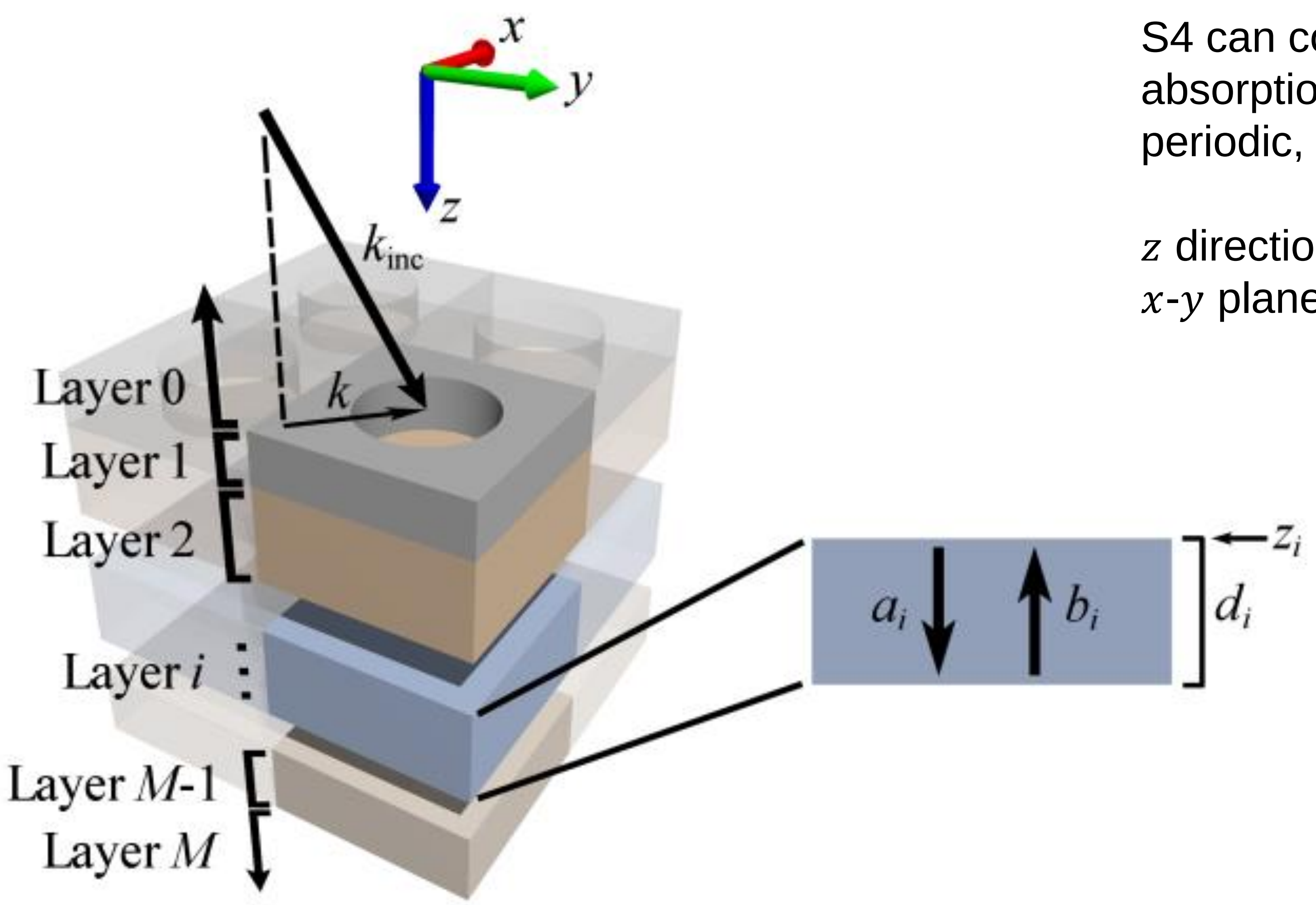
S4 Tutorial

https://nanohub.org/tools/s4sim

S4 – Stanford Stratified Structure Solver



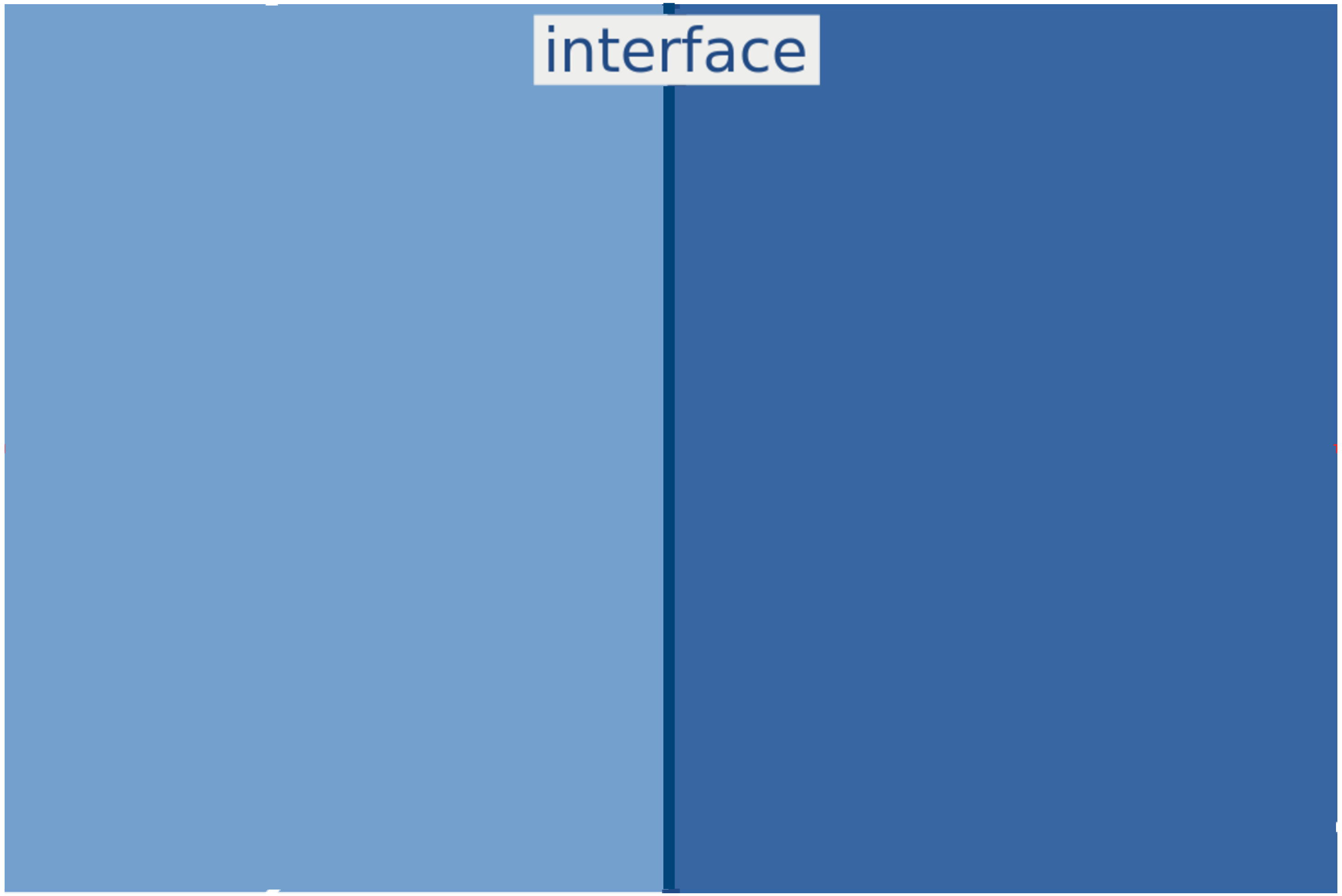
S4 can compute transmission, reflection, or absorption spectra of structures composed of periodic, patterned, planar layers.

