Lorentzian broadening of the transitions, in eV.	(0.01).
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
operator : Determines the type of matrix element to calculate. One of overlap or optical	
orbital_contribution_WF : If true, an output with percentage contribution of each orbital for each eigenfunction will be generated	(false).
output_file : The base name for output files	(matrix_elements.txt).
output_file_suffix : suffix to be added on output files	().
override : If true, then the right wavefunction is overwritten during matrix element calculation	(false).
polarization_phis : Polarization angle phi which can be swept	().
polarization_thetas : Polarization angle theta which can be swept	().
surface_of_regions : surface regions required to calculate boundary conditions	().
sweep : Depending on its value (angle, temp ,ef), a family of absorption plots is generated by sweeping either the polarization angle, the temperature or the Fermilevel.	(undefined).
sweep_values : Number of values of the sweep quantity	(1).
temp : The first entry determines the temperature; if sweep=temp, all subsequent values are the sweep values.	().
tic_toc_name : Name for advanced timing output	($(name)).
transition : If true, a plot with different energy transitions will be generated	(false).
wf_simulation : The name of the simulation from which eigenfunctions and eigenenergies are obtained (e.g. a Schroedinger simulation).	

Table 5.70: Parameters of the MatrixElements Solver.

5.71 MatrixTest1

[TEST] Solver to test matrix multiplications

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
dev__size__increments : This option defines the number of devices/matrices to benchmark	(1).
domain : Domain required for this simulation	().
hermitian__loops : This option defines the number of hermitian transposes to be calculated	(100).

Parameter : Description	Default Value
invert_loops : This option defines the number of matrix inversions to be done in the test	(100).
matrix_source : Name of the source Hamiltonian simulation object defined in the input deck. The test matrix will be obtained from this Hamiltonian simulation.	
mult_loops : This option defines the number of matrix multiplications to be done in the test	(100).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
submatrix_loops : This option defines the number of sub-matrices to extract	(100).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.71: Parameters of the MatrixTest1 Solver.

5.72 Meta55

Meta55Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).

Parameter : Description	Default Value
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).

Parameter : Description	Default Value
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined__dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact__aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence__limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).

Parameter : Description	Default Value
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).

Parameter : Description	Default Value
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).

Parameter : Description	Default Value
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).

Parameter : Description	Default Value
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).

Parameter : Description	Default Value
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).

Parameter : Description	Default Value
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.72: Parameters of the Meta55 Solver.

5.73 Meta5Schottky

Meta5SchottkySolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
consistent_modes : Make transfer matrix modes phases and ordering consistent between different runs. For more details check description of the same option in Propagation	(false).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
number_of_subdomains : number of subdomains for device repartitioner	(1).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
use_scaling_law : use scaling law for Hamiltonian construction or not e.g. wurtzite	(false).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Parameter : Description	Default Value

Table 5.73: Parameters of the Meta5Schottky Solver.

5.74 Meta5_schottky

Meta5_schottkySolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).

Parameter : Description	Default Value
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).

Parameter : Description	Default Value
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
consistent_modes : Make transfer matrix modes phases and ordering consistent between different runs. For more details check description of the same option in Propagation	(false).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).

Parameter : Description	Default Value
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().

Parameter : Description	Default Value
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).

Parameter : Description	Default Value
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).

Parameter : Description	Default Value
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().

Parameter : Description	Default Value
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().

Parameter : Description	Default Value
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).

Parameter : Description	Default Value
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).

Parameter : Description	Default Value
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
number_of_subdomains : number of subdomains for device repartitioner	(1).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	

Parameter : Description	Default Value
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).

Parameter : Description	Default Value
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).

Parameter : Description	Default Value
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
use_scaling_law : use scaling law for Hamiltonian construction or not e.g. wurtzite	(false).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.74: Parameters of the Meta5_schottky Solver.

5.75 MetaBellQTBM

MetaBellQTBMSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).

Parameter : Description	Default Value
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().

Parameter : Description	Default Value
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().

Parameter : Description	Default Value
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).

Parameter : Description	Default Value
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).

Parameter : Description	Default Value
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).

Parameter : Description	Default Value
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).

Parameter : Description	Default Value
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.75: Parameters of the MetaBellQTBM Solver.

5.76 MetaBellQTBM2

MetaBellQTBM2Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).

Parameter : Description	Default Value
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).

Parameter : Description	Default Value
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(().
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(().
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact__aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence__limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug__output : A catch-all option for output that is not necessary unless debugging.	(false).
debug__output__job__list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy__factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).

Parameter : Description	Default Value
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).

Parameter : Description	Default Value
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).

Parameter : Description	Default Value
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	

Parameter : Description	Default Value
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).

Parameter : Description	Default Value
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).

Parameter : Description	Default Value
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.76: Parameters of the MetaBellQTBM2 Solver.

5.77 MetaDummy

MetaDummySolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).

Parameter : Description	Default Value
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).

Parameter : Description	Default Value
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).

Parameter : Description	Default Value
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().

Parameter : Description	Default Value
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).

Parameter : Description	Default Value
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().

Parameter : Description	Default Value
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).

Parameter : Description	Default Value
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.77: Parameters of the MetaDummy Solver.

5.78 MetaLinearDropQTBM

MetaLinearDropQTBMSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).

Parameter : Description	Default Value
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(().
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(().

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().

Parameter : Description	Default Value
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().

Parameter : Description	Default Value
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).

Parameter : Description	Default Value
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).

Parameter : Description	Default Value
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	

Parameter : Description	Default Value
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).

Parameter : Description	Default Value
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.78: Parameters of the MetaLinearDropQTBM Solver.

5.79 MetaOmenQTBM

MetaOmenQTBMSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().

Parameter : Description	Default Value
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact__aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence__limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug__output : A catch-all option for output that is not necessary unless debugging.	(false).
debug__output__job__list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).

Parameter : Description	Default Value
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).

Parameter : Description	Default Value
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).

Parameter : Description	Default Value
fem__source__source__source__contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate__voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous__initial__potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
iteration__output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear__solver__for__bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material__semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).

Parameter : Description	Default Value
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	

Parameter : Description	Default Value
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).

Parameter : Description	Default Value
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).

Parameter : Description	Default Value
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.79: Parameters of the MetaOmenQTBM Solver.

5.80 MetaOmenTransport

MetaOmenTransportSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).

Parameter : Description	Default Value
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).

Parameter : Description	Default Value
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).

Parameter : Description	Default Value
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().

Parameter : Description	Default Value
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).

Parameter : Description	Default Value
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
output_file_suffix : suffix to be added on output files	().
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().

Parameter : Description	Default Value
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).

Parameter : Description	Default Value
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.80: Parameters of the MetaOmenTransport Solver.

5.81 MetaOxide

MetaOxideSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).

Parameter : Description	Default Value
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
adaptive__grid__strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).

Parameter : Description	Default Value
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).

Parameter : Description	Default Value
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).

Parameter : Description	Default Value
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().

Parameter : Description	Default Value
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).

Parameter : Description	Default Value
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).

Parameter : Description	Default Value
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).

Parameter : Description	Default Value
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).

Parameter : Description	Default Value
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).

Parameter : Description	Default Value
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).

Parameter : Description	Default Value
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).

Parameter : Description	Default Value
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.81: Parameters of the MetaOxide Solver.

5.82 MetaPoissonQTBM2

MetaPoissonQTBM2Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.82: Parameters of the MetaPoissonQTBM2 Solver.

5.83 MetaPoissonQTBM3

MetaPoissonQTBM3Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.83: Parameters of the MetaPoissonQTBM3 Solver.

5.84 MetaPoissonQTBM4

MetaPoissonQTBM4Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.84: Parameters of the MetaPoissonQTBM4 Solver.

5.85 MetaPoissonQTBM4UTB

MetaPoissonQTBM4UTBSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Parameter : Description	Default Value

Table 5.85: Parameters of the MetaPoissonQTBM4UTB Solver.

5.86 MetaPoissonQTBM5

MetaPoissonQTBM5Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).

Parameter : Description	Default Value
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).

Parameter : Description	Default Value
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
consistent_modes : Make transfer matrix modes phases and ordering consistent between different runs. For more details check description of the same option in Propagation	(false).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).

Parameter : Description	Default Value
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().

Parameter : Description	Default Value
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The drain contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).

Parameter : Description	Default Value
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).

Parameter : Description	Default Value
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().

Parameter : Description	Default Value
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().

Parameter : Description	Default Value
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).

Parameter : Description	Default Value
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).

Parameter : Description	Default Value
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
number_of_subdomains : number of subdomains for device repartitioner	(1).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).

Parameter : Description	Default Value
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	

Parameter : Description	Default Value
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : Change the number of adaptive grid points when adaptive_grid_strategy = omen. x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).

Parameter : Description	Default Value
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The source contact temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).

Parameter : Description	Default Value
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
transport_direction_cartesian : Vector of transport direction indices within the spatial cartesian coordinate system in the leads. 0 corresponds to x, 1 to y, and 2 to z. Each vector element is for one lead. Currently both leads must be equal. This is used to tell the adaptive grid solver the wave vector component that corresponds to the transport direction in the lead. It is necessary since the transport direction in the lead where the band structure is calculated is periodic and there can be multiple periodic directions, only one of which is transport direction.	((0,0)).
transport_direction_crystal : Vector of transport direction indices within the crystal coordinate system in the leads. Similar to transport_direction_cartesian, but this option specifies transport direction index within the crystal coordinate system. Currently both leads must be equal.	(transport_direction_cartesian).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
use_scaling_law : use scaling law for Hamiltonian construction or not e.g. wurtzite	(false).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.86: Parameters of the MetaPoissonQTBM5 Solver.

5.87 MetaPoissonQTBM5_2top_gates

MetaPoissonQTBM5_2top_gatesSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Ef_approximation : Spatial shape of the Fermi level in the device. This option must be used in conjunction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average. For more details look for the description of this option in Semiclassical.cpp.	(x_linear_drop).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
adaptive_grid_strategy : Method to create adaptive grid given the band structure. Two options possible: omen (like in OMEN code) and adaptive (using this option the total number of energy points can be given as input).	(omen).

Parameter : Description	Default Value
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
bandstructure_smart_parallelization : Allows idle cores in band structure calculation associated with adaptive grid, if the number of cores is larger than the number of k points. Prevents accidental code slow down and bad scaling.	(false).
boundary_regions : required regions as boundary conditions	(().
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
confined_dir : Tells which is confined direction in the channel region for UTB type devices, so that gate boundary can be applied properly in non-planar geometries.	(y).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).

Parameter : Description	Default Value
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).

Parameter : Description	Default Value
do_output_nonlinearpoisson_potential : control poisson node_potential output.	(true).
domain : Domain required for this simulation	().
drain_boundary_region : It denotes the boundary region for drain defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_temperature : The temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial energy threshold. It can be a sharp transition at the threshold or a smooth transition around the threshold, like in OMEN code.	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).

Parameter : Description	Default Value
energy_cutoff_in_kT : Energy cutoff in units of kT applied due to a drop in Fermi distribution for high energies. Option EOffset has precedence if both EOffset and energy_cutoff_in_kT are defined.	(25).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points :	(When adaptive_grid_strategy=omen this option is used for debugging since it allows to cut off the (potentially large) adaptive energy grid to contain only the first n points. When adaptive_grid_strategy=adaptive this option is the total number of energy points in adaptive grid (since with adaptive strategy one can define the total number of energy points in advance).).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_boundary_region : It denotes the boundary region for gate defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
homogeneous_initial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(1.5).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
initial_percent_of_points : Ratio of the total number of energy points that will be used to construct the initial homogeneous energy grid when adaptive_grid_strategy=adaptive. The rest of the points will be constructed inhomogeneously around important features in the energy spectrum (like DOS peaks).	(0.8).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
k_rotational_symmetry : Used in k-mesh construction for whether to use 1D kspace that assumes rotational symmetry (such as effective mass)	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().

Parameter : Description	Default Value
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : Tag of the channel semiconductor material. This material is used to determine the channel work function in order to calculate the actual potential applied to the surface of the oxide (since the actual gate contact is not part of simulation).	(chunk_of_Silicon).
material_semiconductor_name : Defines a list of materials for which confined effective mass, needed for semiclassical solution, is calculated. Only lead materials are possible currently. If the device material is different than the lead material it will be modeled with bulk effective mass. Usage: if both leads have same material this option is equal to that material name. If leads have different materials this option is equal to a list where the first element is the source material and the second element is the drain material.	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).

Parameter : Description	Default Value
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_1D_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(100).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : Define a straight line on which NonlinearPoisson solver interpolates charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
p_noise_potential : Amplitude of the random potential added to the diagonal of the Hamiltonian	(1e-8).
parallelize_adaptive_grid_construction : When Propagation calls AdaptiveGrid to create the grid, it will call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
semiclassical_IV : Solve Poisson/semiclassical charge followed by one-pass QTBM.	(false).
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjunction with Ef_approximation.	(source).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_boundary_region : It denotes the boundary region for source defined in the geometry section. This boundary condition will be applied in poisson solvers, both poisson_device and poisson_device_init	().
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).

Parameter : Description	Default Value
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_temperature : The temperature used in the Schroedinger solver and QTBM in Kelvin.	(300).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.87: Parameters of the MetaPoissonQTBM5_2top_gates Solver.

5.88 MetaPoissonQTBM5_EC

MetaPoissonQTBM5_ECSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact__aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).

Parameter : Description	Default Value
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).

Parameter : Description	Default Value
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).

Parameter : Description	Default Value
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).

Parameter : Description	Default Value
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).

Parameter : Description	Default Value
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).

Parameter : Description	Default Value
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).

Parameter : Description	Default Value
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.88: Parameters of the MetaPoissonQTBM5_EC Solver.

5.89 MetaRGF

MetaRGFSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).

Parameter : Description	Default Value
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(().
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(().
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).

Parameter : Description	Default Value
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).

Parameter : Description	Default Value
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).

Parameter : Description	Default Value
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).

Parameter : Description	Default Value
fem__source__source__source__contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate__voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous__initial__potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
interpolation__power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration__output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear__solver__for__bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).

Parameter : Description	Default Value
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).

Parameter : Description	Default Value
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).

Parameter : Description	Default Value
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.89: Parameters of the MetaRGF Solver.

5.90 MetaSolver

MetaSolverSolver

Parameter : Description	Default Value

Table 5.90: Parameters of the MetaSolver Solver.

5.91 MetafullRGF

MetafullRGFSolver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : Used to let the solver know what is the device region	(()).
average__charge__over__fem__cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).
bands__number__of__nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands__solver__type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary__regions : required regions as boundary conditions	(()).
clean__all__in__reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant__lead__eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact__aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).

Parameter : Description	Default Value
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).

Parameter : Description	Default Value
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).

Parameter : Description	Default Value
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).

Parameter : Description	Default Value
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).

Parameter : Description	Default Value
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).

Parameter : Description	Default Value
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).

Parameter : Description	Default Value
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).

Parameter : Description	Default Value
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).

Parameter : Description	Default Value
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.91: Parameters of the MetafullRGF Solver.

5.92 Metatemp5

Metatemp5Solver

Parameter : Description	Default Value
Dirichlet_nodes_output : If this is true the Dirchlet nodes will be outputted in a .dat files.	(false).
Drain_chemical_potential : The electrochemical potential (in eV) for the drain contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
EOffset : Adaptive energy grid control option for OMEN users.	(25*k*300/q).
Elimit : Adaptive energy grid control option for OMEN users.	(50.e-3).
Emin_tail : Adaptive energy grid control option for OMEN users.	(4.e-3).
MPI_distribution_strategy : Defines the spatial distribution strategy for the repartioner. If solver_type == compression use OMEN.	(OMEN).

Parameter : Description	Default Value
MPI_interleaving : This option enable MPI Interleaving for BC calculation	(false).
Source_chemical_potential : The electrochemical potential (in eV) for the source contact. It is used by QTBM and adaptive grid, unless it is overwritten by Poisson boundary conditions (which happens by default).	(0).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : Used to let the solver know what is the device region	(()).
average_charge_over_fem_cell : This option is used to define how is the atomic charge mapped to the FEM cell (averaged or delta like points).	(true).

Parameter : Description	Default Value
bands_number_of_nodes : The number of k nodes in the transport direction to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	(200).
bands_solver_type : The eigenvalue solver method to be used by the mode contact Schroedinger solvers and the semiclassical Schroedinger (if selfconsistent_algorithm == fast)	(krylovschur).
boundary_regions : required regions as boundary conditions	(()).
clean_all_in_reinit : force to deallocate all matrices when solver is reinitialized	(true).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(0).
contact_aux : The name of the domain to be used by the fast self-consistent algorithm for calculating the Fermi level in the contacts.	(contact_aux).
convergence_limit : convergence limit for iterative solution methods (accuracy of eigenvalues)	(0.0000001).
dE_f : Adaptive energy grid control option for OMEN users.	(4.e-3).
dE_in : Adaptive energy grid control option for OMEN users.	(1.e-3).
dE_sep : Adaptive energy grid control option for OMEN users.	(1.e-4).
debug_output : A catch-all option for output that is not necessary unless debugging.	(false).
debug_output_job_list : A debug output options to output a file that includes the (E,k) tuple load balancing for each MPI rank.	(false).
degeneracy_factor : Sets the degeneracy factor for transmission/current and density calculation in the transport solver. Currently should only be used for transverse k-degeneracy.	(1).

Parameter : Description	Default Value
density_of_states_output : This option controls whether the energy resolved density of states will be written to file.	(false).
density_solver : The Propagator Solver that provides the density for QTBM_Propagation type Solver.	().
derivative_of_density_solver : The Propagator Solver that provides the derivative of density for QTBM_Propagation type Solver.	().
device : Name of the domain associated with the device. This should correspond to the domain in the domain section of the input deck.	(device).
domain : Domain required for this simulation	().
drain_contact : Name of the domain associated with the drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_contact).
drain_drain_contact : Name of the domain associated with the drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_contact).
drain_drain_drain_contact : Name of the domain associated with the drain drain drain contact. This should correspond to the domain in the domain section of the input deck.	(drain_drain_drain_contact).
drain_mode_contact : Name of the domain associated with the drain mode contact. This should correspond to the domain in the domain section of the input deck.	(drain_mode_contact).
drain_voltage : A single value to define the applied drain voltage. To use multiple values use a ramper Solver instead.	(0).
electron_hole_heuristics : Algorithm for distinguishing electrons from holes given the spatial threshold	(sharp).

Parameter : Description	Default Value
electron_hole_model : Use both electrons and holes (with spatial thershold) in QTBM density calculation (eg. for TFET devices)	(false).
energy_grid_constructor : This option is used to set the energy grid constructor. The main use is for telling QTBM what adaptive grid generator is.	().
energy_grid_small_number_of_points : The number of points to actually use in the adaptive energy grid (= n means only first n points will be used from the ordered adaptive grid).	(Not used by default.).
energy_points : The number of energy points to use for a homogenous energy grid.	().
energy_range : The energy range (in eV) to be used by homogenous grid.	().
energy_resolved_current_output : Outputs energy resolved current. Works for wire geometry only.	(true).
energy_resolved_density_output : This option controls whether the energy resolved density will be written to file.	(false).
fem_device : Name of the domain associated with the device finite element method.	(fem_device).
fem_drain_contact : Name of the domain associated with the drain_contact finite element method.	(fem_drain_contact).
fem_drain_drain_contact : Name of the domain associated with the drain_drain_contact finite element method.	(fem_drain_drain_contact).
fem_drain_drain_drain_contact : Name of the domain associated with the drain_drain_drain_contact finite element method.	(fem_drain_drain_drain_contact).
fem_source_contact : Name of the domain associated with the source_contact finite element method.	(fem_source_contact).
fem_source_source_contact : Name of the domain associated with the source_source_contact finite element method.	(fem_source_source_contact).

Parameter : Description	Default Value
fem_source_source_source_contact : Name of the domain associated with the source_source_source_contact finite element method.	(fem_source_source_source_contact
gate_voltage : A single value to define the applied gate voltage. To use multiple values use a ramper Solver instead.	(0).
homogenous_initial_potential : An initial guess for Poisson (when safe mode is used) that should be close to the actual value of the potential. If this value is chosen close to actual this will improve convergence.	(1.5).
interpolation_power : The power used to calculate the weight in inverse distance interpolation	(2).
iteration_output : A catch-all option for intermediate output useful for debugging iterations in Poisson.Some examples of output is Adaptive Grid E,k data and 1D output potential which has potential, density etc. Note: Simulation will take much longer with this option set true. It should only be used when debugging.	(false).
kxmax : if system_type == UTB or system_type == quasi_1D Maximum transverse kx	().
kxmin : if system_type == UTB or system_type == quasi_1D Minimum transverse kx	().
kymax : if system_type == quasi_1D Maximum transverse ky	().
kymin : if system_type == quasi_1D Minimum transverse ky	().
laplacian : Defines the direction in which to solve the linear Poisson in the leads. It is used to propagate device Poisson potential to the leads.	((x)).
linear_solver_for_bc : If the system is a wire with no spin orbit coupling this should be set as true. This will greatly reduce the time the simulation takes.	(false).

Parameter : Description	Default Value
material_semiconductor : This is the tag associated with the semiconductor material defined in the input deck and should match.	(chunk_of_Silicon).
material_semiconductor_name : Should match the name of material in Material section (old geometry definition), or material_name option in Device (metageometry definition). Used for the semiclassical charge solver (initial guess).	(Si).
metal_work_function : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition.	(4.2).
metal_work_function_2 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
metal_work_function_3 : The energy required (in eV) to move an electron from the Fermi Energy to vacuum. It is used by Poisson for the Schottky contact Boundary condition in case of multiple gates.	(4.2).
name : A unique name tag / identifier for the specific instance of the solver.	
no_file_output : This suppresses output files from QTBM	(false).
no_integration_for_transmission : Tells propagation whether to apply an integration weight (based on dimensionality) and integrate transmission.	(false).
noise_potential : Adds a fictitious random noise potential to account for degenerate states in QTBM of the magnitude of this value.	(1e-6).
non_rectangular_energy : This option must be true to define a k-dependent energy mesh. This is necessary for system_type == UTB/quasi_1D	(false).

Parameter : Description	Default Value
number_of_MPI_ranks_in_real_space : The number of processes for Spatial Parallelizing of QTBM	(0).
number_of_eigenvalues : The number of contact bands that the Schroedinger band structure solvers should attempt to find. This option must be high enough to ensure that band solvers find enough converged eigenvalues to satisfy the above option. Otherwise this option does not affect the simulation results.	(150).
number_of_eigenvalues_to_use : The number of contact bands to actually use in the code for constructing the adaptive grid.	(16).
number_of_interpolation_points : The number of the surrounding points that will be considered in the interpolation process.	(1).
number_of_k_points : The number of k nodes in the transverse direction/directions to solve for in the Schroedinger band structure solver for the drain and source mode contact domain.	().
number_of_k_points_for_bandstructure : Number of k-points to be used to genereteate the adaptive energy grid.	(40).
output_file_suffix : suffix to be added on output files	().
output_format_energy_resolved_density : Format for the file output of the energy resolved density (xyz,VTK,DX,Silo)	(VTK).
output_line_corners : To define a line through the device in which to print a poisson_1D_output file that has potential, density etc.	([(0, 1.0862, 1.0862), (20.0, 1.0862, 1.062)]).
particle : Particle type if single particle device.	(electron).
particle_drain : Drain particle type.	(particle).
particle_source : Source particle type.	(particle).

Parameter : Description	Default Value
poisson_max_iterations : Maximum number of iterations to be used by the Poisson algorithm before stopping	(15).
ramper_contact : The name of which contact to apply the ramper_voltage to.	(gate).
ramper_voltage : A list of voltages that are to be applied to the contact defined in ramper_contact	().
ratio_of_LRA : This option is used for a reduced rank matrix method for Transport. It says what ratio of the total eigenvalues to use to transform the source matrix.	(1).
raw_transmission_output : Whether to output Transmission	(false).
regions_adjacent_to_drain_electrode : The region number that is connected to the drain electrode. This allows NEMO5 to know where the drain electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_gate_electrode : The region number that is connected to the gate electrode. This allows NEMO5 to know where the gate electrode is connected and is used for applying boundary conditions in Poisson.	
regions_adjacent_to_source_electrode : The region number that is connected to the source electrode. This allows NEMO5 to know where the source electrode is connected and is used for applying boundary conditions in Poisson.	
residual_criterion : Normalized residual of the Poisson equation. This option and step_size_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
reuse_hamiltonian : If this is true then the base Hamiltonian i.e. without the potential from Poisson will be saved and subsequently loaded when requested. This should save time but uses more memory.	(true).

Parameter : Description	Default Value
scale_inhomogeneous_energy_steps : x means scale energy steps DE as DE*x, i.e. if x<1 the energy grid will be denser, and vice versa.	(1.0).
selfconsistent_algorithm : Algorithm used to solve the nonlinear Poisson equation. Fast uses a semi-classical guess for the first point, previous solution for the second point, and prediction based on previous two solutions for the third and higher points. It also uses a full step size in the Newton method. Safe uses a simple homogenous initial guess for the first point, previous solution as initial guess for other points, and Newton method with protection against divergence.	(safe).
solve_on_single_replica : This should not be changed. Required to avoid communication.	(true).
solver_type : The linear solver to solve the QTBM equation.	(mumps).
source_contact : Name of the domain associated with the source contact. This should correspond to the domain in the domain section of the input deck.	(source_contact).
source_mode_contact : Name of the domain associated with the source mode contact. This should correspond to the domain in the domain section of the input deck.	(source_mode_contact).
source_source_contact : Name of the domain associated with the source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_contact).
source_source_source_contact : Name of the domain associated with the source source source contact. This should correspond to the domain in the domain section of the input deck.	(source_source_source_contact).
source_voltage : A single value to define the applied source voltage. To use multiple values use a ramper Solver instead.	(0.8).

Parameter : Description	Default Value
step_size_criterion : Normalized potential step size between two iterations. This option and residual_criterion have to be satisfied to declare Poisson/transport solver convergence.	(1e-3).
surface_of_regions : surface regions required to calculate boundary conditions	().
system_is_wire_with_no_SO : linear solver used in boundary condition solution while using transfer_matrix method	(mumps).
system_type : Defines the system. It is used to define the transverse kx (if UTB) and the momentum_names. It will define transverse kx and ky if system_type == quasi_1D	(wire).
tb_basis : The tight-binding basis/ band structure model	
temperature : The temperature used in the Schroedinger solver in Kelvin.	(300).
tic_toc_name : Prefix for tic-tocs	($(name)).
use_monomials : Use monomials for TBHamiltonConstructor, set to false when use 2NN.	(true).
with_poisson : This option is used to set whether the simulation should be done with or without self-consistent Poisson.	(false).

