Padre 2.4E
Users Manual

PADRE simulates the electrical behavior of semiconductor devices under steady state, transient conditions or AC small-signal analysis. Multiple devices can be treated, along with lumped element circuit networks. *PADRE* can simulate physical structures of arbitrary geometry (including heterostructures)-with arbitrary doping profiles that can be obtained using analytical functions or directly from multidimensional process simulators such as *PROPHET* or *BICEPS*.

Padre was developed in 1994 at Bell Laboratories by Mark Pinto, R. Kent Smith, and Ashraful Alam. This manual is the original, Bell labs version, which described how to run it in their environment.

Introduction

PADRE simulates the electrical behavior of devices under steady state, transient conditions or AC small-signal analysis. Multiple devices can be treated, along with lumped element circuit networks.

PADRE can simulate physical structures of arbitrary geometry--including heterostructures-with arbitrary doping profiles, which can be obtained using analytical functions or directly from multidimensional process simulators such as *PROPHET* or *BICEPS*.

For each electrical bias, *PADRE* solves a coupled set of partial differential equations (PDEs). A variety of PDE systems are supported which form a hierarchy of accuracy:

- electrostatic (Poisson equation)
- drift-diffusion (including carrier continuity equations)
- energy balance (including carrier temperature)
- electrothermal (including lattice heating)

A variety of physical mechanisms are supported within these formulations, including comprehensive representations of carrier mobility, generation/recombination, and boundary conditions. *PADRE* results can be supplemented by the Monte Carlo device simulator *SMC* which is more appropriate for studying hot carrier effects such as MOS substrate and gate current.

Solutions are represented on a finite element grid within the space of the device. *PADRE* supports box discretizations and general finite element discretizations through non-uniform triangular grids, which can be refined during the simulation process.

Compared with other device simulation programs, *PADRE* has a number of important features:

- Energy balance equations
- Heterostructures
- Enhanced models
- More automatic error-controlled grid generation
- IV predictor-corrector continuation methods
- Modular vector/parallel assembly
- Vector/parallel linear algebra
- Mixed coupled/decoupled nonlinear iterations

More Information

PADRE is accessed using the xtcad command .

PADRE includes a number of utility programs used to process input/output files:

bipad
: Converts process simulator output from *BICEPS* or *PROPHET* into a format suitable for *PADRE*. It also supports simple manipulations, including truncating, stretching and mirroring of device profiles. These facilities are useful, for instance, when converting the process simulator output for a half-MOSFET into a full MOSFET device for electrical simulation.

skel
: An interactive program for building the "skeleton" file that describes the geometry of a device.

tri
: Takes a "skeleton" and produces the initial triangular mesh used for the *PADRE* simulation.

scat2avs
: Converts a 3D output file (produced using the plot.3d command within *PADRE*) to the format required by the AVS visualization system.

pad2a2
: Converts the I-V characteristics in a *PADRE* logfile (produced using the log command within *PADRE*) to the A2 file format used for parameter extraction. Only useful for translating MOS device characteristics.

Contacts

- Mark R. Pinto (research!pinto), MH 6E-306
- Michael J. McLennan (mhcnet!mmc), ALC 2C-226

The REGRID/ADAPT line

Syntax

```
  REGRid  location i/o variable criterion control interpolation
  ADapt   location i/o variable criterion control interpolation
```

Description

The REGRID and ADAPT lines allows refinement or coarsening (unrefining) of a mesh. Any element across which the chosen variable changes by more than a specified tolerance, or in which the chosen variable exceeds a given value, is refined subject to some additional constraints. Conversely coarsening (unrefinement) takes place when a chosen variable changes by less than a specified tolerance or a chosen variable is below a given value. The REGRID line performs the specified grid modification only once whereas the ADAPT line automatically attempts grid adaption at any newly acquired operating point after its occurence, repeating the associated calculation until an adequate grid is achieved.

Parameters

location

```
  X.MIn       =  real
  X.MAx       =  real
  Y.MIn       =  real
  Y.MAx       =  real
  Z.MIn       =  real
  Z.MAx       =  real
  REGion      =  integer  (default:all)
  IGnore      =  integer  (default:none)
  BOX.refine  =  logical  (default is false)
```

The bounds *.MIN and *.MAX are used to limit the regrid; only elements which have nodes which fall inside the box are considered for refinement or coarsening. The REGION parameter has a similar use; only regions specified are adjusted according to the user criterion. (Others may be refined/coarsened as a side effect, to maintain well-shaped elements). The default is to regrid all regions for potential and electric field types, and all semiconductor regions for regrids which depend on the other variables. The parameter IGNORE is similar to REGION, but opposite in effect. Ignored regions are not regrided either according to the user criterion or according to the "obtuse criterion" (see below); nor are they smoothed after regrid. The default is not to ignore any region. BOX.REFINE specifies that regrids should take place inside purely rectangular regions as generated by tri; typically one does not want to regrid these regions as they can have highly anisotropic elements.

i/o

```
  OUTFile    =  filename
  OUT.green  =  filename
  IN.green   =  filename
  NO.green   =  logical    (default is false)
  DOPFile    =  filename
  AScii      =  logical    (default is true)
  STats      =  logical    (default is false)
```

OUTFILE is the binary output mesh file, and is necessary if the mesh is to be used for subsequent runs. A history of the triangle tree is always generated to assist further regriding steps. Its name can be specified by OUT.GREEN, and its default is generated from OUTFILE by concatenating the letters "tt" to the end. No such file will be created if NO.GREEN is specified. Additionally, a triangle tree for the previous mesh (if a tree exists) is used for this regrid. By default, *PADRE* will look for a file with the same name as the current mesh plus "tt" at the end as above. Alternatively, IN.GREEN can be used to implement a different file name. DOPFILE is the name of a file (up to 20 characters) which contains the doping for the device (see DOPING line). Specifying DOPFILE avoids interpolating doping values at any newly created grid points (the default), by using the initial doping specification to redope the structure. ASCII specifies that all mesh files and triangle trees (not DOPFILE) for this line should be done in ascii rather than the default - binary. STATS prints some refinement statistics in the standard *PADRE* output file.

variable is one of:

```
  Potential  =  logical   Mid-gap potential (V)
  QFN        =  logical   Electron quasi-fermi level (V)
  QFP        =  logical   Hole quasi-fermi level (V)
  N.temp     =  logical   Electron temperature (eV)
  P.temp     =  logical   Hole temperature (eV)
  DOping     =  logical   Total net impurity concentration (/cm**3)
  IOn.imp    =  logical   Net ionized impurity concentration (/cm**3)
  ELEctron   =  logical   Electron concentration (/cm**3)
  Hole       =  logical   Hole concentration (/cm**3)
  NET.CHrg   =  logical   Net charge (/cm**3)
  NET.CArr   =  logical   Net carrier concentration (/cm**3)
  MIn.carr   =  logical   Minority carrier concentration (/cm**3)
  P.track    =  logical   Carrier density from particle tracks (/cm**3)
  HEtero     =  logical   Material interfaces (unitless)
  ERror      =  logical   Potential error estimate (V)
```

This parameter selects the discriminatory variable. For ADAPT input lines only the default is ERROR.

criterion

```
  R.STep       =  real
  C.STep       =  real
  CHange       =  logical
  RELAtive     =  logical  (default is false)
  DV.min       =  real     (default is Poisson x.tol)
```

```
R.Threshold  =  real     (default is r.step)
N.Threshold  =  real     (default is 200)
F.Threshold  =  real     (default is 0.1)
REfine       =  logical  (default is false)
COarsen      =  logical  (default is false)
LOCaldop     =  logical  (default is false)
LOGarithm    =  logical  (default is false)
ABsolute     =  logical  (default is false)
```

R.STEP is the actual numerical criterion for performing a element refinement while C.STEP is for coarsening (unrefining). If either the magnitude or change (as specified by CHANGE, which defaults to true unless LOCALDOP is set) in a variable exceeds R.STEP, refinement will take place subject to constraints on element size and level as discussed below; conversely, if the value for this element and its brothers is below C.STEP, coarsening (unrefinement) takes place. RELATIVE specifies that the tolerance be applied in a relative sense; e.g. with a voltage-based criteria, the values of R.STEP and C.STEP are taken as fractions of the maximum potential difference inside the device. To avoid overflow for RELATIVE comparisons to flat potential solutions, a minimum voltage differential is defined by DV.MIN; the default is the Poisson solution tolerance (see METHOD line), as the nonlinear solution is only accurate to this value. Unitless variables (HETERO) do not require criteria, and only refinement can take place there, subject to level and element size control.

The first set of constraints, particularly useful for the ADAPT line, are R.THRESHOLD, F.THRESHOLD and N.THRESHOLD which prohibit any grid modification regardless of R.STEP/C.STEP unless either (1) the largest deviation exceeds the value given by R.THRESHOLD or (2) the number of elements to be coarsened exceeds the absolute quantity N.THRESHOLD or the fraction F.THRESHOLD of the starting number of elements. If only R.STEP is given (typical for a REGRID line), R.THRESHOLD defaults to the same value as R.STEP while if only R.THRESHOLD is given (typical for ADAPT), R.STEP defaults to the 0.95*R.THRESHOLD. N.THRESHOLD defaults to 200 or 10% of the total number of elements in the grid, which ever is larger.

If neither R.STEP, R.THRESHOLD or C.STEP is specified, *PADRE* looks at the REFINE and COARSEN flags to determine which operation is to take place; in this case, the grid will be refined or coarsened until the level and size limits below are met. The parameter LOCALDOP compares the magnitude (default) or difference of a variable against the local doping values; this might for instance be used in conjunction with MIN.CARR to refine in areas within the device that are in high-level injection. LOGARITHM and ABSOLUTE specify that the logarithm and absolute value of the variable be used respectively (with LOGARITHM set, R.STEP/C.STEP will be interpreted as the step in the logarithm). Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM (absolute is taken first).

control

```
MAx.level    =  integer    (default is dynamic)
REL.level    =  integer    (default is 1)
LEv.ignore   =  logical    (default is false)
HMin         =  real       (default is 0)
HDir         =  logical    (default is true)
DEbye        =  real       (default is 0)
FReeze       =  logical    (default is false)
SMooth.k     =  integer    (default is 1)
IT.Smooth    =  integer    (default is 30)
COs.ang      =  real       (default is 2.0)      [Expert]
BOUnd.green  =  logical    (default is true)     [Expert]
G.Semi       =  integer    (default is 1)        [Expert]
G.Ins        =  integer    (default is 1)        [Expert]
COndense     =  character  (default is "all")
REOrder      =  character  (default is false)
3d.refine    =  logical    (default is false)
DZ.level     =  integer    (default is 1)
OFf          =  logical    (default is false)
IT.Resolve   =  integer    (default is 2)
N.Resolve    =  integer    (default is 1.0e-3)
```

MAX.LEVEL is the maximum level any element can reach for a given REGRID line relative to the *initial, unrefined* mesh; The default is to allow MAX.LEVEL to increase to one more than the maximum level in the preceding grid, but can be set to a smaller value to limit refinement. Separately, the relative change in each element's level vis a vis the preceding mesh can be specified using REL.LEVEL. Alternatively, LEV.IGNORE ignores the level limits and proceeds until the criteria are met; LEV.IGNORE can thus be viewed as setting MAX.LEVEL and REL.LEVEL to infinity. Regardless of the discriminatory variable's value, a refinement will not take place if an edge is created that is smaller than HMIN or the product of DEBYE and the local Debye length; a coarsening will not take place beyond level 0. If HDIR=true, only element edges which meet the refinement criteria are checked for length against HMIN and/or DEBYE. FREEZE is a parameter which will prohibit unrefining from the current level. SMOOTH.K and IT.SMOOTH are mesh smoothing options, and CONDENSE is the condense region key, both as described for the MESH line. REORDER reorders the grid nodes and elements after adaption.

COS.ANGLE defines the "obtuse criterion" to limit the creation of obtuse angles in the mesh. If regrid would create a element with an angle whose cos is less than -COS.ANGLE, nodes are added so that this does not occur. The test can be turned off locally by using the ignore parameter; it can be turned off everywhere by using a value of COS.ANG greater than 1. The default is to turn it off everywhere. Similarly, the BOUND.GREEN parameter prevents obtuse elements from being created at a boundary or material interface. G.SEMI and G.INS define the maximum levels of green refinement.

In addition to refinement of triangle faces in each xy-grid plane, 3D.REFINE permits refinement of prisms (adding grid planes) in the z direction. DZ.LEVEL is the maximum number of levels of z-refinement that can be performed relative to the initial state.

IT.RESOLVE, N.RESOLVE and OFF are parameters specific to the ADAPT line. IT.RESOLVE is the maximum number of resolve (readapt) attempts per bias point. N.RESOLVE is the minimum fraction of elements that must be refined in order for a subsequent resolve to be attempted. OFF turns off the effect of a previous ADAPT line; no more adaptive refinement will be attempted after the occurence of a line with OFF specified as true.

interpolation

```
  FEm         =  logical  (default is true)
  DOp.int     =  logical  (default is false)  [Expert]
  1d.integ    =  logical  (default is false)  [Expert]
  DI.Degree   =  real     (default is 3)      [Expert]
  DI.Level    =  integer  (default is 0)      [Expert]
```

The FEM parameter specifies that basis functions equivalents be used to interpolate concentration at new nodes, rather than simple exponents. The parameters DOP.INT, 1D.INTEG, DI.DEGREE and DI.LEVEL refer to doping integration control; see the MESH input line documentation.

Examples

Starting with an initial grid, we refine twice, requesting that all elements with large doping steps be refined:

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid1 DOPF=dopxx1
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1
```

A similar effect could be obtained with just one REGRID line below. In both cases two levels of refinement are done. The first choice is preferable because new doping information is introduced at each level.

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1 MAX.LEVEL=2
```

A better strategy is to use the LEVEL and HMIN parameters, repeating the refinement over and over until on each element either (1) the change in doping is less than 6 orders of magnitude or (2) another refinement will produce a grid spacing less than 0.01mm.

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1
  +       LEV.IGN HMIN=.01
```

A dynamic grid is set up by supplying an ADAPT line. At every bias point solved after this line, the discretization error on the grid will be tested. The refine/unrefine criteria are 9.5mV (from R.THRESHOLD) and 1mV, and the process will be initiated if (1) the potential error exceeds 10mV or (2) the number of unrefinable elements exceeds 500. The process is repeated until these criteria are met, and the final grid is saved as "rmesha" where the terminating "a" is incremented for subsequent meshes.

```
  ADAPT  ERROR R.THRESH=0.01 C.STEP=.001 N.THRESH=500 OUTF=rmesha
```

Regrid on generated carrier density from a particle track. The refinement is performed in all three dimensions if the density changes by more than 1.0e17/cm**3 over any edge in an element.

```
  REGRID   3D P.TR R.STEP=1e17 OUTF=rgrid DOPF=dopfile LEV.IGN
```

The ALLOY line

Syntax

```
  ALloy  definition
```

Description

The ALLOY line defines a means of interpolating defaults of alloy semiconductors from predefined (library or by the user) material types. The alloy thus constructed can be used by subsequent MATERIAL lines to create new material systems.

Parameters

definition:

```
  NAme     =  character
  M1alloy  =  character
  X1alloy  =  real
  M2alloy  =  character
  X2alloy  =  real
  M3alloy  =  character
  X3alloy  =  real
  M4alloy  =  character
  X4alloy  =  real
```

NAME is the name of the alloy to be constructed. Up to four predefined materials can be used to make the alloy. M*ALLOY are the names of the materials, and X*ALLOY are the associated compositional fractions.

Examples

Define an alloy called "sige" from known materials Si and Ge; the compositional fraction *x* is the amount of Ge in the alloy (defined on the MATERIAL line).

```
ALLOY  NAME=sige X1ALLOY=0 M1ALLOY=silicon
+                X2ALLOY=1 M2ALLOY=germanium
```

The ANALYSIS line

Syntax

```
  ANalysis mos general files
```

Description

The ANALYSIS line allows a limited data analysis to be performed on either IV or AC log files.

Parameters

mos

```
  Gate          =  integer
  Source        =  integer
  Drain         =  integer
  BUlk          =  integer
  Subthreshold  =  logical  (default is false)
                   or
  Mos.ft        =  logical  (default is false)
```

The above parameters permit either a subthreshold or Ft analysis to be performed for a MOSFET. The terminals of the MOSFET must be identified using the GATE, SOURCE, DRAIN and BULK parameters. The SUBTHRESHOLD analysis yields (1) extrapolated Vt (2) subthreshold slope and (3) maximum transconductance for an IV file with a fixed Vds; reults are given assuming both linear and saturated bias conditions. The MOSFT analysis computes (1) extrapolated Ft, (2) low frequency current gain and (3) low frequency voltage gain for an MOSFET AC file with a fixed DC bias.

general

```
  VAR1        =  character
  A1          =  real      (default is 1 if VAR1 is specified, else 0)
  B1          =  real      (default is 0)
  C1          =  real      (default is 0)
  N1          =  real      (default is 1)
  LOG1        =  logical   (default is false)
  ABS1        =  logical   (default is false)

  VAR2        =  character
  A2          =  real      (default is 1 if VAR2 is specified, else 0)
  B2          =  real      (default is 0)
  C2          =  real      (default is 0)
  N2          =  real      (default is 1)
  LOG2        =  logical   (default is false)
  ABS2        =  logical   (default is false)
  VAR3         =  character
  A3           =  real      (default is 1 if VAR3 is specified, else 0)
  B3           =  real      (default is 0)
  C3           =  real      (default is 0)
  N3           =  real      (default is 1)
  LOG3         =  logical   (default is false)
  ABS3         =  logical   (default is false)

  FRequency    =  real
  RESULT       =  character
  LOGArithm    =  logical   (default is false)
  ABSOlute     =  logical   (default is false)
  REGression   =  logical   (default is false)
```

VAR1, VAR2 and VAR3 identify data quantities to be manipulated. In the specified output file (below), the quantity given by RESULT will be replaced by

VAR1, VAR2, VAR3 and RESULT are given in the same format as plotted quantities on the PLOT.1D line; e.g. time, frequency, applied biases (VA1, VA2, etc.), actual contact bias (V1, V2, etc.), terminal current (I1, I2, etc.), AC capacitances (C11, C12, C21, etc.), AC conductance (G11, G12, G21, etc.) and AC admittance (Y11, Y12, Y21, etc.). Use FREQUENCY to obtain AC parameters at a single frequency. ABS1, ABS2 and ABS3 indicate that the absolute values of VAR1, VAR2 and VAR3 respectively be taken, while LOG1, LOG2 and LOG3 indicate that the logarithms of these values be evaluated; the absolute value is taken before the logarithm in each case. ABSOLUTE and LOGARITHM specify that the absolute value and logarithm respectively be taken of RESULT. Finally, REGRESSION specifies that a least squares fit be performed on VAR1 vs. VAR2.

files

```
  Infile       =  character
  Outfile      =  character
```

INFILE is the input IV or AC file. The default is to use the log file that is currently open, as with the PLOT.1D line. OUTFILE is the output IV or AC file, including any data manipulation using VAR1, VAR2 and RESULT which can be plotted just like any other IV/AC file. Note also that INFILE could be the result of a previous ANALYSIS line so that a series of data manipulations could be performed on a single IV/AC file.

Examples

The following computes Vt, S and gm from a MOSFET IV file where the gate, source, drain and bulk electrodes are 1,3,4 and 2 respectively:

```
  ANALYSIS  SUBT INF=MOSIV GATE=1 SOURCE=3 DRAIN=4 BULK=2
```

The following does a least square fit to 1/C**2 vs. V where C is the capacitance between electrodes 2 and 1 and V is the bias on electrode 2.

```
ANALYSIS  INF=ACFILE VAR1=C21 RESULT=Y21 OUTF=ACDUM
+         A1=1 N1=-2
ANALYSIS  INF=ACDUM VAR1=V2 VAR2=Y21 REGRESS
```

The CHECK line

Syntax

```
CHeck parameters
```

Description

The CHECK line compares a specified solution against the current solution, returning the maximum and average difference in electrostatic and quasi-fermi potentials. It is particularly useful for comparing solutions that have been obtained on different generations of regrids.

Parameters

parameters

```
Infile      =  filename
Mesh        =  filename
Samemesh    =  logical  (default is false)
Richardson  =  logical  (default is false)
ASCII.In    =  logical  (default is true)
```

INFILE specifies the name of the solution file to compare. MESH is the name of the file containing the mesh for that solution. SAMEMESH indicates the the solution in INFILE used the same mesh as the current solution. RICHARDSON performs a Richardson extrapolation from the two solutions in order to produce an error estimate for a coarse and fine solution. ASCII.IN specifies that the input files are ascii.

Examples

Compare solution "sol1" obtained using mesh "mesh1" against the current solution.

```
CHECK  INFILE=sol1 MESH=mesh1
```

The COMMENT line

Syntax

```
  COMment character string
or
  $ character string
or
  # character string
```

Description

The COMMENT line allows comments to be placed in the *PADRE* input file. *PADRE* ignores the information on the COMMENT line. Note that the "#" can be put on any *PADRE* input line; all information to the left of the character is processed, and that to the right is ignored.

Examples

```
  $  **** This is a comment - wow! ****
  DOPING    CONC=1e16 UNIFORM     # Appended comment
```

The CONTACT line

Syntax

```
  CONTAct  number workfunction special conditions misc
```

Description

The CONTACT line defines the physical parameters of an electrode. If no contact line is supplied for an electrode, it is assumed to be charge-neutral (ohmic). Lumped elements are also specified here.

Parameters

number

```
  NUmber  =  integer
          or
  ALL     =  logical
```

NUMBER gives the number of a previously defined electrode to which properties are to be assigned. Using ALL instead of NUMBER defines properties for all electrodes.

workfunction is one of:

```
  Material            Work function used
    NEutral             (calculated from doping)
    ALUminum            4.17
    P.polysilicon       4.17 + Egap
    N.polysilicon       4.17
    MOLybdenum          4.53
    TUNgsten            4.63
    MO.disilicide       4.80
    TU.disilicide       4.80
                      or
  Workfunction        =  real
  MIN.workfunction    =  real
```

The work function defines a barrier height to majority carriers and is set to the above values for the standard materials, or to the given value WORKFUNCTION (in volts). NEUTRAL (the default) stands for charge-neutral (ohmic). The default barrier height for minority carriers is also computed from the work function; however by using the MIN.WORKFUNCTION parameter, a different barrier height can be established.

special conditions

```
  Surf.rec      =  logical  (default is false)
  N.surf.rec    =  logical  (default is false)
  P.surf.rec    =  logical  (default is false)
  VSURFN        =  real     (default is computed)
  VSURFP        =  real     (default is computed)
  BArrierl      =  logical  (default is false)
  ALPha         =  real     (default is 0)
  BEta          =  real     (default is 0)
                or
  CUrrent       =  real     (default is false)
  Distributed   =  real     (default is false)
                or
  Resistance    =  real     (default is 0)
  CApacitance   =  real     (default is 0)
  Inductance    =  real     (default is 0)
  Distributed   =  real     (default is false)
                or
  Distributed   =  real     (default is false)
                or
  CON.resist    =  real     (default is 0)
                or
  R.Table       =  character
  R.Scale       =  real     (default is 1)
```

If none of the above are specified, this contact is a normal ohmic (Dirichlet) boundary condition. If SURF.REC is specified, finite surface recombination velocities are used for both holes and electrons at the respective contact. N.SURF.REC and P.SURF.REC allow finite recombination velocities for electrons and holes separately (the other carrier is fixed using the standard Dirichlet condition). VSURFN and VSURFP are the electron and hole surface recombination velocities themselves and have the units cm/sec; their defaults are calculated using the associated Richardson constants (see the MATERIAL line). BARRIERL is a flag to turn-on the barrier lowering mechanism, while ALPHA is the linear dipole barrier lowering coefficient. CURRENT denotes a current boundary condition. RESISTANCE specifies a lumped resistance value (ohms), CAPACITANCE a lumped capacitance value (F) and INDUCTANCE a lumped inductance value (H) to be attached, in parallel, to the contact. The current, resistance or capacitance can be equally distributed (rather than lumped) along the contact by supplying the DISTRIBUTED parameter. CON.RESIST is a distributed contact resistance and has an identical effect as a total resistance (RESISTANCE) with the DISTRIBUTED flag, except that the units are given in ohm-cm**2. R.TABLE corresponds to a table of voltages and resistances so that two-terminal lumped elements with non-linear IV characteristics can be attached. R.SCALE is a scale factor that the resistances in the table are multiplied by to account for differences in device width.

misc

```
1d.base  =  logical  (default is false)
CONC0    =  vector
```

1D.BASE specifies that a contact be an internal base contact, in which case only the majority carrier quasi-fermi potential is fixed (to the applied bias, possibly through a lumped element). CONC0 is a vector which allows the user to specify an equilibrium electron and hole density, overriding the default which is based on doping and space charge neutrality.

Examples

Define all electrodes except number 2 to be neutral, and number 2 is aluminum. Besides a workfunction, electrode number 2 also includes finite surface recombination velocities and barrier lowering. Note that the definition on the second line overrides that of the first.