z direction: layered media
x-y plane: periodic patterns

https://web.stanford.edu/group/fan/S4/index.html

nanoHUB

Example 1: Plane Wave Incident on Air-Glass Interface



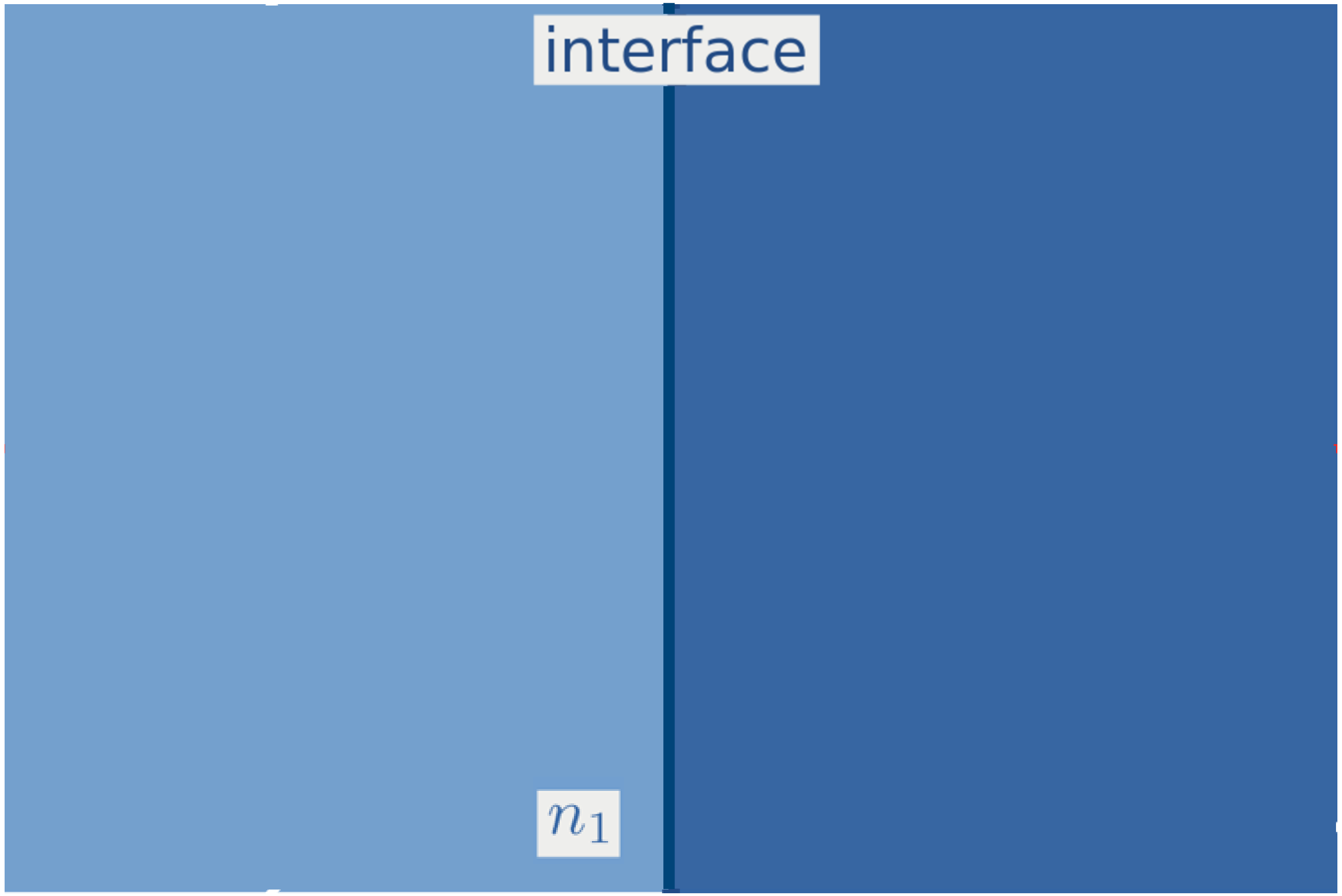
Reflectance *R*:
the fraction of the incident power that is reflected from the interface

Law of reflection

$$\theta_{\mathrm{i}} = \theta_{\mathrm{r}}\,,$$

https://en.wikipedia.org/wiki/Fresnel_equations

Example 1: Plane Wave Incident on Air-Glass Interface



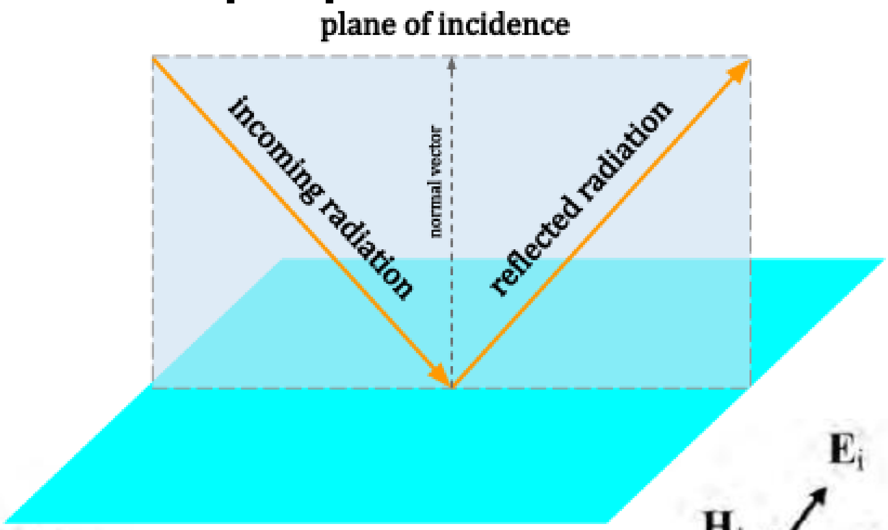
Transmittance *T*:
the fraction of the incident power that is refracted into the second medium