Table 5.92: Parameters of the Metatemp5 Solver.

5.93 ModuleDummy

ModuleDummySolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.93: Parameters of the ModuleDummy Solver.

5.94 ModuleDummyConstruct

ModuleDummyConstructSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.94: Parameters of the ModuleDummyConstruct Solver.

5.95 ModuleDummyNested

[MODULE] Module Dummy Nested, simple class for test that creates nested solvers/modules

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
nested_value : Number of nested solver to be created, it decrease by 1 for each new solver created	(0).
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.95: Parameters of the ModuleDummyNested Solver.

5.96 MultiEqNeq

MultiEqNeqSolver

Parameter : Description	Default Value
$(momentum_names)_constructor : The solver that should provide the $(momentum_name). Some typical examples for transport calculations are the parallelizer for homogenous grid or adaptive grid solver for adaptive grid	
Hamilton_constructor : Name of the Hamiltonian constructor for this module.	().
Hamilton_constructor_$(lead_domain) : The simulation type that is to provide the Hamiltonian and DOFmap as required for the $(lead_domain)	
Hamilton_momenta : A list of extra momenta to be used by the Hamiltonian constructor.	().

Parameter : Description	Default Value
Parallelizer : The simulation that is to provide the energy and possibly k parallelization. In most cases it is of type Green_Propagation and is done within the Propagation network	
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(().
__output : list of solver to trigger output command	(().
__reinit : list of solver to trigger reinit command	(().
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(().
__step : list of solver to trigger step command	(().
__trigger__flag : nested execution of solvers,	(4).
active__regions : The regions that are to be activated, if required, and used by the module that inherits PropagationModule.	
boundary__regions : required regions as boundary conditions	(().
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).
destroy__auxiliary__solvers : If true, destroy auxiliary Schroedinger solvers after retarded self energy is obtained through RetardedSelfModule	(false).

Parameter : Description	Default Value
do_output : Prints the propagator if true.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
domain_0 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_1 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_2 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
domain_3 : domain_3 through domain_0 are inside the lead. They are used for calculating self energy using transfer matrix method. Alternatively, repartition solver can be used.	().
lead_domain : The domain that is connected to this PropagationModule. It can be used, for example, to get the coupling Hamiltonian between this domain and the $(lead_domain)	
memory_thin_calculation : If true, the modules will try to deallocate results as soon as possible. If the result is queried again, it needs to be recalculated. If false, the results are kept in memory.	(true).
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
partitioned_lead_domain : Name of the lead domain which was partitioned using domain_[0-3] or repartition solver.	().
partitions_starting_from_device : Should the partitions start from the device?	(true).
potential_solver : Name of the potential solver	().
result_propagator_name : The name of the propagator that should be returned by the PropagationModule. It should correspond to the name in the NEGF object list for other Propagation solvers	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	($name_retarded_Green_Fermion_dra
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The simulation type that is to prove the Hamiltonian and DOF map of the target_domain.	
target_domain : Name of the domain that the PropagationModule will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature of device in Kelvin. Applied to Schroedinger solver	(300.0).
tic_toc_name : Custom user-defined name for use in timing/memory profiling	($(name)).

Table 5.96: Parameters of the MultiEqNeq Solver.

5.97 NEGF

DEPRECATED.
The implementation of Nonequilibrium Green's Functions (NEGF) has a lot of things in common with the WF implementation. It shares all the parameters except that type is NEGF instead of WF. It also has the same MPI-parallelism. There are additional parameters

connected to scattering:

Parameter : Description	Default Value
__disable_init : flag command to disable initialization	(false).
__disable_output : flag command to disable solver output execution	(false).
__disable_reinit : flag command to disable re-initialization	(false).
__disable_reset_set : flag command to disable re-initialization of sets	(false).
__disable_solve : flag command to disable solver solution	(false).
__disable_step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search_flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
density_iteration : Maximum inner iterations (Green functions - self energies)	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
phonon_factor_atom : Debug option to adjust the electron phonon interaction strength in the deformation potential model. Do not use.	().
scattering : When set to true then a scattering model is turned on based on deformation-potential scattering with phonons.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
symmetric : (default: false) When set to true then matrices are saved in a symmetric format.	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.97: Parameters of the NEGF Solver.

5.98 NEGFG

DEPRECATED.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).

Parameter : Description	Default Value
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
density__iteration : Maximum inner iterations (Green functions - self energies)	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
phonon__factor__atom : Debug option to adjust the electron phonon interaction strength in the deformation potential model. Do not use.	().
scattering : When set to true then a scattering model is turned on based on deformation-potential scattering with phonons.	().
surface__of__regions : surface regions required to calculate boundary conditions	().
symmetric : (default: false) When set to true then matrices are saved in a symmetric format.	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.98: Parameters of the NEGFG Solver.

5.99 NemoMatrixDummy

This solver tests the capability of the NemoMatrixInterface class and its derived classes: PetscMatrixParallelComplex and NMLMatrix.

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
dense__A__cols : Number of columns of dense matrix A, rows of matrix B	(100).
dense__A__rows : Number of rows of dense matrix A	(100).
dense__B__cols : Number of columns of dense matrix B	(100).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
test_NMLMatrix : Test functionality of NMLMatrix objects	(true).
test_PetscMatrixParallelComplex : Test functionality of PetscMatrixParallelComplex objects	(true).
test_dense : Test functionality of dense matrix operations	(true).
test_sparse : Test functionality of sparse matrix operations	(true).
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.99: Parameters of the NemoMatrixDummy Solver.

5.100 NonlinearPoisson

This simulation iterates the Poisson and schrodinger equations to self-consistencyusing a nonlinear predictor-corrector scheme in which the dependence of the density on the potential is predicted. Like this a Newton iteration is carried out to find the electrostatic potential. Such a scheme greatly improves convergence [6]. Note that this simulation object contains both the solution of the Poisson equation and the Newton iteration with any density solver

Parameter : Description	Default Value
CB_initial_shift : If set, then the potential in n-type materials will be set to the conduction band edge plus this number (in eV).	(0.1).
Dirichlet_nodes_output : Outputs actual potential at Dirichlet nodes (e.g. gate).	(false).
Newton_step_only : If true, only a single step rather than the full self-consistentiteration is performed. This option is typically used when the Poisson solver is steered by another solver, e.g.~a transport simulation.	(false).

Parameter : Description	Default Value
VB_initial_shift : If set, then the potential in p-type materials will be set to the valence band edge minus this number (in eV).	(0.1).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
abs_tolerance : Absolute tolerance of the residual used in Non-Linear solver.	(1e-8).
active_regions : required regions to be activated for this solver	(()).
atomistic_output : Save atom-based quantities to file. The list can contain the entries potential, charge, charge_cm-3,free_charge,free_charge_cm-3,doping,doping_cm-3,conduction_band and valence_band.For simulations without grid parallelization VTK is used as format, otherwise Silo	(()).

Parameter : Description	Default Value
average_charge_over_FEM_cell : considers charge to be averaged over a FEM cell	(true).
average_over_cell : Leave this one true (default).	(true).
boundary_condition : A subsection that specifies the boundary condition for the Poisson solver. It contains the following parameters:type:One of ElectrostaticContact, PotentialFromSolver, NormalField.; name: Some arbitrary name.; boundary regions:; voltage: For ElectrostaticContact, the Dirichlet value of the potential.; E-field: For NormalField, the Neumann; D-field value: of the E-field or D-field.	().
boundary_regions : required regions as boundary conditions	(()).
calculate_density_only : If true, calculate the density for a constant potential.	(false).
charge_model : Can be either electron_hole (most cases), in which case doping charges are added to the excess charges from a Schroedinger or other solver, or electron_core_model in which case there are only electrons and these are counter balanced with a fixed positive charge based on the atomic type	(electron_hole).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
confined_band_solvers : Names of band structure solvers that provide the information on conduction and valence band edges due to confienement. Used for printing spatially resolved self-consistent band edges.	().
constant_intial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(0.0).
copy_solution_from : Copy solution from another Poisson simulation.	().

Parameter : Description	Default Value
custom_convergence_test : Provides customized Poisson/transport solver convergence test, avoiding the use of the built in Petsc routine. Current implementation matches the one in OMEN code for ballistic transport.	(false).
density_solver : A list with names of solvers where charge can be found	(()).
dependent_solvers : List of solvers dependent on this Poisson simulation.	(()).
do_input_initial_nonlinearpoisson_potential : See set_initial_potential description	(false).
do_nonlinearpoisson_outputs_xyz_format : If true, then output will be done in the cartesian XYZ format, which is handy for Matlab-processing.	(false).
do_output_nonlinearpoisson_potential : If true, then at the end of NonlinearPoisson, the potential is output in a separate file in FEM format. This potential can be used later to restart the simulation, or to do a fixed potential run	(false).
do_output_potential_each_iteration : When true, on each Schrodinger solve from NonlinearPoisson, the potential is output in a separate file in FEM format. This file is getting updated (overwritten) on each such solve. This potential can be used later to restart thesimulation, or to do the fixed potential run. This is done ifconvergence is not reached because of the crash or too long run, the potential can still be obtained and simulation can be restarted.	(false).
do_outputs_from_density_solver : If true, then the outputs specified in density quantum solver can be obtained at the end of the NonlinearPoisson solution.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
equilibrium_el_chem_pot : See set_initial_potential description	(0.0).

Parameter : Description	Default Value
external_object_name : Set this job to incomplete in the specified external solver	(()).
external_output : Names of solvers external to Poisson for which output will be called.	().
external_solver : Name of the external solver in which job have to be set to incomplete. Job names can be specified in the option external object name.	(()).
fem_output : If equals to free_charge_cm-e it outputs free charge in the units of [cm^-3]	().
field : Electric field. Relevant only if set_initial_field = true.	(()).
friends : if a potential is not found for an specific atom position, Poisson will ask 'friends' solvers if this potential is stored on them	(()).
full_Newton_step : use full newton step for search in Newton solver	(false).
import_initial_potential_from_solver : Import initial potential into this Poisson from some other solver (Poisson or non-Poisson)	(false).
import_on_FEM_grid : Import initial potential on FEM grid (not atomic)	(false).
initial_potential_file_name : Filename related to set_initial_potential	().
initial_potential_point : 3-vector: point in space. Relevant only if set_initial_field = true	(()).
initial_potential_source : Name of the solver from which the initial potential is to be imported.	().
interpolation_power : the power used to calculate the distance weight in interpolation of the free charge density	(2).
ksp_monitor : If true, enables a monitor for Krylov Sub-sPace solver in PETSc package.	(false).
ksp_type : Linear solver type. See the PETSc documentation for possible choices. Recommended are e.g. gmres or bcgs.	(gmres).

Parameter : Description	Default Value
laplacian : Either x, y, z or nothing. If not empty it means only one derivative in laplacian operator is considered, the letter tells which one. Default is empty	(()).
linear_search_alpha : coefficient used in newton raphson search step x(k+1) = x(k) + alpha*dk instead of x(k+1) = xk + dk	(0.0).
linear_search_step_tolerance : the minimum step tolerance of the Newton solver, the search will end if the step is smaller than this number	(0.0).
linear_solver_family : Sets the linear solver family. Current available options are "petsc" and "mumps"	(petsc).
linear_solver_maxit : In case of an iterative linear solver, this is the maximum number of iterations to solve the linear system.	(0).
max_function_evals : Maximum number of Newton right-hand-side evaluations	(1000).
max_iterations : Maximum number of Newton iterations	(100).
max_nonlinear_step : Maximum value of a step in Non-linear solver.	().
name : A unique name tag / identifier for the specific instance of the solver.	
node_potential_output : If true, the potential is written to (sim-name)_nodal_potential.dat	(false).
number_of_interpolation_points : number of surrounding points used in interpolation of the free charge density	(4).
number_of_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(40-100).

Parameter : Description	Default Value
one_dim_output : Interpolates atom-based quantities onto an axis and generates 1D ASCII output compatible with 1D Matlab-plots. The list can contain the entries potential, free_charge_cm-3, doping_cm-3, conduction_band and valence_band.	().
one_dim_output_average : If true then unit cells are used for the 1D discretization along some direction and some averaging is done within the cells. If false then the orthogonal projection of the atomic position serves as the 1D discretization	(true).
output_at_each_iteration : When true, on each Schrodinger solve from NonlinearPoissonthe potential is output in a separate file in FEM format. This file is getting updated (overwritten) on each such solve. This potential can be used later to restart the simulation, or to do the fixed potential run. This is done if convergence is not reached because of the crash or too long run, the potential can still be obtained and simulation can be restarted.	(false).
output_dependent_solvers : List of dependent solvers for which output is needed.	().
output_file_suffix : suffix to be added on output files	().
output_jacobian : Output the Jacobian of Non-linear solver.	(false).
output_line_corners : Define a straight line on which to interpolate charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	().
output_norms : prints the norms as a function of iteration. Useful for debugging.	(true).
output_residual : Output the residual of Non-linear solver.	(true).

Parameter : Description	Default Value
output_yz_integrated_density : If true, output density along x direction, integrated on xy planes.	(true).
pc_type : Preconditioner type. See the PETSc documentation for possible choices. Recommended are e.g. asm for distributed-grid simulations, lu for small systems or jacobi	(jacobi).
petsc_solver_algorithm : linear search algorithm of Newton solver	(ls).
poisson_vtk_mesh : saves the electrostatic potential in vtk format, the potential is then a continuum on a tetrahedral mesh	(false).
potential_at_initial_point : Potential at that point. Relevant only if set_initial_field = true.	(0).
rel_tolerance : Relative residual tolerance of the Newton solver	(1e-6).
restart_data_input_file : File to use for restarting simulation. The file can be written by setting restart_data_output_file option.	().
restart_data_output_file : File in which to save data for restarting simulation. Specify this filename using restart_data_input_file when restart is needed.	().
set_initial_field : If true, then a constant electric field will be set at the beginning.	(false).
set_initial_potential : if true, then some initial potential will be either read in from file (options do_input_initial_nonlinearpoisson_potential is true) or computed (with an equilibrium Fermi level which is 0 by default or is given by equilibrium_el_chem_pot).	(true).
shift_copied_solution_by_voltage : Value by which the solution of Poisson equation is to be shifted.	(0.0).
solve_on_single_replica : Solve Non-linear Poisson on a single replica of the geometry.	(false).

Parameter : Description	Default Value
solve_with_zero_density_first : Solve initial potential with zero density everywhere.	(false).
step_abs_tolerance : Absolute step tolerance of the Newton solver	(1e-10).
step_rel_tolerance : Relative step tolerance of the Newton solver	(1e-10).
stop_if_diverged : Stop the self-consistent loop if potential doesn't converge.	(true).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
tolerance_in_terms_of_doping : Set absolute tolerance = doping tolerance * mesh element volume. Based on user tolerance_in_terms_of_doping	(0.0).
use_average_density_as_a_guess : Must be false (true has a bug)	(false).
use_classical_jacobian : Calculate the Jacobian using the semiclassical charge density with spatially dependent Fermi level.	(false).

Table 5.100: Parameters of the NonlinearPoisson Solver.

5.101 NonlinearPoissonFEM

This simulation iterates the Poisson and schrodinger equations to self-consistencyusing a nonlinear predictor-corrector scheme in which the dependence of the density on the potential is predicted. Like this a Newton iteration is carried out to find the electrostatic potential. Such a scheme greatly improves convergence [6]. Note that this simulation object contains both the solution of the Poisson equation and the Newton iteration with any density solver

Parameter : Description	Default Value
CB_initial_shift : If set, then the potential in n-type materials will be set to the conduction band edge plus this number (in eV).	(0.1).

Parameter : Description	Default Value
Dirichlet_nodes_output : Outputs actual potential at Dirichlet nodes (e.g. gate).	(false).
Newton_step_only : If true, only a single step rather than the full self-consistentiteration is performed. This option is typically used when the Poisson solver is steered by another solver, e.g.˜a transport simulation.	(false).
VB_initial_shift : If set, then the potential in p-type materials will be set to the valence band edge minus this number (in eV).	(0.1).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(().
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
abs_tolerance : Absolute tolerance of the residual used in Non-Linear solver.	(1e-8).
active_regions : required regions to be activated for this solver	(().

Parameter : Description	Default Value
atomistic_output : Save atom-based quantities to file. The list can contain the entries potential, charge, charge_cm-3,free_charge,free_charge_cm-3,doping,doping_cm-3,conduction_band and valence_band.For simulations without grid parallelization VTK is used as format, otherwise Silo	().
average_charge_over_FEM_cell : considers charge to be averaged over a FEM cell	(true).
average_over_cell : Leave this one true (default).	(true).
boundary_condition : A subsection that specifies the boundary condition for the Poisson solver. It contains the following parameters:type:One of ElectrostaticContact, PotentialFromSolver, NormalField.; name: Some arbitrary name.; boundary regions:; voltage: For ElectrostaticContact, the Dirichlet value of the potential.; E-field: For NormalField, the Neumann; D-field value: of the E-field or D-field.	().
boundary_regions : required regions as boundary conditions	().
calculate_density_only : If true, calculate the density for a constant potential.	(false).
charge_model : Can be either electron_hole (most cases), in which case doping charges are added to the excess charges from a Schroedinger or other solver, or electron_core_model in which case there are only electrons and these are counter balanced with a fixed positive charge based on the atomic type	(electron_hole).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).

Parameter : Description	Default Value
confined_band_solvers : Names of band structure solvers that provide the information on conduction and valence band edges due to confienement. Used for printing spatially resolved self-consistent band edges.	().
constant_intial_potential : Initial potential guess for the nonlinear Poisson solver, unless a semiclassical guess is used. Potential on all atoms is set to the same value.	(0.0).
copy_solution_from : Copy solution from another Poisson simulation.	().
custom_convergence_test : Provides customized Poisson/transport solver convergence test, avoiding the use of the built in Petsc routine. Current implementation matches the one in OMEN code for ballistic transport.	(false).
density_solver : A list with names of solvers where charge can be found	(().
dependent_solvers : List of solvers dependent on this Poisson simulation.	(().
do_input_initial_nonlinearpoisson_potential : See set_initial_potential description	(false).
do_nonlinearpoisson_outputs_xyz_format : If true, then output will be done in the cartesian XYZ format, which is handy for Matlab-processing.	(false).
do_output_nonlinearpoisson_potential : If true, then at the end of NonlinearPoisson, the potential is output in a separate file in FEM format. This potential can be used later to restart the simulation, or to do a fixed potential run	(false).

Parameter : Description	Default Value
do_output_potential_each_iteration : When true, on each Schrodinger solve from NonlinearPoisson, the potential is output in a separate file in FEM format. This file is getting updated (overwritten) on each such solve. This potential can be used later to restart thesimulation, or to do the fixed potential run. This is done ifconvergence is not reached because of the crash or too long run, the potential can still be obtained and simulation can be restarted.	(false).
do_outputs_from_density_solver : If true, then the outputs specified in density quantum solver can be obtained at the end of the NonlinearPoisson solution.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
equilibrium_el_chem_pot : See set_initial_potential description	(0.0).
external_object_name : Set this job to incomplete in the specified external solver	(()).
external_output : Names of solvers external to Poisson for which output will be called.	().
external_solver : Name of the external solver in which job have to be set to incomplete. Job names can be specified in the option external object name.	(()).
fem_output : If equals to free_charge_cm-e it outputs free charge in the units of [cm^-3]	().
field : Electric field. Relevant only if set_initial_field = true.	(()).
friends : if a potential is not found for an specific atom position, Poisson will ask 'friends' solvers if this potential is stored on them	(()).
full_Newton_step : use full newton step for search in Newton solver	(false).

Parameter : Description	Default Value
import_initial_potential_from_solver : Import initial potential into this Poisson from some other solver (Poisson or non-Poisson)	(false).
import_on_FEM_grid : Import initial potential on FEM grid (not atomic)	(false).
initial_potential_file_name : Filename related to set_initial_potential	().
initial_potential_point : 3-vector: point in space. Relevant only if set_initial_field = true	(().
initial_potential_source : Name of the solver from which the initial potential is to be imported.	().
interpolation_power : the power used to calculate the distance weight in interpolation of the free charge density	(2).
ksp_monitor : If true, enables a monitor for Krylov Sub-sPace solver in PETSc package.	(false).
ksp_type : Linear solver type. See the PETSc documentation for possible choices. Recommended are e.g. gmres or bcgs.	(gmres).
laplacian : Either x, y, z or nothing. If not empty it means only one derivative in laplacian operator is considered, the letter tells which one. Default is empty	(().
linear_search_alpha : coefficient used in newton raphson search step x(k+1) = x(k) + alpha*dk instead of x(k+1) = xk + dk	(0.0).
linear_search_step_tolerance : the minimum step tolerance of the Newton solver, the search will end if the step is smaller than this number	(0.0).
linear_solver_family : Sets the linear solver family. Current available options are "petsc" and "mumps"	(petsc).
linear_solver_maxit : In case of an iterative linear solver, this is the maximum number of iterations to solve the linear system.	(0).
max_function_evals : Maximum number of Newton right-hand-side evaluations	(1000).

Parameter : Description	Default Value
max_iterations : Maximum number of Newton iterations	(100).
max_nonlinear_step : Maximum value of a step in Non-linear solver.	().
name : A unique name tag / identifier for the specific instance of the solver.	
node_potential_output : If true, the potential is written to (sim-name)_nodal_potential.dat	(false).
number_of_interpolation_points : number of surrounding points used in interpolation of the free charge density	(4).
number_of_output_points : Number of points to be used in plotting along a 1D line. If there are multiple line segments each segment has this number of points. Line, or line segments, are defined by output_line_corners.	(40-100).
one_dim_output : Interpolates atom-based quantities onto an axis and generates 1D ASCII output compatible with 1D Matlab-plots. The list can contain the entries potential, free_charge_cm-3, doping_cm-3, conduction_band and valence_band.	(()).
one_dim_output_average : If true then unit cells are used for the 1D discretization along some direction and some averaging is done within the cells. If false then the orthogonal projection of the atomic position serves as the 1D discretization	(true).
output_at_each_iteration : When true, on each Schrodinger solve from NonlinearPoissonthe potential is output in a separate file in FEM format. This file is getting updated (overwritten) on each such solve. This potential can be used later to restart the simulation, or to do the fixed potential run. This is done if convergence is not reached because of the crash or too long run, the potential can still be obtained and simulation can be restarted.	(false).

Parameter : Description	Default Value
output_dependent_solvers : List of dependent solvers for which output is needed.	(()).
output_file_suffix : suffix to be added on output files	().
output_jacobian : Output the Jacobian of Non-linear solver.	(false).
output_line_corners : Define a straight line on which to interpolate charge density, potential... Form: ([x1,y1,z1],[x2,y2,z2]), where (x1,y1,z1) and (x2,y2,z2) represent the two end points of the line. If there are multiple line segments each segment should be represented with two corners.	(()).
output_norms : prints the norms as a function of iteration. Useful for debugging.	(true).
output_residual : Output the residual of Non-linear solver.	(true).
output_yz_integrated_density : If true, output density along x direction, integrated on xy planes.	(true).
pc_type : Preconditioner type. See the PETSc documentation for possible choices. Recommended are e.g. asm for distributed-grid simulations, lu for small systems or jacobi	(jacobi).
petsc_solver_algorithm : linear search algorithm of Newton solver	(ls).
poisson_vtk_mesh : saves the electrostatic potential in vtk format, the potential is then a continuum on a tetrahedral mesh	(false).
potential_at_initial_point : Potential at that point. Relevant only if set_initial_field = true.	(0).
rel_tolerance : Relative residual tolerance of the Newton solver	(1e-6).
restart_data_input_file : File to use for restarting simulation. The file can be written by setting restart_data_output_file option.	().

Parameter : Description	Default Value
restart_data_output_file : File in which to save data for restarting simulation. Specify this filename using restart_data_input_file when restart is needed.	().
set_initial_field : If true, then a constant electric field will be set at the beginning.	(false).
set_initial_potential : if true, then some initial potential will be either read in from file (options do_input_initial_nonlinearpoisson_potential is true) or computed (with an equilibrium Fermi level which is 0 by default or is given by equilibrium_el_chem_pot).	(true).
shift_copied_solution_by_voltage : Value by which the solution of Poisson equation is to be shifted.	(0.0).
solve_on_single_replica : Solve Non-linear Poisson on a single replica of the geometry.	(false).
solve_with_zero_density_first : Solve initial potential with zero density everywhere.	(false).
step_abs_tolerance : Absolute step tolerance of the Newton solver	(1e-10).
step_rel_tolerance : Relative step tolerance of the Newton solver	(1e-10).
stop_if_diverged : Stop the self-consistent loop if potential doesn't converge.	(true).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
tolerance_in_terms_of_doping : Set absolute tolerance = doping tolerance * mesh element volume. Based on user tolerance_in_terms_of_doping	(0.0).
use_average_density_as_a_guess : Must be false (true has a bug)	(false).

Parameter : Description	Default Value
use_classical_jacobian : Calculate the Jacobian using the semiclassical charge density with spatially dependent Fermi level.	(false).

Table 5.101: Parameters of the NonlinearPoissonFEM Solver.

5.102 NonlocalRGFModule

[MODULE] Implements Recursive Greens Function algorithm for arbitrary amount of offdiagonal blocks

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
active_regions : required regions to be activated for this solver	(().
boundary_regions : required regions as boundary conditions	(().
chemical_potential_for_$(end_leads) : Chemical potential in eV of $(end_leads), usually drain contact	(0.0).
chemical_potential_for_$(start_leads) : Chemical potential in eV of $(start_leads), usually source contact	(0.0).
debug_output : If set to true, produces outputs from Repartition solvers in Leads	(false).
density_Jacobian_output : If true, propagation solver will trigger density Jacobian (derivative) solution and provide visual output file	(false).
density_output : If true, propagation solver will trigger density solution and provide visual output file	(false).
domain : Domain required for this simulation	().
end_leads : name of the lead that will be coupled to the last partition	(end_leads).
end_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green's functionin drain contact	().
lead_surface_green_$(end_leads) : name of the surface green's matrix, gR at the drain end. Usually named like: (end leads solver) Green Fermiondrain contact	().
lead_surface_green_$(start_leads) : name of the surface green's matrix, gL at the source end. Usually named like: (start leads solver) GreenFermion source contact	().
name : A unique name tag / identifier for the specific instance of the solver.	
no_integration_for_transmission : Turns ON or OFF k-space integration in the periodic directions	(false).