```
CONTACT ALL NEUTRAL
CONTACT NUM=2 ALUM SURF BARR
```

Attach a lumped resistor to contact number 2 with a value of 1.0e5 ohms. Include distibuted contact resistance (1.0e-6 ohm-cm**2 ~ Aluminum) on contact 4.

```
CONTACT NUM=2 RESIS=1E5
CONTACT NUM=4 CON.RES=1E-6
```

The CONTINUATION line

Syntax

```
CONTInuation estimate termination bias knee files
```

Description

The CONTINUATION line instructs *PADRE* to trace an IV curve by varying the voltage and current on a given electrode using a continuation method.

Parameters

estimate

```
INitial      =  logical  (default is false)
PREvious     =  logical  (default is true)
PROject      =  logical  (default is false)
EXtrapolate  =  logical  (default is false)
LOcal        =  logical  (default is false)
```

The above parameters are used to specify how the initial guess for the solution is to be obtained. The first bias point for a given structure must have the INITIAL parameter specified. From then *PADRE* will use the previous solution (PREVIOUS) by default. If there are two previous solutions present and equivalent bias steps are taken on any electrodes that are changed, an extrapolation (PROJECT) from the preceding two solutions can be used to sometimes get an improved initial guess, which is particularly good for IV curves that are being traced to a fine resolution. A special Euler projection is also available using the EXTRAPOLATE option which also works quite well for fine IV curve tracing. LOCAL is a special, off-state guess which guesses local values for quasi-Fermi levels based on bias which is typically optimal for for large steps in reverse bias.

termination

```
VMIn       =  real     (default is -max[5V,max(|Vi|])
VMAx       =  real     (default is max[5V,max(|Vi|])
IMIn       =  real     (default is -max[1mA/mm,max(|Ii|])
IMAx       =  real     (default is max[1mA/mm,max(|Ii|])
MAxpnts    =  integer  (default is 30)
VCross     =  real     (default is 0)
ICross     =  real     (default is 0)
N.Vcross   =  real     (default is 1000)
N.Icross   =  real     (default is 1000)
I1.vcross  =  real     (default is -infinity)
I2.vcross  =  real     (default is +infinity)
V1.icross  =  real     (default is -infinity)
V2.icross  =  real     (default is +infinity)
```

A continuation simulation can end on any one of several circumstances. VMIN/VMAX and IMIN/IMAX are the smallest/largest voltage and current that will be allowed at the contact; when exceeded the continuation procedure will terminate. VMIN/VMAX and IMIN/IMAX are also used to scale the IV coordinates so that they always run between 0 and 1. The maximum number of bias points per continuation sequence can be set using MAXPNTS. Termination can also occur if the curve crosses the lines defined by V=VCROSS or I=ICROSS a certain number of times (N.VCROSS for VCROSS, N.ICROSS for ICROSS). The crossings can be further restricted to occur within certain current/voltage limits; for instance a voltage crossing is counted towards the N.VCROSS limit only if the terminal current lies between I1.VCROSS and I2.VCROSS (V1.ICROSS and V2.ICROSS are defined similarly with respect to ICROSS).

bias

```
ELectrode   =  integer
Restart     =  logical  (default is false)
SIGMa0      =  real
V.Up        =  logical  (default is true)
V.Down      =  logical  (default is false)
I.Log       =  logical  (default is false)
V.Log       =  logical  (default is false)
TOL.Sig     =  real     (default is .05)
ERr.term    =  integer  (default is ELECTRODE)
SOl.err     =  logical  (default is false)  [Expert]
L2norm      =  logical  (default is false)  [Expert]
Power       =  real     (default is 2)
SIG.MIN     =  real     (default is 1.0e-8)
SIG.MAX     =  real
SIG.LIMA    =  real     (default is .5)
SIG.LIMB    =  real     (default is 2.0)
SIG.CUT     =  real
NMaxcut     =  integer  (default is 10)
CIrcle      =  logical  (default is false)
N.bias      =  real
P.bias      =  real
```

ELECTRODE specifies the contact which will have IV traced. RESTART specifies that the continuation sequence is a restart from a preceding CONTINUATION line, in which case the step size control is carried on; for new sequences, SIGMA0 specifies the initial step size (pseudo-arc length) for the method (in scaled IV coordinates). V.UP (positive voltage step) and V.DOWN (negative voltage step) specify the direction. I.LOG and V.LOG specify that continuation should be performed on the log of the current and voltage respectively. TOL.SIG specifies the accuracy to which the IV curve is traced; it is used to automatically increase or reduce the arc lengths. By default, the IV characteristic at ELECTRODE is monitored, but a different electrode can be used for error control by setting ERR.TERM. Normally the accuracy is measured in terms of the difference in the linear projection and actual bias point; alternatively, the error can be measured in the solution itself using the infinity (L2NORM=false) or L2 norm (L2NORM=true) by specifying SOL.ERR. POWER specifies the order of the extrapolation error (linear=2). SIG.MIN and SIG.MAX are the minimum and maximum allowed arc lengths while

SIG.LIMA and SIG.LIMB are the minimum and maximum amounts by which the arc length can change; if the change is less than SIG.LIMB, the previous step will be redone. A positive SIG.CUT will set SIG.MAX to SIG.CUT*sigma(fail) if non-convergence is observed for an arc length sigma(fail). NMAXCUT defines the maximum number of cutbacks of an originally desired step in acr length. CIRCLE forces the auxiliary continuation equation to be a circle rather than a perpendicular so that the arc length is limited to a given radius. Finally, N.BIAS and P.BIAS specify fixed quasi-Fermi potentials for carriers (electrons and holes, respectively) that are not being solved for (see the SOLVE line).

knee

```
  KNee         =  logical  (default is false)
  ITol.knee    =  real     (default is .01)
  DVtol.Knee   =  real     (default is .05)
  Bisect       =  logical  (default is false)
```

KNEE specifies that the continuation method is to attempt to capture the first knee (limit point) it finds on the IV curve. ITOL.KNEE and DVTOL.KNEE specify the accuracy to which the knee is desired; ITOL.KNEE is defined as the relative change in consecutive guesses to the knee current, and DVTOL.KNEE is the tolerance in terms of the voltage tangent. BISECT uses a bisection method to find the knee once an IV point has bee acquired on each side; the default is that a much more efficient cubic interpolation procedure is used.

files

```
  Outfile    =  filename
  SAve       =  integer   (default is 1)
  KFile      =  filename
  EFile      =  filename                      [Expert]
  AFile      =  filename                      [Expert]
  Ufile      =  filename                      [Expert]
  Currents   =  logical   (default is true)
  NO.append  =  logical   (default is false)
  AScii      =  logical   (default is true)
```

The OUTFILE parameter optionally specifies the name of the output file for the solution of this bias point. The file names may contain up to 20 characters. If an electrode is stepped so that more than one solution is generated by this line and NO.APPEND is not set, the last non-blank character of the supplied file name will have its ascii code incremented by one for each bias point in succession, resulting in a unique file per saved bias point. By default the solutions are saved at every bias point. Alternatively a frequency of saves can be specified by SAVE, i.e. if SAVE=2, solutions will be saved only for every other bias point. KFILE is the name of the file where the solution at the knee will be stored. If CURRENTS is specified, the electron, hole, and displacement currents, and the electric field, will be computed and stored with the solution. IF ASCII is specified, OUTFILE will be ascii as every bias point. EFILE, AFILE and UFILE are output files containing the nonlinear error/residual, Jacobian matrix and update at every Newton iteration, used for the purposes of debugging.

Examples

The following traces an IV curve using terminal 4. Extrapolation is used to obtain initial guesses, and the simulation will terminate when V4 5 volts or I4 1E-4 Amps/mum. The initial step will be <= .05*5 = .25v on V4 and <= 5E-6A in I4.

```
  CONTIN  ELECT=4 SIGMA=.05 VMAX=5 IMAX=1E-4 TOL.SIG=5E-3 EXTRA
```

The CONTOUR line

Syntax

```
  CONTOur plotted quantity range definition control
```

Description

The CONTOUR line plots contours on a plotted two-dimensional area of the device, as specified by the most recent PLOT.2D line.

Parameters

plotted quantity is one of:

```
  POtential    =  logical  Mid-gap potential
  QFN          =  logical  Electron quasi-fermi level
  QFP          =  logical  Hole quasi-fermi level
  N.temp       =  logical  Electron temperature
  P.temp       =  logical  Hole temperature
  BAND.Val     =  logical  Valence band potential
  BAND.Cond    =  logical  Conduction band potential
  DOping       =  logical  Total net impurity concentration
  IOn.imp      =  logical  Net ionized impurity concentration
  ELectrons    =  logical  Electron concentration
  Holes        =  logical  Hole concentration
  NET.CHarge   =  logical  Net charge concentration
  NET.CArrier  =  logical  Net carrier concentration
  J.Conduc     =  logical  Conduction current
  J.Electr     =  logical  Electron current
  V.Electr     =  logical  Electron velocity
  J.Hole       =  logical  Hole current
  V.Hole       =  logical  Hole velocity
  J.Displa     =  logical  Displacement current
  J.Total      =  logical  Total current
  E.field      =  logical  Electric field
  Recomb       =  logical  Net recombination
  Flowlines    =  logical  Current flow lines
```

The above parameters specify the quantity to be plotted. For vector quantities the magnitude is plotted. Model dependent parameters (current and recombination) are calculated with the models currently defined, *not* with the models that were defined when the solution was computed. This allows the display of, for instance, Auger and Shockley-Read-Hall components of recombination separately. For consistent values of current, the models used in the solution should be specified. The quantity to be plotted has no default.

range definition

```
  MIN.value  =  real
  MAx.value  =  real
  DEl.value  =  real
  NContours  =  integer
  CONstrain  =  logical  (default is true)
```

MIN.VALUE and MAX.VALUE specify the minimum and maximum contours to be plotted. MIN.VALUE and MAX.VALUE default to the minimum and maximum values of the quantity to be plotted over the device (these are printed during execution). DEL.VALUE specifies the interval between contours. Alternatively, NCONTOURS specifies the number of contours to be plotted. If the plot is logarithmic, the minimum and maximum should be given as the logarithmic bounds. CONSTRAIN constrains the contours to lie between min/max values; if CONSTRAIN is turned off, the min and max contours extend out to infinity.

control

```
  LIne.type  =  integer  (default is 1)
  ABsolute   =  logical  (default is false)
  LOgarithm  =  logical  (default is false)
  X.compon   =  logical  (default is false)
  Y.compon   =  logical  (default is false)
  MIX.mater  =  logical  (default is false)
  PAuse      =  logical  (default is false)
  COLor      =  logical  (default is false)
  Grey       =  logical  (default is false)
```

LINE.TYPE defines the plot line type. ABSOLUTE specifies that the absolute value of the variable be taken. For rapidly varying quantities, the LOGARITHM is often more revealing. Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM - the absolute is taken first and there is no danger of negative arguments. X.COMPON and Y.COMPON take the x and y components of a vector quantity, respectively. MIX.MATER specifies that local vector averaging should be done over all materials to which the node belongs as opposed to just the hierarchical choice (see the PRINT line). The PAUSE option causes *PADRE* to stop at the end of the plot during an interactive run so that a hardcopy may be made before continuing. Execution can be resumed by hitting a carriage return. COLOR specifies that color fill (GREY is a grey scale fill), as opposed to simple lines, should be used to delineate contours; the contour colors are chosen by default, but the user can override this scheme using appropriate parameters on the OPTION line.

Examples

The following plots the contours of potential from -1 volts to 3 volts in steps of .25 volts:

```
  CONTOUR  POTEN MIN=-1 MAX=3 DEL=.25
```

In the next example, the log of the doping concentration is plotted from 1.0e10/cm**3 to 1.0e20/cm**3 in multiples of 10. By specifying ABSOLUTE, both the n-type and p-type contours are shown.

```
  CONTOUR  DOPING MIN=10 MAX=20 DEL=1 LOG ABS
```

In the following, current flow lines are plotted. The number of flow lines is 11 so that 10% of the current flows between adjacent lines.

```
  CONTOUR  FLOW NCONT=11
```

The DOPING line

Syntax

```
  DOping profile type location region misc profile specification
```

Description

The DOPING line places impurities in selected regions of the device. The impurity profiles can be specified by analytic functions, the output of one-dimensional process simulators, one-dimensional tables of depth vs. concentration or from preprocessed (via the *BIPAD* interface program) output from two-dimensional process simulators.

Parameters

profile type is one of:

```
  Gaussian     =  logical  (default is false)
  LInear       =  logical  (default is false)
  Uniform      =  logical  (default is false)
  SUprem3      =  logical  (default is false)
  NEW.suprem3  =  logical  (default is false)
  Table.1d     =  logical  (default is false)
```

```
2d.profile    =  logical  (default is false)
BISon         =  logical  (default is false)
RELoad        =  logical  (default is false)
```

GAUSSIAN, LINEAR and UNIFORM are used to analytically describe profile shapes. SUPREM3 is used to get profile information from the original version of the *SUPREM-III* process modeling program. NEW.SUPREM3 reads data files produced by the more recent release of *SUPREM-III*, which supports the "export" output file format. TABLE.1D allows for input of tables of concentration versus depth where each line contains a single depth (in um) followed by its associated concentration (in /cm**3). Positive (negative) numbers are used for N (P) doping concentrations, respectively. The device structure and the doping profile must begin at the same co-ordinate, and must have the same orientation. 2D.PROFILE specifies that a general two-dimensional doping profile is to be loaded using the standard *BIPAD* (or *PADRE*) file format. BISON also indicates a two-dimensional doping profile is to be loaded but using the special "bison" file format created by *BIPAD* from *BICEPS* (version 5.X and predecessors) only. RELOAD specifies the input file will be reloaded from a previously created *PADRE* doping output file.

location-1D

PARAMETER	Uniform	Gaussian/Suprem/Table	
		(x-dir)	(y-dir)
X.Left = *real*	-infinity	*SP*	-infinity
X.Right = *real*	+infinity	X.LEFT	+infinity
Y.Top = *real*	-infinity	-infinity	Y.BOT
Y.Bottom = *real*	+infinity	+infinity	*SP*
Z.Front = *real*	-infinity	-infinity	-infinity
Z.Back = *real*	+infinity	+infinity	+infinity

The box given by the X, Y and Z bounds locates the profile within a device, and defines a volume where the profile is constant (mask edges). Outside this volume, it falls off along the principal axis according to the profile type, and along the lateral axis according to the lateral specifications. The default bounds of the box are chosen depending on the type and principal direction of the profile. In the Gaussian/Suprem case, the bounds default to a plane normal to the principal axis and located at the peak/start of the profile, respectively. This is denoted by the entry SP in the above table.

location-2D

```
X.Origin  =  real  (default is 0.0)
Y.Origin  =  real  (default is 0.0)
Z.Front   =  real  (default is min(z) in device)
Z.Back    =  real  (default is max(z) in device)
```

X.ORIGIN and Y.ORIGIN indicate the (x,y) location within the device where the (0,0) point in the 2D doping profile will be located. The doping concentration at points outside the domain of the 2D profile will be determined by the value at the closest x/y edge.

Z.FRONT and Z.BACK define the minimum and maximum coordinates in the z-dimension in which the 2D profile is to be placed.

region

```
  REGion  =  vector
```

REGION is an optional parameter specifying the region number(s) where doping is to be added. If no region is specified, the entire semiconductor portion of the device is used. In a non-heterostructure mesh (e.g., see the HETERO option on the MESH line), each node may only have one doping value even if it lies on a material interface; in these cases, the determination is made by the hierarchy of nodal material types (semiconductor always outweighs insulator). In heterostructure meshes, a boundary node can have more than one doping concentration, but if the mesh is restored from a file, the RELOAD option is necessary to retain the multiple definition.

misc

```
  INItialize  =  logical   (default is false)
  COmpensate  =  logical   (default is false)
  OUTFile     =  filename
  ASCII.Out   =  logical   (default is true)
  NEgate      =  logical   (default is false)
  ABove       =  real
  BElow       =  real
  ALl         =  logical   (default is false)
  Maxdop      =  real
```

INITIALIZE reinitializes (zeros) the doping in the device. COMPENSATE specifies that the doping should be added to the net concentration as usual, but *subtracted* from the total concentration. This option is useful in removing extra doping in a region without raising the total impurity level which would decrease carrier mobility. The OUTFILE option allows the user to save a copy of all the DOPING lines in a file; the file type is specified by the ASCII.OUT parameter. The first DOPING line should have the OUTFILE parameter, so that the doping information on it and all subsequent DOPING lines are saved in that file. The file can be reread after regrid to calculate the doping on the new grid. The NEGATE parameter negates a specified type of doping (N.TYPE, P.TYPE or ALL). The parameters ABOVE and BELOW are options for NEGATE which specify the doping concentration levels above and below which the negate operation will be applied. If specified, MAXDOP defines a maximum absolute limit for the doping concentrations added by this DOPING line; this parameter may be useful for simulating different activation conditions.

profile specification

The manner in which a profile is specified is dependent on the *profile type*:

- If *profile type*=UNIFORM, the following make up the *profile specification*:

specification

```
COncentration    =  real
N.type/DONor     =  logical  (default is false)
P.type/ACceptor  =  logical  (default is false)
```

CONCENTRATION is the value of the uniform doping level (in atoms/cm**3) and should be positive. The polarity is given by the logical parameters N.TYPE (or DONOR) and P.TYPE (or ACCEPTOR). Doping is introduced in the intersection of the box (see *location-1d* above) and the selected region(s) (see *region* above). The default box for a uniform profile is set up to include the entire device area in the specified region(s).

- If *profile type*=GAUSSIAN or LINEAR, the following make up the *profile specification*:

  profile

  ```
  COncentration   =  real
  DOSe            =  real
  CHaracteristic  =  real
  Junction        =  real
  SLice.lat       =  real
  Z.Slice         =  real
  ```

  These parameters define the doping outside of the constant box defined by the *location-1d* parameters above. CONCENTRATION is the peak concentration, DOSE the total dose, JUNCTION is the location of the junction, and CHARACTERISTIC is either the principal characteristic length for a gaussian profile or the point where the linear profile itself goes to zero. The profile can be specified in three different ways: (1) setting CONCENTRATION and JUNCTION; (2) setting DOSE and CHARACTERISTIC; or (3) setting CONCENTRATION and CHARACTERISTIC. When JUNCTION is used, *PADRE* by default computes a characteristic length by examining the doping at a point half way between the ends of the constant box and at the given depth; a different lateral position can be used for this computation using SLICE.LATERAL (lateral x or y coordinate) and Z.SLICE (z coordinate).

dopant type is one of:

```
N.type/DONor     =  logical  (default is false)
P.type/ACceptor  =  logical  (default is false)
```

These parameters define the polarity of the profile.

2D/3D spread

```
PEak           =  real       (default is 0)
DIRection      =  character  (default is "y")
```

```
Erfc.Lateral    =  logical
RAtio.Lateral   =  real         (default is 0.8)
Lat.char        =  real
Z.Char          =  real         (default is LAT.CHAR)
```

These parameters specify the location of the 1D profile and how it extends in other dimensions. PEAK is the position of the profile peak; the default is placed at the bounds of the constant box (i.e., Y.BOTTOM). DIRECTION is the axis ("x" or "y") along which the profile is directed. By default, the profile in the other ("lateral") dimensions take the same form - e.g., gaussian or linear - as that along the primary axis; ERFC.LATERAL can be used to obtain an error function instead. RATIO.LATERAL gives the ratio of the characteristic length in the lateral dimensions to that defined above. Alternatively, a specific value for this length can be specified using LATERAL.CHAR. Z.CHAR defines a principle characteristic length to be applied in the z-dimension.

If *profile type*=SUPREM3, NEW.SUPREM3 or ASCII.IN, the following make up the *profile specification*:

input file

```
Infile    =  filename
ASCII.IN  =  logical   (default is true)
```

INFILE gives the name of the input file from which the one-dimensional impurity profile is to be taken, and ASCII.IN specifies whether or not this file is in ascii format.

dopant type is one of:

```
BOron             =  logical  (default is false)
PHosphorus        =  logical  (default is false)
ARsenic           =  logical  (default is false)
ANtimony          =  logical  (default is false)
N.type/DONor      =  logical  (default is false)
P.type/ACceptor   =  logical  (default is false)
```

These parameters indicate the use of a particular dopant species/type. For SUPREM-III files, only the those impurities listed are extracted from the input file, where N.TYPE (or DONOR) and P.TYPE (or ACCEPTOR) refer to all n- and p- type species respectively. For 1d doping tables, the parameters simply define the concentration given in the file (e.g., N-TYPE/DONOR adds the values to the net concentrations and P-TYPE/ACCEPTOR subtracts).

2D/3D spread

```
DIRection       =  character  (default is "y")
STArt           =  real       (default is 0)
RAtio.lateral   =  real       (default is 0.8)
Z.Char          =  real
```

These parameters specify where to locate the one-dimensional profile in the device and how to extend it to the second dimension. DIRECTION is the axis ("x" or "y") along which the profile is directed. START is the depth in the specified direction where the profile should start, and should normally be at the surface. The lateral profile is assumed to have the same form as the principal profile, but shrunk/expanded by the factor RATIO.LATERAL. The defaults for the location box are set up as a line, parallel to the surface, and located at START. Z.CHAR defines a principle characteristic length to be applied in the z-dimension.

If *profile type*=2D.PROFILE or BISON, the following make up the *profile specification*:

3d spread

```
Z.Char     =  real    (default is machine epsilon)
Z.2char    =  real    (default is Z.CHAR)
Z.Yoff     =  real
Z.Gamma    =  real    (default is 1.0)
STRech2d   =  vector
```

Z.CHAR defines a principle characteristic length to be applied in the z-dimension. Z.2CHAR is an optional second characteristic length to be used. Z.YOFF defines the y-depth where the breakpoint between characteristic lengths occurs. A tanh smoothing is used between the characteristic length sections, controlled by Z.GAMMA as follows:

such that for larger Z.GAMMA, the grading between Z.CHAR (sigma1) and Z.2CHAR (sigma2) is smoother. STRECH2D = x0, delta-x provides a optional mechanism to expand the profile in the lateral (x) direction, starting at the point x = x0 by an amount delta-x (both in mm).

input file

```
INFile      =  filename
ASCII.IN    =  logical    (default is true)
DIFference  =  logical    (default is false)
```

INFILE gives the name of the input file from which the one-dimensional impurity profile is to be taken, and ASCII.IN specifies whether or not this file is in ascii format. DIFFERENCE specifies that the 2D profile to be included has been obtained by differencing two other profiles; this parameter is used with process simulator file formats which are written in terms of total and net concentration (instead of acceptor and donor) since these cases permit NA, ND < 0.

Examples

A one-dimensional diode with substrate doping 1.0e16/cm**3 and Gaussian profile.

