

# Syntax of the input file

The structure to be simulated as well as all simulation parameters have to be specified in an XML file which can be edited by the users. The contents of such a file are described in the following. More advanced and optional features are also described in the [advanced\\_settings](#) page.

## General syntax

Quotation marks are necessary if an equal sign (=) is present, e.g. AlloyX="Barrier".

```
<?xml version="1.0" encoding="utf-8"?>  
  
<nextnano.NEGF Version="1.1.5">
```

## Comment Section

The Header contains a description of the input file, such as the name of the author of this input file or the version number of this input file. It is useful to add information that describes the content of the input file.

```
<Header>  
  <Author>Thomas Grange</Author>  
  <Content>This is an input file for the nextnano.NEGF code of Thomas  
Grange.  
  [YuSST2010]  
  Injector doping level-dependent continuous-wave operation of InP-based  
QCLs  
  at \lambda = 7.3 micrometer above room temperature  
  J. S. Yu, S. Slivken, M. Razeghi  
  Semiconductor Science and Technology 25, 125015 (2010)  
  http://dx.doi.org/10.1088/0268-1242/25/12/125015  
  </Content>  
</Header>
```

## Main part

### Voltage or temperature sweep (optional)

Here we define the type of sweep: either a voltage sweep (Voltage) or a temperature sweep (Temperature). We define the start and end of our voltage (or temperature) as well as its increment. The units are [mV/period] for the voltage sweep and [K] for the temperature sweep.

```
<SweepParameters>
```

```

    <SweepType>Voltage</SweepType> <!-- SweepType: "Voltage" or
"Temperature" -->
    <!-- Min, Max and delta values in mV for "IV", or in K for "Temperature"-
-->
    <Min>20</Min>    <!-- first/min value -->
    <Max>500</Max>    <!-- last/max value -->
    <Delta>20</Delta>    <!-- increment -->
</SweepParameters>

```

Alternatively a combined temperature-voltage sweep can be set by “Temperature-Voltage” (see the example of code below). In this case, the simulation can be parallelized. <Threads> defines the number of parallel threads (typically the number of CPU cores available). Within each parallel temperature sweep, a standard voltage sweep is performed.

```

    <SweepParameters>
<SweepType>Temperature-Voltage</SweepType>
    <MinV> 50</MinV>
    <MaxV> 60</MaxV>
    <DeltaV> 2</DeltaV>

    <MinT> 25</MinT>
    <MaxT> 300</MaxT>
    <DeltaT> 25</DeltaT>

    <Threads>12</Threads> <!-- Parallelization for Temperature-Voltage sweep
-->
</SweepParameters>

```

## Fixed temperature or voltage

A temperature of the lattice has to be specified (except in the case of a voltage sweep).

```
<Temperature Unit="K"> 300 </Temperature>
```

A potential drop per period has to be specified (except in the case of a temperature sweep).

```

<Bias>
    <PotentialDrop unit="meV">120</PotentialDrop>
</Bias>

```

## Crystal orientation, strain, piezoelectric and pyroelectric effects (optional)

In order to calculate crystal related properties such as strain piezo- and pyro-electric effects, information on the substrate material is required. The crystal structure can be Zincblende or Wurtzite. The lattice constants of the substrate are used to calculate strain. The (*hkl*) Miller indices are needed in order to define the orientation of the substrate on which the heterostructure is grown, i.e. the crystal coordinate system has to be rotated into the simulation coordinate system. The growth direction (simulation axis *z*) is perpendicular to the substrate (*xy* plane). The crystal and thus all

anisotropic material properties are rotated accordingly.

Strain can be included (yes) or excluded (no).

Piezo and pyroelectricity can be included (yes) or ignored (no). Pyroelectricity (also called spontaneous polarization) is only relevant for wurtzite crystals.

```
<Crystal>
  <CrystalStructure>Zincblende</CrystalStructure>

  <Orientation>
    <z_axis> <!-- The heterostructure growth direction is along z (simulation
direction). -->
      <h>0</h> <!-- (hkl) are the Miller indices of the plane
perpendicular to the z direction. -->
      <k>0</k>
      <l>1</l>
    </z_axis>

    <y_axis>
      <h>0</h> <!-- (hkl) are the Miller indices of the plane
perpendicular to the y direction. -->
      <k>1</k>
      <l>0</l>
    </y_axis>
  </Orientation>

  <Material_Substrate>                                <!-- Substrate material-->
    <Name>Al(x)Ga(1-x)As</Name>                        <!-- Al(x)Ga(1-x)As -->
    <Alloy_Composition> 0.15 </Alloy_Composition> <!-- x, i.e. Al0.15Ga0.85As
-->
  </Material_Substrate>

  <Strain>yes</Strain>                                <!-- include strain
(yes/no) -->
  <Piezoelectricity>yes</Piezoelectricity>
  <Pyroelectricity>yes</Pyroelectricity>
</Crystal>
```

If this section <Crystal> is absent from the input file, no strain, no piezo and no pyroelectricity will be taken into account.