Parameter : Description	Default Value
number_of_subdomains : Number of subdomains for RGF calculation. Option for repartition solver	(1).
output_file_suffix : suffix to be added on output files	().
potential_solver : The external potential output from this solver is added to the Hamiltonian	().
start_leads : name of the lead that will be coupled to partition 0	(start_leads).
start_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green'sfunction in the source contact	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature_for_$(end_leads) : Temperature in Kelvin for $(end_leads) contact	(300.0).
temperature_for_$(start_leads) :	(300.0Temperature in Kelvin for $(start_leads) contact).
tic_toc_name : Prefix for tic-tocs	($(name)).
transmission : If true, simulation provides transmission output	(false).

Table 5.102: Parameters of the NonlocalRGFModule Solver.

5.103 NonorthoRGFModule

This module use the RGF algorithm to obtain the Transmission and density for a non-orthogonal basis

Parameter : Description	Default Value
$(lead_surface_green_$(end_leads))_solver : name of the green solver which calculates surface green's matrix, gR at the drain end.	().
G_for_transmission : If this option exists, it specifies the name of the GR matrix which is used to calculate transmission probability	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
chemical_potential_for_$(end_leads) : Chemical potential in eV of $(end_leads), usually drain contact	(0.0).

Parameter : Description	Default Value
chemical_potential_for_$(start_leads) : Chemical potential in eV of $(start_leads), usually source contact	(0.0).
debug_output : If set to true, produces outputs from Repartition solvers in Leads	(false).
do_output : If set to true, output files are generated.	(false).
domain : Domain required for this simulation	().
end_leads : name of the lead that will be coupled to the last partition	(end_leads).
end_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green's functionin drain contact	().
invert_solver : Type of solver used for dense matrix inversions. Options are lapack, petsc, magma, magmamic	(lapack).
lead_surface_green_$(end_leads) : name of the surface green's matrix, gR at the drain end. Usually named like: (end leads solver) Green Fermiondrain contact	().
name : A unique name tag / identifier for the specific instance of the solver.	
no_integration_for_transmission : Turns ON or OFF k-space integration in the periodic directions	(false).
number_of_subdomains : Number of subdomains for RGF calculation. Option for repartition solver	(1).
old_mult_version : old version of matrix multiplication: use super matrix concept in direct_iterative_leads, much slower.	(false).
output_file_suffix : suffix to be added on output files	().
output_raw_transmission_data : If true, simulation provides transmission output	(false).

Parameter : Description	Default Value
partition_direction : Direction used by Repartition solver to perform partitioning of subdomains. See Repartition option "partition_direction" for more info.	(0).
repartition_strategy : Strategy used to set option "strategy" ofpartition the domain by Repartition solver. Available options are: based_on_coupling, based_on_coordinate, bfs, based_on_region	().
start_leads : name of the lead that will be coupled to partition 0	(start_leads).
start_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green'sfunction in the source contact	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature_for_$(end_leads) : Temperature in Kelvin for $(end_leads) contact	(300.0).
temperature_for_$(start_leads) :	(300.0Temperature in Kelvin for $(start_leads) contact).
tic_toc_name : Prefix for tic-tocs	($(name)).
transmission : If true, simulation provides transmission output	(false).
use_smart_dofmap : reorder the device Hamiltonian to be block diagonal shape.	(true).

Table 5.103: Parameters of the NonorthoRGFModule Solver.

5.104 PGAPackOptimization

This simulation do optimizations, by the use of a Genetic Algorithm (PGAPack)

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
crossover__prob : Crossover probability to reproduce next generation	(0.8).
crossover__type : Crossover mechanism to reproduce next generation	(1).
domain : Domain required for this simulation	().

Parameter : Description	Default Value
mutation_prob : Mutation probability to reproduce next generation	(0.2).
mutation_type : Mutation mechanism to reproduce next generation	(2).
name : A unique name tag / identifier for the specific instance of the solver.	
num_cores_by_seed : number of cores used by the evaluation function	(1).
number_of_coords :	(10).
number_of_iterations : number of iterations	(100).
output_file_suffix : suffix to be added on output files	().
pop_size : number of cromosomes in each generation	(100).
random_seed :	(1).
selection_type : Selection mechanism to reproduce next generation	(3).
surface_of_regions : surface regions required to calculate boundary conditions	().
target_solver : Solver that is doing the evaluation	
tic_toc_name : Prefix for tic-tocs	($(name)).
value_range :	

Table 5.104: Parameters of the PGAPackOptimization Solver.

5.105 PartitionRetardedSelfModule

[MODULE] Retarded Self Energy Module Solver including Partitioning

Parameter : Description	Default Value
$(momentum_names)_constructor : The name of the solver which creates the mesh or grid for the corresponding momentum_name	

Parameter : Description	Default Value
Parallelizer : This is the name of the solver which creates the momentum (energy, k etc) grid	
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : This is a list of integers defining the regions of the simulation domain where the solver is solved or contain active atoms	
boundary__regions : required regions as boundary conditions	(()).
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
extra__domain__name__element : Suffix to the name of newly created subdomains	().

Parameter : Description	Default Value
lead_domain : This a list of strings defining the names of the domains which are leads to $domain on which the solver is solved	
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed. Available options are momentum_1D, momentum_2D, momentum_3D, energy, spin, valley.	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
potential_solver : This solver (usually a poisson solver) generates the external potential to be added to the Hamiltonian. Required when possion solver is needed in the lead.	().
result_propagator_name : This is the name of the propagator object which contains the final result of the solver, i.e. G< for all (E,k)	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : Name of Hamilton_constructor for the domain the self energy should be solved	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature (in kelvin) of the device to be simulated	(300).

Parameter : Description	Default Value
tic_toc_name : A name that you want to use for displaying function run times in this solver. The run times are output in NEMO timing xmls in the run folder.	($name).

Table 5.105: Parameters of the PartitionRetardedSelfModule Solver.

5.106 PartitionRetardedSurfaceGreenModule

[MODULE] Forward Retarded Green's Function Module Solver

Parameter : Description	Default Value
$(momentum_names)_constructor : Names assigned to the Hamiltonian used for calculating momentum	
Parallelizer : Point to a Parallelizer solver, where you can define the energy range and how many points you want	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).

Parameter : Description	Default Value
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : The regions are to be used for this simulation. It should be within the region numbers defined in geometry	
boundary_regions : required regions as boundary conditions	(()).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
extra_domain_name_element : Suffix to the name of newly created subdomains.	().
lead_domain : The domain that is connected to this PropagationModule. It can be used, for example, to get the coupling Hamiltonian between this domain and the $(lead_domain)	().
momentum_names : the names user used to define momentum K, like energy, momentum_1D	
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
potential_solver : Name of the simulation that provides the electrostatic Potential	().
result_propagator_name : The name of the propagator that should be returned by the PropagationModule. It should correspond to the name in the NEGF object list for other Propagation solvers	

Parameter : Description	Default Value
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The simulation type that is to prove the Hamiltonian and DOF map of the target_domain.	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.106: Parameters of the PartitionRetardedSurfaceGreenModule Solver.

5.107 Phonons

Solver to calculate phonon modes, spectra, and related quantities
In combination with the VFFStrain simulation this solver allows for the computation of phonon spectra.
Parallelism: Since the phonon simulation uses the matrix that is constructed in VFFStrain it has the same spatial parallelization, though this has not yet been tested. Additionally it distributes the independent problems for each wavevector onto multiple MPI-processes. This is done automatically.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
bulk__file : When set then all the above post-processing quantities are saved into this file.	().
calculate__LO__TO__splitting : If true then the LO-TO splitting is computed.	(false).
calculate__dispersion : If true then the phonon disepersion are calculated as a post-processing step.	(false).
calculate__elastic : If true then the bulk elastic constants are computed from the VFF parameters.	(false).
calculate__grueneisen : If true then the Gruneisen mode parameters at gamma, X, and L are computed as a post-processing step.	(false).
calculate__sound : If true then the speeds of sound are calculated as a post-processing step.	(false).

Parameter : Description	Default Value
calculate_sound_for_Cijk : If true, calculate sound velocity.	(false).
convergence_limit : Convergence limit below which an iterative eigensolver regards the value as converged.	(1e-8).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
eigensolver : Which eigensolver to take (choose from lapack, krylovschur, arpack and others).	(lapack).
eta2_sound : Finite-difference parameter for the speed of sound computation (dimensionless).	(0.001).
eta_LOTO : Optional parameter (in units 2pi/a) to compute gamma-energies slightly away from gamma during the LO-TO splitting computation.	(0.0001).
eta_grueneisen : Finite-difference parameter for the Gruneisen mode	(0.001).
eta_sound : Finite-difference parameter for the speed of sound computation.	(0.001).
max_num_iters : Maximum number of Krylov iterations of the eigensolver.	(300).
monitor_convergence : When set to true then the convergence of the Krylov iteration is monitored in the terminal.	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
num_eigenvalues : Number of eigenvalues to compute (irrelevant for lapack since all eigenvalues are computed there).	(0).
number_of_nodes : This list gives the uniform discretization of each segment set by the qpoints parameter (note that specifying N points means (N-1)) segments.	(()).
output_file : Name of the file where to save the phonon dispersion.	().
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
output_polarization : Filename.	().
preconditioner : Optional choice of preconditioner. Default: lu.	(lu).
qpoints : A list of points in q-space along which the dispersion is calculated.	().
qspace_basis : Sets in which basis the qpoints parameter is specified (cartesian or reciprocal).	(reciprocal).
shift_wavevector_to_1st_BZ : If true then every q-point is shifted back to the first BZ before the computation.	(true).
store_polarization : True or false	(false).
strain_sim : Name of the VFFStrain simulation object. That object sets up the dynamical matrix.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Name of the tic_toc used for recording timing of this solver in NEMO5 timing xml.	($name).

Table 5.107: Parameters of the Phonons Solver.

5.108 Poisson

This simulation solves the linear Poisson equation for a fixed density

Parameter : Description	Default Value
Dirichlet_nodes_output : Outputs actual potential at Dirichlet nodes (e.g. gate).	(false).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
atomistic__output : Save atom-based quantities to file. The list can contain the entries potential, charge, charge_cm-3,free_charge,free_charge_cm-3,doping,doping_cm-3,conduction_band and valence_band.For simulations without grid parallelization VTK is used as format, otherwise Silo	(()).
average__over__cell : Leave this one true (default).	(true).
boundary__regions : required regions as boundary conditions	(()).
charge__model : Can be either electron_hole (most cases), in which case doping charges are added to the excess charges from a Schroedinger or other solver, or electron_core_model in which case there are only electrons and these are counter balanced with a fixed positive charge based on the atomic type	(electron_hole).
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).

Parameter : Description	Default Value
copy_solution_from : Defines another poisson solver that will give a solution, instead of solving. Assumes that the mesh is exactly the same.	().
density_solver : A list with names of solvers where charge can be found	(()).
do_nonlinearpoisson_outputs_xyz_format : If true, then output will be done in the cartesian XYZ format, which is handy for Matlab-processing.	(false).
do_output_nonlinearpoisson_potential : If true, then at the end of NonlinearPoisson, the potential is output in a separate file in FEM format. This potential can be used later to restart the simulation, or to do a fixed potential run	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
fem_output : saves information about the fem output ('potential', 'dielectric_constant')	(()).
friends : if a potential is not found for an specific atom position, Poisson will ask 'friends' solvers if this potential is stored on them	(()).
laplacian : Either x, y, z or nothing. If not empty it means only one derivative in laplacian operator is considered, the letter tells which one. Default is empty	(()).
name : A unique name tag / identifier for the specific instance of the solver.	
node_potential_output : If true, the potential is written to (sim-name)_nodal_potential.dat	(false).
one_dim_output : Interpolates atom-based quantities onto an axis and generates 1D ASCII output compatible with 1D Matlab-plots. The list can contain the entries potential, free_charge_cm-3, doping_cm-3, conduction_band and valence_band.	(()).

Parameter : Description	Default Value
one_dim_output_average : If true then unit cells are used for the 1D discretization along some direction and some averaging is done within the cells. If false then the orthogonal projection of the atomic position serves as the 1D discretization	(true).
output_file_suffix : suffix to be added on output files	().
poisson_vtk_mesh : saves the electrostatic potential in vtk format, the potential is then a continuum on a tetrahedral mesh	(false).
shift_copied_solution_by_voltage : Value by which the solution of Poisson equation is to be shifted.	(0.0).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
use_average_density_as_a_guess : Must be false (true has a bug)	(false).

Table 5.108: Parameters of the Poisson Solver.

5.109 PoolObjectTest1

[TEST] Solver to test Pool objects implementations

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
max__size : Define the max number of persistent objects	(1).
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
size : Define the number of Objects to be created	(20).
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.109: Parameters of the PoolObjectTest1 Solver.

5.110 Propagation

Solver to calculate phonon modes, spectra, and related quantities
This simulation is an attempt to set up a general NEGF framework for electron and phonon systems. It is still under heavy development.

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false)
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(().
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(().
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().

Parameter : Description	Default Value
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().

Parameter : Description	Default Value
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(()).
LRA : true/false, flag to control whether LRA is used or not	(false).
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	

Parameter : Description	Default Value
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).

Parameter : Description	Default Value
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().

Parameter : Description	Default Value
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.110: Parameters of the Propagation Solver.

5.111 PythonSolver

PythonSolverSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable_solve : flag command to disable solver solution	(false).
__disable_step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search_flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.111: Parameters of the PythonSolver Solver.

5.112 QTBMModule

[MODULE] Implements QTBM algorithm and includes capability of self-energy parallelization with MPI

Parameter : Description	Default Value
$(lead_name)_MPI_ranks : Sets the number of MPI ranks to be used to solve specified lead's self-energy	($<number_of_MPI_ranks_in_real_sp
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
debug_output : If set to true, produces outputs from Repartition solvers in device and leads	(false).
domain : Domain required for this simulation	().
drain_contact : Name of drain contact of simulated device	

Parameter : Description	Default Value
end_leads : name of the lead that will be coupled to the last partition of device	(end_leads).
end_leads_$(lead_name) : name of the lead that will be coupled to the last partition of specified lead	(start_leads_$(lead_name)).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
parallelize_leads : If set to true, will distribute self-energy calculation among multiple ranks	(false).
source_contact : Name of source contact of simulated device	
start_leads : name of the lead that will be coupled to partition 0 of device	(start_leads).
start_leads_$(lead_name) : name of the lead that will be coupled to first partition of specified lead	(start_leads_$(lead_name)).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.112: Parameters of the QTBMModule Solver.

5.113 QTBMTaskManagerModule

QTBMTaskManagerModuleSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.113: Parameters of the QTBMTaskManagerModule Solver.

5.114 QTBM_Propagation

QTBM Solver

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(().
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(().
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(().
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(().

Parameter : Description	Default Value
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Drain_chemical_potential : chemical potential at drain contact unit eV	(0.0).
Drain_selfenergy : self energy solver for drain contact, if new_right_hand_side=true	().
Drain_solver : drain self-energy solver that provides the contact mode/phase/velocity	().
Drain_solver_MPI_rank : MPI rank for solving drain contact while using compression	(-1).
Drain_temperature : temperature at drain contact unit K	(300).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : This is the name of the Retarded Green's function used for Transmission calculation.If this options is present in input deck, this GR solver is used for calculating Transmission	().
Hamilton_constructor : spatially distributed Schroedinger solver that defines at the device domain, used for compression algorithm	

Parameter : Description	Default Value
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().
LRA : true/false, flag to control whether LRA is used or not	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(().
OMEN_QTBM : OMEN type QTBM ref M. Luisier, PRB 74, 205323 (2006)	(false).
Parallelizer : Name of the simulation that determines the parallelization	().
Source_chemical_potential : chemical potential at source contact unit eV	(0.0).
Source_selfenergy : self energy solver for source contact, if new_right_hand_side=true	().
Source_solver : source self-energy solver that provides the contact mode/phase/velocity	().
Source_solver_MPI_rank : MPI rank for solving source contact while using compression	(-1).
Source_temperature : temperature at source contact unit K	(300).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
add__readable__Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always__ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical__momenta : analytical integration for k-space	(false).
analytical__momenta__me : average effective mass (in unit of me0) used for analytical integration for k-space	().
analytical__momenta__me__CB__D : Drain-side electron effective mass (in unit of me0) used for analytical integration for k-space	().
analytical__momenta__me__CB__S : Source-side electron effective mass (in unit of me0) used for analytical integration for k-space	().
analytical__momenta__me__VB__D : Drain-side hole effective mass (in unit of me0) used for analytical integration for k-space	().

Parameter : Description	Default Value
analytical_momenta_me_VB_S : Source-side hole effective mass (in unit of me0) used for analytical integration for k-space	().
boundary_regions : required regions as boundary conditions	(()).
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
calculate_transmission : calculate transmission or not	(false).
chemical_potential : value for mass less Bosons	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : Switch to control full deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(()).

Parameter : Description	Default Value
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_lead_eta : constant eta added to diagonal of lead Hamiltonian	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).
current : calculate and output terminal current	(false).
debug_output : output temporary matrices in matlab format for debug purpose	(false).
debug_output_job_list : If set to true, a data file is generated with k-point information and corresponding MPI rank information of each solver in global list	(false).
degeneracy_factor : coefficient for energy/k-space integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_Jacobian_output : calculate and output density Jacobian	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_of_states_output : Switch to print the energy resolved density of states to file.	(false).
density_output : calculate and output density	(false).
density_solver : density solver used to obtain energy (position) resolved density and DOS	().
derivative_of_density_solver : solver that provides the density Jacobian	().
differentiate_over_energy : switch to calculate density(false) or density Jacobian(true)	(false).

Parameter : Description	Default Value
do_output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_domain : drain contact domain name	().
drain_hamilton_constructor : Schroedinger solver for drain contact which gives Hamiltonian	().
drain_inverse_retarded_Green : if Khomyakov's type QTBM is used, define Propagator name for drain inverse green's function	().
drain_lead_domain : if new_right_hand_side is not used, define the lead of drain_contact	().
drain_surface_green : surface green's function solver for drain contact, if new_right_hand_side=false	().
drain_threshold_energy : threshold energy that separates electron/hole for drain; above is electron, below is hole	(-1e10).
electron_hole_model : Switch to account for energy delimiters between electrons and holes. If false, only electrons are considered.	(false).
energy_interpolated_output : enable energy interpolated output	(false).
energy_interpolation_points : number of interpolation points	(100).
energy_resolved_current_output : switch to activate file output of the energy resolved current	(false).
energy_resolved_density_output : If true, output the energy resolved density	(false).
eps_limit : threshold to control modes	(1e-6).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().

Parameter : Description	Default Value
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	((()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : If set to true, fermi distribution is replaced by 1-fermi distribution in density calculation	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	((()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().

Parameter : Description	Default Value
linear_system_solver : mumps/petsc factorization scheme for linear solver	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
memory_slim_QTBM :	().
mkl_pardiso_iparm : Set mkl_pardiso iparm values	(-1).
mkl_pardiso_msglvl : Set mkl_pardiso message level. 0 for non information. 1 for complete information	(0).
mkl_pardiso_threads : Set number of threads for mkl_pardiso execution	(1).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
new_right_hand_side : more efficient right hand side calculation of OMEN type QTBM, avoids explicit solution of surface green's function	(false).
new_source_terms : new source term of Khomyakov's paper ref PRB 72, 035450 (2005)	(false).
no_file_output : only output necessary files	(false).
no_integration_for_transmission : Switch to avoid integration over k-space for transmission.Used for debugging to get integer transmission for homogeneous systems	(false).

Parameter : Description	Default Value
nondistributed_Hamilton_constructor : non-distributed Schroedinger solver that defines at the device domain	
number_of_interpolation_points : Number of interpolation points	(100).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_raw_transmission_data : switch to activate file output of the energy and momentum resolved transmission	(false).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().
ratio_of_eigenvalues_to_be_solved : how many percentages of eigenvalues solved	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).

Parameter : Description	Default Value
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf : RGF execution	(false).
rhs_chunk_size : number of right hand side columns; used to allocate memory for rhs	().
scattering_type : Option to define the type of scattering for scattering self-energies. Possible options so far: "lambda_G_scattering", "deformation_potential".	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solve_on_single_replica : This option controls whether the density is solved on all ranks, or only on the 0th one. This has to be consistent with the same option in Poisson.	(true).
solver_type : linear solver type: mumps/compression	(mumps).
source_domain : source contact domain name	().
source_hamilton_constructor : Schroedinger solver for source contact which gives Hamiltonian	().
source_inverse_retarded_Green : if Khomyakov's type QTBM is used, define Propagator name for source inverse green's function	().

Parameter : Description	Default Value
source_lead_domain : if new_right_hand_side is not used, define the lead of source_contact	().
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
source_surface_green : surface green's function solver for source contact, if new_right_hand_side=false	().
source_threshold_energy : threshold energy that separates electron/hole for source; above is electron, below is hole	(-1e10).
store_atomic_offdiagonals : If this option is present, it implies that the offdiagonal elements are atom-resolved. Otherwise, they are DOF-resolved	(10).
store_offdiagonals : This value determines how many offdiagonal elements of the combined GR matrix need to be stored	(10).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
store_wave_functions : If true then store wave functions within QTBM solver in order to be able to return them by get_data()	(false).
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature of the simulation	(300.0).

Parameter : Description	Default Value
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).

Table 5.114: Parameters of the QTBM_Propagation Solver.

5.115 RGFModule

[MODULE] Implements Recursive Greens Function algorithm

Parameter : Description	Default Value
$(scattering_self_energies)_constructor : constructor for scattering self-energy	().
$(scattering_self_energies)_solver : solver for scattering self-energy	().
BlockDiagonalRGF : If true, Recursive Green's Function Method from JAP 81, 7851 is followed, otherwise JAP 91, 2343	(false).
GPU_offload_ranks : vector of int, rank of GPU support.	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).

Parameter : Description	Default Value
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	().
__output : list of solver to trigger output command	().
__reinit : list of solver to trigger reinit command	().
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	().
__step : list of solver to trigger step command	().
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	().
add__readable__Propagators : list of string, name of propagators that need to be updated.	().
backward__rgf__offload__solver : Solver of type TaskOffload to be used in the "backward" part of RGF which consists heavily of dense matrix-matrix multiplications	().
boundary__regions : required regions as boundary conditions	().
bp__material__tags : controls buttiker probe tag defined in material.	().
calculate__local__current__from__right : solve slab resolved current using right to left formula for RGFv2.	(true).
calculate__spectral__in__backward : solve diagonal of spectral function A in backward RGF.	(true).

Parameter : Description	Default Value
check_current_conservation : check whether current is conserved from slab to slab.	(false).
chemical_potential_for_$(end_leads) : Chemical potential in eV of $(end_leads), usually drain contact	(0.0).
chemical_potential_for_$(start_leads) : Chemical potential in eV of $(start_leads), usually source contact	(0.0).
custom_mult : If set to true, the multiplication, G<=GR*Sigma<*GA is performed only for the diagonal elements of the matrices	(true).
debug_output : If set to true, produces outputs from Repartition solvers in Leads	(false).
density_Jacobian_is_ready : check density Jacobian is ready (solved) or not.	(false).
density_Jacobian_output : If true, propagation solver will trigger density Jacobian (derivative) solution and provide visual output file	(false).
density_of_states_output : output k resolved DOS.	(false).
density_output : If true, propagation solver will trigger density solution and provide visual output file	(false).
domain : Domain required for this simulation	().
end_leads : name of the lead that will be coupled to the last partition	(end_leads).
end_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green's functionin drain contact	().
energy_interpolated_output : do energy interpolation for result or not.	(false).
energy_resolved_density_of_states_output : output energy resolved DOS.	(false).
energy_resolved_density_output : output energy resolved density in individual files.	(false).

Parameter : Description	Default Value
external_Hamilton_Constructor : name of schroedinger solver; if defined, use external schroedinger solver, otherwise construct it in the module.	().
forward_rgf_offload_solver : Solver of type TaskOffload to be used in the "forward" part of RGF which consists heavily of dense matrix inversion	().
free_memory_of_matrix_block : release memory for RGF blocks (forward/backward) when it is not needed anymore.	(true).
invert_solver : Type of solver used for dense matrix inversions. Options are lapack, petsc, magma, magmamic	(lapack).
k_resolved_density_of_states_output : output k resolved DOS.	(false).
k_resolved_density_output : k resolved electron/hole density output.	(false).
lead_surface_green_$(end_leads) : name of the surface green's matrix, gR at the drain end. Usually named like: (end leads solver) Green Fermiondrain contact	().
lead_surface_green_$(start_leads) : name of the surface green's matrix, gL at the source end. Usually named like: (start leads solver) GreenFermion source contact	().
name : A unique name tag / identifier for the specific instance of the solver.	
no_integration_for_transmission : Turns ON or OFF k-space integration in the periodic directions	(false).
number_of_interpolation_points : number of interpolation points for output, e.g. energy resolved density.	(100).
number_of_subdomains : Number of subdomains for RGF calculation. Option for repartition solver	(1).

Parameter : Description	Default Value
old_mult_version : old version: use super matrix concept in direct_iterative_leads, much slower.	(false).
output_along_line : vector of double point, defines a line where energy resolved output is solved.	().
output_along_line_resolution : vector of int, defines the resolution of the line output.	().
output_file_suffix : suffix to be added on output files	().
output_slab_resolved_current : This outputs the slab resolved current as a function of the slab index	(false).
partition_direction : Direction used by Repartition solver. See Repartition option "partition_direction" for more info.	(0).
potential_solver : potential solver of the domain, e.g. poisson	().
repartition_strategy : Strategy used to set option "strategy" of Repartition solver	().
scattering_self_energies : propagator names for scattering self-energy	().
solve_again_for_Jacobian : solve the whole RGF again for density Jacobian.	(false).
spatial_resolved_current : output spatial/slab/atomic resolved current	(false).
start_leads : name of the lead that will be coupled to partition 0	(start_leads).
start_leads_solver : name of the solver(PartitionRetardedSurfaceGreenModule type) which does repartitioning and calcualtes surface green'sfunction in the source contact	().
store_DOS_only : store diagonal of spectral function (DOS) to minimize peak memory.	(true).
subdomains_for_atomic_resolved_current : list of sub-domain indices where atomic resolved current is solved within.	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature_for_$(end_leads) : Temperature in Kelvin for $(end_leads) contact	(300.0).
temperature_for_$(start_leads) :	(300.0Temperature in Kelvin for $(start_leads) contact).
tic_toc_name : Prefix for tic-tocs	($(name)).
transform_local_current : transform of local current (slab resolved).	(true).
transmission : If true, simulation provides transmission output	(false).
use_modified_correlation_equation : Used in RGFv2 for setting whether to assume g< is antihermitian or not. This assumption true in infinite precision but can cause stabilty issue with lambda*G scattering in finite precision.	(true).
use_smart_dofmap : reorder the device Hamiltonian to be block diagonal shape.	(true).