```
DOP   UNIF CONC=1E16 P.TYPE
DOP   GAUSS CONC=1E20 JUNC=0.85 N.TYPE PEAK=0
```

An n-channel MOSFET with Gaussian source and drain. Because the default X.RIGHT is +infinity, for the source we must limit the constant part to X.RIGHT=4, and conversely for the drain. Thus the profile has a constant part along the surface, falls off as an error function towards the gate, and as a gaussian in the direction of the bulk. In both cases, the vertical junction is at 1.3 mm.

```
DOP  UNIF CONC=1E16 P.TYPE
DOP  GAUSS CONC=9E19 N.TYPE
+    X.RIGHT=4 JUNC=1.3 R.LAT=0.6 ERFC.LAT
DOP  GAUSS CONC=9E19 N.TYPE
+    X.LEFT=12 JUNC=1.3 R.LAT=0.6 ERFC.LAT
```

Read a *SUPREM-III* bipolar profile and add it to a uniform substrate concentration. Add doping only to those points lying in regions 1 and 2.

```
COM  *** SUBSTRATE ***
DOP  REGION=1,2 UNIF CONC=1E16 N.TYPE
COM  *** BASE ***
DOP  REGION=1,2 SUPREM BORON R.LAT=0.7 INF=sup3.out1
+    START=0
COM  *** EMITTER ***
DOP  REGION=1,2 SUPREM PHOS  R.LAT=0.8 INF=sup3.out1
+    X.LEFT=12.0 X.RIGHT=13.0 START=0
```

Read a general 2d doping profile and out its origin at (1,0) in the device grid; this profile (A) is loaded as is throughout the entire z-dimension. The second DOPING line loads a second profile (B-A) which is the difference between profile B and A; as a result, for z >= 5um, the device will have profile B. The difference between A and B falls off for z < 5um according to a y-dependent characteristic length; for y << 1um, sigma = 0.25um, whereas for y >> 1um, sigma = 0.5um. Near y = 1um, sigma is determined by the tanh function with a blending coefficient of 0.7.

```
DOP  X.ORIG=1 Y.ORIG=0 2D.PROF INF=sup4.outA
DOP  X.ORIG=1 Y.ORIG=0 2D.PROF DIFF INF=sup4.outB-A
+    Z.FRONT=5 Z.CH=0.25 Z.2CH=0.5 Z.YOFF=1 Z.GAM=0.7
```

Negate the all the n-type doping in the box 0 <= x <= 2.5um and 0 <= y <= 5um.

```
DOP  NEGATE N.TYPE X.L=0 X.R=2.5 Y.TOP=0 Y.BOT=5
```

The ELECTRODE line

Syntax

```
ELEctrode number position adjustment
```

Description

The ELECTRODE line specifies the location of electrodes or allows for adjustment of previously defined electrodes.

Parameters

number

```
Number  =  integer
```

NUMBER is the id number of the electrode to be added or modified. There may be up to ten electrodes, numbered 1,2,3,...,9,0. They may be assigned in any order, but if there are N electrodes, none can have an electrode number above N. There may be multiple definitions for a single electrode number.

location

```
IX.Low   =  integer
IX.High  =  integer
IY.Low   =  integer
IY.High  =  integer
Z.MIn    =  real    (default is min(z) in device)
Z.MAx    =  real    (default is max(z) in device)
Region   =  integer
         or
X.MIn    =  real
X.MAx    =  real
Y.MIn    =  real
Y.MAx    =  real
Z.MIn    =  real    (default is min(z) in device)
Z.MAx    =  real    (default is max(z) in device)
Region   =  integer
```

Electrode extents may be defined in the xy-plane by rectangular indicies or by spatial coordinates. Nodes having x and y indices between IX.LOW and IX.HIGH and between IY.LOW and IY.HIGH respectively or x and y coordinates between X.MIN and X.MAX and between Y.MIN and Y.MAX respectively are designated electrode nodes. In the z-dimension, electrode extents can only be defined by coordinates, Z.MIN and Z.MAX. REGION specifies that this electrode only be applied within the specified region number; the default is every region in the device.

adjustments

```
Clear or INItial  =  logical  (default is false)
INClude           =  integer
Dump or Print     =  integer  (default is specified by NUMBER)
```

CLEAR (or INITIAL) reinitializes (eliminates) currently defined electrodes; if no electrode number is given, the entire electrode data base is initialized. INCLUDE is the number of a second electrode which is to be incorporated into that specified by NUMBER. DUMP (or PRINT) indicates how terminal values associated with the electrode modified by the INCLUDE are to be printed. If DUMP is the same as INCLUDE, the terminal characteristics for those electrode nodes are printed separately even though they now belong to that specified by NUMBER. This option serves to short to electrodes together and is very useful when used with current or continuation boundary conditions.

Examples

Define a typical back-side contact.

```
  ELEC   N=1 IX.LOW=1 IX.HIGH=40 IY.LOW=17 IY.HIGH=17
```

Incorporate all electrode 2 nodes into electrode 3, but print all information separately.

```
  ELEC   NUM=3 INCLUDE=2 DUMP=2
```

The ELIMINATE line

Syntax

```
  ELIminate direction range
```

Description

The ELIMINATE line terminates mesh points along lines in a rectangular grid.

Parameters

direction

```
  X.direction  =  logical
  Y.direction  =  logical
```

These parameters determine whether to eliminate points along vertical or horizontal lines. One must be chosen.

range

```
  IX.Low   =  integer
  IX.High  =  integer
  IY.Low   =  integer
  IY.High  =  integer
```

Points along every second line within the chosen range is removed. Successive eliminations of the same range remove points along every fourth, eighth line, and so on. For horizontal elimination, the vertical bounds should be decreased by one at each re-elimination of the same region, and conversely for vertical eliminations.

Examples

Points along vertical lines between 10 and 20 are removed.

```
  ELIM    Y.DIR IY.LO=10 IY.HI=20 IX.LO=1 IX.HI=8
  ELIM    Y.DIR IY.LO=10 IY.HI=20 IX.LO=1 IX.HI=7
```

The END line

Syntax

```
  ENd
```

Description

The END line specifies the end of a set of *PADRE* input lines. The END line may be placed anywhere in the input deck; all input lines below the occurrence of the END line will be ignored. If an END line is not included, all lines in the input file are processed.

The EXTRACT line

Syntax

```
  EXtract variable bounds file-i/o
```

Description

The EXTRACT line extracts selected electrical data from the solution.

Parameters

variable

```
NET.CHar       =  logical  Integrated net charge
NET.CArr       =  logical  Integrated carrier concentration
RECombination  =  logical  Integrated total recombination
Laser          =  logical  Integrated stimulated emission
Electron       =  logical  Integrated electron concentration
Hole           =  logical  Integrated hole concentration
Metal.ch       =  logical  Integrated charge on a contact
N.Resist       =  logical  n-Resistance of a cross section
P.Resist       =  logical  p-Resistance of a cross section
N.Current      =  logical  n-current through an electrode
P.Current      =  logical  p-current through an electrode
Average        =  logical  (default is false)
```

The net carrier, charge, electron or hole concentrations can be integrated over a section of a device. The charge on a part of an electrode can be calculated, as can the current through that part. This is useful for capacitance studies, in conjunction with the difference mode of the LOAD line. The resistance of a cross sectional structure, for instance a diffused line, can be calculated. For NET.CHAR, NET.CARR, RECOMBINATION and LASER, AVERAGE returns an average value, obtained by dividing the total by the integrated volume.

bounds

```
X.MIn    =  real
X.MAx    =  real
Y.MIn    =  real
Y.MAx    =  real
Z.MIn    =  real
Z.MAx    =  real
Contact  =  integer
REGions  =  integer
N.Type   =  logical  (default is true)
P.Type   =  logical  (default is true)
```

Only nodes falling within X.MIN-X.MAX, Y.MIN-Y.MAX, Z.MIN-Z.MAX are included in the integrations. The default bounds include the entire device. For electrode quantities (current and metal charge) a CONTACT must be chosen; only nodes falling within the bounds and belonging to the contact are included in the integration. REGIONS can be optionally specified, forcing integration only on nodes within the specified bounds that are also part of a particular set of regions. N.TYPE and P.TYPE specify that the extraction be done over n-type and/or p-type material.

file-i/o

```
Outfile  =  filename
```

An optional ascii OUTPUT file can be specified to which the result and bias infor will be written.

Examples

The following extracts the resistance of a p-type line diffused into a lightly doped n substrate. Since the p-conductivity of the substrate is negligible, the bounds of the integration can include the whole device.

```
  EXTRACT  P.RESIST
```

In the next example, the charge on the lower surface of a gate electrode is integrated. There is 0.05 mm of gate oxide on the surface, which is at y=0.

```
  EXTRACT  METAL.CH CONT=1 X.MIN=-2.0 X.MAX=2.0
  +        Y.MAX=-0.0499 Y.MIN=-0.0501
```

The INTERFACE line

Syntax

```
  INTERFace  number parameters
```

Description

The INTERFACE line allows the specification of recombination velocities and fixed charges at previously defined interfaces. Interfaces are accessed by numbers which are either given in an external mesh generator (negative electrode number) or using SURFACE lines.

Parameters

number

```
  Number  =  integer
```

NUMBER selects a given interface by number.

parameters

```
  Qf         =  real       Fixed charge density (/cm**2)
  S.N        =  vector     Electron surface recombination velocity
(cm/sec)
  S.P        =  vector     Hole surface recombination velocity (cm/sec)
  TRap.type  =  character  Trap types
  ETrap      =  vector     Trap energy levels = Et - Ei (eV)
  NTRap      =  vector     Trap densities (/cm**2)
```

Traps are specified as with bulk lifetimes for SRH recombination (see the MATERIAL line), except using surface recombination velocities (S.N, S.P) instead of lifetimes.

Examples

Define fixed charges and recombination velocities on interface number 1.

```
  INTERFACE  NUM=1 QF=1E10 S.N=1E4 S.P=1E4
```

The INTERPOLATE line

Syntax

```
  INTERPolate  files
```

Description

The INTERPOLATE line interpolates the current solution onto a specified mesh and dumps the result to a given output file.

Parameters

files

```
  Mesh       =  filename
  Outfile    =  filename
  ASCII.In   =  logical   (default is true)
  ASCII.Out  =  logical   (default is true)
```

MESH is the name of the file containing the mesh to be interpolated onto. OUTFILE is the output file for the interpolated solution. The formats of the mesh and output files are given by ASCII.IN and ASCII.OUT respectively.

Examples

Interpolate the current solution onto the mesh "xmesh" and save in the file "xout." This file can be then loaded as an initial guess in a simulation run which uses "xmesh" as its mesh.

```
  INTERP  MESH=xmesh  OUTFILE=xout
```

The LINALG line

Syntax

```
  LInalg  method stopping crit misc
```

Description

The LINALG line sets parameters associated with the numerical methods used for the solution of linear algebraic systems. There can be more than one LINALG line in a single simulation, so that parameters can be altered. Subsequent LINALG lines only alter those coefficients specified.

Parameters

method

```
  Dir.def         =  logical
  ITEr.def        =  logical
  MEthod          =  integer vector
  Precondition    =  integer vector
  ACceleration    =  integer vector
  S.Method        =  integer vector
  S.Precondition  =  integer vector
  S.Acceleration  =  integer vector
```

The METHOD, PRECONDITION and ACCELERATION vectors determine the basic method, the preconditioner and the iterative acceleration technique used to perform the linear algebraic solutions. Each of these consists of a set of single digits corresponding exactly to the PDE groupings on the SYSTEM line, defined as follows:

```
  Method        =  1     Direct sparse LU decomposition
                   2     Point iterative
                   3     ABF

  Precondition  =  1     LDU
                   2     ILU
                   3     Block-ILU

  Acceleration  =  1     Conjugate gradients
                   2     Orthomin
                   3     Biconjugate gradients
                   4     CGS
                   5     Generalized conjugate residuals
                   6     GMRES
                   7     CGNE
                   8     LSQR
                   9     USQR
                   10    BICR
                   11    BICG/CGS
```

```
                        12    BICG1
                        13    BICG2
```

Similarly, the S.METHOD, S.PRECONDITIONER and S.ACCELERATION vectors optionally specify the method, preconditioner and acceleration for any plug-in smoothing loops (see GLOOPS on the METHOD line).

With DIR.DEF set, these options are set (reset) to appropriate defaults for direct factorizations (method=1, precon=1, accel=2). With ITER.DEF set, these options are set (reset) to appropriate defaults for iterative solves (method=2, precon=3, accel=3). On the first occurence of a LINALG line, DIR.DEF is defaulted to true; if no LINALG line occurs before the first solve, the options are also defaulted as with DIR.DEF=true.

stopping crit

```
  ITMax     =  integer  (default is 1000)     [Expert]
  NRmax     =  integer  (default is 5)        [Expert]
  LIN.Tol   =  real     (default is 1.0e-6)   [Expert]
  LIN.Atol  =  real     (default is 1.0e-6)   [Expert]
  NLin.tol  =  real     (default is 1.0e-3)   [Expert]
  ITRabf    =  integer  (default is 1)        [Expert]
  ALPHA0    =  real     (default is 1.0e-2)   [Expert]
  MIN.tol   =  real     (default is 1.0e-6)   [Expert]
  AC.Tol    =  real     (default is 1.0e-10)  [Expert]
  AC.Atol   =  real     (default is 1.0e-12)  [Expert]
```

ITMAX is the maximum number of iterations to be performed on each linear system for pure iterative methods; NRMAX sets the maximum number of iterations associated with a Newton-Richardson approximate factorization procedure. LIN.TOL and NLIN.TOL set the relative tolerances to which the linear system must be solved for linear and nonlinear PDEs respectively; LIN.ATOL is an absolute tolerance. The tolerance for nonlinear PDEs is set dynamically based on the residuals of the outer, nonlinear (Newton) iteration using the relative parameter ALPHA0 (times the ratio of the current to the previous residual). MIN.TOL is the minimum tolerance that can be used during the nonlinear iteration, independent of the residual norms. ITRABF sets the maximum number of outer ABF iterations. AC.TOL and AC.ATOL are the relative and absolute linear tolerances respectively for AC linear solves.

misc

```
  MAxfill  =  integer  (default is -1)      [Expert]
  K        =  integer  (default is 1)       [Expert]
  Restart  =  real     (default is 1.0e10)  [Expert]
  SWitch   =  integer  (default is 5)       [Expert]
  LINScal  =  logical  (default is true)    [Expert]
  PTscl    =  logical  (default is true)    [Expert]
  SCale    =  logical  (default is false)   [Expert]
  Bzero    =  logical  (default is true)    [Expert]
```

MAXFILL specifies the maximum amount of fill-in allowed in the LDU (approximate) factorization; with MAXFILL set less than 0, complete fill-in is allowed. K is the number of back vectors for conjugate residual methods. RESTART is a restart tolerance for CGS; SWITCH is the iteration number where the mixed BICG/CGS method switches to CGS. LINSCL activates a scaling which attempts to balance the norms of multiple PDEs. PTSCL is a scaling by unknown, for iterative method only. SCALE specifies that the rows of the linear systems simply be scaled by their diagonals. BZERO zeros Dirichlet rows within the sparse matrix solver; with BZERO=false, the rows are zeroed manually outside.

Examples

For a 2 carrier system, set the linear algebra default to iterative methods, but redefine all the accelerators to use bicg2.

```
SYSTEM   NEWTON CARR=2
LINALG   ITER.DEF ACCEL=13 S.ACCEL=13,13,13
```

The LOAD line

Syntax

```
LOAd  solution files misc
```

Description

The LOAD line loads previous solutions from files for plotting or as initial guesses to other bias points.

Parameters

solution files

```
INFile (or IN1file)  =  filename
IN2file              =  filename
```

The INFILE (or IN1FILE) and IN2FILE parameters specify input files names for solution data and may be up to 20 characters in length. INFILE (or IN1FILE) and IN2FILE represent a present and previous solution respectively. If only one solution is to be loaded (for plotting or as a single initial guess using the PREVIOUS option on the SOLVE line) then INFILE should be used. If two input files are needed to perform an extrapolation for an initial guess (i.e., the PROJECT option on the SOLVE line), IN1FILE and IN2FILE should be used. The solution in IN2FILE is the first to be lost when new solutions are obtained.

misc

```
  Ascii      =  logical  (default is true)
  No.check   =  logical  (default is false)
  DELtav     =  real     (default is 0)
  Outdiff    =  filename
  Differ     =  logical  (default is false)
```

ASCII specifies that any files read or written by the LOAD line should be ascii. NO.CHECK prevents *PADRE* from checking material parameter differences between the loaded files and the values set in the current *PADRE* input file. DELTAV specifies a fixed shift (in volts) for all loaded potentials in the device. The difference between two solutions (IN1FILE-IN2FILE) can be analyzed by reading in both with the mode DIFFER set. The difference is stored; this solution may not be used as an initial guess, or for any purpose other than plotting or extracting data. The difference solution may also be stored in another file using the parameter OUTDIFF.

Examples

The following specifies that a single solution file called SOL.IN should be loaded.

```
  LOAD  INF=SOL.IN
```

In the next example, two solutions are loaded. The present solution is to SOL1.IN and the previous solution is SOL2.IN. We intend to use SOL1.IN and SOL2.IN to project an initial guess for a third bias point.

```
  LOAD IN1F=SOL1.IN IN2F=SOL2.IN
```

Finally, two solutions are loaded, and the difference calculated and stored in a third file.

```
  LOAD IN1F=SOL1.IN IN2F=SOL2.IN DIFF OUTD=SOL1-2
```

The LOG line

Syntax

```
  LOG file specification control
```

Description

The LOG line allows the I-V and/or AC characteristics of a run to be logged to disk. Any I-V or AC data subsequent to the line is saved. If a log file is already open, it is closed and a new file opened.

Parameters

file specification

```
  OUtfile or Ivfile  =  filename
  Acfile             =  filename
```

OUTFILE or IVFILE specify the log file for I-V information. ACFILE specifies the file for AC data. *control*

```
  Last         =  logical  (default is false)
  NO.trap      =  logical  (default is false)
  OFf or NONe  =  logical  (default is false)
```

LAST specifies that only the last bias point be logged for IV files. NO.TRAP prevents logging bias points which were obtained on intermediate bias steps caused by slow convergence (see the METHOD line). OFF (or NONE) turns off IV and AC logging on solves below this line.

Examples

Save the I-V data in a file called IV1 and AC data in AC1.

```
  LOG OUTF=IV1 ACFILE=AC1
```

The LUMP.ELEMENT line

Syntax

```
  LUmp.element  connection quantity
```

Description

The LUMP.ELEMENT line adds a lumped element to the external circuit configuration. New circuit unknown voltages are added to the system as required. Up to 100 elements and 100 nodes can be added.

Parameters

connection

```
  NODE1  =  character
  NODE2  =  character
  NAme   =  character
```

NODE* are labels to identify the desired nodes within the circuit to which the element terminals are attached. If a name is given which is not recognized, *PADRE* creates a new node automatically in the circuit network. NAME is an optional string which can be used to identify the element itself for later reference (e.g. in current plots and printouts); NAME defaults to "Unknown_" followed by the associated lumped element index (by input line order).

For both NODE* and NAME, strings beginning with a digit are reserved for the contacts to the actual device (the PDE simulation domain); names beginning with an "A" are reserved for the terminals where sources are applied (e.g., "A1" refers to external contact number 1). *Only uppercase "A" is allowed for these specifications.*

quantity

```
  REsistance   =  real      (default is infinity)
  Capacitance  =  real      (default is 0)
  Inductance   =  real      (default is 0)
  R.Table      =  character
  R.Scale      =  real      (default is 1)
```

RESISTANCE specifies a lumped resistance value (ohms), CAPACITANCE a lumped capacitance value (F) and INDUCTANCE a lumped inductance value (H) for the lumped element. R.TABLE corresponds to a table of voltages and resistances so that two-terminal lumped elements with non-linear IV characteristics can be added. R.SCALE is a scale factor that the resistances in the table are multiplied by to account for differences in device width. Only one table can be used in a simulation, although it can be reused on separate LUMP.ELEMENT lines an arbitrary number of times, possibly with different R.SCALE.

Examples

The following line makes a parallel RC connection between terminal 1 and its associated internal device contact, followed by a series RC connection between the device contact and terminal 2. The series connection creates a new circuit node called "xx" whose voltage can be accessed in prints and plots. The currents at the element terminals can be accessed later by using the labels "Unknown_1", "Rshunt" and "Unknown_3".