## Material definition and parameters

```
<Materials>
```

### Definition of the materials

The various materials used in the heterostructure have to be declared.

The <Name> has to match one of the available materials that can be found in the [material database](#).

Each material has to be referred by an alias <Alias> (e.g. 'barrier', 'well', ...) which will be used to refer to the material in the following of the input file.

If <Effective\_mass\_from\_kp\_parameters> is yes, the effective mass for the material will be calculated from the k.p parameters in the database. If no, the electron mass will be taken from <ElectronMass Unit="m0"> in the database.

```
<Material>
  <Name>GaAs</Name>                                <!-- Binary material -->
  <Alias>well</Alias>
<Effective_mass_from_kp_parameters>yes</Effective_mass_from_kp_parameters>
</Material>

<Material>
  <Name>Al(x)Ga(1-x)As</Name>                        <!-- Ternary material -->
  <Alloy_Composition>0.15</Alloy_Composition> <!-- alloy composition x -
->
  <Alias>barrier1</Alias>
<Effective_mass_from_kp_parameters>yes</Effective_mass_from_kp_parameters>
</Material>

<Material>
  <Name>Al(x)In(y)Ga(1-x-y)N</Name>                  <!-- Quaternary material -
->
  <Alloy_Composition>0.16</Alloy_Composition> <!-- the first alloy
composition represents x -->
  <Alloy_Composition>0.04</Alloy_Composition> <!-- the second alloy
composition represents y -->
  <Alias>barrier2</Alias>
<Effective_mass_from_kp_parameters>yes</Effective_mass_from_kp_parameters>
</Material>
```

For each material (binary, ternary alloy, quaternary alloy), any parameter calculated from the database can be overwritten in the input file by using the <Overwrite> command. In the example below, conduction band offsets and effective electron masses are overwritten for each of the two materials:

```
<Material>
  <Name>GaAs</Name>
  <Alias>well</Alias>
<Effective_mass_from_kp_parameters>no</Effective_mass_from_kp_parameters>
  <Overwrite>
    <ConductionBandOffset Unit="eV">0.0</ConductionBandOffset>
    <ElectronMass Unit="m0">0.07</ElectronMass>
  </Overwrite>
</Material>
```

```

<Material>
  <Name>Al(x)Ga(1-x)As</Name>
  <Alloy_Composition>0.15</Alloy_Composition>
  <Alias>barrier</Alias>
<Effective_mass_from_kp_parameters>no</Effective_mass_from_kp_parameters>
  <Overwrite>
    <ConductionBandOffset Unit="eV">0.135</ConductionBandOffset>
    <ElectronMass Unit="m0">0.08</ElectronMass>
  </Overwrite>
</Material>

```

Note that `<Overwrite>` has higher priority than `<Overwrite_Material_Database>` (see below). Also note that `<Overwrite>` will overwrite directly the material parameters of an alloy for the composition specified in the input file, while `<Overwrite_Material_Database>` will overwrite the bowing parameters.

## Conduction or valence band offsets

```
<UseConductionBandOffset>yes</UseConductionBandOffset>
```

- If yes, then the conduction band offsets *as defined in the database file or overwritten in the input file* (`Material_Database.xml`) are used.
- If no, then the conduction band offsets are calculated from the valence band offsets, split-off energies and temperature dependent band gaps. This corresponds to the implementation of the nextnano++ software.

In both cases, the conduction band offsets get an additional shift if strain is present.

## Nonparabolicity

There are two models for the effective electron mass.

- a) parabolic effective mass: The effective mass  $m^*$  is independent of energy. This simple model can be sufficient for heterostructures with small conduction band offsets with respect to their bandgaps.

```
<NonParabolicity>no</NonParabolicity>
```

- b) nonparabolic effective mass: The effective mass  $m(E)$  depends on energy  $E$ . This is more realistic model is recommended in general.

```
<NonParabolicity>yes</NonParabolicity>
```

The nonparabolic model requires iteratively solving the Schrödinger equation in contrast to the parabolic model.

```
</Materials>
```

## Overwriting Material Database

One can overwrite the default material parameters from the material database file `Material_Database.xml`.

```
<Overwrite_Material_Database>
  <Material>
    <Name>GaAs</Name>
    <ConductionBandOffset Unit="eV">2.979</ConductionBandOffset>
    <BandGap Unit="eV">1.519</BandGap>
    <EpsStatic>12.93</EpsStatic>
  </Material>
</Overwrite_Material_Database>
```

Note that `<Overwrite>` within the `<Materials>` section has higher priority than `<Overwrite_Material_Database>`.