Table 5.115: Parameters of the RGFModule Solver.

5.116 Ramper

RamperSolver
This is a top-level simulation to run selfconsistent density-potential (or any other) iterations while ramping the voltages applied to contacts.
Parallelism: Ramper can parallelize the computation of voltage points, if desired (see the parallelization option).

Parameter : Description	Default Value
<contact name> : For each contact a list of applied voltages.All these lists should have the same number of entries, or else contain only a single entry at which the contact's voltage will be fixed for the entire simulation.	().
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
call_density_solver_output : If set to true, then the output routine of the density solver is called in each voltage setting. This routine would normally not be called since the density solver is typically not included in the solvers list of the Global section.	().
contacts : A list of the names of all contacts.	().
current_solver :	().
density_solver : Name of the density solver.	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
output_name : Filename base for output files.	().
parallelization : If set to true, then the output routine of the density solver is called in each voltage setting. This routine would normally not be called since the density solver is typically not included in the solvers list of the Global section.	().
potential_solver :	().
rappture_output : If true then some terminal output is generated that allows Rappture (nanoHUB's GUI language) to make status updates after every voltage step.	().
solver : Name of the solver whose solve-routine shall be called at every voltage setting.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.116: Parameters of the Ramper Solver.

5.117 ReadInCharge

Read_in_atomic_charge

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
inputfile : name of the inputfile with atom position and charge	().

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
output : This is a set of strings which defines the output to be generated. Currently, available options are charge and charge_derivative	().
output_file_suffix : suffix to be added on output files	().
potential_solver : name of the potential solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.117: Parameters of the ReadInCharge Solver.

5.118 ReadInCharge3D

Read_in_atomic_charge

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().

Parameter : Description	Default Value
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
fem_domain : Finite element domain that will provide a mesh	
get_atoms : Radius of the atomic density cloud [nm]	(0.5).
inputfile : name of the inputfile with atom position and charge	().
name : A unique name tag / identifier for the specific instance of the solver.	
output : This is a set of strings which defines the output to be generated. Currently, available options are charge and charge_derivative	().
output_file_suffix : suffix to be added on output files	().
potential_solver : name of the potential solver	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.118: Parameters of the ReadInCharge3D Solver.

5.119 ReadInPotential

ReadInPotentialSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
additional__domains : additional domains that needs potential data	( ).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
interpolation__power : parameter for interpolation function	(2).

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_interpolation_points : parameter for interpolation function	(4).
output_file_suffix : suffix to be added on output files	().
potential_file : file that contains x,y,z,v information	
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.119: Parameters of the ReadInPotential Solver.

5.120 ReadInPotentialWithAtoms

ReadInPotentialWithAtomsSolver

Parameter : Description	Default Value

Table 5.120: Parameters of the ReadInPotentialWithAtoms Solver.

5.121 ReadinHamiltonianModule

This module is reading in the Hamiltonian from file and providing it to RGFModule

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Table 5.121: Parameters of the ReadinHamiltonianModule Solver.

5.122 Reinit

This simulation reinitialize a set of solvers

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
solvers : list of solvers that need to be reinitialized.	(()).
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.122: Parameters of the Reinit Solver.

5.123 Repartition

Repartition Solver

Parameter : Description	Default Value
MPI_block_size : This option is used when spatial parallelization is enabled. In that case total number of processors must be divisible by this number	(1).
MPI_distribution : If set to true, spatial parallelization is enabled. In that case, a new domain is created and its atoms are distributed among different MPI processors	(false).
MPI_distribution_strategy : It determines the strategy of distributing this repartition process among several processors. default is set as OMEN	(1).
MPI_interleaving : This option enable BC Interleaving calculation	(false).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().

Parameter : Description	Default Value
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
activate_hydrogen_atoms : If set to true, hydrogen atoms are considered as a part of the domain and used in repartitioning	(false).
active_atoms_only : If set to true, we deal with the active atoms of the domain only	(true).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
coupling_based_on : In Huckel method the coupling is based on interaction	(Coupling).
create_new_domains : Create new domains for the partitions. There are applications where the new domains are not required, e.g. a "hard coded" RGF algorithm	(true).
domain : this is the name of the domain which will be repartitioned	().
domains_coupled_to_end : domains_for_last_rank	(()).
domains_coupled_to_start : domains_for_first_rank	(()).
end_boundary_regions : This is a vector of integers which correspond to the regions that will consist the end atoms for partitioning	().
end_leads : This is the name of the lead coupled to the end of the domain where repartitioning will end	().

Parameter : Description	Default Value
extra_name_element : This phrase will be added as a suffix to the new domain names	(std::string()).
name : A unique name tag / identifier for the specific instance of the solver.	
new_domain_name : This option is used for MPI distribution of the atoms of the domain. A new domain is created by this name and its atoms are distributed for parallelization	($(domain)__distributed).
number_of_subdomains : Total number of subdomains created after partitioning	(0).
output : Outputs data to file regarding domain indices, atomic partitions and partition sizes. Available options are: "domain_index", "atom_partition" and "partition_size"	().
output_file_suffix : suffix to be added on output files	().
partition_direction : 0 corresponds to repartitioning along x-direction or space_orientation_direction1 specified in input deck, 1 corresponds to y and 2 corresponds to z direction	(0).
start_boundary_regions : This is a vector of integers which correspond to the regions that will consist the start atoms for partitioning	().
start_from_one_contact : If set to true and start_leads, end_leads are not specified, it will automatically take first lead as the start_leads and other lead as the end_leads	(std::string()).
start_leads : This is the name of the lead coupled to the start of the domain where repartitioning will start from	().
strategy : Type of strategy use for repartition solver. It currently supports bfs, based_on_coordinate, based_on_unit_cells and based_on_regions	(based_on_coordinate).
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.123: Parameters of the Repartition Solver.

5.124 ResonanceFinder

Solver for finding resonance peaks in RTD structure and setting energy grid accordingly. Resonance peaks are at zeros of [EI-H-Sigma(E)]
This simulation has been ported from older code (rtdnegf Nanohub tool) by Zhengping Jiang. It calculates imaginary eigenvalues of the matrix $(H+\Sigma)$ and provides an energy grid to other simulations that is refined around these values. Use of such a refined energy grid is decisive for transport simulations in structures with sharp energetic resonances, such as RTDs.

Parameter : Description	Default Value
Device_Hamiltonian_constructor : Name of the Hamiltonian solver simulation which will provide information about Hamiltonian matrix.	
Drain_self_energy_constructor : Name of the drain self-energy solver which will provide information about drain self energy matrix.	
Ehi : Parameter used to refine the energy grid around conduction band edge	(0.006).
Emax : Maximum energy of inial guess for Newton's iteration. n energy values are initally guessed for n roots of [EI-H-Sigma(E)], where the dimension of H = n X n	(0.0).
Emin : Minimum energy of inial guess for Newton's iteration. n energy values are initally guessed for n roots of [EI-H-Sigma(E)], where the dimension of H = n X n	(0.0).
Newton_converge : convergence criteria for Newton's iteration which is used to solve the Nonlinear equation det[EI-H-Sigma(E)]	(1.0e-8).
Newton_step : energy step for Newton's iteration which is used to solve the Nonlinear equation det[EI-H-Sigma(E)]	(1.0e-10).

Parameter : Description	Default Value
Source_self_energy_constructor : Name of the source self-energy solver which will provide information about source self energy matrix.	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
exclusion_slabs : (i,j).Slabs 0 to i and j to end are not included in the resonance finding region.	().
gamma : Parameter used to refine the energy grid around conduction band edge	(0.006).

Parameter : Description	Default Value
init_step : Size of the energy segments for applying the SINL algorithm	(0.05).
iteration_limit1 : Maximum number of iterations in Newton refinement	(7).
maximum_resonance_number : max iterations for finding the resonance	(200).
name : A unique name tag / identifier for the specific instance of the solver.	
node_cb : Number of energy points around conduction band edge and 1D density of states	(30).
node_dfermi : Number of energy points for derivative of Fermi level	(0).
node_fermi : Number of energy points around Fermi level	(30).
node_homo : Number of energy points for homogeneous grid	(100).
node_pho_real_res : Number of energy points for real peaks of phonon retarded Green's function	(0).
node_pho_res : Number of energy points for phonon retarded Green's function	(0).
node_real_res : Number of energy points for real part of retarded Green's function	(30).
node_real_res2 : Number of energy points to refine real part of side peak	(0).
node_res : Number of energy points for resonance peaks and also imaginary part of retarded Green's function	(30).
node_res2 : Number of energy points to refine two sides of retarded Green's function peak. This is defined to be side peak	(30).
output : output list, choose from: 1.wf: wave function; 2.energy_grid: energy grid and grid density; 3.EMA_cb_profile: effective mass conduction band profile; 4.resonance: resonance energy	().
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
output_name : name for the output file	(string()).
print_all : If set to 1, debugging messages are printed throughout resonance finder execution	(0).
scale_factor : Scale factor used to calculate determinant for resonance solution at k point	(8.0).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Name for advanced timing output	(get_name()).

Table 5.124: Parameters of the ResonanceFinder Solver.

5.125 ResonanceFinderModule

This is the module for generating adaptive energy grid for transport calculation.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	()).
_output : list of solver to trigger output command	()).
_reinit : list of solver to trigger reinit command	()).
_search_flag : flag command to search nested solver	(2).

Parameter : Description	Default Value
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.125: Parameters of the ResonanceFinderModule Solver.

5.126 ResonanceMesh

ResonanceMesh: Creates a mesh that adapts to resonances of input functions

Parameter : Description	Default Value
$mesh_name+_defining_data : This defines the name of the function	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
initial__mesh__constructor : This defines a solver that gives the first result-independent mesh (e.g. AdaptiveGrid for energy)	(<this solver's name>).
mesh__name : This defines the name of the to-be created mesh	
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
solver__to__find__energy : This defines a propagation that is able to read energy from vectors of NemoMeshPoint	
surface__of__regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.126: Parameters of the ResonanceMesh Solver.

5.127 RetardedSelfModule

[MODULE] This solver calculates the retarded contact self-energies (based on Propagation class)

Parameter : Description	Default Value
$(momentum_names)_constructor : Names assigned to the Hamiltonian used for calculating momentum	
Hamilton_constructor : Names are assgined to Hamiltonian	().
Hamilton_constructor_$(lead_domain) : Names assigned to Hamiltonian for the lead	
Parallelizer : Point to a Parallelizer solver, where you can define the energy range and how many points you want	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).

Parameter : Description	Default Value
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : The regions are to be used for this simulation. It should be within the region numbers defined in geometry	
boundary__regions : required regions as boundary conditions	(()).
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).
destroy__auxiliary__solvers : Destroy auxiliary Schrodginer solvers after done with get_data.	(false).
do__output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
lead__domain : The domain that is connected to this PropagationModule. It can be used, for example, to get the coupling Hamiltonian between this domain and the $(lead_domain)	
linear__system__solver : linear solver type: mumps, petsc, pardiso, superlu	(mumps).
momentum__names : the names user used to define momentum K, like energy, momentum_1D	
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
partitioned_lead_domain : Name of the lead domain which was partitioned using domain_[0-3] or repartition solver.	().
partitions_starting_from_device : Domains 0-3 start from the beginning of the device if true and from end if false.	(true).
potential_solver : Name of the simulation that provides the electrostatic Potential	().
result_propagator_name : The name of the propagator that should be returned by the PropagationModule. It should correspond to the name in the NEGF object list for other Propagation solvers	
surface_greens_function_name : This string refers to the name of the output Matrix or surface green's function generated by the self energy solver	($name_retarded_Green_Fermion_dra
surface_of_regions : surface regions required to calculate boundary conditions	().
target_Hamilton_constructor : The simulation type that is to prove the Hamiltonian and DOF map of the target_domain.	
target_domain : Name of the domain that the self-energy will be used in	
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	().
temperature : Temperature (in kelvin) of the device to be simulated	(300.0).
tic_toc_name : Name for advanced timing output	($(name)).

Table 5.127: Parameters of the RetardedSelfModule Solver.

5.128 Schroedinger

This simulation solves the tight-binding schrodinger equation. It calculates eigenenergies and eigenstates as well as integrates the results to obtain electron and hole densities (if requested).

Band structure models: The band model is specified through the `tb_basis` parameter. Possible choices are the following:

`em`: Effective mass in a current-conserving finite difference discretization due to Frensley (It can be also found in S. Datta's book). The discretization also works for spatially vatying masses. Note that the `crystal_structure` parameter must be set to `simplecubic`.

`kp`: The 8 band k.p method is a perturbation theory based method treating the band structure with high precision but mainly around the vicinity of Γ point. Effect of strain is also treated very effectively without considering the symmetry reduction-induced geometrical and phase factor variation. Note that the `crystal_structure` must be set to `simplecubic`.

`s`: Effective mass in a fake tight-binding implementation. The obtained bulk band structure is parabolic only in the vicinity of Γ. The discretization has problems with inhomogeneous masses. Note that the `crystal_structure` parameter must be set to `simplecubic`.

`sp3sstar`, `sp3sstar_SO`, `sp3d5sstar`, `sp3d5sstar_SO`: 5-, 10- and 20-band tight-binding models. The suffix `_SO` specifies whether the spin-orbit interaction is included or not, doubling the system size. For purely electronic devices (no holes) the results without spin-orbit interaction can sometimes be sufficiently accurate.

`Pz`, `P/D`: 1- and 3-band tight binding models for graphene.

`ExtendedHuckel`: ask M. Povolotskyi about this.

Influence of strain on band structure: In tight-binding the effect of strain on the band structure is derived from the atomic positions. However, it is not sufficient to just do a strain simulation, the influence on the Hamiltonian must be explicitly switched on through the followin `job_list` options:

`include_strain_H`: Modifications according to Boykin's 2002 formulation.

`include_shear_strain_H`: Modifications according to Boykin's 2010 formulation.

When none of these options is included, a distorted crystal system will still produce a different band structure due to the change of bond angles and hence modified direction cosines in the tight-binding formula.
Currently there is no influence of strain on effective mass Hamiltonians.
Instead of doing a separate strain simulation one can superimpose a fixed strain by specifying a matrix in the input deck (see table).
Parallelism: `Schroedinger` employes the spatial parallelism given in the `Partitioning` section of the input deck. Additionally it distributes the k-points onto the different MPI-processes. This is done automatically.

Parameter : Description	Default Value
AP_vca_model : strained SiGe VCA model as implemented by A. Paul, IEEE EDL v31 no4 2010	(false).
BSR_block_size : Size of matrix blocks when using block sparse row (BSR) format with sparse_storage_format option; ignored if format is compressed sparse row (CSR)	(10).
Center_Position : related to region: (min+max)/2 for magnetic_vector_potential A	((0,0,0)).
DOS_broadening : (for DOS calculation) Broadening strength in eV.	(dE).
DOS_broadening_model : (for DOS calculation, default 2) 0: Lorentzian broadening, 1: Exponential broadening, 2: cosh broadening.	(2).
DOS_energy_grid : (for DOS calculation) Determines homogeneous energy grid in the form (Emin,Emax,dE).	(energy_input).
DOS_max_energy : (for DOS calculation) alternative specification of Emax.	(20.0).
DOS_min_energy : (for DOS calculation) alternative specification of Emin.	(-20.0).
DOS_points : (for DOS calculation) alternative specification of the number of energy points.	(100).
DOS_spin_factor : (for DOS calculation, default 1) optional multiplicative factor for DOS.	(1.0).
Dirac_gaussian_width : Width of Dirac delta function used for calculating scattering rate.	(1e-3).
Magnetic_Field : Magnetic field value in the form of (Bx, By, Bz)	().
Magnetic_Gauge : options: Symmetric (default) , Asymmetric_x , Asymmetric_y , Asymmetric_z.	(Symmetric).
Neumann_BC : Set Neumann BC for hamiltonian, which can generate a good basis for EM LRA, implemented by Lang Zeng	(false).

Parameter : Description	Default Value
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Peierls_Phase : true/false for applying peierls phase for nonzero magnetic field..	(true).
State_solver : Gets the external state solver	().
Zeeman_Splitting : true/false for applying zeeman splitting for nonzero magnetic field.	(true).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(().
_output : list of solver to trigger output command	(().
_reinit : list of solver to trigger reinit command	(().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(().
_step : list of solver to trigger step command	(().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(().

Parameter : Description	Default Value
allow_reinit_k_points : When Schroedinger solver is reinitialized, reinitialize the k (wave vector) space object in order to be able to change the values of k points. Implemented for use with band solvers associated with adaptive grid.	(false).
atom_order_from : Determines ordering of atoms in Hamiltonian when repartitioner solver is attached to DOFmap. Most likely, set this to the name of the repartition solver	().
automatic_threshold : If true, then the code automatically determines the energy threshold that separates electrons and holes (i.e. conduction and valence bands) by looking at the bulk band edges and the electrostatic potential. When iterative band structure solvers (like Krylov-Schur) are used this option causes the shift value to be equal to the threshold mentioned above. The actual value of the threshold depends on other options. For more details search for _automatic_threshold variable in the code above. This option is also used for density calculations in Schroedinger solver and not just band structure calculations. Possible values: true/false.	(false).
basis_generator : The (optional) name of the simulation object where the basis set for transforming the Hamiltonian is drawn from.	().
beta_degradation : Numerical parameter if eigensolver uses lanczos	(0.).
boundary_regions : required regions as boundary conditions	()).
brillouin_zone_meshsize : If calculate_brillouin_zone is true, this determines the mesh size. The spacing in every direction, in pi/a	(0.1).
calculate_automatic_shift_subbandbased : implemented by R. Kotlyar. When true, the convergence scheme based on determining automatic shifts from subband energies is specified.	(false).

Parameter : Description	Default Value
calculate_brillouin_zone : When set to true, the entire Brillouin zone will be meshed and computed.	(0.1).
calculate_elastic_overlap : calculates scattering-related form factors. deprecated	(false).
calculate_integrated_electron_density : implemented by R. Kotlyar. These options were added to fix the shift of the eigenvalue solver based on monitoring the subband energies from previ- ous iterations. Usually these shifts are determined by conduc- tion or valence band minima/maxima. Roza found that this convergence scheme works the best for high electric fields in the presence of large local band bendings.	(false).
calculate_integrated_hole_density : implemented by R. Kotlyar. If option is specified and spatially varying threshold = false, the integrated the total carrier density in cm-dim is computed by integration over the confinement dimensions (e.g. for wires this yields 1/cm). As similar option exists for electrons.	(false).
charge_model : In electron_hole you have negatively charged electrons and positively holes. In electron_core you have negatively charged electrons and positively charged ions. Can be electron_core or electron_hole. When set to electron hole (most cases), then an energy threshold separates between electrons and holes. When set to electron core model all states are assumed to be electrons	(electron_hole).
check_every_steps_number : Numerical parameter if eigensolver uses lanczos. Try a value between 20 and 500	(0).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).

Parameter : Description	Default Value
compute_edges_masses : If set to true, some band edges and masses will be computed and written to screen. This option was tested with bulk only and fails to compute correct transverse masses for X- and L- valleys in the presence of spin-orbit interaction.	(0.001).
compute_edges_masses_delta : delta K to calculate numerical derivative of the band dispersion. in pi/a. stepsize for the finite difference scheme employed in compute edges masses,	(0.001).
convergence_limit : Convergence limit for iterative solution methods (accuracy of eigenvalues).	(1e-8).
convergence_method : Numerical parameter if eigensolver uses lanczos. Can be full_convergence or partial_convergence	().
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
cutoff_distance_to_bandedge : Energy cutoff which is added to bandedge for holes and subtracted for electrons. Setting this to some value Delta (in eV) leads to electron thresholds at Ec - ePhi(x) - Alpha and hole thresholds at Ev - e Phi(x) + Delta.	(0.2*(Ec_min-Ev_max)).

Parameter : Description	Default Value
dE_for_elastic_overlap :	(0.0,Warning: option calculate_elastic_overlap is deprecated. Energy threshold for calculating overlap for elastic scattering. Overlap is calculated if difference between energy_for_elastic_overlap option and energy eigenvalue is smaller than dE. Only takes effect if take_all_states_and_no_overlap option is false.).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
direct_order_of_partitions_$neighbor_domain_name$: Orders neighbor domain partitions directly if true, in reverse if false. Considered only with neighbor domain repartitioner is provided.	(true).
domain : Domain required for this simulation	().
doping_density : Doping density of material. Only used if calculate_Fermi_level is included in job list. Doping type (i,n,p) set by option doping_type.	(0.0).
doping_type : Type of doping	(i).
e_xx : When giving strain as input, put epsilon_xx of strain tensor here(for tb_basis = kp only)	(0.01).

Parameter : Description	Default Value
e_xy : When giving strain as input, put epsilon_xy of strain tensor here(for tb_basis = kp only)	(0.00).
e_yy : When giving strain as input, put epsilon_yy of strain tensor here(for tb_basis = kp only)	(0.01).
e_yz : When giving strain as input, put epsilon_yz of strain tensor here(for tb_basis = kp only)	(0.00).
e_zx : When giving strain as input, put epsilon_zx of strain tensor here(for tb_basis = kp only)	(0.00).
e_zz : When giving strain as input, put epsilon_zz of strain tensor here(for tb_basis = kp only)	(0.01).
eff_mass_dump : true: for effective mass calculation only when tb_basis=kp/ false: for disabling it only when tb_basis=kp	(false).
eigen_values_solver : Which eigenvalue solver to use. Setting lapack always com- putes all eigenvalues and is feasible only for very small sys- tems. Recommended choices are krylovschur and arpack. Other choices are arnoldi, jd, gd.	().
electron_chem_pot : Chemical potential used to calculate density of states times Fermi derivative. Has units of eV.	(0.0).
electron_hole_heuristics : Given the atomically resolved energy threshold between electrons and holes, this option tells how to actually determine whether a particle is an electron or hole. Two options available: sharp (sharp transition at the threshold) and omen_smooth (smooth transition around the threshold, like in OMEN code)	(sharp).
electron_temperature : Temperature of the system	(300).
energy_for_elastic_overlap : Energy used for calculating matrix elements for wave functions	(0.0).

Parameter : Description	Default Value
energy_maximum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_minimum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_threshold_for_shift : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enthresh_foresolvershift).
epsilon_matrix : Constant strain tensor matrix, given in laboratory system co- ordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the laboratory system where crystal axis might have been rotated.	(strain_matrix).
epsilon_matrix_crystal : Constant strain tensor matrix, given in crystal system coordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the crystallographic system.	(strain_matrix).
extra_points : Extra K points set by user	(NULL).
fixed_shift : Sets the shift to be used by eigenvalue solvers (except when electron_core model: check option shift).	(0.0).
gaussian_quadrature_order_automatic : Affects k-space integration for periodic directions. If option integration_order is not defined and if gaussian_quadrature_order_automatic = true, then for number_of_k_points <= 22 use Gaussian quadrature with single partition, otherwise use simple rectangular with number of elements equal to number_of_k_points. In both cases the actual number of k points equals number_of_k_points.	(false).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
init_check_steps_number : lanczos option, for checking convergence after how many iterations	(20).
input_shift_subbandbased_for_restart : When true, the shift is read in from file and is used as initial shift to eigensolver. R. Kotylar	(false).
input_shiftfilename_for_restart : Filename for input_shift_subbandbased_for_restart. R. Kotylar	(string()).
input_stress_or_strain : stress: for giving stress as input/ strain: for giving strain as input directly(for tb_basis = kp only)	(strain).
integration_order : Sets the order of integration for Gaussian quadrature method	(1).
is_kp_strain_on : true: for enabling strain calculation with k.p/ false: for disabling strain calculation with k.p(for tb_basis = kp only)	(false).
is_piezoelectric_effect_on : true/false; for enabling piezoelectric calculation with k.p;(for k.p simulation only)	(false).
is_spatially_parallelized : true/false; Whether the device is space parallelized;(for k.p simulation only)	(false).
is_vff_strain_on : true/false; for enabling VFF strain calculation with k.p;(for k.p simulation only)	(false).
job_list : A list that deterimes what is done. Choose from assemble_H, passivate_H, include_strain_H, include_shear_strain_H, calculate_band_structure, electron_density, derivative_electron_density_over_potential, hole_density, derivative_hole_density_over_potential, spin, DOS. assemble_H is activated by any other option automatically.	(job_list).
k0 : k0 vector to solve eigenvalues at this particular k-point	([0 0 0]).

Parameter : Description	Default Value
k_degeneracy : The computed density is multiplied by this number to account for k-space degeneracy. E.g. when kxmax and kymax are set in a simulation with 2D k-space, k degeneracy should be 4.	(1.0).
k_points : Only relevant for calculate band structure. This parame- ter is a list of points in k-space along which the band structure is calculated.	(k_pointsV).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_basis : (Default: cartesian) Sets the basis in which k points is specified. When set to reciprocal the coordinates given in k points are assumed to be w.r.t. the reciprocal lattice vectors b1 =a2xa3 / a1.(a2xa3) etc.	(reciprocal).
kp_band_number : 8 band for direct band gap material / 6 for indirect band gap material / 4 for rhombohedral crystals of TI(for tb_basis = kp only)	(8).
kp_strain_dir1 : If stress is perpendicular to [x y z] plane then input x here(for tb_basis = kp only)	(1).
kp_strain_dir2 : If stress is perpendicular to [x y z] plane then input y here(for tb_basis = kp only)	(0).
kp_strain_dir3 : If stress is perpendicular to [x y z] plane then input z here(for tb_basis = kp only)	(0).
kp_stress_mag : When giving stress as input, put stress in GPa here(for tb_basis = kp only)	(1).
kpoint_to_monitor_subbands_for_shift : The k-point where to look for shift. default is 0. R. Kotylar	(kmagref_foresolvershift).
kpoint_tolerance_to_monitor_subbands_for_shift : The k-tolerance when looking for a k-point on k-grid. Default is 0.001. R. Kotylar	(kmagreftol_foresolvershift).

Parameter : Description	Default Value
kxmax : Upper boundary in 2 pi/a of the simulated kx space. kx ranges from 0 to this number.	(1.0).
kxmin : Lower boundary in 2pi/a range of the simulated kx space. kx begins from zero by default.	(0.0).
kymax : Upper boundary in 2 pi/a of the simulated kx space. ky ranges from 0 to this number.	(1.0).
kymin :	(0.0,Lower boundary in 2pi/a range of the simulated ky space. ky begins from zero by default.).
kzmax : Upper boundary in 2 pi/a of the simulated kz space. kz ranges from 0 to this number.	(1.0).
kzmin :	(0.0,Lower boundary in 2pi/a range of the simulated kz space. kz begins from zero by default.).
lanczos_iseed : Numerical parameter if eigensolver uses lanczos	(0.).
linear_solver : Linear solver employed in the shift-and-invert operation. This should be preferable a direct linear solver since the LU fac- torization can be reused during the Krylov iterations.	().
max_number_iterations : Maximum number of Krylov iterations (irrelevant for lapack).	(0).
maximum_delta_k : Upper bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(1e80).