```
  LUMP.EL  NODE1=A1 NODE2=1 RESIS=1000 CAP=10e-15
  LUMP.EL  NODE1=1 NODE2=xx RESIS=2000 NAME=Rshunt
  LUMP.EL  NODE1=xx NODE2=A2 CAP=20e-15
```

The MATERIAL line

Syntax

```
  MAterial  type bands constants models coeffs
```

Description

The MATERIAL line associates physical parameters with the materials in the mesh. Many of the parameters are default for standard materials. These are included in a library of include files in the *PADRE* src directory. The current value of MATERIAL parameters can be examined using the PRINT line or "print" on the MODELS line. Below, the asterisk "*" stands for either "N" or "P" so that separate coefficients for each can be specified for electrons and holes respectively. Parameters marked with a dagger (++) can have separate values in n-type and p-type material.

Parameters

type

```
  NAme         =  character
  DEFault      =  character
  ALloy        =  character
  COmposition  =  real
  NO.charge    =  logical
  N.Type       =  logical
  P.Type       =  logical
```

NAME is the name of the material to be accessed or defined. DEFAULT is the name of a recognized (library or predefined) material which will be used to default all parameters; alternatively ALLOY is the name of an alloy definition from which the coefficients for this material can be interpolated using the compositional fraction 0 <= COMPOSITION <= 1. For those materials which *PADRE* recognizes, standard library definitions will be used. NO.CHARGE ignores space charge in this region when solving the Poisson equation which can be useful for conductors.

N.TYPE and P.TYPE permit separate coefficient values for a large number of the available material parameters (marked with ++ below). If any other parameters are set on the same MATERIAL input line, they are valid for both n-type and p.type material (a warning is given). When separate values are selected, these are appropriately marked in material coefficient printouts by the addition of an "N" or "P" character to the associated material index. By default, both N.TYPE and P.TYPE are turned on. If the user puts one explicityly on the input line, the other then defaults to false.

bands

```
  EG300      =  real       Energy gap at 300K (eV)
  EGAlpha    =  real       Alpha
  EGBeta     =  real       Beta
  AFfinity   =  real       Electron affinity (eV)
  DECDEV     =  real       DELTA Ec / DELTA Ev
  DECDEG     =  real       DELTA Ec / DELTA Eg
  EC.off     =  real       DELTA Ec (eV)
  Refoff     =  character
  Offtemp    =  real       (default is 300K)
```

The above parameters specify the energy band configurations. Default band offsets are computed using electron affinities. These can be overridden using one of DECDEV, EC.OFF or DECDEG which are referenced to the material named REFOFF; the offset is given at the temperature OFFTEMP and converted to the ambient temperature used in the simulation. The resulting band offsets are printed with all other material parameters.

constants

```
  PErmittivity  =  real   Dielectric permittivity (F/cm)
  Qf            =  real   Fixed bulk charge density (/cm**3)
  NC300         =  real   Conduction band density at 300K (/cm**3)
  NV300         =  real   Valence band density at 300K (/cm**3)
  GCb           =  real   Conduction-band degeneracy factor
  GVb           =  real   Valence-band degeneracy factor
  EDb           =  real   Donor energy level (eV)
  EAb           =  real   Acceptor energy level (eV)
  W2dgas        =  real   Width of 2D gas (mm)
  ARICHN        =  real   Richardson constant for electrons
  ARICHP        =  real   Richardson constant for holes
  VSAT*         =  real   Saturation velocity - 300K (cm/s)
  TAUW*         =  real   Intrinsic low field energy relaxation times
(s)
  MU*       ++  =  real   Intrinsic mobility - 300K (cm**2/s)
  TAU*0     ++  =  vector Intrinsic minority carrier lifetimes (s)
  TAUR0     ++  =  real   Intrinsic radiative lifetime (s)
  AUG*      ++  =  real   Auger coefficient (Cn) (cm**6/s)
  C.helm        =  real   Helmholtz coefficient (default is 0.0)
```

These are basic material constants as defined above. W2DGAS is used only in conjunction with 2dgas statistics (see the MODEL line). The energy relaxation times are only used if no relaxation model (W*.MODEL below) is given; carrier effective masses are currently unused. C.HELM is the coefficient in front of a term, linear in potential, added to the Poisson equation for this region; it is useful in performing contact resiatance studies (see Loh, et.al. EDL, 1985).

models

```
  I*.Model    =  character   Ionized impurity scattering (CONMOB)
  E*.Model    =  character   Velocity saturation model (FLDMOB)
  G*.Model    =  character   Gate-field mobility model (GATMOB)
  C*.Model    =  character   Carrier-carrier scattering model (CCMOB)
  D*.Model    =  character   Diffusivity-field model (FLDDIF)
  W*.Model     =  character   Energy relaxation model (FLDMOB)
  S*.Model     =  character   Energy flux coefficient model
  BGN*.model   =  character   Band-gap narrowing model
```

These are material-dependent model types; the first six correspond to mobility models defined on the MODELS line. W*.MODEL and S*.MODEL are only relevant if carrier temperatures are solved for. Possible choices are:

```
  I*.Model     =  table, analytic, klaasen
  E*.Model     =  caughey, barnes, limit, scharfetter, alley, hansch
  G*.Model     =  pinto, yamaguchi, sun, caughey, schwarz, sgs, lentz,
mujtaba
  C*.Model     =  dorkel, klaasen
  D*.Model     =  mccoll
  W*.Model     =  none, baccarani, lincut, hypcut, npar, general
  S*.Model     =  constant, mc
  BGN*.Model   =  slotboom, cak-sige
```

Descriptions of these models are available in another document. Coefficients for each model can be adjusted below.

coeffs

```
  *.Bgn   ++ =  vector      BGN coefficients
  L*.Mu   ++ =  vector      Lattice mobility coeffs
  II*.Mu  ++ =  vector      Ionized impurity scattering coeffs
  NIN*.Mu ++ =  vector      Neutral impurity scattering coeffs
  CC*.Mu     =  vector      Carrier-carrier scattering coeffs
  E*.Mu      =  vector      Nonlinear drift velocity (vs. field) coeffs
  G*.Mu      =  vector      Gate-field scattering coeffs
  D*.Mu      =  vector      Diffusivity vs. field coeffs
  W*.Mu      =  vector      Carrier temperature mobility coeffs
  W*.Kappa   =  vector      Thermal conductivity proportionality coeffs
  NTAU*   ++ =  real        min-carr. lifetime conc. param (/cm**3)
  B0dir   ++ =  real        Radiative-lifetime coeff (cm**3/s)
  NTAUR   ++ =  real        Radiative-lifetime conc. param (/cm**3)
  GEn.con ++ =  real        Generation constant (eh pairs/cm**3/rad)
  TRap.type  =  character   trap types
  ETrap      =  vector      trap energy levels = Et - Ei (eV)
  NTRap      =  vector      trap densities (/cm**3)
  E*.Ion     =  vector      Impact-ionization field threshs (V/cm)
  A*.Ion     =  vector      Impact-ionization rate (/cm)
  B*.Ion     =  vector      Impact-ionization assym field (V/cm)
  W*.Etpar   =  vector      New energy transport model coefficients
  C.tunnel   =  vector      Band-to-band tunneling coefficients
  Laser   ++ =  vector      Stimulated-emission coeffs
```

These are coefficients for various models, e.g. band-gap narrowing, mobility, lifetime, impact ionization and energy relaxation. The scalar parameters are relatively self-explanatory; see the technical reference manual for more details.

The vector parameters for traps allow for multiple trap levels with different minority lifetimes and densities (the densities are only required for deep-level traps). The type of the traps (standard SRH or deep-level acceptor/donor) is determined by the string TRAP.TYPE, where each character refers to its corresponding element in the vector parameters; possible types are "n" (donor), "p" (acceptor) and "0" (SRH) with a default of "0" for all unspecified. The vector parameters for impact ionization permit multiple field regimes.

The remaining vector quantities define models which require several parameters. See primary documentation for detailed descriptions of model forms and coefficient definitions.

Types of Diffusion Noise Models

```
ns.model    =  character
```

ns.model specifies the types of diffusion model is desired. ns.model=diffmu specifies that differential mobility model is used to compute microscopic noise sources, ns.model=incmu specifies that incremental mobility model is used [default], ns.model=userdiff requires that the user supply the field-dependent diffusivity model, and ns.mode=userbeta requires that the user supply the field dependent beta (ratio of diffusivity and mobility) model. These user defined files are to be included in userdiff.f.

Examples

Specify two trap levels in all silicon regions, one at midgap and the second 0.2eV above, with lifetimes of 1ms and 500ns respectively; the first trap involves standard SRH recombination whereas the second is a deep-level donor trap with a density of 1.0e15/cm**3. Also set the hole ionization coeffs to those suggested by van Overstraeten.

```
MATERIAL  NAME=silicon TAUN0=1e-6,5e-7 TAUP0=1e-6,5e-7 ETRAP=0,0.2
+         N.TRAP=0,1e15 TRAP.TYP=0n
+         EP.ION=1.75e5,4e5 AP.ION=1.58eE6,6.71E5
+         BP.ION=2.036E6,1.693E6
```

Define a material called "likesi" to be silicon with some changes to the gate-field mobility and analytic ionized impurity scattering models for electrons. Also, include the differential mobility model for noise calculations.

```
MATERIAL  NAME=likesi DEF=silicon
+         GN.MOD=sgs    GN.MU=4.75e7,1.74e5,0.125,5.84e14
+         IN.MOD=analyt IIN.MU=55.24,1.072e17,0.733,-3.8
+         NS.MODEL=diffmu
```

Define a material called "sige.3" using a previously defined alloy called "sige"; the compositional fraction is 0.3, hence the material is interpolated as Si(0.7)Ge(0.3) (see the ALLOY line documentation). Override the default (interpolated) energy gap, band offset and bulk mobilities and lifetimes.

```
ALLOY     NAME=sige X1ALLOY=0 M1ALLOY=silicon
```

```
  +                    X2ALLOY=1 M2ALLOY=germanium
  MATERIAL  NAME=sige.3 DEF=sige COMP=0.3
  +         EG300=0.830 DECDEV=0.087 REFOFF=silicon
  +         MUN0=242 MUP0=186 TAUN0=50e-6 TAUP0=50e-6
```

Set separate SRH lifetimes for n-type and p-type material called "polysi". Because the EG300 parameter cannnot have separate n.type and p.type values, it is defined as 1.05eV for both types, eventhough the value is given on a line with an "n.type" modifier.

```
  MATERIAL  NAME=polysi EG300=1.05 TAUN0=6e-6 TAUP0=3e-6 N.TYPE
  MATERIAL  NAME=polysi TAUN0=4e-6 TAUP0=2e-6 P.TYPE
```

The MESH line

Syntax

```
  MESh type specs output adjustments data structure integrate
```

Description

The MESH line either initiates the mesh generation phase or reads a previously generated mesh. Loading a mesh created directly by a previous *PADRE* run will also load all its doping information, although for heterostructures - especially those where doping changes abruptly at region boundaries - using of the doping reload feature (see the DOPING line) is recommended.

Parameters

type

```
  RECtangular  =  logical
  PREvious     =  logical
  Tri          =  logical  (default is false)
```

RECTANGULAR indicates that a rectangular mesh is to be created using X.MESH/Y.MESH lines; the inclusion of the NX/NY parameters (see below) and no other *type* flags defaults RECTANGULAR to true. PREVIOUS is a flag to indicate that a mesh previously generated by *PADRE* is to be loaded; with the INFILE parameter (see below) specified and no other *type* flags set, PREVIOUS defaults to true. TRI specifies that mesh information is to be taken from a standard "tri" mesh generator output file (the "-o" option on the "tri" command); these files do not contain any doping information.

specs

```
  NX           =  integer
  NY           =  integer
```

```
NZ           =  integer   (default is 1)
Width        =  real
INfile       =  filename
ASCII.In     =  logical   (default is true)
CYlindrical  =  logical   (default is false)
```

With the RECTANGULAR flag above, NX defines the number of nodes in the x-direction and NY the number in the y-direction. NZ defines a number grid "planes" in the depth dimension; it can be specified with rectangular or nonrectangular grids in the xy-plane. WIDTH usually defines the width in the depth dimension in mm for 2D simulations (NZ=1) or 3D simulations where no Z.MESH lines are given (in which case the grid planes are spaced uniformly from z=0 to z=WIDTH). For 3D simulations where the depth dimension is fixed, WIDTH becomes a unitless scale factor to multiply all terminal currents/fluxes by; this parameter is convenient to use when only simulating sections of an entire device. With PREVIOUS or TRI, INFILE gives the name of the associated mesh input file to be loaded, and ASCII.IN is a flag to indicate that the file is ascii. CYLINDRICAL sets up a 2D structure (NZ=1) as if it were 3D with cylindrical symmetry about x = xmin.

adjustments

```
PAck.reg    =  logical    (default is false)
Unpack.reg  =  logical    (default is false)
CONdense    =  character  (default is "none")
FLip.y      =  logical
SCale       =  integer    (default is 1)
DIAg.fli    =  logical    (default is false)
SMooth.key  =  integer    (default is 1)
IT.smooth   =  integer    (default is 30)
IGnore      =  integer
PRIce       =  logical    (default is false)   [Expert]
OBt.check   =  logical    (default is true)    [Expert]
```

PACK.REG combines together all (adjoining and disjoint) regions of equivalent material type; UNPACK.REG attempts to perform the inverse, but only on disjoint regions. CONDENSE defines the types of regions to be packed or unpacked; possible inputs are "all" for all types of regions, "ins" for insulating regions only, "semi" for semiconductor regions and "none" to disable the option. FLIP.Y is a flag which will reverse the sign of the y-coordinate, i.e. flip the structure about the "y" axis; this parameter defaults to false, except when TRI or GEOMETRY is specified, as these grid generators work in a right-handed coordinate system. SCALE is a factor by which all the coordinates read from mesh files are multiplied by. With rectangular meshes only, DIAG.FLIP flips the diagonals about the center of the grid; the defaults is to place the diagonals in the same direction. SMOOTH.KEY is an integer which causes mesh smoothing; it digits are read in reverse order and decoded as follows:

```
0    No action.
1    Triangle smoothing, maintaining all region boundaries fixed.
2    Triangle smoothing, maintaining only material boundaries.
3    Node averaging.
```

```
  6    Add node to eliminate obtuse triangles on boundaries.
```

Options 1 and 3 are the most common; 2 is used only if a device has several regions of the same material and the border between the different regions is unimportant. IT.SMOOTH is the maximum number of iterations to be used during triangle smoothing. Only regions not specified in IGNORE (concatonated integer format) are smoothed; if no regions are specified by IGNORE, the default is to smooth only nonrectangular regions.

PRICE uses the obtuse correction suggested by Price which prohibits all negative coupling coefficients. With no smooth key (SMOOTH.K=0), the PRICE option is strongly recommended. OBT.CHECK forces the *PADRE* to assemble only on one side of an obtuse-acute edge, using the net coupling coefficient, rather than adding the components separately.

output

```
  OUTFile    =  filename
  ASCII.Out  =  logical    (default is true)
```

OUTFILE is the *PADRE* format output file to be read by a later run. If ASCII.OUT is set, OUTFILE will be written in ascii, otherwise it will be binary.

data structure

```
  NEclmax      =  integer
  COLor.order  =  character                         [Expert]
  Hetero       =  logical    (default is false)
  Jamin        =  real       (default is -1)        [Expert]
  FAke.edg     =  logical    (default is true)      [Expert]
  MID.Stat     =  logical    (default is true)
  MID.File     =  logical    (default is true)
```

NECLMAX defines a maximum number of elements per color (hence the maximum inner vector length) to be used in the matrix assembly; setting NECLMAX limits the size of the work space necessary which may be necessary to fit big grids on small machines. COLOR.ORDER specifies the colored ordering to be used on equations in the matrix; the choices are "greedy" (2D default) and "reverse" (3D default). HETERO forces the use of the internal heterostructure data format even if materials with different semiconductor band configurations are not given; this option is useful if nodes with multiple local values are desired. JAMIN decouples any nodes whose total coupling coefficient is less than the given value; this parameter can sometimes be useful in conjunction with rectangular grid sections (where the coupling coefficient across a diagonal is zero). FAKE.EDG includes all edges in the matrix arising from nonphysically connected element verticies; e.g. for 3D prisms, there are 9 real edges together with 6 "fake" edges that can be removed (note that removing these edges makes the matrix smaller but also affects the assembly of terms like mobility and impact ionization which require 3D fields; convergence rates may be degraded if these edges are not included). MID.STAT assembles the edge pointer array once and stores it statically for use throughout the program. With MID.STAT turned off, MID.FILE specifies whether the edge array should be dumped to a scratch file (read when needed) or reassembled during every assembly; by turning MID.STAT off, the program takes less memory, but the assembly cost is increased.

integrate

```
DOp.int    =  logical  (default is false)  [Expert]
1d.integ   =  logical  (default is false)  [Expert]
DI.Degree  =  real     (default is 3)      [Expert]
DI.Level   =  integer  (default is 0)      [Expert]
```

DOP.INT indicates that the doping density used in the Poisson equation should be integrated at each node; this option may result in greater accuracy by taking a slightly longer mesh set-up time. 1D.INTEG performs exact 1d integrations of analytic doping specifications. DI.DEGREE and DI.LEVEL control the general 2D doping integration process; DI.DEGREE is the order of the integration quadrature formula to be used, and DI.LEVEL is the maximum number of refinement steps used on each control volume to perform the integration.

Examples

Initiate a rectangular mesh and request it to be stored in mesh1.pg :

```
MESH  RECTANGULAR NX=40 NY=17 OUTF=mesh1.pg
```

Read a geometry file, smooth the mesh, and store the file for a later run (ascii format). The smoothing does several averaging and flipping steps. The digits are read in reverse order, so that the flipping comes first, followed by node averaging, and so on.

```
MESH  TRI INF=geom1 SMOOTH.K=13131 OUTF=mesh1.pg
```

Read a previously generated 2D mesh and add 10 uniformly spaced (over 5mm) grid planes in 3D. Output the new 3D mesh to another file name. If the input mesh already had grid planes (i.e. was 3D), an error would result.

```
MESH  INF=mesh2d NZ=10 WIDTH=5 OUTF=mesh3d
```

The METHOD line

Syntax

```
METhod    non-lin conv psuedo continue damping
          carriers time-dep misc
```

Description

The METHOD line sets parameters associated with miscellaneous numerical methods which at present consist primarily of those associated with nonlinear iteration and the transient discretization. There can be more than one METHOD line in a single simulation, so that parameters can be altered. The default values of the parameters are used on the first occurrence of the METHOD line; subsequent METHOD lines only alter those coefficients specified.

Parameters

non-lin conv

```
  INloops or ITlimit  =  integer      (default is 20)
  Outloops            =  integer      (default is 20)
  MIN.Inner           =  integer      (default is 1)
  MIN.Outer           =  integer      (default is 1)
  Gloops              =  integer      (default is 0)
  XNorm               =  logical      (default is true)
  RHsnorm             =  logical      (default is true)
  L2Norm              =  logical      (default is true)
  X.toler             =  real vector  (default is 1.0e-5)
  RHS.toler           =  real vector  (default is 1.0e-10)
```

The above parameters are used to determine the non-linear convergence of the solution methods. INLOOPS (or ITLIMIT) is the maximum number of allowed inner nonlinear loops, i.e. those performed on each group of PDEs specified on the SYSTEM line. OUTLOOPS specifies the maximum number of outer loops to be performed over all the PDE groups. MIN.INNER and MIN.OUTER force a minimum number of inner and outer iterations respectively regardless of the convergence state. GLOOPS optionally gives a maximum number of plug-in (Gummel) loops to be performed before beginning the specified nonlinear iteration. XNORM and RHSNORM specify the error norms - the nonlinear updates or residuals respectively - to be monitored. X.TOLERANCE and RHS.TOLERANCE specify the convergence tolerances on the XNORM and RHSNORM respectively for each PDE where the vector ordering corresponds to that implied by the digit codes defined for the SYSTEM line. For the update norm (the default), the Poisson and energy updates are measured in absolute units of kT/q, and carrier updates are measured relative to the local carrier concentration. With RHSNORM the norms are scaled such that an equivalent tolerance can be used for each PDE. L2NORM specifies that the RHSNORM is to be computed as an L2 rather than infinity norm.

psuedo continue

```
  TRAp       =  logical    (default is true)
  DGmin      =  real       (default is 3.0e-9)
  A.Trap     =  real       (default is 0.5)
  N.Trap     =  real       (default is 2)
  M.Trap     =  real       (default is 1)
  I.Trap     =  real       (default is 10)
  MAxneg     =  integer   (default is 10)
  DI.Trap    =  real      (default is 1.0e-5)
  DV.Trap    =  real      (default is 1.0e-5)
  OUT.Trap   =  logical   (default is true)
  IGN.Inner  =  logical   (default is false)
  STop       =  logical   (default is true)
```

TRAP specifies the initiation of a "psuedo-continuation" method which attempts to detect if a solution process starts to diverge. If it does, the electrode bias steps taken from the initial guess are reduced by the multiplicative factor A.TRAP (up to a maximum of I.TRAP cutbacks or until DI.TRAP or DV.TRAP are violated - see below). One criterion for determining divergence is that the maximum number of Newton iterations is exceeded with any group of PDEs. Also, *PADRE* can monitor internal quantities - e.g. the norm of the residuals (RHSNORM), the norm of the updates (XNORM) and the size of any damping coefficient - during the Newton iterations if it is clear that convergence is slow (this feature can be disabled by specifying IGN.INNER). Divergence tendencies are ignored if the residual norms are below a small threshold (to account for round-off error), DGMIN. N.TRAP and M.TRAP specify the number of Newton iterations to be ignored before checking and the consecutive number of occurences on which the norm must go up before the TRAP feature is enabled. A trap can also be initiated by reaching a maximum number of consecutive iterations which produce negative concentrations (MAXNEG). DI.TRAP and DV.TRAP specify the minimum size of the allowed bias steps after application of A.TRAP; DV.TRAP is given in volts, while DI.TRAP is defined in a relative sense. OUT.TRAP controls whether solution output files are written for any points computed due to a trap; the file names used are derived from the name given to the originally attempted point, with the string "Xn" (n is a digit, starting from 0) appended. If the TRAP feature is not used, setting the STOP parameter will force the simulator to suspend execution if convergence is not reached.

damping

```
  DAMPEd     =  character  (default is "single")  [Expert]
  ITDamp     =  integer    (default is 1)         [Expert]
  DElta      =  real       (default is 1.0e-6)    [Expert]
  DAMPLoop   =  integer    (default is 10)        [Expert]
  DFactor    =  real       (default is 10.0)      [Expert]
  DPower     =  real       (default is 2)         [Expert]
  DVLimit    =  real       (default is 10.0)      [Expert]
  TRUncate   =  logical    (default is false)     [Expert]
  VMargin    =  real       (default is 0.01*kT/q) [Expert]
```

The DAMPED parameter indicates the use of a more sophisticated damping scheme proposed by Bank and Rose (this is the recommended option, particularly for large bias steps). The method is recommended for single PDE loops (DAMPED="single"), but it can also be performed for coupled groups of PDEs (DAMPED="all"). To turn this damping method off, use DAMPED="none". The Bank/Rose damping scheme is controlled by the parameters ITDAMP, DELTA, DAMPLOOP, DFACTOR and DPOWER which specify the iteration on which to start to damp, the reduction threshold, maximum number of damping loops, update reduction factor and update reduction power respectively. Other damping schemes that can be performed either separately or in conjunction with Bank/Rose include a simple limit on any potential updates (DVLIMIT) and truncation of all potentials so that they lie between the minimum and maximum possible voltages (TRUNCATE) with a margin on either end of VMARGIN.

carriers

```
FIX.qf    =  logical  (default is false)   [Expert]
LINCont   =  logical  (default is true)    [Expert]
LINPois   =  logical  (default is false)   [Expert]
EXPCon    =  integer  (default is -1)      [Expert]
EXPNeg    =  integer  (default is -1)      [Expert]
LOgc      =  integer  (default is -1)      [Expert]
QF        =  integer  (default is -1)      [Expert]
IGN.Neg   =  logical  (default is false)   [Expert]
NEGcon    =  logical  (default is false)   [Expert]
NFact     =  real     (default is 0.1)     [Expert]
CMin      =  real     (default is 0)       [Expert]
NEG.Damp  =  real     (default is -1)      [Expert]
NEG.Tmin  =  real     (default is 0.01)    [Expert]
SLotboom  =  integer  (default is -1)      [Expert]
J0form    =  integer                       [Expert]
JWform    =  integer  (default is 1)       [Expert]
```

FIX.QF fixes the quasi-Fermi potential of each non-solved for carrier to a single value, instead of picking a value based on local bias (see the P.BIAS and N.BIAS parameters on the SOLVE line). LINCONT specifies that the continuity equation be treated as a linear equation with respect to carrier concentrations, even though nonlinear models (or updates) might be used. LINPOIS treats the Poisson as a linear equation - i.e., the carrier densities are assumed to be independent of potential.

By default *PADRE* uses the carrier densities as variables. However EXPCON, EXPNEG, LOGC, QF and SLOTBOOM allows other types of carrier variables during the nonlinear iteration process. QF gives quasi-Fermi potentials, SLOTBOOM gives Slotboom variables (exponentials of quasi-Fermi potentials), while EXPCON, EXPNEG and LOGC both specify the logarithm of the densities. LOGC is implemented during the discretization, but EXPCON and EXPNEG are applied at the update step; further EXPNEG is only applied to those updates which are negative. Each of these parameters is an integer, and only one can be set at a time. A value less than zero indicates the type is off, a value of zero forces the type to be used only during the coupled (non-smoothing) phase of the solution process and a positive value sets the type on for all iterations. With EXPCON, LOGC and QF, it is strongly recommended that a finite number of smoothing loops always be performed (see GLOOPS above).

IGN.NEG, NEGCON, NFACT, CMIN, NEG.DAMP and NEG.TMIN control the treatment of negative concentrations when using the pure density formulation. IGN.NEG tells *PADRE* that a solution that looks as though it has converged, yet has negative values is treated as converged. NEG.CON forces *PADRE* to take some action during the Newton iteration if a negative concentration occurs. With NFACT <= 0, negative concentrations are corrected by using an update equivalent to that used by EXPCON and EXPNEG (log of densities). With NFACT > 0, the offending densities are obtained from their previous values by multiplying by NFACT (hence recommended to be less than 1). Further, any adjustment is forced to yield a concentration >= CMIN. With NEG.DAMP > 0, a damping coefficient is computed to make the updated concentration go down by a factor NEG.DAMP; the damping coefficient is limited to be greater than NEG.TMIN.

J0FORM controls the form of the Scharfetter-Gummel current discretization used; with J0FORM=1, a standard concentration based form is used while for J0FORM=2, a potentially more accurate, quasi-Fermi based implementation is employed. J0FORM defaults to 2 only when QF is activated. JWFORM controls the discretization of temperature along element edges in the expression for energy flux. With JWFORM=0, a simple average is used while for JWFORM=1, the temperature is integrated exactly assuming a linear form.

time-dep