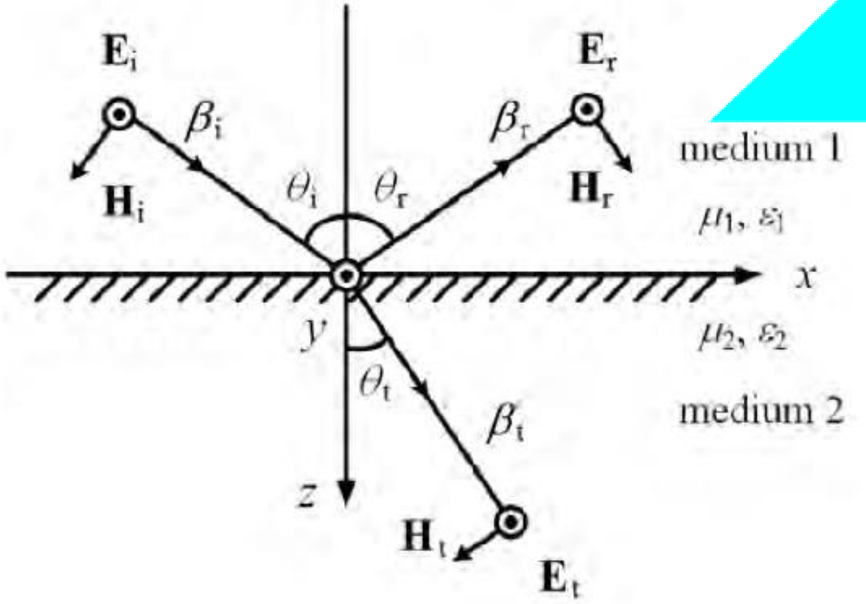
Snell's law

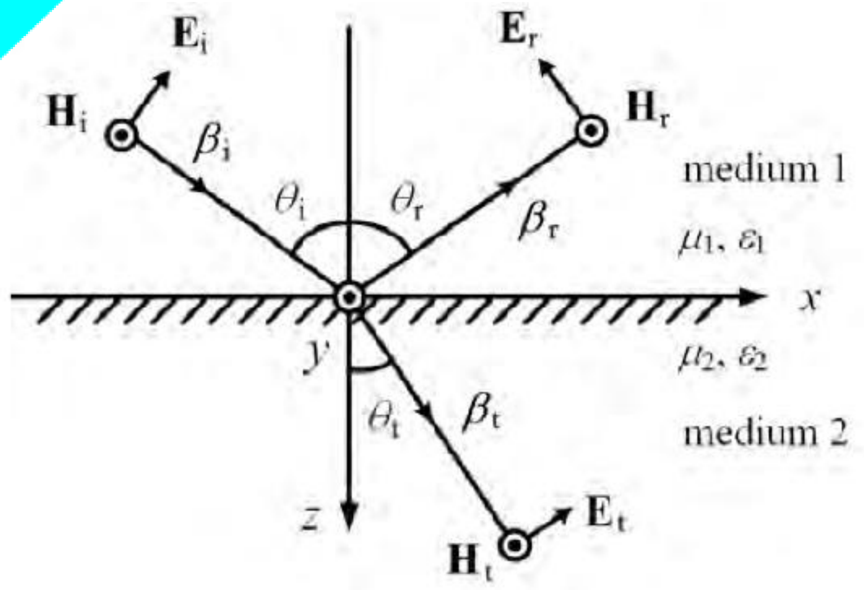
$$n_1 \sin\theta_\mathrm{i} = n_2 \sin\theta_\mathrm{t}\,.$$

https://en.wikipedia.org/wiki/Fresnel_equations

s-/p- polarization





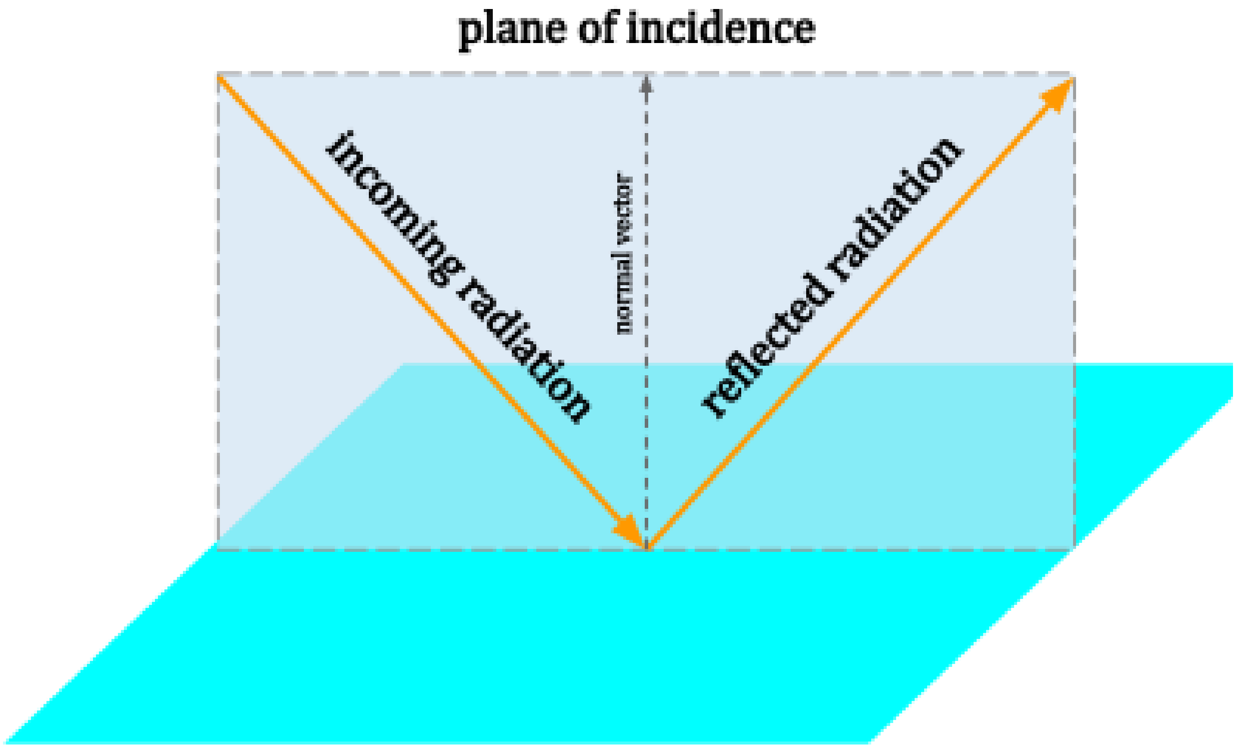


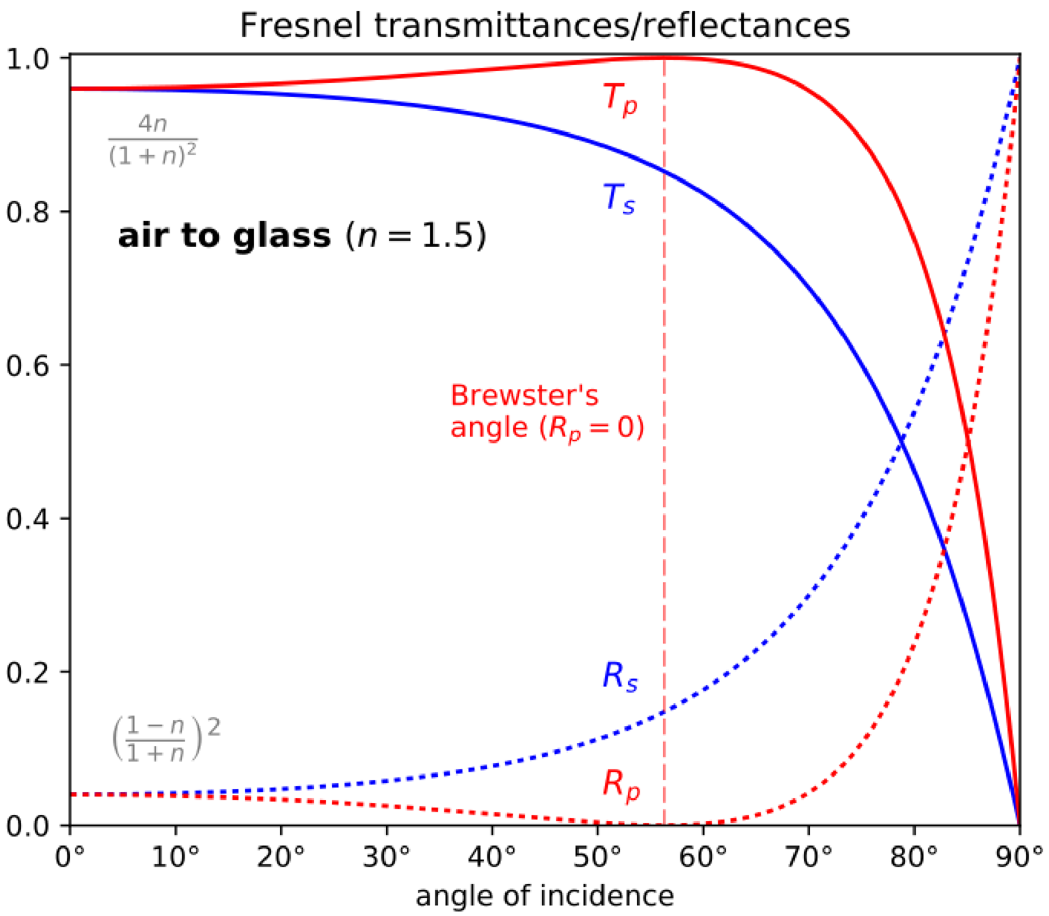
s polarization (TE):
electric field is perpendicular to the plane of incidence