Parameter : Description	Default Value
maximum_scattering_angle : Option connected to computation of scattering matrix elements	(180.0).
minimum_delta_k : Lower bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(0.0).
minimum_scattering_angle : Option connected to computation of scattering matrix elements	(0.0).
modes_for_current : Relevant for current calculation. Define the number of modes to be extracted from E-k while calc. ballistic TOB current.	().
monitor_convergence : When set to true, terminal output related to the Krylov iteration is generated.	(false).
mpd : Specialized numerical option. Refer to the SLEPc user manual for the meaning.	(0).
mumps_ordering : Relevant only for eigen values solver=mumps. Choose between pord and metis.	().
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : Specialized numerical option (size of the Krylov subspace).	(0).
no_of_spatial_division : No of cores used for parallelizing the device(for tb_basis = kp only)	(256).
noise_potential : noise potential used in qtbm to break degenerate modes.	(0.0).
number_of_eigenvalues : Number of eigenvalues to compute (irrelevant for lapack).	(0).
number_of_eigenvalues_to_use : Number of requested eigenvalues	(0).
number_of_k_points : This list gives the uniform discretization of each segment set by the k points parameter (note that specifying N points means N - 1 segments). For density calculations, setting this parameter to 0 leads to computation of k = 0 only and application of an analytical formula that assumes parabolic subbands.	(10).

Parameter : Description	Default Value
number_of_nodes : number of nodes for each k-segment	(()).
output : Choose from Hamiltonian, energies, k-points, DOS, electron density, electron density VTK, hole density, hole density VTK, ion density, eigenfunctions, eigenfunctions k0, eigenfunctions VTK, eigenfunctions VTK k0, eigenfunctions Silo, eigenfunctions Silo k0, spin.	(output_list_v).
output_eigenstate_energy : Energy around which eigenpair has to be output.	(0.0).
output_file_suffix : suffix to be added on output files	().
output_k_point : Value of k-point dumped as output	(k0).
output_precision : Accuracy of saved eigenvalues within output file.	(12).
output_shift_subbandbased_for_restart : When true, at the end of simulation the file can be output with the last eigensolver shift information. Using this shift and the last potential, the next simulation can be restarted from the last state of simulation. R. Kotylar.	(false).
p_parameter : It's the block size for block Lanczos. If the degeneracy of the eigenvalues is known, then set it to that value, otherwise start with 4.	(0).
parallelize_here : When set to false, some other simulation (like Propagator) can determine the parallelization.	(true).
potential_solver : The (optional) name of the simulation object where the electrostatic potential is drawn from.	().
preconditioner : This option controls how the eigenvalue problem is transformed before solution Possible options are: "lu" "mumps"...work ongoing . (default: lu) Preconditioner employed in the shift-and-invert operation.	(lu).

Parameter : Description	Default Value
project_DOS_ : This option allows to calculate the project density of states	(false).
rate_of_shift_changes_between_iterations : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enrate_foresolvershift).
reinit_parallelize : When Schroedinger solver is reinitialized, reinitialize the MPI parallelization as well.	(false).
resolution : The resolution to be used for the eigensolver	(0.0).
reuse_hamiltonian : if this option is false, hamiltonian would be recalculated every iteration, for different k always it is recalculated	(true).
seed : seed for noise potential.	(-1).
set_valley : set valley name to get Material properties	(DEFAULT).
shift : The eigensolver will search for eigenvalues around the shift value. This seems to only be used for electron_core model. fixed_shift is used otherwise.	(0.001).
solve_on_single_replica : Gather density to the single rank only	(false).
solver_transformation_type : (default: sinvert) Use sinvert for shift-and-invert, shift for shift.	(sinvert).
sort_like_electrons : This option controls the way the eigenenergies are sorted. When true (electrons) then ascending and when false (holes) then descending. Possible values: true/false.	(true if job list has electron density or band structure or electron core model. false if job list has hole density.).
sparse_storage_format : Sparse storage format in which PETSc will allocate sparse matrices; options available are "csr" (compressed sparse row) and "bsr" (block sparse row)	(csr).

Parameter : Description	Default Value
spatially_varying_threshold : If true, then the threshold energy discriminating between electrons and holes will vary spatially according to the elec- trostatic potential. Needed in Schroedinger-Poisson simulations.	(false).
spin_resolved : if this option is enabled all parameters will be spin dependant Up/Down	(false).
strain_calc_option : hydrostatic: for calculating the effect of hydrostatic strain/ biaxial: for calculating the effect of biaxial strain/ uniaxial: for calculating the effect of uniaxial strain (use this option only when giving strain as input)(for tb_basis = kp only)	(uniaxial).
strict_threshold : Solve for either conduction (electron density) or valence (hole density) bands depending on option sort_like_electrons. Possible values: true/false.	(true if job list has electron density or hole density. false if job list has band structure or if electron_core charge model is used.).
surface_of_regions : surface regions required to calculate boundary conditions	().
take_all_states_and_no_overlap : This needs the option "calculate_elastic_overlap" to be true. Allows to switch the Schroedinger from solving the scattering overlap for a subset of states to collect all states and make them ready for other solvers to get via get_data. Primary purpose of this is to support the fitting of TB to DFT including the TB wavefunctions	(false).
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	(sp3).
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Tic toc name to be used for saving timing results for this simulation.	().

Parameter : Description	Default Value
tolerance : The judgement of the convergence. Once the target reaches the tolerance then end the loop.	(0.0).
use_monomials : calculate monomials table based on the orbitals	(true).
use_neighbour_repartitioner_$neighbor_domain_name$: Repartitioner solver name for neighbor domain.	()

Table 5.128: Parameters of the Schroedinger Solver.

5.129 SchroedingerDensity

SchroedingerDensitySolver

Parameter : Description	Default Value
Emax : Maximum value for the energy grid	
Emin : Minimum value for the energy grid	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).

Parameter : Description	Default Value
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
automatic_energy_window : Energy windows based on conduction band and fermi levels	(false).
boundary_regions : required regions as boundary conditions	(()).
chemical_potential : Full chemical potential [eV]	
debug_output : Do debugging output?	(false).
domain : Domain required for this simulation	().
integration_order : Integration order over the k-mesh	(1).
kxmax : Right boundary of k-space kx axis	(1.0).
kxmin : Left boundary of k-space kx axis	(0.0).
kymax : Right boundary of k-space ky axis	(1.0).
kymin : Left boundary of k-space ky axis	(0.0).
kzmax : Right boundary of k-space kz axis	(1.0).
kzmin : Left boundary of k-space kz axis	(0.0).
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_k_points : Number of nodes in a k-mesh along principal axis	(10).
number_of_kx_points : Number of nodes in a k-mesh along kx axis	(number_of_k_points).
number_of_ky_points : Number of nodes in a k-mesh along ky axis	(number_of_k_points).
number_of_kz_points : Number of nodes in a k-mesh along kz axis	(number_of_k_points).
output_file_suffix : suffix to be added on output files	().

Parameter : Description	Default Value
passivation : apply the surface state passivation empirical self-energy	(true).
potential_solver : Solver that can provide electric potential on atoms	
strain_correction : strain correction jobs for Schroedinger	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.129: Parameters of the SchroedingerDensity Solver.

5.130 Schroedinger_huckel

This simulation solves the tight-binding schrodinger equation. It calculates eigenenergies and eigenstates as well as integrates the results to obtain electron and hole densities (if requested).

Parameter : Description	Default Value
AP_vca_model : strained SiGe VCA model as implemented by A. Paul, IEEE EDL v31 no4 2010	(false).
BSR_block_size : Size of matrix blocks when using block sparse row (BSR) format with sparse_storage_format option; ignored if format is compressed sparse row (CSR)	(10).
Center_Position : related to region: (min+max)/2 for magnetic_vector_potential A	((0,0,0)).
DOS_broadening : (for DOS calculation) Broadening strength in eV.	(dE).
DOS_broadening_model : (for DOS calculation, default 2) 0: Lorentzian broadening, 1: Exponential broadening, 2: cosh broadening.	(2).

Parameter : Description	Default Value
DOS_energy_grid : (for DOS calculation) Determines homogeneous energy grid in the form (Emin,Emax,dE).	(energy_input).
DOS_max_energy : (for DOS calculation) alternative specification of Emax.	(20.0).
DOS_min_energy : (for DOS calculation) alternative specification of Emin.	(-20.0).
DOS_points : (for DOS calculation) alternative specification of the number of energy points.	(100).
DOS_spin_factor : (for DOS calculation, default 1) optional multiplicative factor for DOS.	(1.0).
Dirac_gaussian_width : Width of Dirac delta function used for calculating scattering rate.	(1e-3).
Magnetic_Field : Magnetic field value in the form of (Bx, By, Bz)	().
Magnetic_Gauge : options: Symmetric (default) , Asymmetric_x , Asymmetric_y , Asymmetric_z.	(Symmetric).
Neumann_BC : Set Neumann BC for hamiltonian, which can generate a good basis for EM LRA, implemented by Lang Zeng	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Peierls_Phase : true/false for applying peierls phase for nonzero magnetic field..	(true).
State_solver : Gets the external state solver	().
Zeeman_Splitting : true/false for applying zeeman splitting for nonzero magnetic field.	(true).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).

Parameter : Description	Default Value
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
allow__reinit__k__points : When Schroedinger solver is reinitialized, reinitialize the k (wave vector) space object in order to be able to change the values of k points. Implemented for use with band solvers associated with adaptive grid.	(false).
atom__order__from : Determines ordering of atoms in Hamiltonian when repartitioner solver is attached to DOFmap. Most likely, set this to the name of the repartition solver	().

Parameter : Description	Default Value
automatic_threshold : If true, then the code automatically determines the energy threshold that separates electrons and holes (i.e. conduction and valence bands) by looking at the bulk band edges and the electrostatic potential. When iterative band structure solvers (like Krylov-Schur) are used this option causes the shift value to be equal to the threshold mentioned above. The actual value of the threshold depends on other options. For more details search for _automatic_threshold variable in the code above. This option is also used for density calculations in Schroedinger solver and not just band structure calculations. Possible values: true/false.	(false).
basis_generator : The (optional) name of the simulation object where the basis set for transforming the Hamiltonian is drawn from.	().
beta_degradation : Numerical parameter if eigensolver uses lanczos	(0.).
boundary_regions : required regions as boundary conditions	(().
brillouin_zone_meshsize : If calculate_brillouin_zone is true, this determines the mesh size. The spacing in every direction, in pi/a	(0.1).
calculate_automatic_shift_subbandbased : implemented by R. Kotlyar. When true, the convergence scheme based on determining automatic shifts from subband energies is specified.	(false).
calculate_brillouin_zone : When set to true, the entire Brillouin zone will be meshed and computed.	(0.1).
calculate_elastic_overlap : calculates scattering-related form factors. deprecated	(false).

Parameter : Description	Default Value
calculate_integrated_electron_density : implemented by R. Kotlyar. These options were added to fix the shift of the eigenvalue solver based on monitoring the subband energies from previ- ous iterations. Usually these shifts are determined by conduc- tion or valence band minima/maxima. Roza found that this convergence scheme works the best for high electric fields in the presence of large local band bendings.	(false).
calculate_integrated_hole_density : implemented by R. Kotlyar. If option is specified and spatially varying threshold = false, the integrated the total carrier density in cm-dim is computed by integration over the confinement dimensions (e.g. for wires this yields 1/cm). As similar option exists for electrons.	(false).
charge_model : In electron_hole you have negatively charged electrons and positively holes. In electron_core you have negatively charged electrons and positively charged ions. Can be electron_core or electron_hole. When set to electron hole (most cases), then an energy threshold separates between electrons and holes. When set to electron core model all states are assumed to be electrons	(electron_hole).
check_every_steps_number : Numerical parameter if eigensolver uses lanczos. Try a value between 20 and 500	(0).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
compute_edges_masses : If set to true, some band edges and masses will be computed and written to screen. This option was tested with bulk only and fails to compute correct transverse masses for X- and L- valleys in the presence of spin-orbit interaction.	(0.001).

Parameter : Description	Default Value
compute_edges_masses_delta : delta K to calculate numerical derivative of the band dispersion. in pi/a. stepsize for the finite difference scheme employed in compute edges masses,	(0.001).
convergence_limit : Convergence limit for iterative solution methods (accuracy of eigenvalues).	(1e-8).
convergence_method : Numerical parameter if eigensolver uses lanczos. Can be full_convergence or partial_convergence	().
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).
cutoff_distance_to_bandedge : Energy cutoff which is added to bandedge for holes and subtracted for electrons. Setting this to some value Delta (in eV) leads to electron thresholds at Ec - ePhi(x) - Alpha and hole thresholds at Ev - e Phi(x) + Delta.	(0.2*(Ec_min-Ev_max)).

Parameter : Description	Default Value
dE_for_elastic_overlap :	(0.0,Warning: option calculate_elastic_overlap is deprecated. Energy threshold for calculating overlap for elastic scattering. Overlap is calculated if difference between energy_for_elastic_overlap option and energy eigenvalue is smaller than dE. Only takes effect if take_all_states_and_no_overlap option is false.).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
direct_order_of_partitions_$neighbor_domain_name$: Orders neighbor domain partitions directly if true, in reverse if false. Considered only with neighbor domain repartitioner is provided.	(true).
domain : Domain required for this simulation	().
doping_density : Doping density of material. Only used if calculate_Fermi_level is included in job list. Doping type (i,n,p) set by option doping_type.	(0.0).
doping_type : Type of doping	(i).
e_xx : When giving strain as input, put epsilon_xx of strain tensor here(for tb_basis = kp only)	(0.01).

Parameter : Description	Default Value
e_xy : When giving strain as input, put epsilon_xy of strain tensor here(for tb_basis = kp only)	(0.00).
e_yy : When giving strain as input, put epsilon_yy of strain tensor here(for tb_basis = kp only)	(0.01).
e_yz : When giving strain as input, put epsilon_yz of strain tensor here(for tb_basis = kp only)	(0.00).
e_zx : When giving strain as input, put epsilon_zx of strain tensor here(for tb_basis = kp only)	(0.00).
e_zz : When giving strain as input, put epsilon_zz of strain tensor here(for tb_basis = kp only)	(0.01).
eff_mass_dump : true: for effective mass calculation only when tb_basis=kp/ false: for disabling it only when tb_basis=kp	(false).
eigen_values_solver : Which eigenvalue solver to use. Setting lapack always com- putes all eigenvalues and is feasible only for very small sys- tems. Recommended choices are krylovschur and arpack. Other choices are arnoldi, jd, gd.	().
electron_chem_pot : Chemical potential used to calculate density of states times Fermi derivative. Has units of eV.	(0.0).
electron_hole_heuristics : Given the atomically resolved energy threshold between electrons and holes, this option tells how to actually determine whether a particle is an electron or hole. Two options available: sharp (sharp transition at the threshold) and omen_smooth (smooth transition around the threshold, like in OMEN code)	(sharp).
electron_temperature : Temperature of the system	(300).
energy_for_elastic_overlap : Energy used for calculating matrix elements for wave functions	(0.0).

Parameter : Description	Default Value
energy_maximum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_minimum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_threshold_for_shift : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enthresh_foresolvershift).
epsilon_matrix : Constant strain tensor matrix, given in laboratory system co- ordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the laboratory system where crystal axis might have been rotated.	(strain_matrix).
epsilon_matrix_crystal : Constant strain tensor matrix, given in crystal system coordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the crystallographic system.	(strain_matrix).
extra_points : Extra K points set by user	(NULL).
fixed_shift : Sets the shift to be used by eigenvalue solvers (except when electron_core model: check option shift).	(0.0).
gaussian_quadrature_order_automatic : Affects k-space integration for periodic directions. If option integration_order is not defined and if gaussian_quadrature_order_automatic = true, then for number_of_k_points <= 22 use Gaussian quadrature with single partition, otherwise use simple rectangular with number of elements equal to number_of_k_points. In both cases the actual number of k points equals number_of_k_points.	(false).
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).

Parameter : Description	Default Value
init_check_steps_number : lanczos option, for checking convergence after how many iterations	(20).
input_shift_subbandbased_for_restart : When true, the shift is read in from file and is used as initial shift to eigensolver. R. Kotylar	(false).
input_shiftfilename_for_restart : Filename for input_shift_subbandbased_for_restart. R. Kotylar	(string()).
input_stress_or_strain : stress: for giving stress as input/ strain: for giving strain as input directly(for tb_basis = kp only)	(strain).
integration_order : Sets the order of integration for Gaussian quadrature method	(1).
is_kp_strain_on : true: for enabling strain calculation with k.p/ false: for disabling strain calculation with k.p(for tb_basis = kp only)	(false).
is_piezoelectric_effect_on : true/false; for enabling piezoelectric calculation with k.p;(for k.p simulation only)	(false).
is_spatially_parallelized : true/false; Whether the device is space parallelized;(for k.p simulation only)	(false).
is_vff_strain_on : true/false; for enabling VFF strain calculation with k.p;(for k.p simulation only)	(false).
job_list : A list that deterimes what is done. Choose from assemble_H, passivate_H, include_strain_H, include_shear_strain_H, calculate_band_structure, electron_density, derivative_electron_density_over_potential, hole_density, derivative_hole_density_over_potential, spin, DOS. assemble_H is activated by any other option automatically.	(job_list).
k0 : k0 vector to solve eigenvalues at this particular k-point	([0 0 0]).

Parameter : Description	Default Value
k_degeneracy : The computed density is multiplied by this number to account for k-space degeneracy. E.g. when kxmax and kymax are set in a simulation with 2D k-space, k degeneracy should be 4.	(1.0).
k_points : Only relevant for calculate band structure. This parame- ter is a list of points in k-space along which the band structure is calculated.	(k_pointsV).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_basis : (Default: cartesian) Sets the basis in which k points is specified. When set to reciprocal the coordinates given in k points are assumed to be w.r.t. the reciprocal lattice vectors b1 =a2xa3 / a1.(a2xa3) etc.	(reciprocal).
kp_band_number : 8 band for direct band gap material / 6 for indirect band gap material / 4 for rhombohedral crystals of TI(for tb_basis = kp only)	(8).
kp_strain_dir1 : If stress is perpendicular to [x y z] plane then input x here(for tb_basis = kp only)	(1).
kp_strain_dir2 : If stress is perpendicular to [x y z] plane then input y here(for tb_basis = kp only)	(0).
kp_strain_dir3 : If stress is perpendicular to [x y z] plane then input z here(for tb_basis = kp only)	(0).
kp_stress_mag : When giving stress as input, put stress in GPa here(for tb_basis = kp only)	(1).
kpoint_to_monitor_subbands_for_shift : The k-point where to look for shift. default is 0. R. Kotylar	(kmagref_foresolvershift).
kpoint_tolerance_to_monitor_subbands_for_shift : The k-tolerance when looking for a k-point on k-grid. Default is 0.001. R. Kotylar	(kmagreftol_foresolvershift).

Parameter : Description	Default Value
kxmax : Upper boundary in 2 pi/a of the simulated kx space. kx ranges from 0 to this number.	(1.0).
kxmin : Lower boundary in 2pi/a range of the simulated kx space. kx begins from zero by default.	(0.0).
kymax : Upper boundary in 2 pi/a of the simulated kx space. ky ranges from 0 to this number.	(1.0).
kymin :	(0.0,Lower boundary in 2pi/a range of the simulated ky space. ky begins from zero by default.).
kzmax : Upper boundary in 2 pi/a of the simulated kz space. kz ranges from 0 to this number.	(1.0).
kzmin :	(0.0,Lower boundary in 2pi/a range of the simulated kz space. kz begins from zero by default.).
lanczos_iseed : Numerical parameter if eigensolver uses lanczos	(0.).
linear_solver : Linear solver employed in the shift-and-invert operation. This should be preferable a direct linear solver since the LU fac- torization can be reused during the Krylov iterations.	().
max_number_iterations : Maximum number of Krylov iterations (irrelevant for lapack).	(0).
maximum_delta_k : Upper bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(1e80).

Parameter : Description	Default Value
maximum_scattering_angle : Option connected to computation of scattering matrix elements	(180.0).
minimum_delta_k : Lower bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(0.0).
minimum_scattering_angle : Option connected to computation of scattering matrix elements	(0.0).
modes_for_current : Relevant for current calculation. Define the number of modes to be extracted from E-k while calc. ballistic TOB current.	().
monitor_convergence : When set to true, terminal output related to the Krylov iteration is generated.	(false).
mpd : Specialized numerical option. Refer to the SLEPc user manual for the meaning.	(0).
mumps_ordering : Relevant only for eigen values solver=mumps. Choose between pord and metis.	().
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : Specialized numerical option (size of the Krylov subspace).	(0).
no_of_spatial_division : No of cores used for parallelizing the device(for tb_basis = kp only)	(256).
noise_potential : noise potential used in qtbm to break degenerate modes.	(0.0).
number_of_eigenvalues : Number of eigenvalues to compute (irrelevant for lapack).	(0).
number_of_eigenvalues_to_use : Number of requested eigenvalues	(0).
number_of_k_points : This list gives the uniform discretization of each segment set by the k points parameter (note that specifying N points means N - 1 segments). For density calculations, setting this parameter to 0 leads to computation of k = 0 only and application of an analytical formula that assumes parabolic subbands.	(10).

Parameter : Description	Default Value
number_of_nodes : number of nodes for each k-segment	(()).
output : Choose from Hamiltonian, energies, k-points, DOS, electron density, electron density VTK, hole density, hole density VTK, ion density, eigenfunctions, eigenfunctions k0, eigenfunctions VTK, eigenfunctions VTK k0, eigenfunctions Silo, eigenfunctions Silo k0, spin.	(output_list_v).
output_eigenstate_energy : Energy around which eigenpair has to be output.	(0.0).
output_file_suffix : suffix to be added on output files	().
output_k_point : Value of k-point dumped as output	(k0).
output_precision : Accuracy of saved eigenvalues within output file.	(12).
output_shift_subbandbased_for_restart : When true, at the end of simulation the file can be output with the last eigensolver shift information. Using this shift and the last potential, the next simulation can be restarted from the last state of simulation. R. Kotylar.	(false).
p_parameter : It's the block size for block Lanczos. If the degeneracy of the eigenvalues is known, then set it to that value, otherwise start with 4.	(0).
parallelize_here : When set to false, some other simulation (like Propagator) can determine the parallelization.	(true).
potential_solver : The (optional) name of the simulation object where the electrostatic potential is drawn from.	().
preconditioner : This option controls how the eigenvalue problem is transformed before solution Possible options are: "lu" "mumps"...work ongoing . (default: lu) Preconditioner employed in the shift-and-invert operation.	(lu).

Parameter : Description	Default Value
project_DOS_ : This option allows to calculate the project density of states	(false).
rate_of_shift_changes_between_iterations : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enrate_foresolvershift).
reinit_parallelize : When Schroedinger solver is reinitialized, reinitialize the MPI parallelization as well.	(false).
resolution : The resolution to be used for the eigensolver	(0.0).
reuse_hamiltonian : if this option is false, hamiltonian would be recalculated every iteration, for different k always it is recalculated	(true).
seed : seed for noise potential.	(-1).
set_valley : set valley name to get Material properties	(DEFAULT).
shift : The eigensolver will search for eigenvalues around the shift value. This seems to only be used for electron_core model. fixed_shift is used otherwise.	(0.001).
solve_on_single_replica : Gather density to the single rank only	(false).
solver_transformation_type : (default: sinvert) Use sinvert for shift-and-invert, shift for shift.	(sinvert).
sort_like_electrons : This option controls the way the eigenenergies are sorted. When true (electrons) then ascending and when false (holes) then descending. Possible values: true/false.	(true if job list has electron density or band structure or electron core model. false if job list has hole density.).
sparse_storage_format : Sparse storage format in which PETSc will allocate sparse matrices; options available are "csr" (compressed sparse row) and "bsr" (block sparse row)	(csr).

Parameter : Description	Default Value
spatially_varying_threshold : If true, then the threshold energy discriminating between electrons and holes will vary spatially according to the elec- trostatic potential. Needed in Schroedinger-Poisson simulations.	(false).
spin_resolved : if this option is enabled all parameters will be spin dependant Up/Down	(false).
strain_calc_option : hydrostatic: for calculating the effect of hydrostatic strain/ biaxial: for calculating the effect of biaxial strain/ uniaxial: for calculating the effect of uniaxial strain (use this option only when giving strain as input)(for tb_basis = kp only)	(uniaxial).
strict_threshold : Solve for either conduction (electron density) or valence (hole density) bands depending on option sort_like_electrons. Possible values: true/false.	(true if job list has electron density or hole density. false if job list has band structure or if electron_core charge model is used.).
surface_of_regions : surface regions required to calculate boundary conditions	().
take_all_states_and_no_overlap : This needs the option "calculate_elastic_overlap" to be true. Allows to switch the Schroedinger from solving the scattering overlap for a subset of states to collect all states and make them ready for other solvers to get via get_data. Primary purpose of this is to support the fitting of TB to DFT including the TB wavefunctions	(false).
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	(sp3).
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Tic toc name to be used for saving timing results for this simulation.	().

Parameter : Description	Default Value
tolerance : The judgement of the convergence. Once the target reaches the tolerance then end the loop.	(0.0).
use_monomials : calculate monomials table based on the orbitals	(true).
use_neighbour_repartitioner_$neighbor_domain_name$: Repartitioner solver name for neighbor domain.	()

Table 5.130: Parameters of the Schroedinger_huckel Solver.

5.131 Schroedinger_intra_atomic

This simulation solves the tight-binding schrodinger equation. It calculates eigenenergies and eigenstates as well as integrates the results to obtain electron and hole densities (if requested).

Parameter : Description	Default Value
AP_vca_model : strained SiGe VCA model as implemented by A. Paul, IEEE EDL v31 no4 2010	(false).
BSR_block_size : Size of matrix blocks when using block sparse row (BSR) format with sparse_storage_format option; ignored if format is compressed sparse row (CSR)	(10).
Center_Position : related to region: (min+max)/2 for magnetic_vector_potential A	((0,0,0)).
DOS_broadening : (for DOS calculation) Broadening strength in eV.	(dE).
DOS_broadening_model : (for DOS calculation, default 2) 0: Lorentzian broadening, 1: Exponential broadening, 2: cosh broadening.	(2).
DOS_energy_grid : (for DOS calculation) Determines homogeneous energy grid in the form (Emin,Emax,dE).	(energy_input).
DOS_max_energy : (for DOS calculation) alternative specification of Emax.	(20.0).