```
  2ndorder     =  logical  (default is true)
  TR.print     =  logical  (default is false)
  TAuto        =  logical  (default is true)
  TOLR.time    =  real     (default is 1.0e-2)
  TOLA.time    =  real     (default is 1.0e-10)
  L2Tnorm      =  logical  (default is true)
  RMs.tnorm    =  logical  (default is true)         [Expert]
  DT.min       =  real     (default is 1.0e-16)
  T.LIMA       =  real     (default is 0.2)
  T.LIMB       =  real     (default is 2.0)
  EXTrapolate  =  logical  (default is false)
```

The above parameters control time integration. 2NDORDER specifies that the second-order TR-BDF2 discretization of Bank, et.al. be used as opposed to first-order backward differences. TR.PRINT forces the terminal characteristics (e.g. currents) to be printed after the TR segment of the composite step for TR-BDF2 as well as the BDF2 segment. TAUTO forces *PADRE* to select time-steps automatically from the local truncation error estimates. TOLR.TIME and TOLA.TIME are the relative and absolute local truncation error bounds to control the timestep. L2TNORM specifies that the error norms be L2 as opposed to infinity norms for calculating the time-steps; RMS.TNORM specifies whether the error norm is a roo-mean squared value or the ratio of error and solution norms. DT.MIN is the minimum time-step allowed in seconds, while T.LIMA and T.LIMB are the minimum time-step cutback and maximum time-step increase respectively. EXTRAPOLATE uses a second-order extrapolation to compute initial guesses for successive time-steps.

misc

```
  AUtonr         =  logical   (default is true)
  NRCriterion    =  real      (default is 0.1)
  NRLoop         =  real      (default is 2)
  PRint          =  logical   (default is false)
```

```
ERr.estimate  =  logical   (default is false)
LIN.Proj      =  logical   (default is false)
LIN.Fail      =  logical   (default is false)
OFfmu         =  logical   (default is true)     [Expert]
MU.Tol        =  real      (default is 1.0e7)    [Expert]
MU.Dir        =  real      (default is 1.0e20)   [Expert]
MU.Neg        =  logical   (default is true)     [Expert]
W.muoff       =  logical   (default is false)    [Expert]
AC.method     =  character (default is "lu")
```

The AUTONR option initiates a Newton-Richardson algorithm which attempts to minimize the number of LU factorizations per bias point associated with direct linear solves. NRCRITERION is the ratio by which the norm from the previous Newton loop must go down in order to be attempt to use the same Jacobian (i.e., LU decomposition) for the current Newton loop; this test is performed on Newton loops numbered NRLOOP and above. If the same factorization is indeed used (indicated by a "*" character in the output file), yet the linear system residual does not converge in a given amount of loops (see LINALG input line), the Jacobian will be refactored anyway (indicated by a "%" character in the output file).

PRINT prints the terminal fluxes/currents after each continuity iteration; if this parameter is not set, the terminal fluxes/currents are only printed after the solution converges. ERR.ESTIMATE gives an estimate of the discretization error after each bias point using the error indicator described on the REGRID/ADAPT lines. The LIN.PROJ parameter specifies linear rather than exponential projections for the carrier concentrations during initial guess setup. LIN.FAIL forces a nonlinear convergence error if the tolerance on the linear solution cannot be met in the maximum number of interations (see LINALG line); with TRAP on, such an occurrence will lead to a smaller bias step. OFFMU directs *PADRE* to turn off field-dependent mobility derivatives in in the full Newton system under certain conditions, defined by MU.TOL, MU.DIR and MU.NEG. MU.TOL and MU.DIR are multiplicative factors to be used with RHS.TOL to determine when all the derivatives or just the directional derivatives are to be invoked. MU.NEG omits mobility derivatives any time the previous iteration resulted in updates that would have yielded negative concentrations. W.MUOFF specifies that checks should be made for carrier-temperature (rather than field dependencies for drift-diffusion) derivatives before including.

AC.METHOD specifies the method for solving the linear systems arising from ac small-signal analysis. For arbitrary frequencies, a full matrix solve is recommended ("lu"); for low frequencies (e.g., below ft), a substantial amount of cpu time can be saved by using a linear expansion ("lin") or relaxation ("sor").

Examples

The following specifies that damping for all isolated PDEs, and the Poisson error tolerance should be 1.0e-12. Note that because XNORM defaults to true, XNORM must be turned off to use the residual norm as a convergence criterion. If XNORM=FALSE was not specified, both the rhs and update norms would be printed, but the update norm would be used to determine convergence.

```
  METHOD  DAMPED=SINGLE RHS.TOL=1.e-12 RHSNORM XNORM=FALSE
```

The next example illustrates the trap feature. The first SOLVE line solves for the initial, zero bias case. On the second SOLVE line, we attempt to solve for V2=3, V3=5 volts. If such a large bias change caused divergence here, the bias step would be multiplied by ATRAP(0.5); i.e., an point at (V2=1.5, V3=2.5 volts) would be attempted before retrying V2=3, V3=5 volts again. If the intermediate point diverges as well, *PADRE* continues to reduce the step by 0.5 (next is V2=0.75, V3=1.25 volts) up to 4 times. The intermediate solutions will be saved in output files based on the name given to the actual bias point attempted; the string "X0" is appended to the first, and the last character is incremented similar to that for normal bias stepping. For example, if two intermediate steps to V2/V3=3/5 volts were required, they would be stored in "outaX0" and "outaX1" while V2/V3=3/5 volts would be stored in "outa".

```
  METHOD  TRAP ATRAP=0.5
  SOLVE   INIT
  SOLVE   V2=3 V3=5 OUTFILE=outa
```

In the next example, the second-order transient discretization is used, but the relative LTE criterion, 1.0e-3, is chosen smaller than the default. Newton-Richardson is also specified. Note that because the Jacobian is exact for the second part (BDF-2) of the composite time-step, there should be very few factorizations for the BDF-2 interval when AUTONR is specified.

```
  METHOD  TOLR.TIME=1E-3 AUTONR
```

The MODELS line

Syntax

```
  MOdels  forms general scattering drive terms numerical
```

Description

The MODELS line sets the temperature for the simulation and specifies model flags to indicate the inclusion of various physical mechanisms and models. Some non-material dependent model parameters are also specified here.

Parameters

forms

```
C1.Sign  =  integer  (default is -1)
C2.Sign  =  integer  (default is +1)
C1.Type  =  integer  (default is 1)
C2.Type  =  integer  (default is 2)
JTherm   =  logical  (default is true)
ET.form  =  logical  (default is false)
```

C1.SIGN and C2.SIGN control the sign of the electric charge associated with carrier 1 and 2; C1.TYPE and C2.TYPE control the type (1=electrons, 2=holes) associated with carriers 1 and 2. JTHERM controls the inclusion of the thermoelectric current in the energy balance models. ET.FORM specifies a new form of the energy transport equations be used (see D. Chen, et. al.); ET.FORM=false is the standard energy balance formulation.

general

```
SRh          =  logical  (default is false)
AUger        =  logical  (default is false)
DIrect       =  logical  (default is false)
DEeptrap     =  logical  (default is false)
CONLife      =  logical  (default is false)
IMpact       =  logical  (default is false)
TUnneling    =  logical  (default is false)
Laser        =  logical  (default is false)
BGn          =  logical  (default is false)
STatistics   =  logical  (default is "boltzmann")
INcomplete   =  logical  (default is false)
Print        =  logical  (default is false)
TEmperature  =  real     (default is 300K)
```

SRH, AUGER and DIRECT specify Shockley-Read-Hall, Auger and direct radiative recombination respectively. DEEPTRAP activates acceptor and donor deep-level traps, where the trap occupancy is equivalent to SRH however with a charge density which is included in the Poisson equation. SRH, DEEPTRAP and DIRECT can optionally use concentration-dependent lifetimes by specifying CONLIFE. IMPACT incorporates an impact ionization generation term. TUNNELING specifies that a band-to-band tunneling model is turned on. LASER is a recombination model which simulates the effects of stimulated-emission.

BGN is band-gap narrowing. STATISTICS indicates the type of carrier statistics to be used ("boltzmann", "fermi" or "2dgas"), while INCOMPLETE indicates that incomplete-ionization of impurities should be accounted for. PRINT prints the status of all models and a variety of coefficients and constants. TEMPERATURE is the ambient temperature and should be specified in Kelvin units.

scattering

```
CONMob    =  logical  (default is false)
CCmob     =  logical  (default is false)
FLDMob    =  logical  (default is false)
```

```
GAtmob     =  logical  (default is false)
FLDDif     =  logical  (default is false)
NEutral    =  logical  (default is false)
E.Region   =  integer  (default is all regions)
G.Region   =  integer  (default is all regions)
D.Region   =  integer  (default is all regions)
```

The parameters above specify which carrier transport models are to be used. CONMOB is concentration-dependent, ionized impurity scattering model. CCMOB includes a carrier-carrier scattering term in the mobility, and NEUTRAL incorporates a neutral impurity scattering effect (re. Sklar). FLDMOB specifies a nonlinear drift velocity-field model (i.e., a parallel field or temperature dependent mobility), GATMOB includes a vertical field dependence and FLDDIF includes a field-dependent, diffusivity to mobility ratio. E.REGION, G.REGION and D.REGION apply the velocity saturation, gate-field and diffusivity-field models respectively only within selected regions of the device.

drive terms

```
I.Current  =  character  (default is "jmag")
E.Drive    =  character  (default is "qf")
G.Drive    =  character  (default is "exqf")
D.Drive    =  character  (default is "qf")
I.Drive    =  character  (default is "eoj")
```

I.CURRENT specifies the way carrier currents are computed in the expression for impact ionization; "jmag" forces the use of the actual magnitude of electron and hole currents, and "jsat" uses the product of the local densities and respective saturation velocities (times q). The parallel (for drift-diffusion), gate, diffusivity field and impact ionization models require a driving force or field term. E.DRIVE, G.DRIVE, D.DRIVE and I.DRIVE are character strings which define various possibilities:

```
E.Drive  =  eoj     E-j dot-product
            eoqf    E-(grad qf) dot-product
            ex      X-component of E (abs value)
            ey      Y-component of E (abs value)
            emag    Magnitude of E
            qf      Magnitude of (grad qf)
            qfb     E-Magnitude of (grad qf) with numerical smooth

G.Drive  =  exj     E-j cross-product
            exqf    E-(grad qf) cross-product
            ex      X-component of E (abs value)
            ey      Y-component of E (abs value)
            emag    Magnitude of E

D.Drive              (same choices as E.DRIVE)

I.Drive              (same choices as E.DRIVE)
```

For energy-balance simulations, each of these models (except gate-field) have temperature dependencies and the driving forces are ignored.

Noise Models

```
gen.nois    =  logical  (default is false)
diff.noi    =  logical  (default is false)
1overf.n    =  logical  (default is false)
```

The parameters above specify which type of noise sources are desired for device noise calculations.

numerical

```
EEth.mu   =  real     (default is 1 V/cm)        [Expert]
GEth.mu   =  real     (default is 1 V/cm)        [Expert]
DVside    =  real     (default is 1.0e-6 kT/q)   [Expert]
GZth.mu   =  real     (default is 0.03mm)
JTh.mu    =  real     (default is 1.0e-6)        [Expert]
AJMin     =  real     (default is 1.0e-20)       [Expert]
AJRoff    =  real     (default is 1.0)           [Expert]
II.Cut    =  real     (default is -10)           [Expert]
II.Hyp    =  real     (default is 5.0e-3)        [Expert]
II.Tanh   =  real     (default is 50)            [Expert]
II.Sig    =  real     (default is 500)           [Expert]
E.Vertex  =  logical  (default is false)         [Expert]
Qf.np0    =  real     (default is 1.0e-5)        [Expert]
```

EETH.MU, GETH.MU, DVSIDE, GZTH.MU and JTH.MU are thresholds used in the field-dependent models. EETH.MU and GETH.MU are minimum fields used in the parallel and gate field mobility models (basically a zero field offset for a hyperbola). DVSIDE is a minimum potential difference across an edge, used in any field computation. GZTH.MU is a threshold on the distance from an insulator-semiconductor interface where the gate field models are included. JTH.MU, AJMIN and AJROFF are low current thresholds used in `E . j`, `E x j` and `|J|` computations (see E.DRIVE, et. al. above); JTH.MU is a threshold relative to the maximum current density over the device, AJMIN is an absolute quantity (in A-cm**2), and AJROFF is a multiplicative factor applied to internally computed maximum roundoff currents (in A-cm**2). II.CUT specifies an initial ionization threshold field be applied for fields below the cutoff for a given material; the field used is |II.CUT|*beta0, where beta0 is the threshold in the first field interval (see B*.ION on the MATERIAL line). II.HYP > 0 specifies that a hyperbolic approximation to the (log) piecewise-linear ionization rates be used; the value of II.HYP defines the relative offset of the rate at the intersection of adjacent intervals, and II.TANH defines the multiplicative term in front of the argument to a tanh function which blends consecutive hyperbolas. II.SIG performs the same function as II.TANH, but for II.HYP <= 0 (the lower II.TANH or II.SIG, the smoother - but less accurate - the adjusted function becomes). E.VERTEX computes fields at verticies instead of a single value over the entire element. QF.NP0 defines the minimum concentration for use in the numerical smooth qf driving force estimates; the value given is multiplied by the local intrinsic concentration.

Examples

Run a simulation with Shockley-Reed-Hall recombination using concentration dependent lifetimes. Field-dependent models are only applied in regions 1 and 2 however. The ambient temperature is 425K.

```
  MODELS  TEMP=425 SRH FLDMOB E.REGION=12
```

The MOVIE line

Syntax

```
  MOvie area-definition plotted-quantity control
```

Description

The MOVIE line maintains a view2d format file, suitable for making movies, of a selected quantity which is continuously logged for all subsequent bias points.

Parameters

area-definition

```
  X.MIn  =  real
  X.MAx  =  real
  Y.MIn  =  real
  Y.MAx  =  real
  Z.pos  =  real  (default is 0)
```

The above parameters define the rectangular area of the device to be plotted. The default area is a rectangle around the entire device in the xy-plane closest to the z-coordinate specified by Z.POS (in mm).

plotted-quantity is one of:

```
  POtential    =  logical   Mid-gap potential
  QFN          =  logical   Electron quasi-fermi level
  QFP          =  logical   Hole quasi-fermi level
  N.temp       =  logical   Electron temperature
  P.temp       =  logical   Hole temperature
  BAND.Val     =  logical   Valence band potential
  BAND.Cond    =  logical   Conduction band potential
  DOping       =  logical   Doping
  ELectrons    =  logical   Electron concentration
  Holes        =  logical   Hole concentration
  NET.CHarge   =  logical   Net charge concentration
  NET.CArrier  =  logical   Net carrier concentration
  J.Conduc     =  logical   Conduction current
```

```
J.Electr     =  logical   Electron current
V.Electr     =  logical   Electron velocity
J.Hole       =  logical   Hole current
V.Hole       =  logical   Hole velocity
J.Displa     =  logical   Displacement current
J.Total      =  logical   Total current
E.field      =  logical   Electric field
Recomb       =  logical   Net recombination
FLowlines    =  logical   Current flow lines
```

The above parameters specify the quantity to be plotted. For vector quantities the magnitude is plotted. Model dependent parameters (current and recombination) are calculated with the models currently defined, *not* with the models that were defined when the solution was computed. This allows the display of, for instance, Auger and Shockley-Read-Hall components of recombination separately. For consistent values of current, the models used in the solution should be specified. The quantity to be plotted has no default.

control

```
Outfile    =  character
NX         =  integer   (default is 256)
NY         =  integer   (default is 256)
FRequency  =  integer   (default is 1)
ABsolute   =  logical   (default is false)
LOgarithm  =  logical   (default is false)
X.compon   =  logical   (default is false)
Y.compon   =  logical   (default is false)
MIx.mater  =  logical   (default is false)
MIN.value  =  real
MAx.value  =  real
No.fill    =  logical   (default is false)
```

OUTFILE is the name of the movie output file which contains data discretized on an NX-by-NY grid. FREQUENCY gives the frequency at which movie files are made; e.g. with FREQUENCY=2, only every other bias point would be saved. ABSOLUTE specifies that the absolute value of the variable be taken. For rapidly varying quantities, the LOGARITHM is often more revealing. Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM - the absolute is taken first and there is no danger of negative arguments. X.COMPON and Y.COMPON take the x and y components of a vector quantity, respectively. MIX.MATER specifies that local vector averaging should be done over all materials to which the node belongs as opposed to just the hierarchical choice (see the PRINT line). MIN.VALUE and MAX.VALUE specify the minimum and maximum values to be plotted. The NO.FILL option will force *PADRE* to draw the device area plotted to scale; if this option is not specified, the plot will fill the screen, and the triangles will appear distorted.

Examples

Start a movie file called VMOVIE that saves the potential distribution in the entire device at every third bias point.

```
  MOVIE   OUTF=VMOVIE POTEN FREQ=3
```

The OPTIONS line

Syntax

```
  Options  run control size plot-control color
```

Description

The OPTIONS line sets options for an entire run.

Parameters

run control

```
  NEws               =  logical    (default is false)
  G.debug or Debug   =  logical    (default is false)
  N.debug            =  logical    (default is false)
  NS.debug           =  logical    (default is false)
  CPUStat            =  logical    (default is false)
  CPUFile            =  character  (default is "padre.cpu")
  MAX.cpu            =  real       (default is -1)
  MOde               =  character  (default is "2.3")
```

NEWS prints news about the current *PADRE* version. G.DEBUG and N.DEBUG print debugging information to standard output. G.DEBUG prints general information, NS.debug prints debugging information regarding noise problems, and N.DEBUG outputs more specific numerical parameters. CPUSTAT prints a cpu profile of the run to a file called CPUFILE. If MAX.CPU >= 0, a maximum cpu time for the run is given in seconds. MODE defines the program mode (defaults, methods, constants) for compatibility with older versions; currently "2.1" and "2.3" are available.

size

```
  SIze.buf  =  real     (default is 3.25e6)
  WEe.size  =  logical  (default is false)
  MEd.size  =  logical  (default is false)
  Big.size  =  logical  (default is false)
  WOw.size  =  logical  (default is false)
  CRush     =  logical  (default is false)
```

```
Lindump    =  logical  (default is false)
NO.trcnt   =  logical  (default is false)
```

SIZE.BUF, WEE.SIZE, MED.SIZE, BIG.SIZE and WOW.SIZE define the size of the internal *PADRE* data buffer. SIZE.BUF is given in 8-byte words; other parameters are pre-defined (WEE=10M, MED=2.45M, BIG=3.25M, WOW=20M). If no parameter is given, the default SIZE.BUF is used; otherwise SIZE.BUF defaults to 0, and the user selection overrides (for >1 options, the largest is used). Nondefault sizes should be supplied before the first non TITLE or COMMENT line. CRUSH uses a compact ascii output format for all mesh and solution files. LINDUMP and NO.TRCNT attempt to save space for machines which have tight memory restrictions. LINDUMP forces *PADRE* to dump all nonessential arrays to scratch files before all linear solves; NO.TRCNT specifies that no transient or continuation steps will be performed, reducing memory required.

plot-control

```
HP2648      =  logical  (default true)
HP2623      =  logical  (default false)
Tek4107     =  logical  (default false)
4014        =  logical  (default false)
Vt240       =  logical  (default false)
PIc         =  logical  (default false)
Splot       =  logical  (default false)
POstscript  =  logical  (default false)
X.Screen    =  real     (default 10 inches)
Y.Screen    =  real     (default  5 inches)
X.Offset    =  real     (default  0 inches)
Y.Offset    =  real     (default  0 inches)
```

The first 8 parameters are used to change the plotting language. Currently, the Hewlett-Packard HP2648 and HP2623 graphics terminals, the Tektronix 4107 color graphics terminal, the Tektronix 4014, the DEC VT240 and the pic and postscript languages are supported. For 1D plots, the SPLOT format is also available. On color terminals, different line types are implemented as different colors; on the black and white monitors dot and line patterns are used. X.SCREEN is the physical width of the screen and Y.SCREEN is the height. They are set automatically depending on the plot device, but can be altered for special effects (e.g. split screen plots). The offset from the bottom left corner of the screen may be set using X.OFFSET and Y.OFFSET.

color

```
COL.gamma  =  real       (default is 1.5)
COLOrfile  =  character
C1.color   =  integer    (default is 6)
C2.color   =  integer    (default is 7)
C3.color   =  integer    (default is 8)
C4.color   =  integer    (default is 9)
C5.color   =  integer    (default is 10)
C6.color   =  integer    (default is 11)
```

```
  C7.color   =  integer   (default is 12)
  C8.color   =  integer   (default is 13)
  C9.color   =  integer   (default is 14)
  C0.color   =  integer   (default is 15)
```

COL.GAMMA is the gamma value used to determine the rgb levels for a rainbow level scheme. COLORFILE gives a list of colors to be used in rgb (one set per line). C1.COLOR, C2.COLOR, etc. specify color types for color plotters which need them (e.g. TEK4107).

Examples

The following sets up a plot for a Tektronix terminal, using a small centered window. Cpu information is also logged to the default file.