p polarization (TM):
electric field is parallel to the plane of incidence

Example 1: Plane Wave Incident on Air-Glass Interface

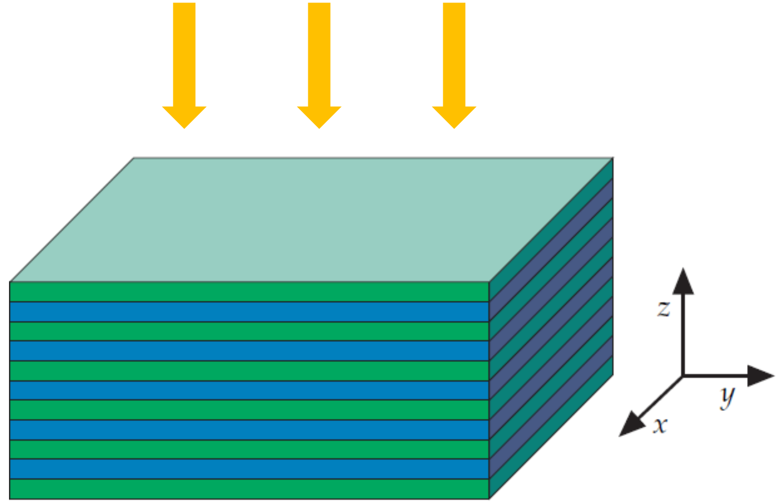
$$R_s = \left| \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \right|^2 \qquad R_p = \left| \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} \right|^2$$

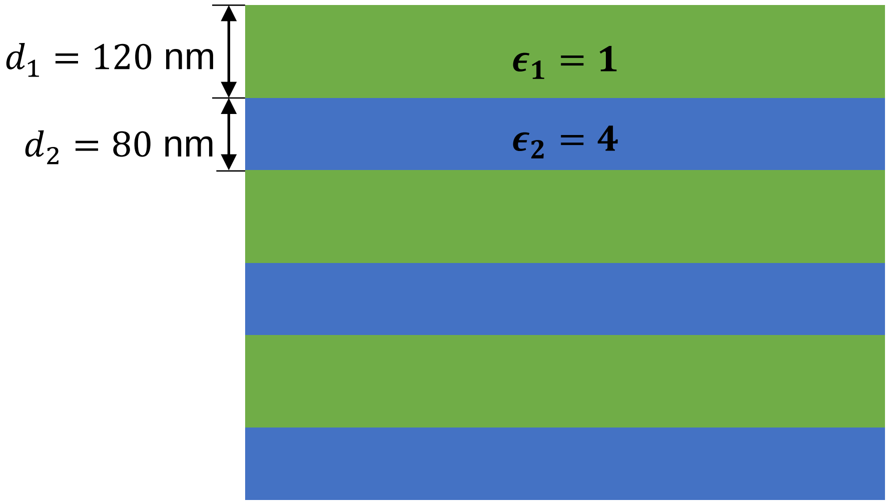


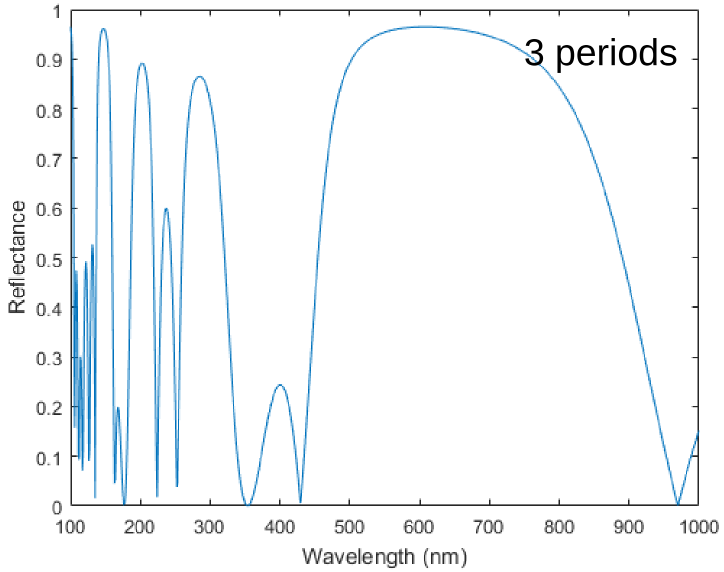


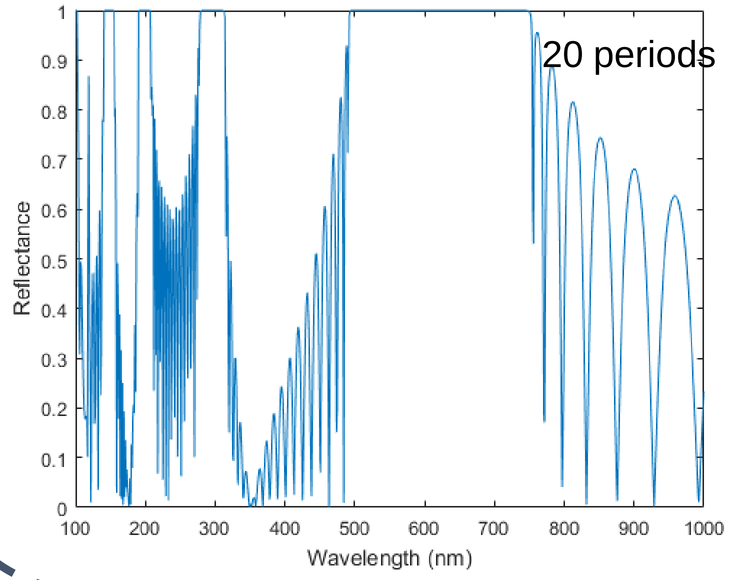
https://en.wikipedia.org/wiki/Fresnel_equations