## Definition of the struture

Within the `<Superlattice>` section, the relevant region of the device is defined. Periodic boundary condition are applied to the defined region.

```
<Superlattice>
```

## Layers and thicknesses

The layer(s) of the structure have to be defined.

```
<Layer> <!-- #1 -->
  <Material>barrier</Material>
  <Thickness unit="nm">4.1</Thickness>
</Layer>

<Layer> <!-- #2 -->
  <Material>well</Material>
  <Thickness unit="nm">16.0</Thickness>
</Layer>
<Layer> <!-- #3 -->
  <Material>barrier</Material>
  <Thickness unit="nm">4.3</Thickness>
</Layer>

<Layer> <!-- #4 -->
  <Material>well</Material>
  <Thickness unit="nm">8.9</Thickness>
</Layer>
```

```

    <Layer> <!-- #5 -->
    ...
</Layer>

    <Layer> <!-- #6 -->
    ...
</Layer>

```

## Doping

We can specify a homogeneous doping between starting and end point. The doping can be in barriers or wells or both. Origin = start of first barrier. The units for `<Doping_Density>` are according to `<Doping_Specification>` where an integer is used in order to choose how the doping density is specified:

- = 2D equivalent density per period in [ $\text{cm}^{-2}$ ]
- 1 = 3D doping density in the doped region in [ $\text{cm}^{-3}$ ]
- 2 = Averaged 3D doping density over the whole structure in [ $\text{cm}^{-3}$ ]

```

<Doping>

  <DopingStart unit="nm"> 1.4 </DopingStart> <!-- with respect to origin -
->
  <DopingEnd    unit="nm"> 9.8 </DopingEnd>    <!-- with respect to origin -
->

  <Doping_Specification> 0 </Doping_Specification>
  <Doping_Density> 16.8e10 </Doping_Density>

</Doping>

```

One can have several doping regions in the structure.

```

<Doping> doping #1
  <DopingStart unit="nm"> 1.4 </DopingStart> <!-- with respect to origin -
->
  <DopingEnd    unit="nm"> 9.8 </DopingEnd>    <!-- with respect to origin -
->
  <Doping_Specification> 0 </Doping_Specification>
  <Doping_Density> 16.8e10 </Doping_Density>
</Doping>
<Doping> doping #2
  <DopingStart unit="nm"> ... </DopingStart> <!-- with respect to origin -
->
  <DopingEnd    unit="nm"> ... </DopingEnd>    <!-- with respect to origin -
->
  <Doping_Specification> 0 </Doping_Specification>
  <Doping_Density> 16.8e10 </Doping_Density>
</Doping>

```

Note that the doping densities should be realistic (of the order  $1e10 \text{ cm}^{-2}$  instead of  $0.1 \text{ cm}^{-2}$ ) to produce broadening. Intrinsic doping is not yet implemented.

### Definition of a 'Tight-Binding' basis for the analysis (optional)

In order to analyze the results, the user can define a 'tight-binding' basis by specifying modules in which the wavefunctions will be confined. To this purpose, the user has to define separations using the command `<Analysis_Separator>` (see example below). Each separator has the effect of decoupling the wavefunctions from the two sides. Between each separator, a distinct module is defined in which the Schrödinger equation will be solved separately. In other words, the wavefunctions of the tight-binding basis are not eigenstates of the full structure, but they are instead eigenstates on each separated module. The new set of basis states will be displayed in a folder named "TightBinding".

For example, in the case of a QCL with an injector barrier and a collector barrier, one can define two separators in the following way.

```
<Analysis_Separator>
  <Separator_Position>2.0</Separator_Position> <!-- position of the
collector barrier with respect to origin -->
</Analysis_Separator>

<Analysis_Separator>
  <Separator_Position>22.0</Separator_Position> <!-- position of the
injector barrier with respect to origin -->
</Analysis_Separator>
```

In this case there will be two defined regions per period. The laser states will be defined in between these two separators, while other states will be defined outside. An arbitrary number of analysis separators can be defined (the minimum is one). In general, it is recommended to define a separator for each thick barrier at which a resonant tunneling process is expected. This kind of "tight-binding basis" has been discussed by Sushil Kumar and Qing Hu in [Phys. Rev. B 80, 245316](#) (see Fig. 1b).

```
</Superlattice>
```

### Scattering processes

Here the parameters for scattering processes are defined.

```
<Scattering>
```

Material parameters for the scattering processes are taken from a single material (optical phonon energy, dielectric constants, mass density, sound speed...). This material has to be specified using `<Material_for_scattering_parameters>`.

```
<Material_for_scattering_parameters>well</Material_for_scattering_parameters>
```

## Interface roughness scattering

For the interface auto correlation type, two options are possible.