```
  OPTIONS   TEK4107 X.S=6 Y.S=5 X.Off=1 Y.OFF=0.5
  +         CPUSTAT
```

The PLOT.1D line

Syntax

```
  PLOT.1d  segment function axes plot-control data-control
```

Description

The PLOT.1D line plots a specific quantity along a line segment through the device (mode A), or plots an I-V curve of data (mode B).

Parameters

segment

```
  X.Start or A.X  =  real
  Y.Start or A.Y  =  real
  X.End or B.X    =  real
  Y.End or B.Y    =  real
  Z.pos           =  real      (default is 0)
  COordinate      =  character
```

The above parameters define the Cartesian coordinates of the start (A.X,A.Y) and the end (B.X,B.Y) of the line segment along which the specified quantity is to be plotted. By default, the data is plotted as a function of distance from the start (A) on the z-plane closest to the z-coordinate given by Z.POS; the data can alternatively be plotted against a particular coordinate ("x", "y" or "z") along the same segment. The line segment may not be defaulted, and it is required in mode A.

function is one of:

```
POTential    =  logical   Mid-gap potential
QFN          =  logical   Electron quasi-fermi level
QFP          =  logical   Hole quasi-fermi level
N.temp       =  logical   Electron temperature
P.temp       =  logical   Hole temperature
DOping       =  logical   Total net impurity concentration
IOn.imp      =  logical   Net ionized impurity concentration
ELectrons    =  logical   Electron concentration
Holes        =  logical   Hole concentration
NET.CHarge   =  logical   Net charge concentration
NET.CArrier  =  logical   Net carrier concentration
J.Conduc     =  logical   Conduction current
J.Electr     =  logical   Electron current
V.Electr     =  logical   Electron velocity
J.Hole       =  logical   Hole current
V.Hole       =  logical   Hole velocity
J.Displa     =  logical   Displacement current
J.Total      =  logical   Total current
E.Field      =  logical   Electric field
REcomb       =  logical   Net recombination
BAND.Val     =  logical   Valence band potential
BAND.Con     =  logical   Conduction band potential
               or
X.Axis       =  character
Y.Axis       =  character
Freq         =  real
INFile       =  character
```

The above parameters specify the quantity to be plotted. There is no default. In mode A, one of the solution variables is plotted versus distance into the device. For vector quantities, the magnitude is plotted. In mode B, terminal characteristics can be plotted against each other by choosing the value to be plotted on each axis (XAXIS=,YAXIS=). Quantities available for plotting include applied device biases (XAXIS/YAXIS=VA1, VA2, ..., VA9, VA0), actual contact bias which may differ from applied bias in the case of lumped element boundary conditions (V1, V2, etc.), terminal current (I1, I2, etc.), AC capacitances (C11, C12, C21, etc.), AC conductance (G11, G12, G21, etc.) and AC admittance (Y11, Y12, Y21, etc.). Also, voltages at circuit nodes can be plotted as "Vnode_name" where "node_name" is the name of the node, and currents for labeled lumped elements can be plotted by name as "Iname" where "name" is the lumped element name (see LUMP.ELEMENT line). Additionally, any of the voltages or currents can be plotted versus time for transient simulations, and any AC quantity can be plotted versus frequency. For plotting AC parameters versus bias at a particular frequency, use the FREQUENCY parameter. The values plotted are the I-V or AC data of the present run, provided a log is being kept (see the LOG line). Alternatively, a different log file can be loaded with INFILE.

axes

```
RIght.axis  =  logical    (default is false)
SHort.axis  =  logical    (default is false)
```

```
MIN.value   =  real
MAx.value   =  real
X.MIn       =  real
X.MAX       =  real
X.Scale     =  real        (default is 1)
Y.Scale     =  real        (default is 1)
UNScale     =  logical     (default is false)
X.Label     =  character
Y.Label     =  character
X.MARk      =  real        (default is 0)
Y.Mark      =  real        (default is 0)
Title       =  logical     (default is true)
```

RIGHT.AXIS plots the y-axis on the right, and SHORT.AXIS shortens the length of the x-axis, which allows the labels for a right-side y-axis to appear. MIN.VALUE and MAX.VALUE specify minimum and maximum values for the ordinate of the graph; their defaults are found automatically from the data to be plotted. X.MIN and X.MAX allow minimum/maximum values for the abscissa to be specified (the defaults are just the min/max abscissa values in the data to be plotted). The x and y values on the plot are multiplied by X.SCALE and Y.SCALE before plotting. By default, some quantities are further scaled before plotting (e.g. time in ms, ps, ns, etc.); UNSCALE prohibits this scaling. X.LABEL and Y.LABEL allow the user to specify his/her own labels for the x and y axes respectively. X.MARK and Y.MARK put dotted lines at x=X.MARK and y=Y.MARK respectively which is sometimes useful if the origin of the plot is not in the lower left corner. TITLE places the current simulation title (see TITLE line) on the plot.

plot-control

```
NO.Clear   =  logical   (default is false)
NO.Axis    =  logical   (default is false)
UNChanged  =  logical   (default is false)
NO.End     =  logical   (default is false)
NO.Order   =  logical   (default is false)
ORder.y    =  logical   (default is false)
UNIque     =  real      (default is 1.0e-6)
POInts     =  logical   (default is false)
NO.Line    =  logical   (default is false)
PAuse      =  logical   (default is false)
LIne.type  =  integer   (default is 1)
OUtfile    =  character
AScii      =  logical   (default is false)
```

NO.CLEAR indicates that the screen is not to be cleared before the current plot so that several curves can be plotted on the same axis. NO.AXIS indicates that the axes for the graph are not to be plotted. UNCHANGED is a synonym for NO.AXIS and NO.CLEAR, but additionally it forces the use of the previous axis bounds so that a number of curves can easily be put on the same axis. *PADRE* by default will order the plot coordinates by abscissa value; this ordering will result in unusual plots for IV curves with negative resistance, for example. NO.END keeps the output file open after the plot is finished so that it can be appended to by a subsequent plot. The NO.ORDER parameter forces *PADRE* to plot the data points as they naturally occur while the ORDER.Y parameter orders by ordinate (y) value. UNIQUE is a relative factor used to determine whether data points coincide and hence can be stripped. POINTS marks the data points on the plotted curve, while NO.LINE specifies that the points not be connected by a line. The PAUSE option causes *PADRE* to stop at the end of the plot so that a hardcopy may be made before continuing. Execution can be resumed by hitting a carriage return. LINE.TYPE specifies the line type for the plotted curve. OUTFILE generates an output file containing the plot in a format to be dumped on a specified terminal screen or plotter. ASCII specifies that this output file have an ascii tabular format.

data-control

```
ABsolute     =  logical  (default is false)
LOgarithm    =  logical  (default is false)
X.Log        =  logical  (default is false)
DEcibels     =  logical  (default is false)
INTegral     =  logical  (default is false)
NEGative     =  logical  (default is false)
INVerse      =  logical  (default is false)
D.order      =  real     (default is 0)
X.Component  =  logical  (default is false)
Y.Component  =  logical  (default is false)
MIx.mater    =  logical  (default is false)  [Expert]
SPline       =  logical  (default is false)
NSpline      =  logical  (default is 100)
```

ABSOLUTE specifies that the absolute value of the variable be taken. For rapidly varying quantities, the LOGARITHM (X.LOG) is often more revealing. Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM - the absolute is taken first and there is no danger of negative arguments. DECIBELS converts the function to a measure of gain in decibels defined by

INTEGRAL plots the integral of the specified ordinate, and NEGATIVE negates the ordinate values; INVERSE plots the value of the function raised to the (-1) power. D.ORDER0 specifies that the derivative of the function (order=D.ORDER) with respect to the ordinate be plotted. X.COMPONENT and Y.COMPONENT force the x and y components respectively of any vector quantities to be plotted as opposed to the default total magnitude. MIX.MATER specifies that local vector averaging should be done over all materials to which the node belongs as opposed to just the hierarchical choice (see the PRINT line). The SPLINE option indicates that spline-smoothing should be performed on the data using NSPLINE interpolated points (maximum is 500).

Examples

The following plots a graph of potential along a straight line from (0.0,0.0) to (5.0,0.0):

```
PLOT.1D  POTEN A.X=0 A.Y=0 B.X=5 B.Y=0
```

In the next example, the log of the electron concentration is plotted from (1.0,-0.5) to (1.0,8.0) with bounds on the plotted electron concentration of 1.0e10 and 1.0e20. A spline interpolation is performed with 300 interpolated points. The non-spline-interpolated points are marked.

```
PLOT.1D  ELECT LOG A.X=1 A.Y=-.5 B.X=1 B.Y=8
+        MIN=10 MAX=20 SPLINE NSPL=300 POINTS
```

In the following example, the current in contact 1 is plotted as a function of contact 2 voltage, then the curve is compared with a previous run.

```
PLOT.1D  X.AXIS=V2 Y.AXIS=I1
PLOT.1D  X.AXIS=V2 Y.AXIS=I1 INF=logf0 UNCH
```

The following plots the actual contact voltage on a contact versus the applied voltage.

```
PLOT.1D  X.AXIS=V3 Y.AXIS=VA3
```

Finally, the following shows a plot of two capacitance components versus the log of frequency. A different line type is chosen for the second component.

```
PLOT.1D  X.AXIS=FREQ Y.AXIS=C21 X.LOG
PLOT.1D  X.AXIS=FREQ Y.AXIS=C31 X.LOG UNCH LINE=4
```

The PLOT.2D line

Syntax

```
PLOT.2d area-definition quantity control linetypes
```

Description

The PLOT.2D line plots quantities in a specified two-dimensional area of the device. A PLOT.2D line is required before performing a contour plot (see CONTOUR line) in order to obtain the plot boundaries.

Parameters

area-definition

```
X.MIn   =  real
X.MAx   =  real
Y.MIn   =  real
Y.MAx   =  real
Z.pos   =  real      (default is 0)
TOp     =  logical   (default is false)
```

```
BOTtom  =  logical  (default is false)
LEft    =  logical  (default is false)
Right   =  logical  (default is false)
```

The above parameters define the rectangular area of the device to be plotted. The default area is a rectangle around the entire device in the xy-plane closest to the z-coordinate specified by Z.POS (in mm). Alternatively one can plot the TOP xz-plane, the BOTTOM xz-plane, the LEFT most yz-plane or the RIGHT most yz-plane.

quantity

```
GRid or Mesh  =  logical  (default is false)
OBtuse        =  logical  (default is false)
CROsses       =  logical  (default is false)
BOUndary      =  logical  (default is false)
Interface     =  integer  (default is 1)
DEpl.edg      =  logical  (default is false)
Junction      =  logical  (default is false)
```

The GRID (or MESH) option plots the grid, including lines delineating elements. OBTUSE colors in all obtuse triangles in the displayed grid. CROSSES plots crosses at the locations of grid points. BOUNDARY indicates that internal and external device boundaries are to be plotted. INTERFACE controls which boundaries are plotted as follows:

```
0    Outer boundaries
1    Outer boundaries and dissimilar material interfaces
2    Outer boundaries and all interfaces
```

DEPL.EDG indicates that depletion edges are to be plotted (note: depletion edges can only be plotted after a solution is present). The JUNCTION option specifies that the junctions from the doping profiles are to be plotted.

control

```
NO.TIc       =  logical  (default is false)
NO.TOp       =  logical  (default is false)
NO.Fill      =  logical  (default is false)
NO.Clear     =  logical  (default is false)
NO.End       =  logical  (default is false)
NO.Diag      =  logical  (default is false)
LAbels       =  logical  (default is false)
TITle        =  logical  (default is false)
Flip.x       =  logical  (default is false)
TILt         =  logical  (default is false)
A.Elevation  =  real     (default is 30)
A.Azimuth    =  real     (default is -30)
Pause        =  logical  (default is false)
SPline       =  logical  (default is false)
NSpline      =  integer  (default is 100)
CRIter       =  real
GEomfile     =  character
```

```
  Outfile      =  character
```

NO.TIC indicates that tic marks are not to be included around the plotted area. NO.TOP indicates that tic marks are not to be put on the top of the plotted region. The NO.FILL option will force *PADRE* to draw the device area plotted to scale; if this option is not specified, the plot will fill the screen, and the triangles will appear distorted. NO.CLEAR specifies that the screen is not to be cleared before plotting. NO.END keeps the output file open after the plot is finished so that it can be appended to by a subsequent plot. NO.DIAG leaves diagonals out of rectangular grids that are plotted using the GRID option. LABELS makes room for color contour labels on the right side of the plot device, and TITLE prints the title of the run (from the TITLE line) at the top of the plot. FLIP.X flips the plot about the y-axis; i.e., it negates all x coordinates so that the plot is mirrored. TILT specifies that the plot be tilted by the angles A.ELEVATION and A.AZIMUTH (in degrees). The PAUSE option causes *PADRE* to stop at the end of the plot so that a hardcopy may be made before continuing. Execution can be resumed by hitting a carriage return. SPLINE indicates that spline interpolation should be used for all contours using NSPLINE points. GEOMFILE specifies the name of a skel format file which will be extracted from each plotted contour (including the junction) using the criterion CRITER; each contour is put in a separate file (last ascii character of GEOMFILE is incremented). These can be used to improve initial grids. OUTFILE generates a binary plotfile which can be dumped directly on a specified screen or plotter.

linetypes

```
  L.Elect  =  integer
  L.Deple  =  integer
  L.Junct  =  integer
  L.Bound  =  integer
  L.Grid   =  integer
  COlor    =  logical  (default is false)
  Grey     =  logical  (default is false)
```

L.ELECT, L.DEPLE, L.JUNCT, L.BOUND and L.GRID set line types for electrodes, depletion edges, junctions, region boundaries and grid, respectively. COLOR specifies color fills, and GREY specifies grey scale fills.

Examples

The following plots the entire grid to scale with tic marks:

```
  PLOT.2D  GRID NO.FILL
```

In the next example, the device and region boundaries, junctions and depletion edges are plotted in the rectangular area bounded by 0 < x < 5um and 0 < y < 10um. The plot is allowed to fill the screen and tic marks are not included along the top of the plot.

```
  PLOT.2D  X.MIN=0 X.MAX=5 Y.MIN=0 Y.MAX=10
  +        JUNCT BOUND DEPL NO.TOP
```

Make a 3d plot of junctions on front face, top and right side. Put all faces in a single output file using NO.END and NO.CLEAR. Note that contours could also be included in 3D plots by inclusion after each respective PLOT.2D line.

```
  PLOT.2    NO.TIC NO.FILL BOUND JUNC OUTF=plt3d     # front
  +         Z.POS=0 TILT TITLE=f L.ELECT=1 NO.END
  PLOT.2    NO.TIC NO.FILL BOUND JUNC OUTF=plt3d     # right
  +         RIGHT TILT NO.CLEAR TITLE=f L.ELECT=1 NO.END
  PLOT.2    NO.TIC NO.FILL BOUND JUNC OUTF=plt3d     # top
  +         TOP TILT NO.CLEAR TITLE=f L.ELECT=1
```

The PLOT.3D line

Syntax

```
  PLOT.3d plotted-quantity control
```

Description

The PLOT.3D line dumps 3D scatter files for use in the external Coughran/Grosse 3D visualization system in Murray Hill. Scatter files can also be converted to the AVS unstructured cell data (UCD) format via the external "scat2avs" utility.

Each PLOT.3D line makes a single file which may have up to 5 quantities over the grid.

Parameters

plotted-quantity is up to 5 of:

```
  POtential    =  logical   Mid-gap potential
  QFN          =  logical   Electron quasi-fermi level
  QFP          =  logical   Hole quasi-fermi level
  N.temp       =  logical   Electron temperature
  P.temp       =  logical   Hole temperature
  BAND.Val     =  logical   Valence band potential
  BAND.Cond    =  logical   Conduction band potential
  DOping       =  logical   Doping
  ELectrons    =  logical   Electron concentration
  Holes        =  logical   Hole concentration
  NET.CHarge   =  logical   Net charge concentration
  NET.CArrier  =  logical   Net carrier concentration
  J.Conduc     =  logical   Conduction current (not yet implemented)
  J.Electr     =  logical   Electron current (not yet implemented)
  V.Electr     =  logical   Electron velocity (not yet implemented)
  J.Hole       =  logical   Hole current (not yet implemented)
  V.Hole       =  logical   Hole velocity (not yet implemented)
  J.Displa     =  logical   Displacement current (not yet implemented)
  J.Total      =  logical   Total current (not yet implemented)
  E.field      =  logical   Electric field
  REComb        =  logical   Net recombination
  FLowlines     =  logical   Current flow lines (not yet implemented)
```

The above parameters specify the quantity to be plotted. For vector quantities the magnitude is plotted unless specific vector components are requested. Model dependent parameters (current and recombination) are calculated with the models currently defined, *not* with the models that were defined when the solution was computed. This allows the display of, for instance, Auger and Shockley-Read-Hall components of recombination separately. For consistent values of current, the models used in the solution should be specified. The quantity to be plotted has no default.

control

```
  Outfile         =  character
  REGion          =  vector
  IGn.region      =  vector
  Semiconductor   =  logical  (default is true)
  INsulator       =  logical  (default is true)
  ABsolute        =  logical  (default is false)
  LOgarithm       =  logical  (default is false)
  X.compon        =  logical  (default is false)
  Y.compon        =  logical  (default is false)
  Z.compon        =  logical  (default is false)
  MIx.mater       =  logical  (default is false)
```

OUTFILE is the name of the scatter plot file. REGION gives a list of regions to be included; the default is every region. IGN.REGION gives a list of regions to be ignored; the default is none. SEMICONDUCTOR includes semiconductor regions, while INSULATOR includes insulator regions; setting either of these false excludes these regions from the plot. ABSOLUTE specifies that the absolute value of the variable be taken. For rapidly varying quantities, the LOGARITHM is often more revealing. Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM - the absolute is taken first and there is no danger of negative arguments. X.COMPON, Y.COMPON and Z.COMPON request the x, y and z components of a vector quantity, respectively. MIX.MATER specifies that local vector averaging should be done over all materials to which the node belongs as opposed to just the hierarchical choice (see the PRINT line).

Examples

The following line dumps a scatter plot of the electrostatic potential, electron density and hole density at the last bias point solved into a file called "plt.wmc".

```
  PLOT.3D   POTEN ELECT HOLE OUTFILE=plt.wmc
```

The following line creates a scatter plot file with three quantities, the potential, the x-component of the electric field, and the y-component of the electric field.

```
PLOT.3D   POTEN E.FIELD X.COMP Y.COMP OUTFILE=plt.field
```

The PRINT line

Syntax

```
PRint location quantity flags
```

Description

The PRINT line prints specific quantities at points within a defined area of the device.

Parameters

location

```
Region   =  vector  (default is all regions)
Z.MIn    =  real    (default is min(z) in device)
Z.MAx    =  real    (default is max(z) in device)
        and
X.MIn    =  real
X.MAx    =  real
Y.MIn    =  real
Y.MAx    =  real
         or
IX.Low   =  integer
IX.High  =  integer
IY.Low   =  integer
IY.High  =  integer
```

The above parameters define area in which the points of interest lie. The default area is the entire device. REGION gives an optional vector of region numbers. In the z-dimension, extents can be defined by coordinates, Z.MIN and Z.MAX (in mm). In the x and y dimensions, extents can be defined by physical coordinates X.MIN, X.MAX, Y.MIN and Y.MAX, or by the bounding indicies IX.LOW, IX.HIGH, IY.LOW and IY.HIGH (valid only for rectangular meshes).

quantity

```
POints    =  logical
Elements  =  logical
Geometry  =  logical
Solution  =  logical
P.SOL1    =  logical
P.SOL2    =  logical
Current   =  logical
```

```
P.CURR1   =  logical
P.CURR2   =  logical
QUe       =  logical
P.QUE1    =  logical
P.QUE2    =  logical
Jcomp     =  logical
P.Jcomp1  =  logical
P.Jcomp2  =  logical
MObility  =  logical
MAterial  =  logical
```

The above parameters specify the quantities to be plotted. Any or all may be specified, and each defaults to false. POINTS prints node information (coordinates, doping, etc.). ELEMENTS prints information on the triangular elements (number, nodes, material). GEOMETRY prints geometrical information on the triangles. SOLUTION prints the *present* solution (psi, n, p and carrier temperatures or quasi-fermi potentials - see below), while P.SOL1 and P.SOL2 print the *previous* two solutions. CURRENT prints currents (electron, hole, conduction, displacement and total) at each node for the present solution; P.CURR1 and P.CURR2 print currents for previous solutions. QUE prints space charge, recombination and electric field for the present solution; P.QUE1 and P.QUE2 print the same quantities for the previous two solutions. JCOMP prints total mobility and current components - drift, diffusion, etc. - estimated by taking centered differences (rather than Scharfetter-Gummel, so these may not add to the same values as elsewhere). MOBILITY prints mobility components as defined by activated scattering mechanisms at each point. MATERIAL prints material information (permittivity, band-gap, etc.), including the value of the concentration dependent mobility and lifetime (if specified) at each point.

flags

```
X.Component  =  logical  (default is false)
Y.Component  =  logical  (default is false)
Z.Component  =  logical  (default is false)
QF           =  logical  (default is false)
ELectrons    =  logical  (default is true)
Holes        =  logical  (default is true)
Velocity     =  logical  (default is false)
No.order     =  logical  (default is false)
MIx.mater    =  logical  (default is false)  [Expert]
```

X.COMPONENT, Y.COMPONENT and Z.COMPONENT specify how any of the various vector quantities (currents, fields) should be printed. The default is the magnitude of the vector as a whole. X.COMPONENT specifies that the magnitude of the x-component of all vectors be printed; Y.COMPONENT specifies the y-component; and Z.COMPONENT specifies the z-component (assuming the structure is 3D). Only one (or none) of these can be specified on a single line. By default, the SOLUTION parameter above will print carrier temperatures; the QF parameter can be used to replace the temperature columns with respective quasi-Fermi potentials. ELECTRONS and HOLES specify the carriers used when printing current and mobility components. VELOCITY scales all local currents by the local carrier density, thus printing velocities rather than currents where appropriate. Inclusion of NO.ORDER will list grid point data in the order used in the internal data structures, rather than the default geometric (left-to-right, top-to-bottom) ordering. MIX.MATER specifies that local vector averaging should be done over all materials to which the node belongs as opposed to just the hierarchical choice (given on the output to the POINTS option).

Examples

The following prints the physical coordinates, doping and region/electrode information for points along the 10th x grid line, from the 1st to the 20th y grid lines.

```
  PRINT  POINTS IX.LO=10 IX.HI=10 IY.LO=1 IY.HI=20
```

In the next example, solution information is printed for 0 < x < 1mm and 0 < y < 2mm.

```
  PRINT  SOLUTION X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=2
```

The P.TRACK line

Syntax

```
  P.track  location form density/radius integrate time
```

Description

The P.TRACK line defines a particle track which can subsequently be activated on a transient simulation sequence (see the "seu" option on the SOLVE line).

Parameters

location

```
  ORIgin    =  vector  (default is [0,0,0])
  ENdpoint  =  vector  (default is [0,0,0])
  TRack     =  vector  (default is [0,0,0])
```

ORIGIN defines the starting [x,y,z] coordinates of the particle track. The remainder of the track can be defined using the ENDPOINT - again as [x,y,z] - or by TRACK which is given as [r, theta, phi] where r is a depth (in mm) and theta, phi are strike angles to the yz- and xz-planes.

form

```
  Uniform      =  logical  (default is false)
  Gaussian     =  logical  (default is false)
  Power        =  logical  (default is false)
  CUt.power    =  logical  (default is false)
  RAnge.power  =  real     (default is infinity)
```

The form of the generation function generated by the particle track is given by one of the parameters UNIFORM, GAUSSIAN, POWER or CUT.POWER; if none of the options are specified, the default is taken as UNIFORM. All the distributions have radial symmetry about the track. The radius of the uniform profile or characteristic length of the gaussian can vary along the track, as described below. The power distribution decays from the track as (1+|r|/sigma)**-n where n and sigma are defined below. CUT.POWER is a power distribution whose extent radially ends at RANGE.POWER (in mm).

density/radius

```
  Density    =  real
  Volume     =  logical  (default is false)
  Radius     =  real
  SIgma      =  real
  EXp.power  =  real
  TAble      =  character
```

The density and radius/sigma along the track can be defined either: (1) as constants using DENSITY, RADIUS, SIGMA or (2) by a table of (distance,density,radius/sigma) values in a file called TABLE. The density units are taken as linear densities (eh-pairs/mm) if VOLUME=false, or local volume densities (eh-pairs/cm**3) if VOLUME=true. Distances are in mm. For power distributions, the exponent n (see above) is defined by EXP.power or as a fourth column in the input file, if TABLE is specified.

integrate

```
  ORDer     =  integer  (default is 5)
  Level     =  integer  (default is 0)
  Conserve  =  logical  (default is true)
  E.thresh  =  real     (default is 0.01)
```

ORDER is the quadrature order applied to the integration of the charge over control volume pairs in each element; the highest order implemented is 5. For more accuracy, the control volume subsections can be uniformly refined LEVEL times. The CONSERVE option scales the specified density to assure the the total dose matches the exact integral of the specified radial function; with CONSERVE off, the dose may not be preserved on coarse grids due to discretization error. The exact and numerically integrated rates are always printed in the output file (the scale factor is taken from these rates if CONSERVE is on). E.THRESH defines a threshold (as a fraction of the total dose) generation at electrodes which will induce an error; this charge will by default be assigned over the remainder of the track.

time

```
  TIme  =  real vector
```

By default, the charge generated by the particle track is distributed equally (square pulse) over an intervals specified on SOLVE lines. TIME gives the decay rate tau and peak tp (in sec) for a quasi-Gaussian distribution in time:

As with the pulse, the Gaussian is included on all SOLVE lines where SEU is turned on.

Examples

Define a particle track of length 15mm with a constant linear density of 1.8E6 e-h pairs/mm. The charge will be distributed uniformly outward from the track to a radius of 0.05mm. In time, the charge is generated as a one-sided Gaussian with a decay rate of 30ps and a peak at t=10ps.

```
  P.TRACK   UNIF RADIUS=.05 DENSITY=1.8e6 ORIGIN=0,0,0
  +         END=0,15,0  TIME=30e-12,10e-12
```

Define a particle track of length 10mm which has a Gaussian distribution laterally in space with density and radius along the track specified in the file "gentbl". The charge will be generated as a pulse in time, with pulse width determined on the SOLVE line. Extra accuracy in the integration is achieved by using 2 levels of refinement of the control volumes.