Example 2: 1D Photonic Crystal



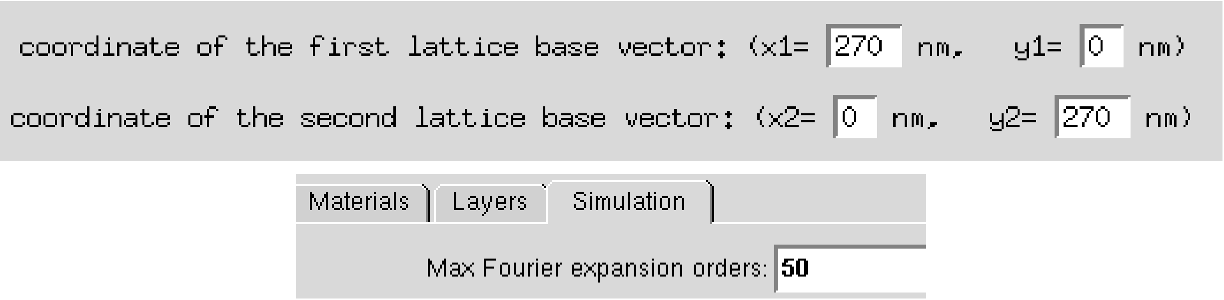






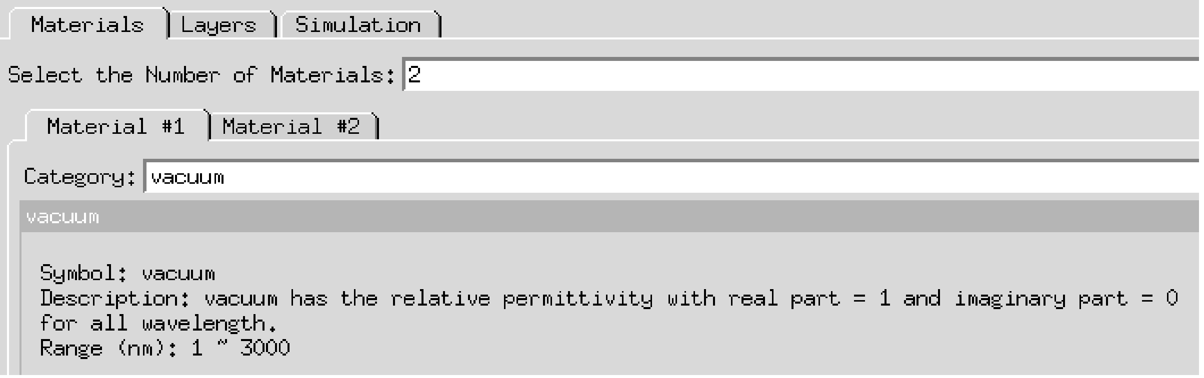
Graphical Interface vs. Control File

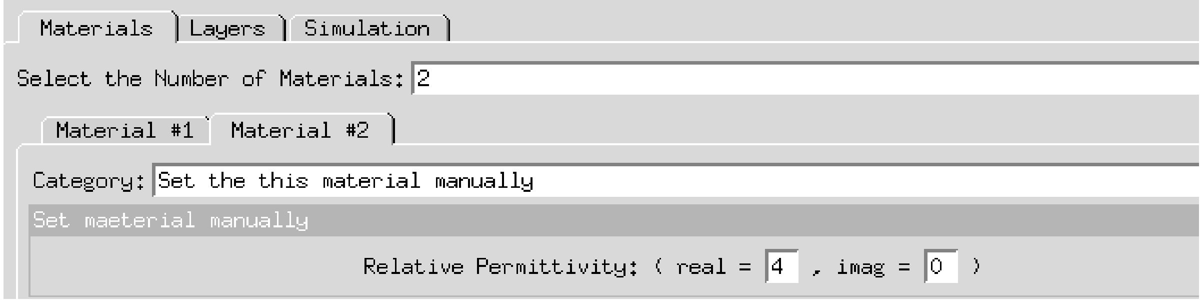
Obtain a new simulation object.
Set the lattice and number of basis functions.




```
S = S4.NewSimulation()   -- create new simulation object
S:SetLattice({3.000000,0.000000}, -- lattice basis vector (x1,y1)
             {0.000000,3.000000}) -- lattice basis vector (x2,y2)
S:SetNumG(50)   -- number of basis functions
```

Define all materials.



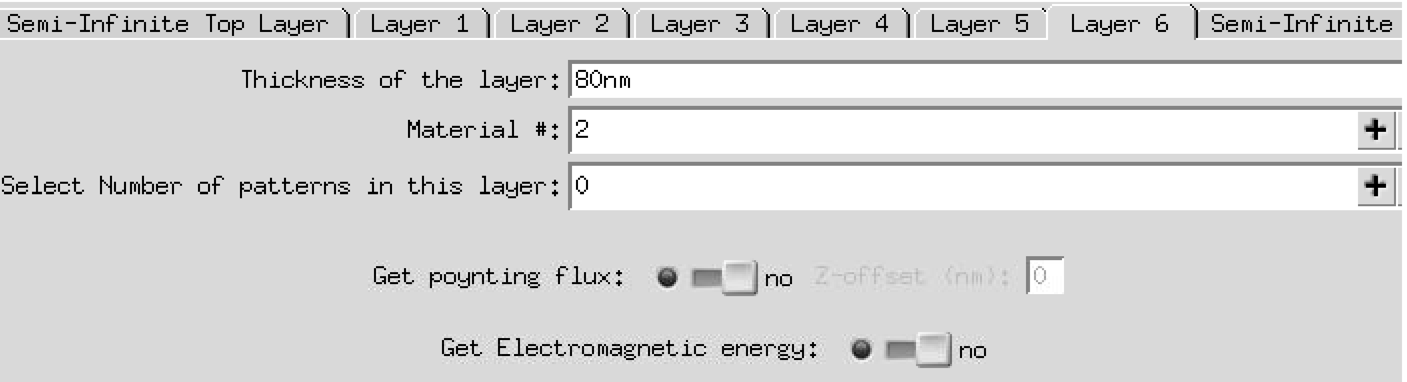



```
S:AddMaterial("vacuum",       -- material name
        {1.000000,0.000000}) -- real and imag parts of permittivity
S:AddMaterial("material_2", {4.000000,0.000000})
```

nanoHUB

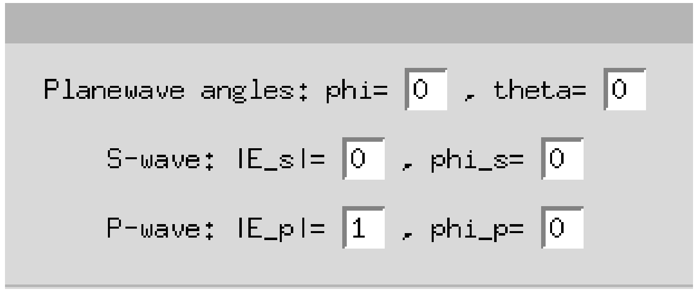
Graphical Interface vs. Control File

Add all layers.




```
S:AddLayer(
    'Layer_Above',   -- layer name
    0.000000,        -- thickness
    'vacuum')        -- background material
S:AddLayer('layer_1', 1.333333, 'vacuum')
S:AddLayer('layer_2', 0.888889, 'material_2')
S:AddLayer('layer_3', 1.333333, 'vacuum')
S:AddLayer('layer_4', 0.888889, 'material_2')
S:AddLayer('layer_5', 1.333333, 'vacuum')
S:AddLayer('layer_6', 0.888889, 'material_2')
S:AddLayerCopy(
    'Layer_Below',   -- new layer name
    0.000000,        -- thickness
    'Layer_Above')   -- layer to copy
```

Specify the excitation mechanism.




```
S:SetExcitationPlanewave(
    {0.000000,0.000000},   -- incidence angles (spherical coordinates):
                           -- phi (polar angle) in [0,180),
                           -- theta (azimuthal angle) in [0,360)
    {0.000000,0.000000},   -- s-polarization amplitude and phase in degrees
    {1.000000,0.000000})   -- p-polarization amplitude and phase in degrees
```

Graphical Interface vs. Control File

Create array before sweeping over all frequencies.