Parameter : Description	Default Value
DOS_min_energy : (for DOS calculation) alternative specification of Emin.	(-20.0).
DOS_points : (for DOS calculation) alternative specification of the number of energy points.	(100).
DOS_spin_factor : (for DOS calculation, default 1) optional multiplicative factor for DOS.	(1.0).
Dirac_gaussian_width : Width of Dirac delta function used for calculating scattering rate.	(1e-3).
Magnetic_Field : Magnetic field value in the form of (Bx, By, Bz)	().
Magnetic_Gauge : options: Symmetric (default) , Asymmetric_x , Asymmetric_y , Asymmetric_z.	(Symmetric).
Neumann_BC : Set Neumann BC for hamiltonian, which can generate a good basis for EM LRA, implemented by Lang Zeng	(false).
OMEN_style_transverse_k_points : Construct the same transverse k (wave vector) grid as in OMEN. This was necessary in order to do benchmarking against OMEN. For now implemented for UTB only (1D transverse k).	(false).
Peierls_Phase : true/false for applying peierls phase for nonzero magnetic field..	(true).
State_solver : Gets the external state solver	().
Zeeman_Splitting : true/false for applying zeeman splitting for nonzero magnetic field.	(true).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
allow__reinit__k__points : When Schroedinger solver is reinitialized, reinitialize the k (wave vector) space object in order to be able to change the values of k points. Implemented for use with band solvers associated with adaptive grid.	(false).
atom__order__from : Determines ordering of atoms in Hamiltonian when repartitioner solver is attached to DOFmap. Most likely, set this to the name of the repartition solver	().
automatic__threshold : If true, then the code automatically determines the energy threshold that separates electrons and holes (i.e. conduction and valence bands) by looking at the bulk band edges and the electrostatic potential. When iterative band structure solvers (like Krylov-Schur) are used this option causes the shift value to be equal to the threshold mentioned above. The actual value of the threshold depends on other options. For more details search for _automatic_threshold variable in the code above. This option is also used for density calculations in Schroedinger solver and not just band structure calculations. Possible values: true/false.	(false).

Parameter : Description	Default Value
basis_generator : The (optional) name of the simulation object where the basis set for transforming the Hamiltonian is drawn from.	().
beta_degradation : Numerical parameter if eigensolver uses lanczos	(0.).
boundary_regions : required regions as boundary conditions	(()).
brillouin_zone_meshsize : If calculate_brillouin_zone is true, this determines the mesh size. The spacing in every direction, in pi/a	(0.1).
calculate_automatic_shift_subbandbased : implemented by R. Kotlyar. When true, the convergence scheme based on determining automatic shifts from subband energies is specified.	(false).
calculate_brillouin_zone : When set to true, the entire Brillouin zone will be meshed and computed.	(0.1).
calculate_elastic_overlap : calculates scattering-related form factors. deprecated	(false).
calculate_integrated_electron_density : implemented by R. Kotlyar. These options were added to fix the shift of the eigenvalue solver based on monitoring the subband energies from previ- ous iterations. Usually these shifts are determined by conduc- tion or valence band minima/maxima. Roza found that this convergence scheme works the best for high electric fields in the presence of large local band bendings.	(false).
calculate_integrated_hole_density : implemented by R. Kotlyar. If option is specified and spatially varying threshold = false, the integrated the total carrier density in cm-dim is computed by integration over the confinement dimensions (e.g. for wires this yields 1/cm). As similar option exists for electrons.	(false).

Parameter : Description	Default Value
charge_model : In electron_hole you have negatively charged electrons and positively holes. In electron_core you have negatively charged electrons and positively charged ions. Can be electron_core or electron_hole. When set to electron hole (most cases), then an energy threshold separates between electrons and holes. When set to electron core model all states are assumed to be electrons	(electron_hole).
check_every_steps_number : Numerical parameter if eigensolver uses lanczos. Try a value between 20 and 500	(0).
chem_pot : Defines the chemical potential for the Fermi distribution.	(0.0).
compute_edges_masses : If set to true, some band edges and masses will be computed and written to screen. This option was tested with bulk only and fails to compute correct transverse masses for X- and L- valleys in the presence of spin-orbit interaction.	(0.001).
compute_edges_masses_delta : delta K to calculate numerical derivative of the band dispersion. in pi/a. stepsize for the finite difference scheme employed in compute edges masses,	(0.001).
convergence_limit : Convergence limit for iterative solution methods (accuracy of eigenvalues).	(1e-8).
convergence_method : Numerical parameter if eigensolver uses lanczos. Can be full_convergence or partial_convergence	().
custom_transverse_k_points : Transverse k (wave vector) grid which has denser points around the center of the Brillouin zone. This option is used along with dense_grid_ratio and grid_density options. For now implemented for UTB only (1D transverse k).	(false).

Parameter : Description	Default Value
cutoff_distance_to_bandedge : Energy cutoff which is added to bandedge for holes and subtracted for electrons. Setting this to some value Delta (in eV) leads to electron thresholds at Ec - ePhi(x) - Alpha and hole thresholds at Ev - e Phi(x) + Delta.	(0.2*(Ec_min-Ev_max)).
dE_for_elastic_overlap :	(0.0,Warning: option calculate_elastic_overlap is deprecated. Energy threshold for calculating overlap for elastic scattering. Overlap is calculated if difference between energy_for_elastic_overlap option and energy eigenvalue is smaller than dE. Only takes effect if take_all_states_and_no_overlap option is false.).
dense_grid_ratio : The ratio of the total number of k (wave vector) points that will belong to the denser grid. Look at description of custom_transverse_k_points for more details.	(0.2).
direct_order_of_partitions_$neighbor_domain_name$: Orders neighbor domain partitions directly if true, in reverse if false. Considered only with neighbor domain repartitioner is provided.	true).
domain : Domain required for this simulation	().
doping_density : Doping density of material. Only used if calculate_Fermi_level is included in job list. Doping type (i,n,p) set by option doping_type.	(0.0).

Parameter : Description	Default Value
doping_type : Type of doping	(i).
e_xx : When giving strain as input, put epsilon_xx of strain tensor here(for tb_basis = kp only)	(0.01).
e_xy : When giving strain as input, put epsilon_xy of strain tensor here(for tb_basis = kp only)	(0.00).
e_yy : When giving strain as input, put epsilon_yy of strain tensor here(for tb_basis = kp only)	(0.01).
e_yz : When giving strain as input, put epsilon_yz of strain tensor here(for tb_basis = kp only)	(0.00).
e_zx : When giving strain as input, put epsilon_zx of strain tensor here(for tb_basis = kp only)	(0.00).
e_zz : When giving strain as input, put epsilon_zz of strain tensor here(for tb_basis = kp only)	(0.01).
eff_mass_dump : true: for effective mass calculation only when tb_basis=kp/ false: for disabling it only when tb_basis=kp	(false).
eigen_values_solver : Which eigenvalue solver to use. Setting lapack always com- putes all eigenvalues and is feasible only for very small sys- tems. Recommended choices are krylovschur and arpack. Other choices are arnoldi, jd, gd.	().
electron_chem_pot : Chemical potential used to calculate density of states times Fermi derivative. Has units of eV.	(0.0).
electron_hole_heuristics : Given the atomically resolved energy threshold between electrons and holes, this option tells how to actually determine whether a particle is an electron or hole. Two options available: sharp (sharp transition at the threshold) and omen_smooth (smooth transition around the threshold, like in OMEN code)	(sharp).

Parameter : Description	Default Value
electron_temperature : Temperature of the system	(300).
energy_for_elastic_overlap : Energy used for calculating matrix elements for wave functions	(0.0).
energy_maximum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_minimum : Eigenslover Lanczos will search the energy within energy_maximum & energy_minimum	(0.0).
energy_threshold_for_shift : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enthresh_foresolvershift).
epsilon_matrix : Constant strain tensor matrix, given in laboratory system co- ordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the laboratory system where crystal axis might have been rotated.	(strain_matrix).
epsilon_matrix_crystal : Constant strain tensor matrix, given in crystal system coordinates. One needs to include strain in the job list in order to use this. Superimposed strain matrix in the crystallographic system.	(strain_matrix).
extra_points : Extra K points set by user	(NULL).
fixed_shift : Sets the shift to be used by eigenvalue solvers (except when electron_core model: check option shift).	(0.0).
gaussian_quadrature_order_automatic : Affects k-space integration for periodic directions. If option integration_order is not defined and if gaussian_quadrature_order_automatic = true, then for number_of_k_points <= 22 use Gaussian quadrature with single partition, otherwise use simple rectangular with number of elements equal to number_of_k_points. In both cases the actual number of k points equals number_of_k_points.	(false).

Parameter : Description	Default Value
grid_density : How many times denser will the dense part of the transverse k (wave vector) grid be. Look at description of custom_transverse_k_points for more details.	(1.0).
init_check_steps_number : lanczos option, for checking convergence after how many iterations	(20).
input_shift_subbandbased_for_restart : When true, the shift is read in from file and is used as initial shift to eigensolver. R. Kotylar	(false).
input_shiftfilename_for_restart : Filename for input_shift_subbandbased_for_restart. R. Kotylar	(string()).
input_stress_or_strain : stress: for giving stress as input/ strain: for giving strain as input directly(for tb_basis = kp only)	(strain).
integration_order : Sets the order of integration for Gaussian quadrature method	(1).
is_kp_strain_on : true: for enabling strain calculation with k.p/ false: for disabling strain calculation with k.p(for tb_basis = kp only)	(false).
is_piezoelectric_effect_on : true/false; for enabling piezoelectric calculation with k.p;(for k.p simulation only)	(false).
is_spatially_parallelized : true/false; Whether the device is space parallelized;(for k.p simulation only)	(false).
is_vff_strain_on : true/false; for enabling VFF strain calculation with k.p;(for k.p simulation only)	(false).
job_list : A list that deterimes what is done. Choose from assemble_H, passivate_H, include_strain_H, include_shear_strain_H, calculate_band_structure, electron_density, derivative_electron_density_over_potential, hole_density, derivative_hole_density_over_potential, spin, DOS. assemble_H is activated by any other option automatically.	(job_list).

Parameter : Description	Default Value
k0 : k0 vector to solve eigenvalues at this particular k-point	([0 0 0]).
k_degeneracy : The computed density is multiplied by this number to account for k-space degeneracy. E.g. when kxmax and kymax are set in a simulation with 2D k-space, k degeneracy should be 4.	(1.0).
k_points : Only relevant for calculate band structure. This parame- ter is a list of points in k-space along which the band structure is calculated.	(k_pointsV).
k_rotational_symmetry : Assumes rotational symmetry in a 2D k-space	(false).
k_space_basis : (Default: cartesian) Sets the basis in which k points is specified. When set to reciprocal the coordinates given in k points are assumed to be w.r.t. the reciprocal lattice vectors b1 =a2xa3 / a1.(a2xa3) etc.	(reciprocal).
kp_band_number : 8 band for direct band gap material / 6 for indirect band gap material / 4 for rhombohedral crystals of TI(for tb_basis = kp only)	(8).
kp_strain_dir1 : If stress is perpendicular to [x y z] plane then input x here(for tb_basis = kp only)	(1).
kp_strain_dir2 : If stress is perpendicular to [x y z] plane then input y here(for tb_basis = kp only)	(0).
kp_strain_dir3 : If stress is perpendicular to [x y z] plane then input z here(for tb_basis = kp only)	(0).
kp_stress_mag : When giving stress as input, put stress in GPa here(for tb_basis = kp only)	(1).
kpoint_to_monitor_subbands_for_shift : The k-point where to look for shift. default is 0. R. Kotylar	(kmagref_foresolvershift).

Parameter : Description	Default Value
kpoint_tolerance_to_monitor_subbands_for_shift : The k-tolerance when looking for a k-point on k-grid. Default is 0.001. R. Kotylar	(kmagreftol_foresolvershift).
kxmax : Upper boundary in 2 pi/a of the simulated kx space. kx ranges from 0 to this number.	(1.0).
kxmin : Lower boundary in 2pi/a range of the simulated kx space. kx begins from zero by default.	(0.0).
kymax : Upper boundary in 2 pi/a of the simulated kx space. ky ranges from 0 to this number.	(1.0).
kymin :	(0.0,Lower boundary in 2pi/a range of the simulated ky space. ky begins from zero by default.).
kzmax : Upper boundary in 2 pi/a of the simulated kz space. kz ranges from 0 to this number.	(1.0).
kzmin :	(0.0,Lower boundary in 2pi/a range of the simulated kz space. kz begins from zero by default.).
lanczos_iseed : Numerical parameter if eigensolver uses lanczos	(0.).
linear_solver : Linear solver employed in the shift-and-invert operation. This should be preferable a direct linear solver since the LU fac- torization can be reused during the Krylov iterations.	().
max_number_iterations : Maximum number of Krylov iterations (irrelevant for lapack).	(0).

Parameter : Description	Default Value
maximum_delta_k : Upper bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(1e80).
maximum_scattering_angle : Option connected to computation of scattering matrix elements	(180.0).
minimum_delta_k : Lower bound of norm of delta_k to calculate scattering rate in units of 1/nm.	(0.0).
minimum_scattering_angle : Option connected to computation of scattering matrix elements	(0.0).
modes_for_current : Relevant for current calculation. Define the number of modes to be extracted from E-k while calc. ballistic TOB current.	().
monitor_convergence : When set to true, terminal output related to the Krylov iteration is generated.	(false).
mpd : Specialized numerical option. Refer to the SLEPc user manual for the meaning.	(0).
mumps_ordering : Relevant only for eigen values solver=mumps. Choose between pord and metis.	().
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : Specialized numerical option (size of the Krylov subspace).	(0).
no_of_spatial_division : No of cores used for parallelizing the device(for tb_basis = kp only)	(256).
noise_potential : noise potential used in qtbm to break degenerate modes.	(0.0).
number_of_eigenvalues : Number of eigenvalues to compute (irrelevant for lapack).	(0).
number_of_eigenvalues_to_use : Number of requested eigenvalues	(0).

Parameter : Description	Default Value
number_of_k_points : This list gives the uniform discretization of each segment set by the k points parameter (note that specifying N points means N - 1 segments). For density calculations, setting this parameter to 0 leads to computation of k = 0 only and application of an analytical formula that assumes parabolic subbands.	(10).
number_of_nodes : number of nodes for each k-segment	(().
output : Choose from Hamiltonian, energies, k-points, DOS, electron density, electron density VTK, hole density, hole density VTK, ion density, eigenfunctions, eigenfunctions k0, eigenfunctions VTK, eigenfunctions VTK k0, eigenfunctions Silo, eigenfunctions Silo k0, spin.	(output_list_v).
output_eigenstate_energy : Energy around which eigenpair has to be output.	(0.0).
output_file_suffix : suffix to be added on output files	().
output_k_point : Value of k-point dumped as output	(k0).
output_precision : Accuracy of saved eigenvalues within output file.	(12).
output_shift_subbandbased_for_restart : When true, at the end of simulation the file can be output with the last eigensolver shift information. Using this shift and the last potential, the next simulation can be restarted from the last state of simulation. R. Kotylar.	(false).
p_parameter : It's the block size for block Lanczos. If the degeneracy of the eigenvalues is known, then set it to that value, otherwise start with 4.	(0).
parallelize_here : When set to false, some other simulation (like Propagator) can determine the parallelization.	(true).

Parameter : Description	Default Value
potential_solver : The (optional) name of the simulation object where the electrostatic potential is drawn from.	().
preconditioner : This option controls how the eigenvalue problem is transformed before solution Possible options are: "lu" "mumps"...work ongoing . (default: lu) Preconditioner employed in the shift-and-invert operation.	(lu).
project_DOS_ : This option allows to calculate the project density of states	(false).
rate_of_shift_changes_between_iterations : To determine the position of the next shift to be passed to eigensolver based on subband energies in previous iteration. R. Kotylar.	(enrate_foresolvershift).
reinit_parallelize : When Schroedinger solver is reinitialized, reinitialize the MPI parallelization as well.	(false).
resolution : The resolution to be used for the eigensolver	(0.0).
reuse_hamiltonian : if this option is false, hamiltonian would be recalculated every iteration, for different k always it is recalculated	(true).
seed : seed for noise potential.	(-1).
set_valley : set valley name to get Material properties	(DEFAULT).
shift : The eigensolver will search for eigenvalues around the shift value. This seems to only be used for electron_core model. fixed_shift is used otherwise.	(0.001).
solve_on_single_replica : Gather density to the single rank only	(false).
solver_transformation_type : (default: sinvert) Use sinvert for shift-and-invert, shift for shift.	(sinvert).

Parameter : Description	Default Value
sort_like_electrons : This option controls the way the eigenenergies are sorted. When true (electrons) then ascending and when false (holes) then descending. Possible values: true/false.	(true if job list has electron density or band structure or electron core model. false if job list has hole density.).
sparse_storage_format : Sparse storage format in which PETSc will allocate sparse matrices; options available are "csr" (compressed sparse row) and "bsr" (block sparse row)	(csr).
spatially_varying_threshold : If true, then the threshold energy discriminating between electrons and holes will vary spatially according to the elec- trostatic potential. Needed in Schroedinger-Poisson simulations.	(false).
spin_resolved : if this option is enabled all parameters will be spin dependant Up/Down	(false).
strain_calc_option : hydrostatic: for calculating the effect of hydrostatic strain/ biaxial: for calculating the effect of biaxial strain/ uniaxial: for calculating the effect of uniaxial strain (use this option only when giving strain as input)(for tb_basis = kp only)	(uniaxial).
strict_threshold : Solve for either conduction (electron density) or valence (hole density) bands depending on option sort_like_electrons. Possible values: true/false.	(true if job list has electron density or hole density. false if job list has band structure or if electron_core charge model is used.).
surface_of_regions : surface regions required to calculate boundary conditions	().

Parameter : Description	Default Value
take_all_states_and_no_overlap : This needs the option "calculate_elastic_overlap" to be true. Allows to switch the Schroedinger from solving the scattering overlap for a subset of states to collect all states and make them ready for other solvers to get via get_data. Primary purpose of this is to support the fitting of TB to DFT including the TB wavefunctions	(false).
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	(sp3).
temperature : Temperature (in kelvin) of the device to be simulated	(300).
tic_toc_name : Tic toc name to be used for saving timing results for this simulation.	().
tolerance : The judgement of the convergence. Once the target reaches the tolerance then end the loop.	(0.0).
use_monomials : calculate monomials table based on the orbitals	(true).
use_neighbour_repartitioner_$neighbor_domain_name$: Repartitioner solver name for neighbor domain.	()

Table 5.131: Parameters of the Schroedinger_intra_atomic Solver.

5.132 Self_Propagation

Self_energy Solver

Parameter : Description	Default Value
$(NEGF_object_list)_MPI_rank : This associates an MPI rank to a given NEGF_object. If this MPI rank does not agree with the given number, the NEGF_object is ignored.	(1).
$(NEGF_object_list)_constructor : For each element in NEGF_object_list a constructor needs to be defined.	(()).

Parameter : Description	Default Value
$(NEGF_object_list)_energy_resolved_output_one_per_file : If true, the energy resolved output will be dumped to different files for different energies.	(false).
$(NEGF_object_list)_equivalent : This defines an alias for a given NEGF_object.	().
$(NEGF_object_list)_inversion : This defines a method for the inversion of a inverse retarded Green's function.	().
$(NEGF_object_list)_non_solver : To be used in iterators: this option deletes this solver from the solver list of a given NEGF_object_list.	(()).
$(NEGF_object_list)_solver : For each element in NEGF_object_list a solver needs to be defined.	(()).
$(NEGF_object_list)_store_atomic_offdiagonals : This defines the amount of atomic/nonlocal offdiagonals of the Green's function to keep in memory.	(1).
$(NEGF_object_list)_store_offdiagonals : This defines the amount of offdiagonals of the Green's function to keep in memory.	(1).
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
$(momentum_names)_granularity : For each mesh in option "momentum_names", a minimum amount of MPI ranks per mesh point can be defined.	(1).
$(momentum_names)_mesh_corners : If this is the constructor of mesh "A", the mesh corners for a linear mesh can be defined.	(()).
$(momentum_names)_parent : If more than one entry is in option "momentum_names", this option sets the parallelization hierarchy: the parent is the superior mesh.	().
$(momentum_names)_points : If this is the constructor of mesh "A", the number of mesh points "A_points" can be defined.	(10).

Parameter : Description	Default Value
$(momentum_names)_strategy : "A_strategy" sets the parallelization strategy for the mesh "A".	(Scatter).
Bands:BandEdge:Ec : initialize the band edge for the conduction band	(0).
GPU_offload_ranks : Contains a vector of MPI ranks. RGFModule/GeneralLeadModule offload matrix multiplies and matrix inversions to GPU for the ranks specified. NEMO5 should have been compiled withthe MAGMA library (http://icl.cs.utk.edu/magma/) for option for this to work.	().
G_for_transmission : Get the retarded Green's function of the device (either the only retarded Green's function within this propagation, or read in from inputdeck	().
Hamilton_constructor : Name of the simulation that generates the Hamiltonian.This defines a solver that knows about the X-coordinates (degrees of freedom)	().
Hamilton_constructor$(#) : used in transfer matrix method: This defines a Hamilton_constructor for a given subdomain number #=(0..4).	().
Hamilton_constructor0 : schroedinger solver for 0th block	().
Hamilton_constructor1 : schroedinger solver for 1st block	().
Hamilton_constructor2 : schroedinger solver for 2nd block	().
Hamilton_constructor3 : schroedinger solver for 3rd block	().
Hamilton_constructor4 : schroedinger solver for 4th block	().
Hamilton_momenta : Names of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	(().

Parameter : Description	Default Value
LRA : use LRA only for transfer_matrix ref: NEMO5_BC.ppt	(false).
MPI_interleaving :	().
NEGF_object_list : This option defines all Propagators relevant for this Propagation. Supported keywords are: retarded or lesser, Green or self, Fermion or Boson, contact + self, scattering + self, inverse + retarded + Green	(()).
Parallelizer : Name of the simulation that determines the parallelization	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
add_readable_Propagators : To be used in iterators: This option adds NEGF_objects to the list of readable Propagators.	(()).
always_ready : Typically used for combine_Propagation: This option sets a solver to be always ready to hand over data.	(false).
analytical_momenta : If this option is present, analytical k-integration is done for density calculation. Supported values of this options are 1D and 2D depending on 1D or 2D k-integration	().
average_eps0 :	(12.0).
average_epsinf :	(10.0).
boundary_regions : required regions as boundary conditions	(()).
bp_analytical_momenta : specify the analytical momenta type, include 1D, 2D and nanowire case	().
bp_chem_potential_direct : initializing the chemical potential values directly reads in a vector of values equal to the number of Buettiker probes	().
bp_chem_potential_linear : setup the chemical potential corners for the Buettiker probe	(No default).
bp_coupling : setup the BP coupling value, include real and imaginary part	().
bp_current_threshold : Threshold for Buettiker Probe Newton Raphson to be met.	(1e-15).
bp_full_transmission_calculation : output the type for the transmission calculation	(false).
bp_initial_chemical_potential_model : choose the model to initialize the chemical potential in each probe	(zeros).
bp_material_tags : Name of materials which contain relevant Buettiker probes.	(No default).
bp_probe_model : Specify the type of Buettiker probe models	().
bp_trans_map_storage : setup the matrix type for the transmission map, include sparse and dense	(sparse).

Parameter : Description	Default Value
calculate_Jacobian : Switch to solve the density (false) or the derivative of the density (true) with respect to the Poisson potential. Typically needed for debugging.	(false).
calculate_optical_absorption : If true, the optical absorption is solved and written to file. UNDER DEVELOPEMENT!	(false).
chemical_potential : chemical potential for lead	(0.0).
chemical_potential1 : chemical potential of lead 1	(0.0).
chemical_potential2 : chemical potential for lead 2	(0.0).
clean_all_in_reinit : if this option is enabled, all matrices will be destroyed on the re-initialization process. by default some matrices are reused	(false).
clean_up : clean up propagators, set job to be not ready	(false).
condense_memory_of : Switch to control partial deletion of all Propagation member variables during reinitialization. Typically needed by developers.	(().
consistent_modes : If true transfer matrix method will make mode phases and mode ordering consistent between different runs. Phase and magnitude of the first mode matrix row will be set to zero, while ordering of propagating modes is according to velocities and of decaying modes according to phase factor's (i.e. eigenvalue's) magnitude then phase.	(false).
constant_eta : artificial dephasing usually used in direct_iterations method	(0.0).
constant_lead_eta : eta added to diagonal of lead Hamiltonian used in transfer_matrix	(1e-12).
convergence_limit : convergence limit for eigensolver if not lapack	(1e-8).

Parameter : Description	Default Value
current : If true, current density is solved and written to file.	(false).
debug_output_job_list : output all temporary matrix for debug purpose	(false).
deformation_potential : Potential for the scattering of electrons on lattice deforming phonons in eV	(1.0).
degeneracy_factor : band degeneracy factor. This value is used in calculating the integration weight for analytical k-integration	(1).
delete_combine : If set to true, propagator matrices are deleted during re-initialization process	(false).
density_of : This is the name of The G lesser solver which is used for density calculation	().
density_output : If true, the density is solved and written to file.	(false).
differentiate_over_energy : If set to true,density jacobian will be solved by G< solver	(false).
do_output : output all propagator matrix into matlab	(false).
do_output_scattering_rate : calculate and output scattering rate	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
drain_location : temporary location for the drain to calculate the initial chemical potential map	(0).
energy_interpolated_output : enable energy interpolated output	(true).
eps_limit : threshold to control modes	(1e-6).
equilibrium_model : lesser self-energy calculation	(true).
exact_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact lesser Green's function	().

Parameter : Description	Default Value
exact_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the exact retarded Green's function	().
free_memory_of : Set of NEGF objects that can be erased once the writeable object of this Propagation is solved. Typically needed by developers.	(()).
half_way_lesser_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward lesser Green's function	().
half_way_retarded_green : This option is required for solving the local current density in the case of RGF+scattering.It defines the name of the forward retarded Green's function	().
hole_distribution : specify the fermi distribution type, include electron and hole	(false).
ignore_readable_Propagators : To be used in iterators: This option deletes NEGF_objects to the list of readable Propagators.	(()).
imag_limit : threshold to control modes	(2e-6).
imaginay_part_of_density_prefactor2 : This is for debugging only! Imaginary part of the extra prefactor for the density calculation.	(-1.0).
integration_to_get_density : If set to false, integration weight(used for k-integration) is multiplied by volume to get density in terms of charge per volume	(false).
iterative_damping : exponential damping ratio for constant_eta	(0.5).
kxmax : Maximum k point in the x vector	(1.0).
kymax : Maximum k point in the y vector	(1.0).
kzmax : Maximum k point in the z vector	(1.0).
lambda :	(0.0).