```
  P.TRACK   GAUSS TABLE=gentbl LEVEL=2 ORIGIN=0,0,0
  +         END=5.7735,5.7735,5.7735
```

The REGION line

Syntax

```
  REGIon number position material
```

Description

The REGION line defines material names and/or geometric extents of regions in a mesh. Every element must be defined to be some material.

Parameters

number

```
  NUmber  =  integer
  NEwnum  =  integer
```

NUMBER selects a given region by number. NEWNUM defines this region (or subregion) to have a new region number; this feature is most useful to define nonuniformities in 3D regions (see example below).

position

```
IX.Low    =  integer
IX.High   =  integer
IY.Low    =  integer
IY.High   =  integer
X.Min     =  real    (default is min(x) in device)
X.Max     =  real    (default is max(x) in device)
Y.Min     =  real    (default is min(y) in device)
Y.Max     =  real    (default is max(y) in device)
Z.Min     =  real    (default is min(z) in device)
Z.Max     =  real    (default is max(z) in device)
```

These parameters give a location within a domain where a region is to be defined. IX.LOW, IX.HIGH, IY.LOW and IY.HIGH are the indices of a box in a rectangular mesh. X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN, Z.MAX are limits given by physical coordinates. For extents in x and y, one must choose either the index limits or the physical limits; for the z extent, only the physical limits are permitted.

material

```
NAme or Material   =  character
SEmiconductor      =  logical   (default is false)
INsulator          =  logical   (default is false)
  or
SILicon            =  logical   (default is false)
GAas               =  logical   (default is false)
GErmanium          =  logical   (default is false)
SEmiconductor      =  logical   (default is false)
Oxide or SIO2      =  logical   (default is false)
NItride or SI3n4   =  logical   (default is false)
SApphire           =  logical   (default is false)
INsulator          =  logical   (default is false)
```

The first set of parameters give the accepted way of defining material types in the current version of *PADRE*. NAME (or MATERIAL) defines the material name for a given region. The name may correspond to a material *PADRE* knows about (such as "silicon" for Si, "gaas" for GaAs, "germanium" for Ge, "sio2" for SiO2, "si3n4" for nitride, "sapphire", "poly" for polysilicon, "aga45" for AlGaAs [Al/(Al+Ga)=.45], "iga53" for InGaAs [In/(In+Ga)=.53]) or a new material which will be defined on a later MATERIAL line. SEMICONDUCTOR defines the material as a semiconductor, meaning that carrier transport will be explicitly computed within these regions. INSULATOR defines the material as a pure insulator with zero carrier densities; only displacement currents are thus accounted for in these regions.

The second set of (logical) parameters are included for compatibility with older versions. Only one of these parameters can be given. SILICON, GAAS, GERMANIUM and SEMICONDUCTOR define different semiconductors (SEMICONDUCTOR defines a generic type); OXIDE, NITRIDE, SAPPHIRE and INSULATOR are different insulators.

Examples

The following defines a semiconductor silicon region extending from nodes 1 to 25 in the x direction and nodes 4 to 20 in the y direction :

```
  REGION  NUM=1 IX.LO=1 IX.HI=25 IY.LO=4 IY.HI=20 NAME=SILICON SEMI
```

Note that region lines are cumulative in effect so that the following defines one insulating oxide region comprised of two separate strips.

```
  REGION  NUM=1 IX.LO=4  IX.HI=5  IY.LO=1 IY.HI=20 NAME=OXIDE INS
  REGION  NUM=1 IX.LO=36 IX.HI=37 IY.LO=1 IY.HI=40 NAME=OXIDE INS
```

Define a region number 1 as a semiconductor material called "base". Redefine that part of the region 1 extending beyond z=8.4 to be region 2 which is given oxide as a material type.

```
  REGION  NUM=1 SEMI MATER=base
  REGION  NUM=1 NEW=2 Z.MIN=8.4 INSU NAME=sio2
```

The REGRID/ADAPT line

Syntax

```
  REGRid  location i/o variable criterion control interpolation
  ADapt   location i/o variable criterion control interpolation
```

Description

The REGRID and ADAPT lines allows refinement or coarsening (unrefining) of a mesh. Any element across which the chosen variable changes by more than a specified tolerance, or in which the chosen variable exceeds a given value, is refined subject to some additional constraints. Conversely coarsening (unrefinement) takes place when a chosen variable changes by less than a specified tolerance or a chosen variable is below a given value. The REGRID line performs the specified grid modification only once whereas the ADAPT line automatically attempts grid adaption at any newly acquired operating point after its occurence, repeating the associated calculation until an adequate grid is achieved.

Parameters

location

```
X.MIn       =  real
X.MAx       =  real
Y.MIn       =  real
Y.MAx       =  real
Z.MIn       =  real
Z.MAx       =  real
REGion      =  integer  (default:all)
IGnore      =  integer  (default:none)
BOX.refine  =  logical  (default is false)
```

The bounds *.MIN and *.MAX are used to limit the regrid; only elements which have nodes which fall inside the box are considered for refinement or coarsening. The REGION parameter has a similar use; only regions specified are adjusted according to the user criterion. (Others may be refined/coarsened as a side effect, to maintain well-shaped elements). The default is to regrid all regions for potential and electric field types, and all semiconductor regions for regrids which depend on the other variables. The parameter IGNORE is similar to REGION, but opposite in effect. Ignored regions are not regrided either according to the user criterion or according to the "obtuse criterion" (see below); nor are they smoothed after regrid. The default is not to ignore any region. BOX.REFINE specifies that regrids should take place inside purely rectangular regions as generated by tri; typically one does not want to regrid these regions as they can have highly anisotropic elements.

i/o

```
OUTFile    =  filename
OUT.green  =  filename
IN.green   =  filename
NO.green   =  logical   (default is false)
DOPFile    =  filename
AScii      =  logical   (default is true)
STats      =  logical   (default is false)
```

OUTFILE is the binary output mesh file, and is necessary if the mesh is to be used for subsequent runs. A history of the triangle tree is always generated to assist further regriding steps. Its name can be specified by OUT.GREEN, and its default is generated from OUTFILE by concatenating the letters "tt" to the end. No such file will be created if NO.GREEN is specified. Additionally, a triangle tree for the previous mesh (if a tree exists) is used for this regrid. By default, *PADRE* will look for a file with the same name as the current mesh plus "tt" at the end as above. Alternatively, IN.GREEN can be used to implement a different file name. DOPFILE is the name of a file (up to 20 characters) which contains the doping for the device (see DOPING line). Specifying DOPFILE avoids interpolating doping values at any newly created grid points (the default), by using the initial doping specification to redope the structure. ASCII specifies that all mesh files and triangle trees (not DOPFILE) for this line should be done in ascii rather than the default - binary. STATS prints some refinement statistics in the standard *PADRE* output file.

variable is one of:

```
Potential  =  logical   Mid-gap potential (V)
QFN        =  logical   Electron quasi-fermi level (V)
QFP        =  logical   Hole quasi-fermi level (V)
N.temp     =  logical   Electron temperature (eV)
P.temp     =  logical   Hole temperature (eV)
DOping     =  logical   Total net impurity concentration (/cm**3)
IOn.imp    =  logical   Net ionized impurity concentration (/cm**3)
ELEctron   =  logical   Electron concentration (/cm**3)
Hole       =  logical   Hole concentration (/cm**3)
NET.CHrg   =  logical   Net charge (/cm**3)
NET.CArr   =  logical   Net carrier concentration (/cm**3)
MIn.carr   =  logical   Minority carrier concentration (/cm**3)
P.track    =  logical   Carrier density from particle tracks (/cm**3)
HEtero     =  logical   Material interfaces (unitless)
ERror      =  logical   Potential error estimate (V)
```

This parameter selects the discriminatory variable. For ADAPT input lines only the default is ERROR.

criterion

```
R.STep       =  real
C.STep       =  real
CHange       =  logical
RELAtive     =  logical  (default is false)
DV.min       =  real     (default is Poisson x.tol)
R.Threshold  =  real     (default is r.step)
N.Threshold  =  real     (default is 200)
F.Threshold  =  real     (default is 0.1)
REfine       =  logical  (default is false)
COarsen      =  logical  (default is false)
LOCaldop     =  logical  (default is false)
LOGarithm    =  logical  (default is false)
ABsolute     =  logical  (default is false)
```

R.STEP is the actual numerical criterion for performing a element refinement while C.STEP is for coarsening (unrefining). If either the magnitude or change (as specified by CHANGE, which defaults to true unless LOCALDOP is set) in a variable exceeds R.STEP, refinement will take place subject to constraints on element size and level as discussed below; conversely, if the value for this element and its brothers is below C.STEP, coarsening (unrefinement) takes place. RELATIVE specifies that the tolerance be applied in a relative sense; e.g. with a voltage-based criteria, the values of R.STEP and C.STEP are taken as fractions of the maximum potential difference inside the device. To avoid overflow for RELATIVE comparisons to flat potential solutions, a minimum voltage differential is defined by DV.MIN; the default is the Poisson solution tolerance (see METHOD line), as the nonlinear solution is only accurate to this value. Unitless variables (HETERO) do not require criteria, and only refinement can take place there, subject to level and element size control.

The first set of constraints, particularly useful for the ADAPT line, are R.THRESHOLD, F.THRESHOLD and N.THRESHOLD which prohibit any grid modification regardless of R.STEP/C.STEP unless either (1) the largest deviation exceeds the value given by R.THRESHOLD or (2) the number of elements to be coarsened exceeds the absolute quantity N.THRESHOLD or the fraction F.THRESHOLD of the starting number of elements. If only R.STEP is given (typical for a REGRID line), R.THRESHOLD defaults to the same value as R.STEP while if only R.THRESHOLD is given (typical for ADAPT), R.STEP defaults to the 0.95*R.THRESHOLD. N.THRESHOLD defaults to 200 or 10% of the total number of elements in the grid, which ever is larger.

If neither R.STEP, R.THRESHOLD or C.STEP is specified, *PADRE* looks at the REFINE and COARSEN flags to determine which operation is to take place; in this case, the grid will be refined or coarsened until the level and size limits below are met. The parameter LOCALDOP compares the magnitude (default) or difference of a variable against the local doping values; this might for instance be used in conjunction with MIN.CARR to refine in areas within the device that are in high-level injection. LOGARITHM and ABSOLUTE specify that the logarithm and absolute value of the variable be used respectively (with LOGARITHM set, R.STEP/C.STEP will be interpreted as the step in the logarithm). Since many of the quantities may become negative, *PADRE* actually uses

to avoid overflow. To get the true logarithm of a quantity, specify ABSOLUTE and LOGARITHM (absolute is taken first).

control

```
  MAx.level    =  integer    (default is dynamic)
  REL.level    =  integer    (default is 1)
  LEv.ignore   =  logical    (default is false)
  HMin         =  real       (default is 0)
  HDir         =  logical    (default is true)
  DEbye        =  real       (default is 0)
  FReeze       =  logical    (default is false)
  SMooth.k     =  integer    (default is 1)
  IT.Smooth    =  integer    (default is 30)
  COs.ang      =  real       (default is 2.0)      [Expert]
  BOUnd.green  =  logical    (default is true)     [Expert]
  G.Semi       =  integer    (default is 1)        [Expert]
  G.Ins        =  integer    (default is 1)        [Expert]
  COndense     =  character  (default is "all")
  REOrder      =  character  (default is false)
  3d.refine    =  logical    (default is false)
  DZ.level     =  integer    (default is 1)
  OFf          =  logical    (default is false)
  IT.Resolve   =  integer    (default is 2)
  N.Resolve    =  integer    (default is 1.0e-3)
```

MAX.LEVEL is the maximum level any element can reach for a given REGRID line relative to the *initial, unrefined* mesh; The default is to allow MAX.LEVEL to increase to one more than the maximum level in the preceding grid, but can be set to a smaller value to limit refinement. Separately, the relative change in each element's level vis a vis the preceding mesh can be specified using REL.LEVEL. Alternatively, LEV.IGNORE ignores the level limits and proceeds until the criteria are met; LEV.IGNORE can thus be viewed as setting MAX.LEVEL and REL.LEVEL to infinity. Regardless of the discriminatory variable's value, a refinement will not take place if an edge is created that is smaller than HMIN or the product of DEBYE and the local Debye length; a coarsening will not take place beyond level 0. If HDIR=true, only element edges which meet the refinement criteria are checked for length against HMIN and/or DEBYE. FREEZE is a parameter which will prohibit unrefining from the current level. SMOOTH.K and IT.SMOOTH are mesh smoothing options, and CONDENSE is the condense region key, both as described for the MESH line. REORDER reorders the grid nodes and elements after adaption.

COS.ANGLE defines the "obtuse criterion" to limit the creation of obtuse angles in the mesh. If regrid would create a element with an angle whose cos is less than -COS.ANGLE, nodes are added so that this does not occur. The test can be turned off locally by using the ignore parameter; it can be turned off everywhere by using a value of COS.ANG greater than 1. The default is to turn it off everywhere. Similarly, the BOUND.GREEN parameter prevents obtuse elements from being created at a boundary or material interface. G.SEMI and G.INS define the maximum levels of green refinement.

In addition to refinement of triangle faces in each xy-grid plane, 3D.REFINE permits refinement of prisms (adding grid planes) in the z direction. DZ.LEVEL is the maximum number of levels of z-refinement that can be performed relative to the initial state.

IT.RESOLVE, N.RESOLVE and OFF are parameters specific to the ADAPT line. IT.RESOLVE is the maximum number of resolve (readapt) attempts per bias point. N.RESOLVE is the minimum fraction of elements that must be refined in order for a subsequent resolve to be attempted. OFF turns off the effect of a previous ADAPT line; no more adaptive refinement will be attempted after the occurence of a line with OFF specified as true.

interpolation

```
  FEm         =  logical  (default is true)
  DOp.int     =  logical  (default is false)  [Expert]
  1d.integ    =  logical  (default is false)  [Expert]
  DI.Degree   =  real     (default is 3)      [Expert]
  DI.Level    =  integer  (default is 0)      [Expert]
```

The FEM parameter specifies that basis functions equivalents be used to interpolate concentration at new nodes, rather than simple exponents. The parameters DOP.INT, 1D.INTEG, DI.DEGREE and DI.LEVEL refer to doping integration control; see the MESH input line documentation.

Examples

Starting with an initial grid, we refine twice, requesting that all elements with large doping steps be refined:

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid1 DOPF=dopxx1
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1
```

A similar effect could be obtained with just one REGRID line below. In both cases two levels of refinement are done. The first choice is preferable because new doping information is introduced at each level.

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1 MAX.LEVEL=2
```

A better strategy is to use the LEVEL and HMIN parameters, repeating the refinement over and over until on each element either (1) the change in doping is less than 6 orders of magnitude or (2) another refinement will produce a grid spacing less than 0.01mm.

```
  REGRID  LOG DOPING R.STEP=6 OUTF=grid2 DOPF=dopxx1
  +       LEV.IGN HMIN=.01
```

A dynamic grid is set up by supplying an ADAPT line. At every bias point solved after this line, the discretization error on the grid will be tested. The refine/unrefine criteria are 9.5mV (from R.THRESHOLD) and 1mV, and the process will be initiated if (1) the potential error exceeds 10mV or (2) the number of unrefinable elements exceeds 500. The process is repeated until these criteria are met, and the final grid is saved as "rmesha" where the terminating "a" is incremented for subsequent meshes.

```
  ADAPT  ERROR R.THRESH=0.01 C.STEP=.001 N.THRESH=500 OUTF=rmesha
```

Regrid on generated carrier density from a particle track. The refinement is performed in all three dimensions if the density changes by more than 1.0e17/cm**3 over any edge in an element.

```
  REGRID   3D P.TR R.STEP=1e17 OUTF=rgrid DOPF=dopfile LEV.IGN
```

The SOLVE line

Syntax

```
  SOlve estimate gen bias transient ac i/o
```

Description

The SOLVE line instructs *PADRE* to perform a solution for one or more specified bias points.

Parameters

estimate

```
  INitial   =  logical  (default is false)
  PREvious  =  logical  (default is true)
  PROject   =  logical  (default is false)
  EUler     =  logical  (default is false)
  LOcal     =  logical  (default is false)
```

The above parameters are used to specify how the initial guess for the solution is to be obtained. The first bias point for a given structure must have the INITIAL parameter specified. From then *PADRE* will use the previous solution (PREVIOUS) by default. If there are two previous solutions present and equivalent bias steps are taken on any electrodes that are changed, an extrapolation (PROJECT) from the preceding two solutions can be used to sometimes get an improved initial guess, which is particularly good for IV curves that are being traced to a fine resolution. If a full Jacobian is available, another (single point) projected guess is EULER which also works quite well for fine IV curve tracing. LOCAL is a special, off-state guess which guesses local values for quasi-Fermi levels based on bias which is typically optimal for for large steps in reverse bias.

bias

```
  V1         =  real
  I1         =  real
  V2         =  real
  I2         =  real
   .
   .
   .
  V9         =  real
  I9         =  real
  V0         =  real
  I0         =  real
  VStep      =  real     (default is 0.0)
  IStep      =  real     (default is 0.0)
  NSteps     =  integer  (default is 0)
  MULTIply   =  logical  (default is false)
  ELectrode  =  integer
  N.bias     =  real
  P.bias     =  real
  CH.mult    =  real     (default is 1.0)
```

The parameters V1, V2, ..., V9, V0 represent the bias voltages (for non-current boundaries only) and I1, I2, ..., I9, I0 represent the terminal currents in units of Amps/um (for current boundary contacts only - see CONTACT line) applied at contacts 1, 2, ..., 9,

0. The defaults for these parameters are the potentials (currents) from the previous bias point. Note that for the INITIAL bias point, 0 volts will be assumed for any voltage that is not specified. VSTEP (ISTEP) is a voltage (current) increment to be added (or multiplied if MULTIPLY is set) to one or more electrodes, as specified by the integer assigned to ELECTRODE. If more than one electrode is to be stepped, ELECTRODE should then be an n-digit integer, where each of the n-digits is a separate electrode number (and if there are 10 electrodes, don't put electrode 0 first in the sequence!). NSTEPS is the number of bias increments (steps) to be taken; i.e., if VSTEP (ISTEP) is specified, the specified electrode is incremented NSTEPS times. N.BIAS and P.BIAS specify fixed quasi-Fermi potentials for carriers (electrons and holes, respectively) that are not being solved for. If N.BIAS or P.BIAS are not specified, then *PADRE* either will choose local quasi-Fermi potentials based on bias and doping if the FIX.QF parameter is set on the METHOD line, will set the quasi-Fermi levels where applicable to values which produce the least amount of free carriers (maximum bias for electrons and minimum bias for holes). CH.MULT is a multiplicative factor used to adjust all Helmoholtz coefficients (see C.HELM on the MATERIAL line) in the device. *gen*

```
  Generation  =  real       (default is 0)
  DOse.rad    =  real       (default is 0)
  ABsorption  =  real       (default is 0)
  DIr.gen     =  character  (default is "y")
  PK.gen      =  real       (default is 0)
  REg.gen     =  vector
```

GENERATE and DOSE.RAD specify blanket radiation terms; the superposition of the two is applied. GENERATE is given in units /s-cm**3 while DOSE.RAD is specified in *rad*; the corresponding generation rate is obtained from DOSE.RAD multiplication with the material generation constant (see GEN.CON on the MATERIAL line). ABSORPTION defines an absorption coefficient for the radiation (in /um). If an absoption coefficient is specified, DIR.GEN defines the coordinate axis along which the radiation penetrates, and PK.GEN defines the position along that axis where the peak generation rate occurs; the rate therefore falls of exponentially, according to teh value of ABSORPTION, away from PK.GEN in both directions along the DIR.GEN axis. Finally, by default the generation terms are applied to all regions; specific regions can be targeted using REG.GEN.

transient

```
  TSTEp        =  real     (default is 0)
  TSTOp        =  real
  TDelta       =  real
  TEnd.refine  =  logical  (default is false)
  RAmptime     =  real     (default is 0)
  ENdramp      =  real     (default is 0)
  SEu          =  logical  (default is false)
  DT.seu       =  real
  G.tau        =  real
```

TSTEP is the time-step to be taken. For automatic time-step runs (see the METHOD line), TSTEP is used to select the first time step only. TSTOP specifies the end of the time interval to be simulated so that if the simulation begins at t = t0, it will end at t = TSTOP. TDELTA defines a stopping time relative to the start of the execution of the line it occurs; i.e., the simulation will stop at at t = t0+TDELTA. If equal time-steps are desired, set *TAuto=false* in the *method* command line. TEND.REFINE refines the timestep as it approaches the end of a transient sequence using DELTA t = (DELTA t(end)) / iceil(epsilon + DELTA t(end) / DELTA t) where DELTA t(end) = TSTOP - t.

RAMPTIME and ENDRAMP apply any bias changes as linear ramps. RAMPTIME specifies a ramp interval in seconds; i.e., the ramp will begin at t = t0 and end at t = t0 + RAMPTIME. Alternatively, ENDRAMP specifies the exact end of the ramp in running time; i.e. the ramp will start at t = t0 and end at t = ENDRAMP.

The SEU flag activates single event upset simulation mode, defined by previous P.TRACK lines. For those particle tracks which have been given a time dependent generation rate on the respective P.TRACK lines, t(gen) = 0 corresponds to t0. Without defined time dependencies, the track charge is generated as a pulse over DT.SEU; the default for DT.SEU is determined by TSTOP (DT.SEU=TSTOP-t0) or TDELTA (DT.SEU=TDELTA) depending on which is set. G.TAU defines the rise or fall time for the applied blanket radiation rate (GENERATION or DOSE.RAD) in sec; the default is to apply or remove the generation immediately.

ac

```
AC.analysis  =  logical   (default is false)
FRequency    =  real
FStep        =  real      (default is 0)
MULT.freq    =  logical   (default is false)
NFsteps      =  integer   (default is 0)
VSs          =  real      (default is 0.1*kT/q)
TErminal     =  integer   (default is all)
S.omega      =  real      (default is 1.0)
MAx.inner    =  integer   (default is 25)
TOlerance    =  real      (default is 1.0e-5)
```

AC.ANALYSIS indicates AC sinusoidal small-signal analysis be performed after the DC condition is solved. FREQUENCY is the frequency (in Hz) at which the analysis is performed. The analysis can be repeated at a number of frequencies (without resolving DC) using FSTEP which is a frequency increment, added to the previous frequency by default or multiplied by setting MULT.FREQ. NFSTEPS defines the number of increments. VSS is the magnitude of the applied small-signal bias; TERMINAL is the contact to which AC bias is applied. More than one contact number may be specified (via concatenation), but each will be solved separately. Each contact that is specified yields a column of the admittance matrix as defined by equation (2.17). If the sor ac method is used (see METHOD line), S.OMEGA is the relaxation parameter, MAX.INNER is the maximum number of sor iterations and TOLERANCE is the sor convergence criterion.

noise analysis

```
  noise.anal   =  logical   (default is false)
  i.noise      =  logical   (default is false)
  e.noise      =  integer   (default is false)
```

NOISE.ANALYSIS indicates that the user wants to compute the noise parameters for a multiterminal device. PADRE can compute both diffusion noise (including nonlinear processes) as well as the generation-recombination noise based on the specification in the model line. Noise.analysis is performed after both DC and AC conditions are solved for. The frequency (in Hz) spacing of the AC solutions are preserved. I.NOISE=t specifies that ac short-circuit current noise parameters are desired, I.NOISE=f specifies that ac open-circuit voltage noise parameters are desired. Also, see the OPTIONS, MATERIAL, and MODEL lines. E.NOISE is the electrode which is ac-grounded.

i/o

```
  Outfile    =  filename
  Currents   =  logical   (default is true)
  NO.append  =  logical   (default is false)
  AScii      =  logical   (default is true)
  SAve       =  integer   (default is 1)
  T.SAve     =  vector
  EFile      =  filename                          [Expert]
  AFile      =  filename                          [Expert]
  Ufile      =  filename                          [Expert]
  Nfile      =  character
  MOstrans   =  integer   (default is 0)
```

OUTFILE optionally specifies the name of the output file for the solution of this bias point. The file names may contain up to 20 characters. If an electrode is stepped so that more than one solution is generated by this line and NO.APPEND is not set, the last non-blank character of the supplied file name will have its ascii code incremented by one for each bias point in succession, resulting in a unique file per bias point. If CURRENTS is specified, the electron, hole, and displacement currents, and the electric field, will be computed and stored with the solution. IF ASCII is specified, OUTFILE will be ascii as opposed to binary. By default the solutions are saved at every bias point. Alternatively a frequency of saves can be specified by SAVE, i.e. if SAVE=2, solutions will be saved only for every other bias point. Alternately T.SAVE gives a list of times at which solutions will be written for transient simulations. EFILE, AFILE and UFILE are output files containing the nonlinear error/residual, Jacobian matrix and update at every Newton iteration, used for the purposes of debugging. NFILE contains noise parameters bewteen various pairs of terminals. MOSTRANS0 defines the gate of a MOSFET for which static estimates of transconductance, capacitance and cutoff frequency will be printed based on the current and previous solution values.

Examples

The following performs an initial bias point, saving the solution to the data file OUT0:

```
SOLVE  INIT OUTF=OUT0
```

In the next example, bias stepping is illustrated. The two solve lines produce the following bias conditions:

Bias point #	V1	V2	V3
1	0.0	0.5	-0.5
2	1.0	0.5	0.0
3	2.0	0.5	0.0
4	3.0	0.5	0.0
5	4.0	0.5	0.0
6	5.0	0.5	0.0

The solutions for these bias points will be saved to the files OUT1, OUTA, OUTB, OUTC, OUTD and OUTE. Note that the initial guess for the first bias point is obtained directly from the preceding solution because the PREVIOUS option was specified. The initial guesses for bias points 2 and 3 will also be obtained as if PREVIOUS had been specified since two electrodes (numbers 1 and 3) had their biases changed on bias point 2. However, for bias points 4, 5 and 6, *PADRE* will use a projection to obtain an initial guess since starting with bias point 4, both of its preceding solutions (bias points 2 and 3) only had the same electrode bias (number 1) altered.