The wavelength range is: min= 100nm , max= 1000nm , step= 2nm

```
frequency = {0.900000, 0.882353, 0.865385, 0.849057, 0.833333, 0.818182, 0.803571, 0.789474, 0.775862, 0.762712,...
real_eps_1 = {1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,...
imag_eps_1 = {0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,...

real_eps_2 = {4.000000, 4.000000, 4.000000, 4.000000, 4.000000, 4.000000, 4.000000, 4.000000, 4.000000, 4.000000,...
imag_eps_2 = {0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,...
```

Loop over all frequencies

```
for i = 1, 451 do
    freq = frequency[i];
    S:SetFrequency(freq)

    S:SetMaterial('vacuum', {real_eps_1[i], imag_eps_1[i]});
    S:SetMaterial('material_2', {real_eps_2[i], imag_eps_2[i]});

    incidence_flux, reflection_flux_vacuum = S:GetPoyntingFlux('Layer_Above', -- layer in which to get Poynting flux
                                                               0.000000)   -- z-offset
    reflection_flux_vacuum = (-1) * reflection_flux_vacuum / incidence_flux; -- normalize reflection flux by incidence flux

    transmission_flux = S:GetPoyntingFlux('Layer_Below', 0.000000)
    transmission_flux_vacuum = transmission_flux / incidence_flux; -- normalize transmission flux by incidence flux

    incidence_flux_vacuum = incidence_flux / incidence_flux;    -- normalize incidence flux itself

    print(freq .. '\t' .. incidence_flux_vacuum .. '\t' .. reflection_flux_vacuum .. '\t' .. transmission_flux_vacuum);
end
```

Specify the operating frequency

Reset material properties (for dispersive media)

Obtain desired output

Print output

FAQ: Reduced Unit

In S4, the speed of light is assumed to be 1. Then $f = 1/\lambda$.

reduced units -> period: default

type in a number or the word "default".
Default means to set the period to 0.9 * wavelength_min.
If you set it to 1, the frequency will have a unit of c/nm, where c is the speed of light.

By setting "default", normalization constant = $0.9 \times \lambda_{min} = 0.9 \times 100nm = 90nm$

Wavelength = $100nm, 102nm, 104nm, \ldots, 998nm, 1000nm$

Wavelength in reduced unit = $\frac{100}{90}, \frac{102}{90}, \frac{104}{90}, \ldots, \frac{998}{90}, \frac{1000}{90}$

Frequency in reduced unit = $\frac{90}{100}, \frac{90}{102}, \frac{90}{104}, \ldots, \frac{90}{998}, \frac{90}{1000}$

```
frequency = {0.900000, 0.882353, 0.865385, 0.849057, 0.833333, 0.818182, 0.803571, 0.789474, 0.775862, 0.762712,…
```

FAQ: Reduced Unit

Basis vector coordinate $\frac{270nm}{90nm} = 3$

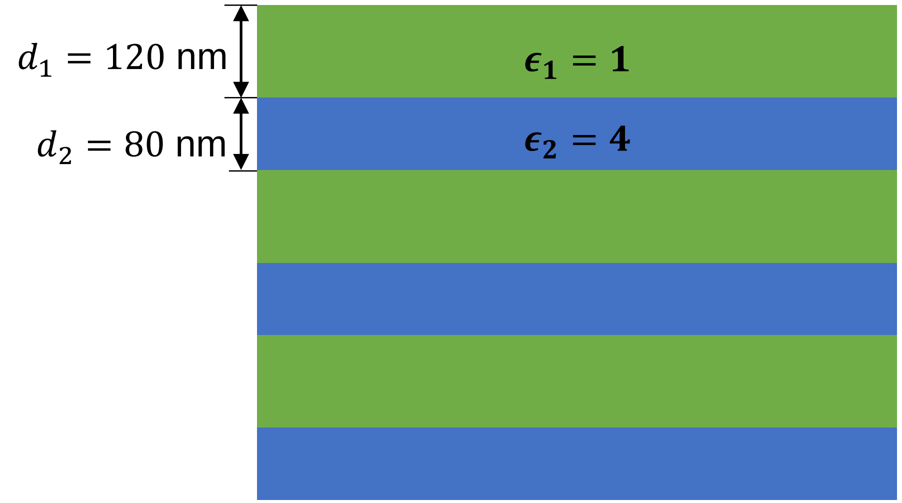
coordinate of the first lattice base vector: (x1= 270 nm, y1= 0 nm)

coordinate of the second lattice base vector: (x2= 0 nm, y2= 270 nm)

```
S:SetLattice({3.000000,0.000000},{0.000000,3.000000})
```

Layer thickness: $d_1 = 120nm$, $d_2 = 80nm$

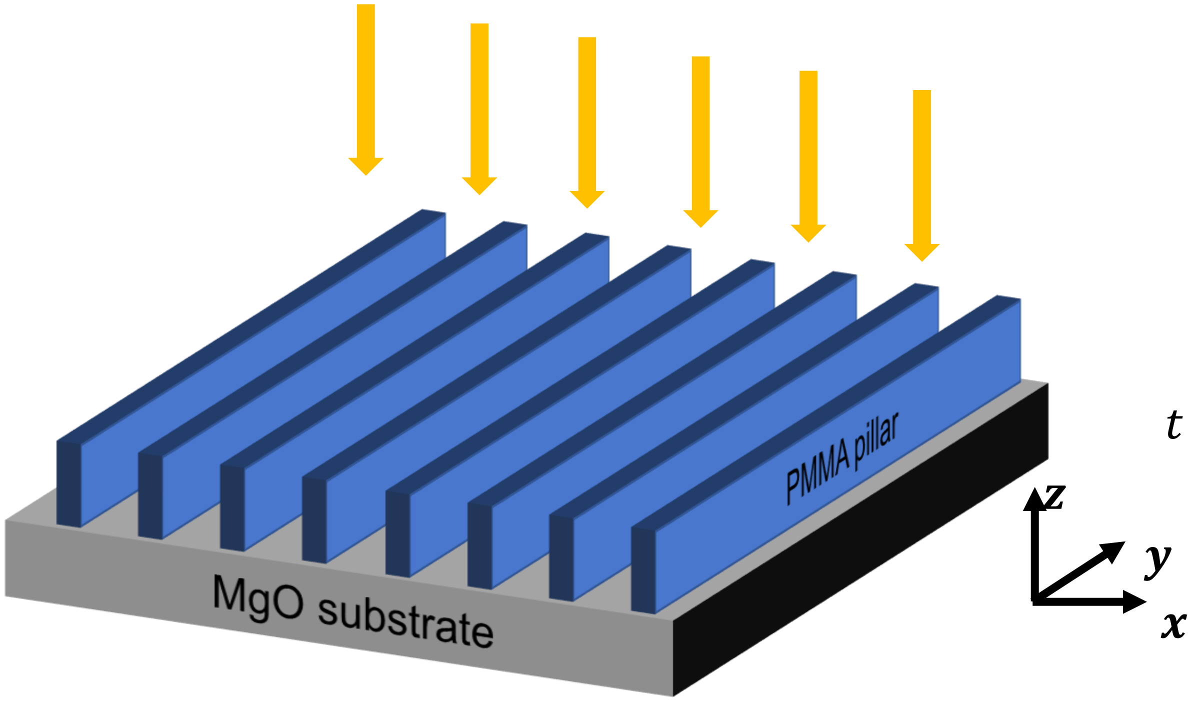
In reduced unit: $\frac{d_1}{90nm} = 1.333$, $\frac{d_2}{90nm} = 0.88889$