Parameter : Description	Default Value
lead_domain$(#) : used in transfer matrix method: This defines a the neighbor domain for a given subdomain number #=(0..4).	().
lead_domain0 : lead domain of subdomain0	().
lead_domain1 : lead domain of subdomain1	().
lead_domain2 : lead domain of subdomain2	().
lead_domain3 : lead domain of subdomain3	().
lead_domain4 : lead domain of subdomain4	().
linear_system_solver : linear solver options: petsc/mumps ref: http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/index.html	(mumps).
linsolver : This option defines the linear solver option for the eigensolver used in transfer matrix method	().
local_current : If true, the local current density is solved and written to file.	(false).
max_number_iterations : iterations for eigensolver if not lapack	(300).
maximum_Sancho_iterations : Maximum number of iterations for the Sancho-Rubio method of solving the contact self-energy.	(50).
momentum_names : Vector of names for 'momenta', i.e. k,E,...	(()).
monitor_convergence : This option is used by the eigen solver in transfer matrix method to monitor the convergence of the solver	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : It stands for number of column vectors. This is the eigensolver option used in transfer matrix method	(2*$number_of_set_num).
no_integration_for_transmission : If set to false, the solver multiplies transmission function with an integration factor depending on 1D or 2D analytical k-integration	(false).

Parameter : Description	Default Value
nondistributed_Hamilton_constructor : This is the name of the Schroedinger solver for the device domain which constructs the Hamiltonian based on tb basis	($Hamilton_constructor).
number_of_interpolation_points : Number of interpolation points	(100).
number_of_k_points_for_scattering_rate : number of k points for scattering rate calculations	(30).
output_Ek : output Ek for the lead Hamiltonian from transfer_matix method	(false).
output_complexEk : bool type, output complex E-k	(false).
output_eigenvalues : output eigenvalues/phase factor/mode velocity	(false).
output_file_suffix : suffix to be added on output files	().
output_format : Defines the output of atom or orbital resolved data. Possible options are: "VTK", "xyz", "DX", "Silo"	(VTK).
output_suffix : A string that is used as part of all filenames output	().
parallelization_scheme : Parallelization scheme: Top_Down or Bottom_Up	(top_down).
parallelize_adaptive_grid_construction : When calling adaptive grid constructor to create the grid, call it with all transverse k points at once. This allows parallelization of adaptive grid construction when there are enough cores available, instead of calculating the adaptive grid for every transverse k point in serial.	(false).
phonon_energy :	(0.0).
phonon_temperature :	(300).
precond_type : This defines the type of the preconditioner of eigensolver for transfer matrix method	().

Parameter : Description	Default Value
ratio_of_eigenvalues_to_be_solved : percentage of number of eigenvalues (0~1.0) to be solved in transfer_matrix method	(1.0).
real_part_of_density_prefactor2 : This is for debugging only! Real part of the extra prefactor for the density calculation.	(0.0).
remove_internal_meshes : If true delete dynamic memory associated with internally generated meshes when the solver is reinitialized.	(true).
remove_meshes : If true delete dynamic mesh memory associated with external mesh constructors when the solver is reinitialized.	(true).
retarded_lead_method : self-energy solver options: direct_iterations/transfer_matrix/transfer_matrix_avoid_surface_greens_func we usually use transfer_matrix_avoid_surface_greens_function for QTBM ref:M.Luisier, PRB 74, 205323 (2006)/NEMO5_BC.ppt	
reuse_parallel_hierarchy : DEPRECATED	(false).
rgf_band : get the conduction band edge for RGF method, this part is under development	().
rgf_eta : get the imaginary part of coupling constant for RGF method, this part is under development	().
rgf_override : Checks for RGF override, this function is under development	().
scattering_energy_interval : energy interval for scattering	(0.0).
scattering_momentum_interval : momentum interval for scattering	(1e20).
scattering_type :	().
selfenergy1_for_transmission : This option defines the name of the first of two contact self-energies needed for the transmission between these two contacts.	().

Parameter : Description	Default Value
selfenergy2_for_transmission : This option defines the name of the second of two contact self-energies needed for the transmission between these two contacts.	().
shift :	().
skip_step_6 : This option is relevant during self energy calculation using transfer matrix method. If this option is set to false, original Gr matrix is used, otherwise, a dummy Gr is generated	(false).
solver_type : lapack for all eigenvalues, krylovschur for partial eigenvalues	(lapack).
source_location : temporary location for the source to calculate the initial chemical potential map	(0).
source_matrix_name : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the small matrices to be stored	().
stationary_Propagation : time dependent NEGF or not, so far only stationary work	(true).
store_atomic_offdiagonals :	(0).
store_offdiagonals :	(0).
store_result_in : In some cases, several smaller matrices(e.g., left connected gR) need to be combined into a large matrix (GR). In this case, this option defines the name of the combined large matrix	().
surface_of_regions : surface regions required to calculate boundary conditions	().
target_domain_Hamiltonian_constructor : In RGF method, small gR matrices are solved block by block. This option defines the name of the Schroedinger solver which determines the DOF number of these small blocks	().
temperature : temperature for lead	(300).

Parameter : Description	Default Value
temperature1 : temperature of lead1 or source used in fermi distribution or other temperature dependent calculations	(300.0).
temperature2 : temperature of lead2 or drain used in fermi distribution or other temperature dependent calculations	(300.0).
tic_toc_name : prefix for advanced tic-toc system	($name).
top_most_MPI_strategy : Can be cluster (default) or scatter. Refer to the N5 paper[1] for an illustration of these strategies.	(cluster).
transform_type : transform_type for eigensolver	(none).
transmission : If true, the transmission is solved and written to file.	(false).
wire_nonSO_basis : if true, solve double matrix instead of complex matrix, only worked for wire and no spin tb_basis	(false).

Table 5.132: Parameters of the Self_Propagation Solver.

5.133 Selfconsistent

SelfconsistentSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable_step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	().
__output : list of solver to trigger output command	().
__reinit : list of solver to trigger reinit command	().
__search_flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	().
__step : list of solver to trigger step command	().
__trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.133: Parameters of the Selfconsistent Solver.

5.134 SelfconsistentBornModule

SelfconsistentBornModuleSolver

Parameter : Description	Default Value
G_for_transmission : Green function propagator for transmission calculation.	().

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
assure__current__conservation : After the given bias point is convergence with respect to Poisson. This option will allow an additional Selfconsistent born loop to be perfomed where the current conservataoin is the criterion.	(false).
backward__rgf__offload__solver : Solver of type TaskOffload to be used in the "backward" part of RGF which consists heavily of dense matrix-matrix multiplications	().
boundary__regions : required regions as boundary conditions	(()).

Parameter : Description	Default Value
check_current_conservation : check whether current is conserved from slab to slab.	(false).
current : output total/terminal current.	(false).
current_conservation_tolerance : Criterion used whether to say current is conserved or not.	(false).
density_of_states_output : output density of states.	(false).
density_output : output density.	(false).
disable_blocking_operations : strictly a debugging option. Used to minimize sychronization points for testing scaling. Should not be used in any real production simulations.	(false).
divergence_charge_threshold : Used in charge convergence check for setting the maximum norm before a divergence is detected (since Selfconsistent Born iterations should converge monotonically). When the max norm goes larger than this value then the selfconsistent Born iterations are concluded and returned to Poisson.	(1.0).
domain : Domain required for this simulation	().
end_leads_solver : self-energy solver name for drain lead.	().
energy_interpolated_output : do energy interpolation for result or not.	(false).
energy_resolved_density_of_states_output : output energy resolved DOS.	(false).
energy_resolved_density_of_states_output_one_per_file : If true dumps energy resolved output on one separate file per energy	(false).
energy_resolved_density_output : energy resolved electron/hole density.	(false).
energy_resolved_density_output_one_per_file : output energy resolved density in individual files.	(false).

Parameter : Description	Default Value
external_Hamilton_Constructor : name of schroedinger solver; if defined, use external schroedinger solver, otherwise construct it in the module.	().
external_lesser_self_energies : lesser self-energy propagator names that produce sigma<.	().
external_retarded_self_energies : self-energy propagator names that produce sigmaR.	().
fixed_number_of_iterations : If the Born iteration needs to be run for a fixed number of iterations, this option should be set to true	(false).
forward_rgf_offload_solver : Solver of type TaskOffload to be used in the "forward" part of RGF which consists heavily of dense matrix inversion	().
k_resolved_density : output k resolved density.	(fasle).
k_resolved_density_of_states_output : output k resolved DOS.	(false).
k_resolved_density_output : k resolved electron/hole density output.	(false).
k_resolved_density_output_one_per_file : output k resolved density in individual files.	(false).
lead_domains : vector of string defines the name of leads.	().
local_current_density : output local current, slab resolved.	(false).
max_Poisson_residual_change : The max change of the residual from Poisson before we say the previous scattering self energy should not be used for 1st iteration and the ballistic case should be solved.	(false).
max_norm_threshold : Used in charge convergence check for setting the minimum value of charge to check in the max norm of convergence criterion. This helps from checking values close to 0 that might fluctuate a lot between iterations	(1E-4).

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
no_integration_for_transmission : do sum over raw transmission rather than integration over k-space.	(false).
number_of_interpolation_points : int, number of interpolation points for line output.	().
output_along_line : vector of double point, defines a line where energy resolved output is solved.	().
output_along_line_resolution : vector of int, defines the resolution of the line output.	().
output_file_suffix : suffix to be added on output files	().
output_raw_transmission_data : output raw transmission e.g. E-k resolved transmission for utb.	(false).
output_slab_resolved_current : This outputs the slab resolved current as a function of the slab index	(false).
output_slab_resolved_energy_current : This outputs the slab resolved current as a function of the slab index for each energy	(false).
output_solver : name of the solvers that produce the output: GR, G<, current, etc	().
potential_solver : potential solver of the domain, e.g. poisson	().
readin_coupling_table : Debug option in order to read in the energies that couple to other energies.	(false).
readin_poisson_iteration_number : For use with the dynamical convergence when the simulation is restarted. Allows the user to input the previous Poisson iteration. To be automated in the future.	(false).

Parameter : Description	Default Value
readin_poisson_residual_change : For use with the convergence charge criterion when the simulation is restarted. Allows the user to input the previous change in Poisson residual. To be automated in the future.	(false).
reduced_real_space_transformation : Setting to true activates transformation to a reduced real space of scattering self energies when a low rank approximation method is applied to Green's functions	(false).
scattering_in_lead : whether solve scattering in the lead also.	(false).
scattering_types : type of phonon scattering, e.g. Lamda*G scattering or deformation_potential, etc.	().
spatial_resolved_current : output spatial/slab/atomic resolved current	(false).
start_leads_solver : self-energy solver name for source lead.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : tight binding basis.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
total_charge_tolerance : Criterion with respect to change in charge tolerance used whether to stop the Self-consistent Born iterations.	(false).
transmission : output total/integrated transmission.	(true).
use_RGF : use RGF to solve retarded/lesser Green's function, otherwise use direct inversion.	(false).
use_interpolated_scattering : If this is set to true then the previous scattering self energy will be used whenever possible	(false).

Parameter : Description	Default Value
use_modified_correlation_equation : Used in RGFv2 for setting whether to assume g< is antihermitian or not. This assumption true in infinite precision but can cause stabilty issue with lambda*G scattering in finite precision.	(true).

Table 5.134: Parameters of the SelfconsistentBornModule Solver.

5.135 Semiclassical

This simulation calculates the semiclassical charge density and Jacobian.
This simulation can replace a more involved compuation of the density in some or all regions of the structure by the simple semiclassical relationship

$$n(x) = N_c \, \mathcal{F}_{0.5} \left(\frac{E_F \quad (E_c \quad e \quad (x))}{kT} \right). \tag{5.2}$$

This relation assumes a single parabolic conduction band with band edge E_c. $\mathcal{F}_{0.5}$ is the Fermi integral of order 0.5. N_c is the effective density of states which is computed from the effective mass and the band gap.

Parameter : Description	Default Value
Ef_approximation : Spatial shape of the Fermi level in the device. Fermi level is constant in equilibrium_regions and varies according to this option elsewhere (i.e. channel). This option must be used in conjuction with small_coordinate_contact. Possible options x_linear_drop, y_linear_drop, z_linear_drop, average.	(average).
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).

Parameter : Description	Default Value
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
contacts__control : Used when Ef_approximation = average. This is a vector input with the same order of elements as in equilibrium_contacts. If there are contacts given by fermilevel_solver that are not included in equilibrium_contacts their control is 1.0.	(All contacts have equal and full control (1.0) of the device Fermi level.).
dimensionality : The density of states dimensionality. Default: 3.	().
domain : Domain required for this simulation	().
eff__mass__materials : Defines a list of materials for which confined effective mass will be calculated. The list ordering corresponds to option fermi_level_solver_names which is a vector of vector. Each material may have up to 2 fermi level solvers, for n and p doping type.	().
equilibrium__contacts : Names of the contacts. Should have the same names as the contact names given by fermilevel_solver.	

Parameter : Description	Default Value
equilibrium_regions : Vector of vector. Regions in the simulation domain associated with contacts in equilibrium_contacts. Each contact can have more than one region. Fermi level is constant in these regions.	
fermi_level_solver_names : Vector od vector. Femi level solvers are used in the process of confined effective mass calculation. The vector element ordering corresponds to eff_mass_materials, where each material may have up to 2 fermi level solvers, one for n and one for p doping type.	().
fermilevel : The globally constant Fermilevel EF.	().
fermilevel_solver : Should provide contact names and list of bias applied to each contact (in the same order). This is usually Poisson solver.	().
name : A unique name tag / identifier for the specific instance of the solver.	
no_free_charge_regions : Regions with no free charge.	().
non_equilibrium_regions : If there are non-equilibrium regions defined, then the electron/hole Fermi level in them is equal to electron/hole Fermi level in the contacts defined in quasi_fermi_level_contacts (first contact is for electrons, second for holes). Contact names should be the same as given by fermilevel_solver. When this is used then average or linear_drop Ef_approximation is not used.	().
output_file_suffix : suffix to be added on output files	().
potential_solver : The solver from which the electrostatic potential phi can be obtained.	().

Parameter : Description	Default Value
quasi_fermi_level_contacts : If there are non-equilibrium regions defined, then the electron/hole Fermi level in them is equal to electron/hole Fermi level in the contacts defined in quasi_fermi_level_contacts (first contact is for electrons, second for holes). Contact names should be the same as given by fermilevel_solver. When this is used then average or linear_drop Ef_approximation is not used.	().
small_coordinate_contact : Contact with smaller coordinate in the direction of Fermi level linear drop. This option must be used in conjuction with Ef_approximation.	(source_contact).
surface_of_regions : surface regions required to calculate boundary conditions	().
temperature : The temperature T in Kelvin (enters the Fermi function).	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.135: Parameters of the Semiclassical Solver.

5.136 SemiclassicalFEM

SemiclassicalFEMSolver

Parameter : Description	Default Value

Table 5.136: Parameters of the SemiclassicalFEM Solver.

5.137 SeparateDomains

SeparateDomainsSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	().
__output : list of solver to trigger output command	().
__reinit : list of solver to trigger reinit command	().
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	().
__step : list of solver to trigger step command	().
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	().
boundary__regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().
surface__of__regions : surface regions required to calculate boundary conditions	().
tic__toc__name : Prefix for tic-tocs	($(name)).

Parameter : Description	Default Value

Table 5.137: Parameters of the SeparateDomains Solver.

5.138 SimpleExample

A toy example solver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	().
__output : list of solver to trigger output command	().
__reinit : list of solver to trigger reinit command	().
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	().
__step : list of solver to trigger step command	().
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	().
boundary__regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().

Parameter : Description	Default Value
mean_filename : The path to the file to store the mean	(mean.txt).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
output_mean : Whether or not to store the mean	
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.138: Parameters of the SimpleExample Solver.

5.139 SimulationModules

[DEPRECATED] Simulation Modules
This helper combines various simulations into logical units to ease input deck creation.

Parameter : Description	Default Value
$(momentum_names)_constructor : For each mesh in option "momentum_names", a mesh constructor has to be defined.	().
Hamilton_momenta : ames of the momentum that the Hamilton_constructor needs as an input to create the Hamiltonian (such as k-space).	((()).
Module_type : type of module to be created	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
chem__pot : Defines the chemical potential for the Fermi distribution.	(0.0).
do__output : Switch to allow output of the Propagator matrices. Typically needed for debugging.	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
hamilton__constructor : Name of the solver that contains the hamiltonian	().
job__list : A list that deterimes what is done. Choose from assemble_H, passivate_H, include_strain_H, include_shear_strain_H, calculate_band_structure, electron_density, derivative_electron_density_over_potential, hole_density, derivative_hole_density_over_potential, spin, DOS. assemble_H is activated by any other option automatically.	(()).

Parameter : Description	Default Value
k_points : Only relevant for calculate band structure. This parameter is a list of points in k-space along which the band structure is calculated.	().
k_space_basis : Sets the basis in which k points is specified. When set to reciprocal the coordinates given in k points are assumed to be w.r.t. the reciprocal lattice vectors b1 =a2xa3 / a1.(a2xa3) etc.	(reciprocal).
lead_direction : Lead direction	(().
lead_domain : The domain that is connected to this PropagationModule. It can be used, for example, to get the coupling Hamiltonian between this domain and the $(lead_domain)	().
max_num_iterations :	(15Max number of Poisson iterations).
momentum_names : This is a list of strings defining the possible momenta of the propagator entities for which a mesh or grid will be constructed.	(().
name : A unique name tag / identifier for the specific instance of the solver.	
number_of_nodes : number of nodes for each k-segment	(().
output_file_suffix : suffix to be added on output files	().
parallelize_here : When set to false, some other simulation (like Propagator) can determine the parallelization.	(false).
prefix : prefix name to be assigned to Domains	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tb_basis : The tight-binding basis / band structure model (sp3, sp3d5sstar_SO, etc...).	
temperature : Temperature (in kelvin) of the device to be simulated	().

Parameter : Description	Default Value
threshold_energy : DEPRECATED, this option was removed from Schroedinger	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.139: Parameters of the SimulationModules Solver.

5.140 Step

This simulation increase an iterator in a set of solvers

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).

Parameter : Description	Default Value
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
solvers : list of solvers that need to be stepped.	(()).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.140: Parameters of the Step Solver.

5.141 StrainDomain

StrainDomainSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).

Parameter : Description	Default Value
_output : list of solver to trigger output command	().
_reinit : list of solver to trigger reinit command	().
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	().
_step : list of solver to trigger step command	().
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	().
boundary_regions : required regions as boundary conditions	().
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.141: Parameters of the StrainDomain Solver.

5.142 StrainVFF

StrainVFFSolver

Parameter : Description	Default Value
: If true then the strain tensor components are calculated according to the method of Pryor and Zunger as a post-processing step.	().
_disable_init : flag command to disable initialization	(false).

Parameter : Description	Default Value
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
absolute__tol : Absolute tolerance convergence criterion of the Newton iteration.	().
active__regions : required regions to be activated for this solver	(()).
anharm__cutoff : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
anharm__smoothing : (anharmonic Areshkin only) dimensionless numerical smoothing parameter.	().
anharm__theta__limit : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
boundary__regions : required regions as boundary conditions	(()).
calculate__epsilon : bool, if true, calculate epsilon.	(false).

Parameter : Description	Default Value
coulomb_cutoff_radius : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interaction is neglected. This parameter influences thesparsity of thematrix and memory consumption.	().
coulomb_ewald_Kcut : (coulomb only) (3D-periodic (bulk) simulations only) integer cutoff outside which the long-range contribution is neglected,in units of reciprocal lattice vectors.	().
coulomb_ewald_Rcut : (coulomb only) (3D-periodic (bulk) simulations only) cutoff in real-space	().
domain : Domain required for this simulation	().
ewald_sigma : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interactionis neglected. This parameter influences the sparsity of thematrix and memory consumption.	().
hessian_info : If true then some numerical debug terminal output is created.	().
linear_solver : Which linear solver is employed in the Newton iteration. See the PETSc manual for possible choices - gmres is preferred.	().
linsolver_monitor : If true then the convergence of the iterative linear solver is monitored as terminal output.	().
linsolvers_max_iters : Maximum number of linear solver iterations (in the case of an iterative solver) for each Newton step.	().
max_num_iters : Maximum number of Newton iterations.	().
models : A list that may contain the following: harmonic or anharmonic Lazarenkova or anharmonic Areshkin or anharmonic Sui; stretch-bend; cross-stretch; coplanar-2ndNN; coulomb.	().

Parameter : Description	Default Value
modify_ideal_length : (coulomb only) If set to true, then the parameter d which in the absence of Coulomb interaction has the meaning of an ideal bond length r0 is modified such that the enrgy functional retains its minimum at r0 .	().
name : A unique name tag / identifier for the specific instance of the solver.	
num_harmonic_iters : Taking the harmonic model while the Newton iteration is far away from the solution improves convergence and speeds up the simulation. This number specifies for how many iterations this is done.	().
output_file_suffix : suffix to be added on output files	().
phonon_solver : The name of the Phonons simulation object, if present. Existence of this option enables the set-up of the dynamical matrix.	().
preconditioner : Which preconditioner is employed in the Newton iteration. See the PETSc manual for possible choices - asm is preferred.lu does not work for simulations with grid-parallelization.	().
relative_tol : Relative tolerance convergence criterion of the Newton iteration.	().
silo_file : When set then the atomic displacement and possibly the strain tensor are saved to a Silo file.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
test_mat_vec_mult : Debug routine to test matrix vector multiplication using the Hessian of the strain simulation.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
vtk_file : When set then the atomic displacement and possibly the strain tensor are saved to a VTK file. Works only for simulations without grid parallelization.	().
vtk_storage : Choose from ascii or binary.	().

Parameter : Description	Default Value

Table 5.142: Parameters of the StrainVFF Solver.

5.143 Structure

This simulation outputs the atomistic structure of a certain domain to file

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__atoms__only : if True, then any non-active atoms (eg H-passivation atoms connected to non-periodic surfaces) will not be saved	(false).
active__regions : required regions to be activated for this solver	(()).

Parameter : Description	Default Value
boundary_regions : required regions as boundary conditions	(()).
crystal_additions :	(false).
crystal_viewer_planes : Crystal Viewer nanoHUB tool needs plane polygon data in Angstroms-set to true	(false).
domain : The name of the domain on which the simulation is carried out. corresponding the name parameter in the Domain section	
info_file : A file into which some crystallographic information is saved	().
miller_coloring :	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
output_format : One of silo, vtk, jmol, pdb or (for simple_cubic lattices only) dx. If the structure is distributed onto several MPI-processes, only Silo is an option	(vtk).
structure_file : The file into which the domain is saved	(structure.vtk).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Tic Toc name	($name).
unit_cell_file :	(unit_cell.vtk).

Table 5.143: Parameters of the Structure Solver.

5.144 TDSManager

Module for managing TDSchroedinger solver.

Parameter : Description	Default Value
$(quantum_numbers)_constructor : Solvers that construct the grids for each quantum number given by the option quantum_numbers.	
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
device_Hamiltonian_constructor : Solver that constructs the device Hamiltonian. For now this option is only necessary in the test code when it is necessary to give the time-depedendent test potential to the Hamiltonian constructor.	
domain : Domain required for this simulation	().

Parameter : Description	Default Value
initial_condition_solver : Solver that provides initial condition, e.g. wave functions or Green functions at initial time.	
lead_solvers : Solvers that calculate open boundary conditions for each boundary (lead).	(empty list).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
print_atomic_resolved_data : Specifies whether to print atomic resolved data. Should be true only if there are small number of time steps and quantum numbers. Possible vaues: true/false.	(false).
quantum_numbers : List with names of all good quantum numbers for this simulation. The final results are obtained by integrating over these quantum numbers. Quantum numbers will be parallelized. Parallelization tree hierarchy will be determined internally. Possible values: energy, momentum_1D, momentum_2D, valley.	
surface_of_regions : surface regions required to calculate boundary conditions	().
test_case : Determines which test code is to be used. 1 <= test_case <= N, where N is the number of implemented tests in this solver.	(1).
test_device_type : Specifies whether the device is closed (no current-carrying leads) or open. Possible vaues: closed and open.	(closed).
test_final_time : Final simulation time used in tests.	(1.e-2 [ps]).
test_number_of_time_steps : Number of time steps used in tests.	(10).
test_potential_parameters : Vector of test time-dependent potential parameters: spatial width [nm], amplitude [eV], period [ps].	((2.0,-100.e-3,1.0)).

Parameter : Description	Default Value
test_potential_type : Type of test time-dependent potential variation: periodic or rising. If periodic then the last parameter in test_potential_parameters is the period, while if rising then it is the rise time after which the potential stays constant.	(periodic).
test_print_ratio : Ratio of total number of time points to print. If test_print_ratio is x, then every 1/x time will be printed. x must be between 0.0 and 1.0.	(1.0).
tic_toc_name : Prefix for tic-tocs	($(name)).
time_evolution_solver : Solver that does the time evolution of a physical object, like wave function or Green's function.	

Table 5.144: Parameters of the TDSManager Solver.

5.145 TDSchroedinger

Solver for the linear time-dependent Schroedinger equation.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).

Parameter : Description	Default Value
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
debug__output__files : Defines for which jobs in job_list the debug output should be printed. This is a list with names corresponding to the names in job_list. This option will in general cause printing of several matrices for each unique combination of quantum numbers, so the number of quantum number sets should be low to reduce storage size and print time.	(empty).
device__data : Solver that provides device Hamiltonian. Currently API as in Schroedinger is assumed.	
diagonal__shift__only : Speed up system assembly job when only device potential changes.	(false).
domain : Domain required for this simulation	().
equation__system__type : Type of time-dependent equation system to be solved. Possible values: mixed_DOF, mixed_DOF_transformation, orbital_DOF.	(mixed_DOF).
injected__qndomain : Name of QNDomain for injected space. This is domain of quantum numbers that are injected into device. At least there must be injection from one full set of quantum numbers of one lead. QNDomain contains boundary index as quantum number. This option can only be set at run-time by another solver.	().