```
SOLVE  PREV V1=0 V2=.5 V3=-.5 OUTF=OUT1
SOLVE  PROJ V1=1 V2=.5 V3=0 VSTEP=1 NSTEPS=4
+      ELECT=1 OUTF=OUTA
```

Here is a case where two electrodes are stepped (2 and 3). The bias points solved for will be (0,0,1), (0,.5,1.5), (0,1,2) and (0,2,3). *PADRE* will use the PROJECT option to predict an initial guess for the third and fourth bias points since the bias voltages on both electrodes 2 and 3 have been altered by the same amount between each point.

```
SOLVE  PROJ V1=0 V2=0 V3=1 VSTEP=.5 NSTEPS=2 ELECT=23
SOLVE  PROJ V2=2 V3=3
```

If no new voltages are specified and a VSTEP is included, the first bias point solved for is the preceding one incremented appropriately by VSTEP. This is illustrated by repeating the above example as a three line sequence:

```
SOLVE  V1=0 V2=0 V3=1
SOLVE  PROJ VSTEP=.5 NSTEPS=2 ELECT=23
SOLVE  PROJ VSTEP=1  NSTEPS=1 ELECT=23
```

The following sequence is an example of a time-dependent solution. The METHOD line specifies the second-order discretization and automatic time-step selection option, along with Newton-Richardson. The first SOLVE line then computes the solution for a device with 1 volt on V1 and 0 on V2 in steady-state. The second SOLVE line specifies that V1 is to be ramped to 2 volts over a period of 10ns and is left on until 25 ns. Each solution is written to a file; the name of the file is incremented in a manner similar to that described above for a dc simulation (UP1, UP2, etc.). Note that an initial time step had to be specified on this line. The third SOLVE line ramps V1 down from 2 volts to -1 volts in 20 ns (end of ramp is at t = 45ns). The device is then solved at this bias for another 55 ns (out to 100 ns). Note that again each solution is saved in a separate file (DOWN1, DOWN2, etc.) and that no initial time-step was required since one had been estimated from the last transient solution for the previous SOLVE line. Finally, the fourth SOLVE line performs the steady-state solution at V1=-1 and V2=0.

```
METHOD 2ND TAUTO AUTONR
SOLVE  V1=1 V2=0
SOLVE  V1=2 TSTART=1E-12 TSTOP=25E-9 RAMPTIME=10E-9
+      OUTF=UP1
SOLVE  V1=-1 TSTOP=100E-9 RAMPTIME=20E-9 OUTF=DOWN1
SOLVE  V1=-1 V2=0
```

The following is a single event upset simulation. Assuming a particle track have been defined using a previous P.TRACK line and that no G(t) dependence was defined on that line (see the P.TRACK line documentation), the track charge will be generated uniformly over the first 20ps of the simulation which will last until 1us.

```
SOLVE  SEU TSTART=1e-14 TSTOP=1e-6 DT.SEU=20e-12
```

The following is a blanket radiation simulation using rise and fall times. Starting from a dc solution (V1=-10 volts), a generation rate of 1.0e24 eh pairs/cm**3, falling exponentially away from y=0 with a characteristic length of 0.01mm, is applied to regions 1 and 2. The rise time of the pulse is 1ps. At t=1ns, the pulse is removed with a fall time of 5ps, and the simulation is continued until t=1ms.

```
SOLVE  V1=-10
SOLVE  GEN=1E24 REG.GEN=1,2 ABSORP=0.01 DIR.GEN=y
+      G.TAU=1E-12 TSTEP=1E-14 TSTOP=1E-9
SOLVE  G.TAU=5E-12 TSTOP=1E-6
```

Finally, an AC example is presented. Assume the device to be simulated has 3 electrodes. Starting from solved DC conditions at V1 = 0, 0.5, 1.0, 1.5 and 2.0 volts, 10 mV AC signals of frequency 1 MHz, 10 MHz, 100 MHz, 1 GHz, 10 GHz and 100GHz are applied to each electrode in the device. And for each of these frequencies, also compute the current noise parameters, and write the output in the "noisefile". Note that the number of AC solutions to be performed is 5*6*3=90.

```
SOLVE  PROJ V1=0 V2=0 V3=0 VSTEP=0.5 NSTEPS=4 ELECT=1
+      AC FREQ=1E6 FSTEP=10 MULT.F NFSTEP=5 VSS=0.01
+      NOISE  I.NOISE=t  NFILE=noisefile
```

The SPREAD line

Syntax

```
  SPread direction region specifics
```

Description

The SPREAD line provides a way to distort rectangular grids in the vertical direction to follow surface and junction contours. SPREAD is very useful in reducing the amount of grid for some specific problems, most notably MOSFET's. The SPREAD line is somewhat complicated; it is suggested to follow the supplied examples very carefully (see the MOSFET example in the *PISCES* manual).

Parameters

direction is one of:

```
  LEft   =  logical  (default is false)
  Right  =  logical  (default is false)
```

LEFT and RIGHT specify that the left and right-hand sides of the grid respectively be distorted.

region

```
  Width  =  real
  Upper  =  integer
  LOwer  =  integer
```

WIDTH specifies the width from the left or right edge (depending on the LEFT and RIGHT parameters) of the distorted area. The actual x-coordinate specified by WIDTH (min[x] + WIDTH for LEFT, max[x] - WIDTH for RIGHT) will lie in the middle of the transition region between the distorted and undistorted grid regions. UPPER and LOWER specify the upper and lower y-grid lines between which the distortion will take place.

specifics

```
  Y.Lower    =  real
             or
  Thickness  =  real
            and
  Vol.ratio  =  real  (default is 0.44)
  Encroach   =  real  (default is 1.0)
  GRAding    =  real  (default is 1.0)
  GR1        =  real  (default is 1.0)
  GR2        =  real  (default is 1.0)
  Middle     =  real
  Y.Middle    =  real
```

The Y.LOWER and THICKNESS parameters define the distorted grid region; only one should be supplied. Y.LOWER is the physical location in the distorted region at which the line specified by LOWER will be moved. The line specified by UPPER is not moved. THICKNESS is the thickness of the distorted region; THICKNESS will usually move the positions of both the UPPER and LOWER grid lines (unless VOL.RATIO is set to 0 or 1). VOL.RATIO specifies the ratio of the downward displacement of the lower grid line to the net increase in thickness. The default is 0.44 so that oxide-silicon interfaces are correct. VOL.RATIO is ignored if Y.LOWER is specified. ENCROACH is a factor which defines the abruptness of the transition between distorted and non-distorted grid. The transition region becomes more abrupt with smaller ENROACH factors (the minimum is 0.1). An important note: depending on the characteristics of the undistorted grid, very bad triangles (long, thin and obtuse) may result if ENCROACH is set too low. GRADING specifies a grid ratio (identical to the RATIO parameter on the X.MESH and Y.MESH lines) to produce a non-uniform grid in the distorted region. As alternative to a single grading parameter, GR1 and GR2 can be specified along with the y grid line MIDDLE and location Y.MIDDLE so that GR1 is used as the grading in the spread region from UPPER to MIDDLE and GR2 is the grading from MIDDLE to LOWER.

Examples

The following spreads what was previously a uniform 400 Angstroms of oxide to 1000 Angstroms on the left side of the device. This will result in a net increase in thickness of 600 Angstroms of oxide. Because the default VOL.RATIO is used, 0.44*(600) = 264 Angstroms of the net increase will lie below the original 400 Angstroms and 0.56*(600) = 336 Angstroms of the net increase will lie above the original 400 Angstroms. The width of the spread region is 0.5mm and the oxide taper is quite gradual because of the high encroachment factor. The grid is left uniform in the spread region.

```
  $ *** Mesh definition ***
  MESH    NX=30 NY=20 RECT
  X.M     N=1  L=0
  X.M     N=30 L=2
  Y.M     N=1  L=-.04
  Y.M     N=5  L=0
  Y.M     N=20 L=1 R=1.4
  $ *** Thin oxide ***
  REGION  X.L=1 X.H=30 Y.L=1 Y.H=5
  $ *** Silicon substrate ***
  REGION  X.L=1 X.H=30 Y.L=5 Y.H=20
  $ *** Spread ***
  SPREAD  LEFT WIDTH=0.5 UP=1 LO=5 THICK=0.1 ENC=1.3
```

In the next example, the right side of the grid is distorted in order to follow a junction contour. Assume that the initial grid is defined as above. Y.LOWER is used so that there is no increase in the size of the device, just grid redistribution. With Y.LOWER set to the junction, the ENCROACH parameter should be chosen such that the lower grid line

(LOWER=10) follows the junction as closely as possible. Note that the grid is graded so that the grid lines are spaced closer together as they approach the junction. Because the point specified by WIDTH on the SPREAD line lies in the middle of the transition region, it should be chosen to be slightly larger than the width of the doping "box" (WIDTH < X.LEFT - X.RIGHT = 0.5mm).

```
$ *** Doping ***
DOPING  UNIFORM N.TYPE CONC=1E15
DOPING  GAUSS P.TYPE X.LEFT=1.5 X.RIGHT=2
+       PEAK=0 CONC=1e19 RATIO=.75 JUNC=0.3
$ *** Spread ***
SPREAD  RIGHT WIDTH=0.7 UP=5 LO=10 Y.LO=0.3
+       ENC=1.2 GRAD=0.7
```

The SURFACE line

Syntax

```
SUrface  type  location
```

Description

The SURFACE line defines the location of an interface or electrode number along region boundaries for later access. Both electrodes and interfaces can also be defined in an external mesh generator (interfaces have negative electrode numbers). Electrodes can also be defined by the ELECTRODE line, but the SURFACE line restricts the electrode edges to conform to a region interface.

Parameters

type

```
Interface  =  logical
Electrode  =  logical
Number     =  integer
```

INTERFACE and SURFACE specify what kind of surface is to be defined. NUMBER defines the electrode or interface number.

location

```
X.Min  =  real     (default is min(x) in device)
X.Max  =  real     (default is max(x) in device)
Y.Min  =  real     (default is min(y) in device)
Y.Max  =  real     (default is max(y) in device)
Z.Min  =  real     (default is min(z) in device)
Z.Max  =  real     (default is max(z) in device)
```

```
  REG1   =  integer  (default is 0)
  REG2   =  integer  (default is 0)
```

X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN and Z.MAX define a bounding box, measured in um. If REG1=REG2=0, any outer edge of the device within this box is incorporated into the surface. With nonzero REG1 and REG2, any boundary between REG1 and REG2 that falls in the box is incorporated.

Examples

Define an interface between regions 1 and 4 within a box which extends through the entire z-dimension.

```
  SURFACE  INTERFACE NUM=1 REG1=1 REG2=4
  +        X.MIN=-4 X.MAX=4 Y.MIN=-0.5 Y.MAX=4
```

The SYSTEM line

Syntax

```
  System  equations coupling options
```

Description

The SYSTEM line defines which partial differential equations (PDEs) are to be solved and how the equations are to be coupled in the nonlinear iteration.

Parameters

equations

```
  CArriers       =  integer  (default is 0)
  Electrons      =  logical
  Holes          =  logical
  N.temperature  =  logical
  P.temperature  =  logical
```

The above parameters define the PDEs to be solved. The Poisson equation is always solved, and optionally one can specify that continuity and/or energy balance PDEs be solved for the carriers (the corresponding PDE for carrier momentum always reduces to a closed form expression in the current version of *PADRE* and hence need not be explicitly selected). The continuity equations can be selected by setting CARRIERS to the number of carriers (0, 1 or 2) with the single carrier defined through the ELECTRONS and/or HOLES parameter; the default is ELECTRONS for CARRIERS=1. The parameters

N.TEMPERATURE and P.TEMPERATURE select the equations for average electron and hole energy respectively.

coupling

```
NEwton    =  logical
          or
Gummel    =  logical
          or
COupling  =  integer vector
```

NEWTON specifies that all the equations be solved simulaneously in a single set of nonlinear iterations. GUMMEL forces each of the specified PDEs to be solved separately (e.g., plug-in iteration) so that in general there is an inner nonlinear iteration for each PDE and an outer iteration over all the equations. Alternatively, one can specify a general nonlinear construct (with arbitrary groupings of PDEs) using the COUPLING parameter. In this case the PDEs are selected using an integer code as follows:

```
1    Poisson equation
2    Electron continuity equation
3    Hole continuity equation
4    Electron energy balance equation
5    Hole energy balance equation
```

Groups of equations to be fully coupled are specified by concatenating digits, and individual groups of equations are separated by commas.

options

```
PRint      =  logical  (default is false)
Symmetric  =  logical  (default is true)   [Expert]
```

The PRINT parameter indicates that information about the memory allocated for the run should be printed to the *PADRE* standard output file. SYMMETRIC specifies that the symmetric block matrix data structure should be used wherever possible.

Examples

The following specifies a PDE system for a simulation with only holes and using the Gummel method:

```
SYSTEM  GUMMEL CARR=1 HOLES
```

Solve the Poisson equation and the continuity and energy balance equations for electrons. Define an iteration strategy which couples the Poisson and continuity equations directly, but treats the energy balance equation separately.

```
SYSTEM  ELECTRONS N.TEMP COUPLING=12,4
```

The TITLE line

Syntax

```
  Title character string
```

Description

The TITLE line specifies a title (up to 60 characters) to be used in *PADRE* standard output.

Examples

```
  TITLE  *** 0.1um MOSFET - overshoot calculation ***
```

The VECTOR line

Syntax

```
  Vector plotted-quantity control
```

Description

The VECTOR line plots vector quantities over an area of the device defined by the previous PLOT.2D line.

Parameters

plotted-quantity is one of:

```
  J.Conduc   =  logical   Conduction current
  J.Electr   =  logical   Electron current
  V.Electr   =  logical   Electron velocity
  J.Hole     =  logical   Hole current
  V.Hole     =  logical   Hole velocity
  J.Displa   =  logical   Displacement current
  J.Total    =  logical   Total current
  E.field    =  logical   Electric field
```

control

```
  LOgarithm  =  logical   (default is false)
  MInimum    =  real      (default is 0)
  MAximum    =  real      (default is 0)
  Scale      =  real      (default is 1)
  Clipfact   =  real      (default is 0.1)
  LIne.type  =  integer   (default is 1)
```

By default, vectors are scaled linearly by the maximum magnitude of the quantity of interest over the grid; alternatively LOGARITHM specifies logarithmically-scaled vectors, scaled by the minimum (non-zero) magnitude. The minimum and maximum values - both of which are printed during execution of a plot - can be set using MINIMUM and MAXIMUM so that two bias conditions or devices can be plotted with the same scale. SCALE a constant scale factor to be multiply all magnitudes by. CLIPFACT is a threshold below which vectors are not plotted. LINE.TYPE specifies the vector line type for plotting.

Examples

Plot electron and hole currents over a device:

```
  PLOT.2D BOUN NO.FILL
  VECTOR J.ELEC LINE=2
  VECTOR J.HOLE LINE=3
```

The X.MESH, Y.MESH, Z.MESH lines

Syntax

```
  X.Mesh  node  location  density
  Y.Mesh  node  location  density
  Z.Mesh  node  location  density
```

Description

The X.MESH and Y.MESH lines specify the location of grid lines in a rectangular mesh. Z.MESH lines specify the locations of grid planes in the depth dimension to either rectangular or nonrectangular 2D grids.

Parameters

node

```
  Node  =  integer
```

This is the number of the line in the mesh. At present, there can be at most 300 lines in either direction. Lines are assigned consecutively, beginning with the first and ending with the last.

location

```
Location   =  real
```

This is where to locate the line. The location is interpreted in microns.

density

```
Ratio          =  real
Density or H   =  real
```

These parameters define the grid line spacing or density; only one may be specified. DENSITY (or H) gives an exact density (in microns). RATIO gives the ratio to use when interpolating lines between those given in the X.MESH/Y.MESH input. The spacing grows/shrinks by RATIO in each subinterval and should usually lie between 0.667 and 1.5.

Examples

Space grid lines closely around a junction (at 0.85mm) in a 1D diode:

```
Y.MESH N=1 LOC=0.0
Y.MESH N=20 LOC=0.85 RATIO=0.75
Y.MESH N=40 LOC=2 RATIO=1.333
```

Overview

A *bipad* input deck is composed of one or more of the following commands:

- 2dload - specifies the input/output files
- edit - performs geometric editing operations
- electrode - defines electrodes

An example deck looks like <a href="../ex-nmos-bipad.html>this.

The 2DLOAD line

Syntax

```
  2dload process-simulator files control
```

Description

The 2DLOAD line specifies the input and output files and some control for the [1] interface program.

Parameters

process-simulator

Specifies which process simulator to use:

```
  OBiceps  = logical  (default is true)
  BICeps5  = logical  (default is false)
  SUpremiv = logical  (default is false)
```

files

Specifies file information:

```
  INfile   = character
  SKelfile = character
  OUtfile  = character
  BISon    = character
```

INFILE is the output file from the given process simulator. SKELFILE is a skel format file containing the extracted boundary of the device structure. OUTFILE and BISON give doping output files with differing formats for use in [2]. OUTFILE is the general format file suitable for both [3] and [4]. BISON is a special doping file which can only be used with [5] output. The primary advantage in using the BISON format is a reduction in the file size (and hence i/o time consumed).

control

Controls which modify geomerty:

```
  Gcriter      =  real     (default is 0.05)
```

```
Reset.origin  =  logical  (default is false)
EStart        =  integer
Mos           =  logical  (default is false)
H.inv         =  integer  (default is 0.0025)
EPs.mos       =  integer  (default is 0.01)
Y.mos         =  integer
HY.bot        =  integer
NX.mos        =  integer  (default is 15)
```

GCRITER is the extraction criterion for obtaining the skel output file. The higher GCRITER, the less points are used (if GCRITER is zero, all the points in the [6] grid on boundaries are kept). RESET.ORIGIN puts the x-origin (x=0) back at the extreme left edge of the device after performing and subsequent editing. ESTART gives the starting electrode number for any electrodes found in the process simulator output file; the default is to ignore these definitions. MOS outputs a special skeleton for devices with inversion layers under semiconductor-insulator interfaces, and H.INV, HY.BOT, EPS.MOS, Y.MOS and NX.MOS are parameters specific to the MOS skeleton extraction that define a box of rectangular grid under the gate region(s). H.INV defines the vertical grid spacing directly under the gate, HY.BOT defines the vertical grid spacing at the bottom of the box, EPS.MOS sets the offset of the box from the junctions, Y.MOS defines the box depth and NX.MOS specifies the number of grid lines in the horizontal direction within the box. The default for Y.MOS is selected based on the values of the other parameters.

Examples

The following uses a [7] file called "boutfil" to make a doping file called "dop2d" and a skel file called "skel1". The extraction criterion is 0.1.

```
2DLOAD  BICEPS5 INF=boutfil OUTF=dop2d SKEL=skel1 GCRIT=0.1
```

The EDIT line

Syntax

```
EDit type control
```

Description

The EDIT line performs geometric editing on a 2D device structure obtained from a preceding 2DLOAD line. Any number of EDIT cards may be used, and each is applied in the sequence in which it occurs in the [1] input deck.

Parameters

type

One of:

```
EXpand  =  logical  (default is true)
Strech  =  logical  (default is false)
DELEte  =  logical  (default is false)
```

The EXPAND option allows the device structure (geometry and doping profile) to be extended in x or y by specifying a new minimum or maximum value. STRECH and DELETE are used to strech and delete an internal part of the structure in x only.

If *type*=EXPAND, the following control arguments are relevant:

One of:

```
X.MIn   =   real
X.MAx   =   real
Y.MIn   =   real
Y.MAx   =   real

  and

Mirror   =   logical  (default is false)
```

The device structure is enlarged in x or y by setting one of X.MIN, X.MAX, Y.MIN or Y.MAX to a value beyond its current limit. The MIRROR option, used only in conjuction with X.MIN and X.MAX, does the expansion by mirroring both the geometry and doping inside the device.

If *type*=STRECH, the following control arguments are relevant:

```
X.1   =  real
DELX  =  real
```

The device is streched at x=X.1 by a distance DELX.

If *type*=DELETE, the following control arguments are relevant:

```
X.1   =  real
X.2   =  real
```

The device area between x=X.1 and x=X.2 is deleted.

Examples

Assuming an initial device that goes from x=0 to x=2um, the following sequence of cards (1) expands the device in the -x direction out to x=-1um by mirroring and (2) streches the device at x=0 by 0.5um.

```
  EDIT   EXPAND X.MIN=-1 MIRROR

  EDIT   STRECH X.1=0 DELX=.5
```

The ELECTRODE line

Syntax

```
  ELEctrode number position adjustment
```

Description

The ELECTRODE line specifies the location of new electrodes in a [1] generated structure. The electrodes will be incorporated in the skel format output file specified on a preceding 2DLOAD line.

Parameters

number

```
        Number  =  integer
```

NUMBER is the id number of the electrode to be added or modified: There may be up to ten electrodes, numbered 1,2,3,...,9,0. They may be assigned in any order, but if there are N electrodes, none can have an electode number above N. There may be multiple definitions for a single electrode number.

location

```
        X.MIn   =  real     (default is left edge)
        X.MAx   =  real     (default is right edge)
        Top     =  logical  (default is false)
        Bottom  =  logical  (default is false)
```

Electrode extents are defined in the x-direction by spatial coordinates. TOP and BOTTOM specify the y location. If TOP is specified, the electrode is placed on top of the oxide layer (if it exists). If BOTTOM is specified, the electrode is placed on the back-side. If neither TOP nor BOTTOM is given, then the electrode is put at the $Si-SiO sub 2$ interface.

Examples

- Define contact all along the back-side:
- `    ELEC   N=1 BOTTOM`
- Put an electrode at the top silicon surface between x=1.25um and x=1.5um.
- `    ELEC   NUM=3 X.MIN=1.25 X.MAX=1.5`