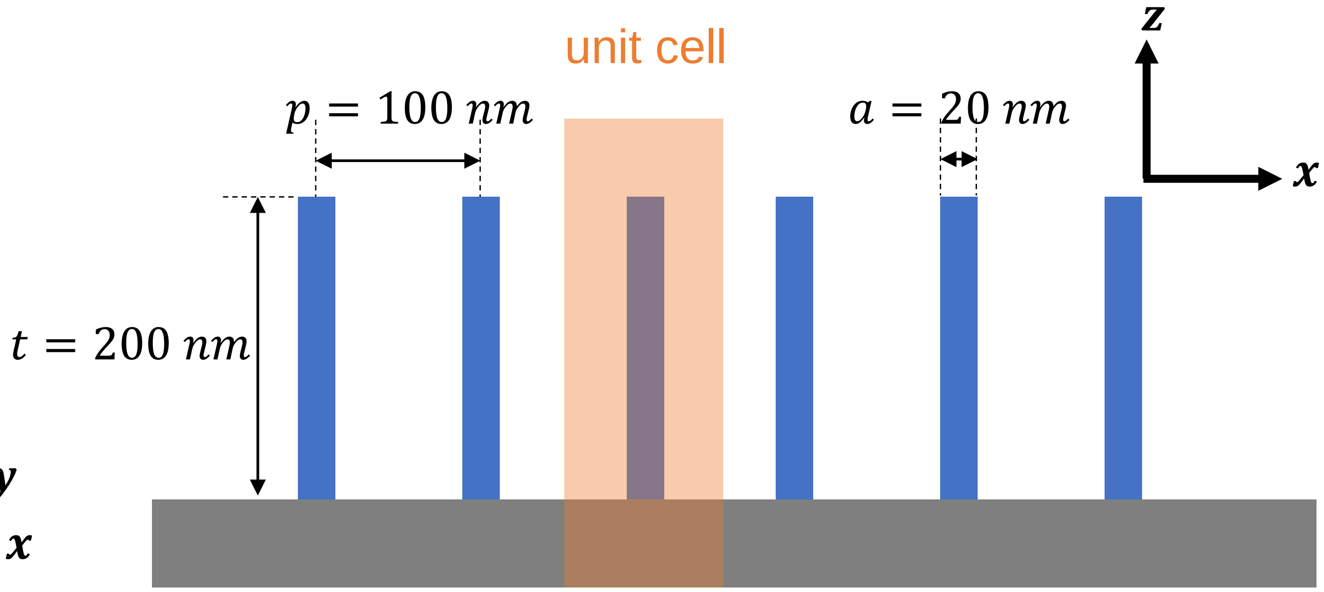



```
S:AddLayer('Layer_Above', 0.000000, 'vacuum')
S:AddLayer('layer_1', 1.333333, 'vacuum')
S:AddLayer('layer_2', 0.888889, 'material_2')
S:AddLayer('layer_3', 1.333333, 'vacuum')
S:AddLayer('layer_4', 0.888889, 'material_2')
S:AddLayer('layer_5', 1.333333, 'vacuum')
S:AddLayer('layer_6', 0.888889, 'material_2')
S:AddLayerCopy('Layer_Below', 0.000000, 'Layer_Above')
```