Parameter : Description	Default Value
iterative_linear_solver_type : Type of petsc iterative linear solver to be used.	(gmres).
job_list : Defines the list of jobes to be executed by TDSchroedinger. Possible values: boundaryx, where x must be consecutive numbers starting at 1, assemble_system, time_step.	(empty list).
lead_Hamiltonian : List of solvers that are Hamiltonian constructors for each lead. The order of solvers should correspond to boundary job name numbering in job_list.	
lead_data : List of solvers that provide data for each boundary: lead modes/phases and Hamiltonians. Currently solvers of type PartitionRetardedSelfModule are assumed. The order of solvers should correspond to boundary job name numbering in job_list.	
lead_lead_domain : List of domain names that are leads of leads. The order of names should correspond to boundary job name numbering in job_list.	
linear_solver_family : Type of linear solver: iterative or direct. If iterative then the user can specify which petsc solver to use by iterative_linear_solver_type and which preconditioner by preconditioner_type. If direct, mumps with LU decompositions is used.	(iterative).
message_rank : Rank on which solver messages are printed.	(0).
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
preconditioner_type : Type of petsc preconditioner to be used with iterative linear solver.	(bjacobi in parallel and ilu in serial).

Parameter : Description	Default Value
scattered_qndomain : Name of QNDomain for scattered space. This is domain of quantum numbers that the injected quantum number/s can scatter to due to time-dependent Hamiltonian. In the limiting case without scattering this QNDomain contains only injected quantum number/s. QNDomain contains boundary index as quantum number. This option can only be set at run-time by another solver.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
time_stepping_solver : Defines the numerical solver for time-integration.	(PETSC_CN).
wavefunction_data : Solver that provides wavefunction data at initial or previous times. API must be as in TDSManager.	

Table 5.145: Parameters of the TDSchroedinger Solver.

5.146 TaskOffload

TaskOffloadSolver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).

Parameter : Description	Default Value
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
coprocessors__per__node : Number of coprocessors per node on current system	(1).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
number__of__cpu__threads : Set number of threads to be spawned on CPU for pcl_lapack_zgemm tiling	(16).
number__of__nodes : Set total number of nodes; Option is required when set_number_of_nodes is true	
offload : Set whether offload solver should offload to a coprocessor	
offload__to : Type of coprocessor to offload to; supported options are "MIC" and "GPU"	(CPU).
offloading__rank__load : Workload of MPI processes that offload to a coprocessor; currently not supported	(1).
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
processes_per_node : Number of MPI processes per node on current system	(16).
set_number_of_nodes : Manually set total number of nodes; this may be required on systems where MPI ranks are distributed non-contiguously when each host has more than one coprocessor	(false).
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.146: Parameters of the TaskOffload Solver.

5.147 TopOfBarrier

TopOfBarrierSolver
This is Mark Lundstrom's famous top-of-the-barrier model [7]. It allows for the rapid calculation of I-V curves and other characteristics of ballistic nanotransistors gien that the density of states (DOS) and the average carrier velocity (vavg) is known from another simulation. Alternatively the DOS and vavg it can be computed by this simulation given knowledge of the E(k).

Parameter : Description	Default Value
Cdrain : The drain-channel capacitance, in F/mdim . Can be set to 0 for ideal electrostatics.	().
Cgate : The gate-channel capacitance, in F/mdim . It is unusual to specify it like this, therefore a specification with EOT can be made instead.	().
Csource : The source-channel capacitance, in F/mdim . Can be set to 0 for ideal electrostatics.	().
DOS_file : Name of a file containing the density of states in the format E D(E).	().
EF : The Fermilevel of the source contact. Raising this results in shifting the ID(VG)-curve to the left.	().

Parameter : Description	Default Value
EOT : Effective oxide thickness of the dielectric separating the gate contact and the channel: CG = (3.9epsilon)/(EOT).	().
EOT_QW : (if tripleQW is used) The capacitance between two wells will be CQW = (3.9epsilon)/(EOTQW)	().
Ek_file : Name of a file containing E(k) in the format kx ky kz E1 E2... Specifying this option overrides DOS file and vavg file. This file is typically generated by a Schroedinger simulation (ending energies.dat).	().
Emax : (if Ek file is used, in eV) Energy maximum of the regular grid set up for D(E) and vavg(E).	().
Emin : (if Ek file is used, in eV) Energy minimum of the regular grid set up for D(E) and vavg(E).	().
Utol : Numerical parameter to adjust the convergence criterion in the solution for U .	().
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).

Parameter : Description	Default Value
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
bands : number of bands desired for calculation, counted from bottom of conduction band or from top of valence band.	().
boundary_regions : required regions as boundary conditions	(()).
broadening : (if Ek file is used, in eV) Every energy eigenvalue is broadened by this amount to smoothen the DOS and vavg curves.	().
dE : spacing of D(E), vavg(E) energy grids	().
dUmax : Numerical parameter to adjust the maximum update per Newton step in the solution for U .	().
dim : The dimensionality of the problem: 1 for a nanowire or fin geometry, 2 for a thin body. This must be in agreement with Ek file if that option is used.	().
domain : Domain required for this simulation	().
double_gate : (default: false) If true, then CG will be doubled CG = (3.9epsilon)/(EOT)	().
drain_bias : Vector containing the voltages applied to the drain contact.	().
electron : Set this to true if doing electrons.	().
file : The output file where all results will be written to.	().
files : file where Idrain, Qtop, Uscf, CQ are stored	().
gate_bias : Vector containing the voltages applied to the gate contact. Must have the same length as drain bias.	().

Parameter : Description	Default Value
gradient : (if Ek file is used) Direction in which the velocity is computed. The direction is specified in terms of the reciprocallattice vectors and should have dim components.	().
gradient_dir : direction of gradient in terms of periodicity vectors	().
hole : Set this to true if doing holes.	().
k_deg : (if Ek file is used) The computed DOS is multiplied by thisnumber to account for degeneracy in k-space. Like this itsuffices typically to compute half or 1/4 of the Brillouin zone.	().
max_iters : Numerical parameter to adjust the maximum number of Newton iterations in the solution for U .	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
spin_deg : (if Ek file is used) The computed DOS is multiplied by this number to account for spin degeneracy.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
temperature : The temperature in Kelvin.	().
threshold : threshold point to distinguish between elec & holes. Ideally a number in bandgap.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
tripleQW : (default: false) If true, then a special model will be turned on that accounts for capacitances between the three quantumwells.	().
vavg_file : Name of a file containing the energy-resolved average velocity in the format E vavg(E). The energy grid must be the same as for the DOS.	().

Table 5.147: Parameters of the TopOfBarrier Solver.

5.148 Transformation

TransformationSolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
atom__order__from : Determines ordering of atoms in Hamiltonian when repartitioner solver is attached to DOFmap. Most likely, set this to the name of the repartition solver	().

Parameter : Description	Default Value
basis_generator : This defines where the basis to be transformed comes from. It can either be the Transformation solver or another solver. If it is the transformation solver, then a reduced order matrix is formed based on the ratio_of_eigenvalues_to_be_solved option. Or if transformation_method = modespace, the number_of_eigenmodes multiplied by the the number of slabs	().
basis_matrix : The propagator that provides the original source matrix to be used for Transformation	().
basis_set_at_energies : If a solver is looking for basis functions, the solver can specify what energy range to use for finding/generating the basis set by setting this option	().
boundary_regions : required regions as boundary conditions	(().
convergence_limit : The tolerance value that needs to be met before the iterative eigensolver will stop and claim convergence	(1e-8).
debug_output : A catch-all for intermediate results such as matrices, energies, etc that are only outputted if this is set true. Setting this to true can have the adverse effect of slowing down the code and should only be used while debugging.	(false).
density_multiplication_factor : This sets the degeneracy factor scaling the density.	().
domain : Domain required for this simulation	().
get_momentum_from : If basis_set_at_energies exists the solver need to find out where to get the momentum from. This option should give the name of the Propagator where the momentum to be used can found. This must be used in conjunction with basis_set_at_energies. The solver for this Propagator is specified as 'propagator_name'_solver	().

Parameter : Description	Default Value
max_number_iterations : The maximum number of iterations to be used by the eigensolver. If this value is reached the eigensolver will stop	(300).
monitor_convergence : For output from the eigensolver to be shown this value should be set to true	(false).
name : A unique name tag / identifier for the specific instance of the solver.	
ncv : This is used by the Krylov iterative solvers. It defines the size of the Krylov subspace	().
number_of_eigenmodes : Only used with the modespace method. Should be set to the number of eigenmodes to be calculated for each slab.	(0).
output_file_suffix : suffix to be added on output files	().
precond_type : This option controls how the eigenvalue problem is tranformed before solution by the solver_type. Possible options are 'lu', 'mumps'. Other options exist and an exhaustive list can be found in SLEPc documentation	().
ratio_of_eigenvalues_to_be_solved : This is used to reduce the source matrix by creating a Transformation matrix based on this value. The solver will only solve the fraction of the total eigenvalues based on this option. A value of 1 will solve all eigenvalues of the source matrix.	().
save_basis_set : This is used to let the Transformation solver wheter to save the basis set between calls. It is only relevant when the the basis_generator is set to Transformation solver type.	(false).
set_up_Propagation : This is used when the basis_matrix is 'Hamiltonian'. This option is used to get the momentum from Propagator (usually fake) for use with Schroedinger	().

Parameter : Description	Default Value
shift : This value is used by the transform_type to reposition the eigenvalue problem around this value. Increased convergence hasbeen found when using this value near a good initial guess for eigenvalues	(0.0).
solver_type : Used to choose which eigensolver type to use. For an exhaustive list of options see SLEPc documentation.	(lapack).
source_matrix_constructor : This option sets where the source matrix is coming from. The source matrix is a Propagator commonly used examples are 'inverse_retarded_Green_fermiond'. The source_matrix_constructor has a closely related option, the solver. The solver is defined as 'source_matrix_name'_solver. For the 'inverse_retarded_Green_fermiond' example it would be 'inverse_retarded_Green_fermiond_solver'.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
transform_type : Main choices are none, shift or sinvert. These transforms are used by the eigensolver to shift the eigenvalue problem to a zero point near the value given by shift.	(none).
transformation_method : This lets the solver know if this is the modespace transformation method or not. If this is equal to 'modespace' then a different method of finding the basis matrix is used. Mainly, eigenmodes are calculated based off of number_of_eigenmodes and the number of slabs. This is then used to form a block diagonal Transformation matrix for transformation of the source matrix into a Modespace basis.	().

Table 5.148: Parameters of the Transformation Solver.

5.149 TranslateSelfEnergy

TranslateSelfEnergySolver

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
boundary__regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output__file__suffix : suffix to be added on output files	().

Parameter : Description	Default Value
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.149: Parameters of the TranslateSelfEnergy Solver.

5.150 UnderBarrier

UnderBarrierSolver

Parameter : Description	Default Value
Cdrain : The drain-channel capacitance, in F/mdim . Can be set to 0 for ideal electrostatics.	().
Cgate : The gate-channel capacitance, in F/mdim . It is unusual to specify it like this, therefore a specification with EOT can be made instead.	().
Csource : The source-channel capacitance, in F/mdim . Can be set to 0 for ideal electrostatics.	().
DOS_file : Name of a file containing the density of states in the format E D(E).	().
EF : The Fermilevel of the source contact. Raising this results in shifting the ID(VG)-curve to the left.	().
EOT : Effective oxide thickness of the dielectric separating the gate contact and the channel: CG = (3.9epsilon)/(EOT).	().
EOT_QW : (if tripleQW is used) The capacitance between two wells will be CQW = (3.9epsilon)/(EOTQW)	().
Ek_file : Name of a file containing E(k) in the format kx ky kz E1 E2... Specifying this option overrides DOS file and vavg file. This file is typically generated by a Schroedinger simulation (ending energies.dat).	().

Parameter : Description	Default Value
Emax : (if Ek file is used, in eV) Energy maximum of the regular grid set up for D(E) and vavg(E).	().
Emin : (if Ek file is used, in eV) Energy minimum of the regular grid set up for D(E) and vavg(E).	().
Utol : Numerical parameter to adjust the convergence criterion in the solution for U .	().
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
active__regions : required regions to be activated for this solver	(()).
bands : number of bands desired for calculation, counted from bottom of conduction band or from top of valence band.	().

Parameter : Description	Default Value
boundary_regions : required regions as boundary conditions	(()).
broadning : (if Ek file is used, in eV) Every energy eigenvalue is broadened by this amount to smoothen the DOS and vavg curves.	().
dUmax : Numerical parameter to adjust the maximum update per Newton step in the solution for U .	().
dim : The dimensionality of the problem: 1 for a nanowire or fin geometry, 2 for a thin body. This must be in agreement with Ek file if that option is used.	().
domain : Domain required for this simulation	().
double_gate : (default: false) If true, then CG will be doubled CG = (3.9epsilon)/(EOT)	().
drain_bias : Vector containing the voltages applied to the drain contact.	().
electron : Set this to true if doing electrons.	().
file : The output file where all results will be written to.	().
gate_bias : Vector containing the voltages applied to the gate contact. Must have the same length as drain bias.	().
gradient : (if Ek file is used) Direction in which the velocity is computed. The direction is specified in terms of the reciprocallattice vectors and should have dim components.	().
hole : Set this to true if doing holes.	().
k_deg : (if Ek file is used) The computed DOS is multiplied by thisnumber to account for degeneracy in k-space. Like this itsuffices typically to compute half or 1/4 of the Brillouin zone.	().
max_iters : Numerical parameter to adjust the maximum number of Newton iterations in the solution for U .	().
name : A unique name tag / identifier for the specific instance of the solver.	

Parameter : Description	Default Value
output_file_suffix : suffix to be added on output files	().
spin_deg : (if Ek file is used) The computed DOS is multiplied by this number to account for spin degeneracy.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
temperature : The temperature in Kelvin.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
tripleQW : (default: false) If true, then a special model will be turned on that accounts for capacitances between the three quantumwells.	().
vavg_file : Name of a file containing the energy-resolved average velocity in the format E vavg(E). The energy grid must be the same as for the DOS.	().

Table 5.150: Parameters of the UnderBarrier Solver.

5.151 Unfold

Unfold Solver ...

Parameter : Description	Default Value
band_solver : Name of the schroedinger/phonon solver	().
debug : debug=true will dump out useful debug info	(false).
k_points : Single point to be specified in schroedinger solver ... cartesian	().
k_space_basis : Needs to be cartesian	().
maximum_energy : Maximum energy to be considered	(0.0).
min_cum_prob_for_band : Minimum probability that defines a band	(0.9).

Parameter : Description	Default Value
min_resolvable_gap : Minimum resolvable gap between bands	(1e-6).
minimum_energy : Minimum energy to be considered	(0.0).
only_run_bandguess : Should only the band guessing algo be run?	(false).
qpoints : Single point to be specified in phonon solver ... cartesian	(()).
qspace_basis : Needs to be cartesian	(()).
refine_tight : bool type use to define the solution of peaks	(false).
small_cell : small_cell_kspace will be saved in this name	(()).
strain_type : Only uses none. relaxed is ignored, strained not implemented	().
tiny_probability : Smallest probability that should be considered zero	(1e-3).

Table 5.151: Parameters of the Unfold Solver.

5.152 VFFStrain

VFFStrainSolver
This simulation relaxes the atomic positions by minimizing the lattice energy computed via the valence force field (VFF) model. Depsite the misleading simulation name, several models for the energy functional are available through a mix-and-match approach steered by the input deck.
Parallelism: The spatial parallelization given in the Partitioning section of the input deck is employed to distribute the Hessian matrix as well as the displacement and right-hand-side vectors onto multiple MPI processes. Like this a simulation of a 44-million InAs quantum dot has been achieved using 216 CPUs.

Parameter : Description	Default Value
: If true then the strain tensor components are calculated according to the method of Pryor and Zunger as a post-processing step.	().

Parameter : Description	Default Value
__disable__init : flag command to disable initialization	(false).
__disable__output : flag command to disable solver output execution	(false).
__disable__reinit : flag command to disable re-initialization	(false).
__disable__reset__set : flag command to disable re-initialization of sets	(false).
__disable__solve : flag command to disable solver solution	(false).
__disable__step : flag command to disable change of sets when iterators are defined	(false).
__init : list of solver to trigger init command	(()).
__output : list of solver to trigger output command	(()).
__reinit : list of solver to trigger reinit command	(()).
__search__flag : flag command to search nested solver	(2).
__solve : list of solver to trigger solve command	(()).
__step : list of solver to trigger step command	(()).
__trigger__flag : nested execution of solvers,	(4).
absolute__tol : Absolute tolerance convergence criterion of the Newton iteration.	().
active__regions : required regions to be activated for this solver	(()).
anharm__cutoff : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
anharm__smoothing : (anharmonic Areshkin only) dimensionless numerical smoothing parameter.	().
anharm__theta__limit : (anharmonic Lazarenkova only) dimensionless numerical cutoff parameter.	().
boundary__regions : required regions as boundary conditions	(()).
calculate__epsilon : bool, if true, calculate epsilon.	(false).

Parameter : Description	Default Value
coulomb_cutoff_radius : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interaction is neglected. This parameter influences thesparsity of thematrix and memory consumption.	().
coulomb_ewald_Kcut : (coulomb only) (3D-periodic (bulk) simulations only) integer cutoff outside which the long-range contribution is neglected,in units of reciprocal lattice vectors.	().
coulomb_ewald_Rcut : (coulomb only) (3D-periodic (bulk) simulations only) cutoff in real-space	().
domain : Domain required for this simulation	().
ewald_sigma : (coulomb only) For simulations without periodicity this is a cutoff radius in nm outside which the Coulomb interactionis neglected. This parameter influences the sparsity of thematrix and memory consumption.	().
hessian_info : If true then some numerical debug terminal output is created.	().
linear_solver : Which linear solver is employed in the Newton iteration. See the PETSc manual for possible choices - gmres is preferred.	().
linsolver_monitor : If true then the convergence of the iterative linear solver is monitored as terminal output.	().
linsolvers_max_iters : Maximum number of linear solver iterations (in the case of an iterative solver) for each Newton step.	().
max_num_iters : Maximum number of Newton iterations.	().
models : A list that may contain the following: harmonic or anharmonic Lazarenkova or anharmonic Areshkin or anharmonic Sui; stretch-bend; cross-stretch; coplanar-2ndNN; coulomb.	().

Parameter : Description	Default Value
modify_ideal_length : (coulomb only) If set to true, then the parameter d which in the absence of Coulomb interaction has the meaning of an ideal bond length r0 is modified such that the enrgy functional retains its minimum at r0 .	().
name : A unique name tag / identifier for the specific instance of the solver.	
num_harmonic_iters : Taking the harmonic model while the Newton iteration is far away from the solution improves convergence and speeds up the simulation. This number specifies for how many iterations this is done.	().
output_file_suffix : suffix to be added on output files	().
phonon_solver : The name of the Phonons simulation object, if present. Existence of this option enables the set-up of the dynamical matrix.	().
preconditioner : Which preconditioner is employed in the Newton iteration. See the PETSc manual for possible choices - asm is preferred.lu does not work for simulations with grid-parallelization.	().
relative_tol : Relative tolerance convergence criterion of the Newton iteration.	().
silo_file : When set then the atomic displacement and possibly the strain tensor are saved to a Silo file.	().
surface_of_regions : surface regions required to calculate boundary conditions	().
test_mat_vec_mult : Debug routine to test matrix vector multiplication using the Hessian of the strain simulation.	().
tic_toc_name : Prefix for tic-tocs	($(name)).
vtk_file : When set then the atomic displacement and possibly the strain tensor are saved to a VTK file. Works only for simulations without grid parallelization.	().
vtk_storage : Choose from ascii or binary.	().

Parameter : Description	Default Value

Table 5.152: Parameters of the VFFStrain Solver.

5.153 ValleyGrid

Valley Grid Solver

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().

Parameter : Description	Default Value
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).
valleys : list of valleys names	(()).

Table 5.153: Parameters of the ValleyGrid Solver.

5.154 WF

DEPRECATED.
Open-boundary wavefunctions are superior to the NEGF formalism when it comes to the computation of ballistic transport properties. Simulation times are roughly 10 times faster. The physics is, however, limited to ballistic simulations (no scattering).
Parallelism: The wavefunction solver has a three-level parallelization: energies, k-points, and a spatial decomposition that is based on the numebr of contacts (using a generalized recursive algorithm). In compination with Ramper this yields a 4D parallelization. Parallelization is done automatically except for the voltages.

Parameter : Description	Default Value
_disable_init : flag command to disable initialization	(false).
_disable_output : flag command to disable solver output execution	(false).
_disable_reinit : flag command to disable re-initialization	(false).
_disable_reset_set : flag command to disable re-initialization of sets	(false).
_disable_solve : flag command to disable solver solution	(false).
_disable_step : flag command to disable change of sets when iterators are defined	(false).
_init : list of solver to trigger init command	(()).

Parameter : Description	Default Value
_output : list of solver to trigger output command	(()).
_reinit : list of solver to trigger reinit command	(()).
_search_flag : flag command to search nested solver	(2).
_solve : list of solver to trigger solve command	(()).
_step : list of solver to trigger step command	(()).
_trigger_flag : nested execution of solvers,	(4).
active_regions : required regions to be activated for this solver	(()).
boundary_regions : required regions as boundary conditions	(()).
domain : Domain required for this simulation	().
name : A unique name tag / identifier for the specific instance of the solver.	
output_file_suffix : suffix to be added on output files	().
surface_of_regions : surface regions required to calculate boundary conditions	().
tic_toc_name : Prefix for tic-tocs	($(name)).

Table 5.154: Parameters of the WF Solver.

Chapter 6

Material Parameter Names

This chapter is a reference on which material parameters are relevant for which solver.

`Ramper`, `ResonanceFinder`, `Propagation`, `Angel`, `MatrixElements` simulations:
These simulations are either dysfunctional or they do not directly implement a physical equation. Hence there are no material parameters associated with them.

6.1 `Structure`, `Brillouin`

These simulations depend only on the atomic grid and hence are only dependent on the parameters forming the grid. All these parameters are located in the `Lattice` group of the material database.

6.2 `VFFStrain`, `Phonons`

In the following table `X` always denotes the name of an atom, i.e. Si, Ga, As etc. All parameters are located in the `Lattice` group in the material database.

6.3 `Poisson`, `NonlinearPoisson`

For the Poisson equation only the static dielectric constant is needed.

Crystal structure	Parameter
simple_cubic	a_lattice
diamond, zincblende	a_lattice
wurtzite	a_lattice, a_wurtzite, c_wurtzite
Bi2Te3	a_lattice, c_lattice, u_lattice, v_lattice

Table 6.1: Material parameters relevant for atomistic grid construction.

Parameter	Description
strain_alpha	Keating's bond-stretching parameter .
strain_beta OR	Keating's bond-bending parameter .
strain_beta_X	If a separate parameter strain_beta_X is present for all atom types X than those will be used instead.
strain_gamma OR	Cross-stretch interaction parameter.
strain_cross_stretch_X	If separate parameters are present for all atom types then those will be used.
strain_delta OR	Stretch-bend interaction parameter.
strain_stretch_bend_X	If separate parameters are present for all atom types then those will be used.
strain_nu	Second-nearest-neighbor coplanar interaction parameter.
strain_anharm_...	Anharmonicity parameters used in the Lazarenkova, Areshkin or Sui/Herman anharmonicity models.
X_charge	Fractional point charge sitting at the atomic locations that induces a Coulomb interaction.
X_mass	Atom mass in amu., used only in dynamical matrix (phonons)

Table 6.2: Material parameters relevant for the VFFStrain and Phonons simulations.

Crystal structure	Parameter
simple_cubic,diamond, zincblende	Lattice:epsilon_dc
wurtzite, Bi2Te3	Not yet implemented.

Table 6.3: Material parameters relevant for the Poisson and NonlinearPoisson simulations.

6.4 Schroedinger, WF, NEGF

These simulations are all centered around the electronic Hamiltonian, and thus the following parameters are needed:

Tight-binding or effective-mass parameters for the Hamiltonian.

For tight-binding simulations where an analytical formula is used to compute the density from $k = 0$ results, effective-mass parameters are also used.

The nomenclature for tight-binding parameters is consistent and straightforward to interpret. For example, `E_Sstar_Ga` denotes the onsite energy of the s orbital and `V_S_P_Sigma_As_Ga` is the -coupling element between the arsenic s-orbital and the gallium p-orbital. Note that when entries are symmetric (e.g. `V_S_S_Sigma_X_Y` must always be the same as `V_S_S_Sigma_Y_X`) then only a single entry is used where the atomic sorts `X` and `Y` are ordered alphabetically. Effective-mass simulations are currently implemented with a fake tight-binding model using a single s-orbital.

Parameter	Description
tight-binding parameters	The parameter sets are located in the group `Bands:TB:<tbmodel>:param_XYZ`, where `<tbmodel>` corresponds to the model given in the input deck (e.g. `sp3sstar_SO`) and `param_XYZ` is the name of the employed parameter set (e.g. `param_Klimeck`).
`Ec`, `Ev`	Located in `Bands:BandEdge`. These parameters are used for the distinction between electrons and holes.
`mstar_{c,v}_dos`	Located in `Bands:BandEdge`. These parameters are used in effective-mass simulations and when computing the density fom $k = 0$-results assuming a parabolic transverse dispersion.

Table 6.4: Material parameters relevant for the `Schroedinger`, `WF` and `NEGF` simulations.

Choice of tight-binding parameter sets

This choice can be made by setting `Bands:TB:<tbmodel>:param_set = param_XYZ` in the `Material` section in the input deck.

Valence band offsets

Valence band offsets can be adjusted either directly in the material file (typically `all.mat`) by changing the parameter `Bands:BandEdge:VBO` or `Bands:TB:<tbmodel>:param_XYZ:VBO`, or they can be adjusted by setting the same parameter in the corresponding `Material` section in the input deck. The onsite tight-binding energies will then be shifted by a constant value.

6.5 Semiclassical

All the Semiclassical material parameters are located in the group Bands:BandEdge.

Parameter	Description
Ec, Ev	Bulk conduction and valence band edge.
mstar_c_dos, mstar_v_dos	Density-of-states masses.

Table 6.5: Material parameters relevant for the Semiclassical simulation.

Bibliography

[1] S. Steiger, M. Povolotskyi, T. Kubis, H.-H. Park and G. Klimeck, Manuscript accepted to IEEE Transactions of Nanotechnology (2011).

[2] R. Lake, G. Klimeck, R. C. Bowen and D. Jovanovic, J. Appl. Phys. **81**, 7845 (1997).

[3] G. Klimeck *et al.*, IEEE Trans. Electr. Dev. **9**, 2079 (2007).

[4] G. Klimeck *et al.*, IEEE Trans. Electr. Dev. **9**, 2090 (2007).

[5] M. Luisier, A. Schenk, W. Fichtner and G. Klimeck, Phys. Rev. B **74**, 205323 (2006).

[6] A. Trellakis, A. T. Galick, A. Pacelli and U. Ravaioli, J. Appl. Phys. **81**, 7880 (1997).

[7] A. Rahman, J. Guo, S. Datta and M. Lundstrom, Electron Devices, IEEE Transactions on **50**, 1853 (2003).