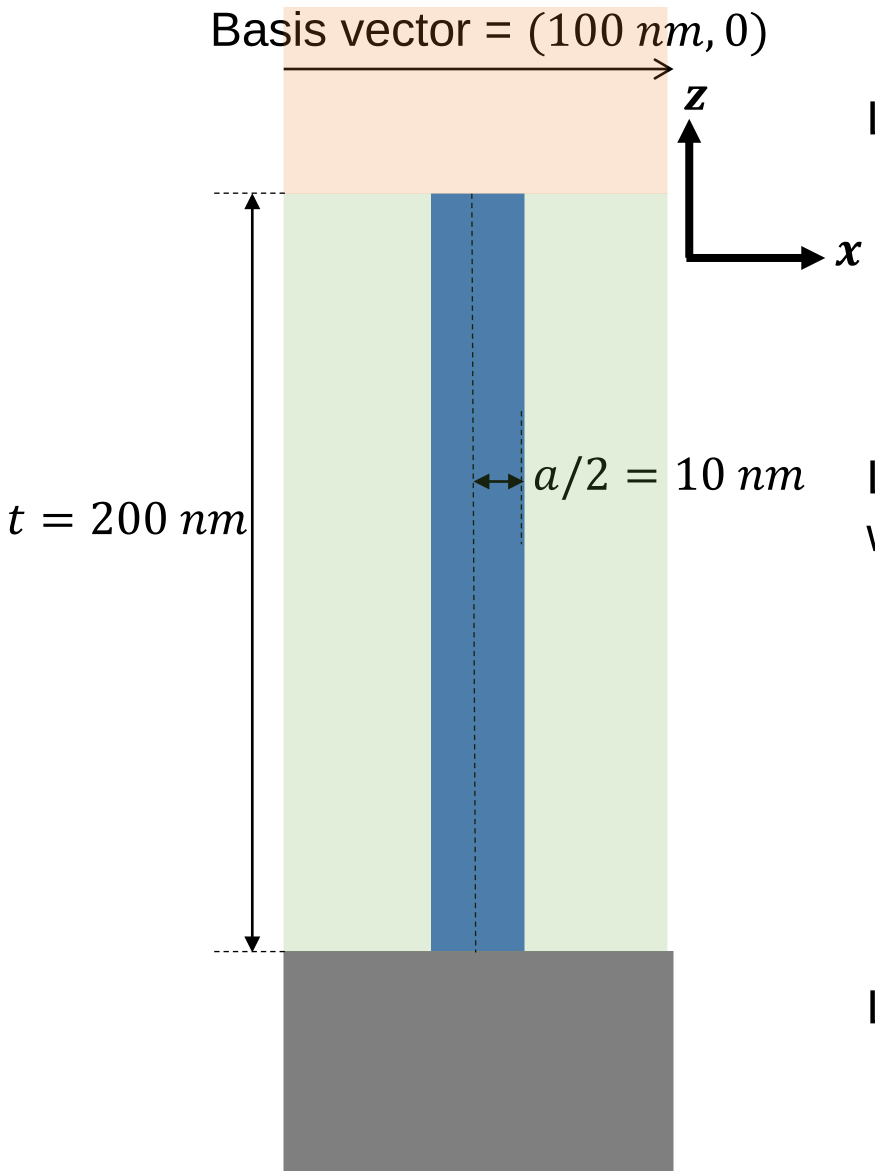
nanoHUB

Example 3: 1D Grating





Example 3: 1D Grating

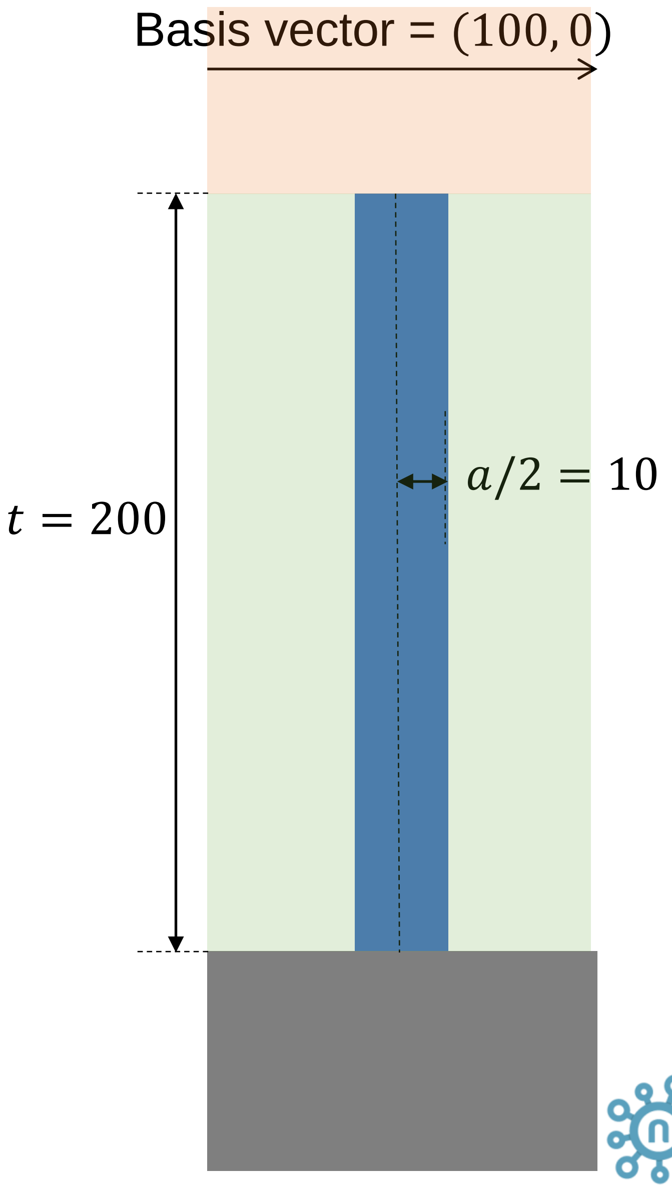


Layer 1: semi-infinite (vacuum)

Layer 2: vacuum background with PMMA pattern

Layer 3: semi-infinite (MgO)

Set normalization constant = 1nm



Example 3: 1D Grating

```
S = S4.NewSimulation()
S:SetLattice({100,0}, {0,0}) -- 1D lattice
S:SetNumG(50)
```

Obtain a new simulation object.
Set the lattice and number of basis functions.

```
-- Material definition
S:AddMaterial("Vacuum", {1,0}) -- real and imag parts
S:AddMaterial("PMMA", {2.25,0})
S:AddMaterial("MgO", {3,0})
```

Define materials.

```
-- Layer definition
S:AddLayer(
    'AirAbove',  --name
    0,           --thickness
    'Vacuum')    --background material
```

Add layers.

```
S:AddLayer('Middle', 200, 'Vacuum')
S:SetLayerPatternRectangle('Middle',    -- which layer to alter
                           'PMMA',      -- material in rectangle
                           {0,0},       -- center
                           0,           -- tilt angle (degrees)
                           {10, 0})     -- half-widths
```

Modify patterns.

```
S:AddLayer(
    'MgOBelow',  --name
    0,           --thickness
    'MgO')    --background material
```

Example 3: 1D Grating

```
-- Excitation
S:SetExcitationPlanewave(
    {0,0},  -- incidence angles (spherical coordinates: phi in [0,180], theta in [0,360])
    {1,0},  -- s-polarization amplitude and phase (in degrees)
    {1,0})  -- p-polarization amplitude and phase
```

Specify the excitation mechanism.

```
-- Sweep frequency
for freq=0.001,0.01,0.0005 do
    S:SetFrequency(freq)

    incidence_flux, reflection_flux = S:GetPoyntingFlux('AirAbove', 0)
    reflection_flux = (-1) * reflection_flux / incidence_flux;

    transmission_flux, back_incidence_flux = S:GetPoyntingFlux('MgOBelow', 0)
    transmission_flux = transmission_flux / incidence_flux

    print(freq .. '\t' .. reflection_flux .. '\t' .. transmission_flux);
end
```

Sweep over all frequencies.
Obtain and output results.

Wavelength = $100nm \sim 1000nm$
Wavelength in reduced unit = $100 \sim 1000$
Frequency in reduced unit = $0.01 \sim 0.001$

Additional Examples

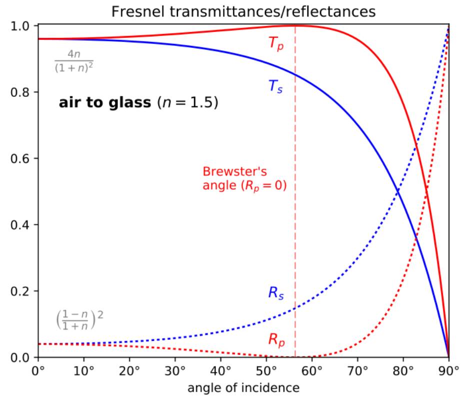
master / S4 / examples /

victorliu Reorganizing and adding examples.

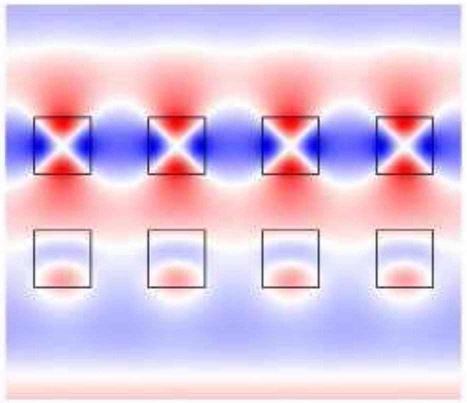
..
- 0d
- 1d
- 2d
- C_api
- Integrate
- MPI_example
- convergence
- interpolator
- magneto
- patterns
- polarization_basis
- simple

0d:
- Antonoyiannakis_PRB_60_1999
- Sakaguchi_OptComm_162_64_1999
- fabry_perot

fabry_perot:
- fresnel.lua
- slab_resonances.lua
- tir.lua
- tir_field.lua
- transmission_spectrum.lua



1d:
- Bi_OE_18_11969_2010
- Christ_PRB_70_125113_2004
- Liu_OE_17_21897_2009
- Oliva_OE_19_14735_2011
- Pietarinen_OE_14_2583_2006
- README



patterns:
- circle.lua
- composite.lua
- ellipse.lua
- nonorth.lua
- polygon.lua
- rectangle.lua

https://github.com/victorliu/S4/tree/master/examples

https://web.stanford.edu/group/fan/S4/lua_api.html

nanoHUB