- = exponential:  $\delta^2 \exp(-\frac{r}{\lambda})$
- 1 = Gaussian:  $\delta^2 \exp(-\frac{r^2}{\lambda^2})$

```
<Interface_Roughness>
  <Amplitude_in_Z unit="nm"> 0.1 </Amplitude_in_Z>

  <InterfaceAutoCorrelationType> 0 </InterfaceAutoCorrelationType>
  <Correlation_Length_in_XY unit="nm"> 8 </Correlation_Length_in_XY>

</Interface_Roughness>
```

Recommendation: One may play with the interface roughness, for which mid-IR QCLs are very sensitive.

To switch off interface roughness scattering, one has to specify:

```
<Amplitude_in_Z unit="nm"> 0 </Amplitude_in_Z>
```

Currently it is required additionally that

```
<Correlation_Length_in_XY unit="nm"> 8 </Correlation_Length_in_XY>
```

must be a nonzero value even in case interface roughness is switched off by using

```
<Amplitude_in_Z unit="nm"> 0 </Amplitude_in_Z>
```

## Diffuse interfaces

The interfacial width, as defined in our paper [Phys. Rev. Applied 13, 044062 \(2020\)](#), can be specified in the following way:

```
<Interface_Roughness>
  <InterfaceWidth unit="nm">0.8</InterfaceWidth>
  <Amplitude_in_Z unit="nm">0.12</Amplitude_in_Z>
  <InterfaceAutoCorrelationType>1</InterfaceAutoCorrelationType> <!--
Correlation type: 0=Exponential, 1=Gaussian, 2 = Generalized expression with
Hurst parameter -->
  <Correlation_Length_in_XY unit="nm">8.0</Correlation_Length_in_XY>
  <AxialCorrelationLength>0.4</AxialCorrelationLength>
</Interface_Roughness>
```

Note that in this case, the axial correlation length is needed to fully specified the interface roughness. Otherwise, if not specified or set to zero, the standard approach for calculating interface roughness scattering is used.

## Threading dislocations

To be documented

## Acoustic phonons

Comment: Acoustic phonons are in general not efficient. They can be neglected in most cases. The maximum acoustic phonon energy, `<AcousticPhonon_Scattering_EnergyMax>`, is the maximum energy acoustic phonons efficiently couple to electrons, which is 2 or 3 meV.

```
<AcousticPhonon_Scattering> no </AcousticPhonon_Scattering>
<AcousticPhonon_Scattering_EnergyMax unit="meV"> 2.5
</AcousticPhonon_Scattering_EnergyMax>
```

## Optical phonons

(optional) Sometimes it might be useful to investigate the influence of the longitudinal-optical (LO) phonon interaction on the full NEGF simulation. Then one can artificially tune the LO-phonon scattering rates (self-energies in practice, or equivalently the square of the coupling to LO-phonons). The microscopic value of the square of the LO phonon matrix element is then multiplied by the scaling factor `<LO_Phonon_Coupling_Strength>`.

```
<Tune_LO_Phonon_Scattering>yes</Tune_LO_Phonon_Scattering>
  <!-- tune the coupling constant due to LO phonons (Fröhlich coupling
constant)
      from the microscopically calculated value -->

  <LO_Phonon_Coupling_Strength>1.0</LO_Phonon_Coupling_Strength>
      <!-- 1.0 is the value of the normal calculation -
->
```

A value of 1.0 will have no effect, while larger (smaller) values will increase (decrease) the LO-phonon self-energy proportionally respectively.

## Charged impurities

There are 3 models available for the effective temperature of the electrons involved in electrostatic screening, namely 1, 2 and 3 according to the desired model.

```
<Model_Temperature_for_Screening> 1 </Model_Temperature_for_Screening>
```

The screening involved in charged impurity scattering is modeled by a homogeneous electron gas, with a temperature taken as the average electron temperature (model 2) for screening (self-consistent calculation), i.e. the screening temperature is equal to the calculated average electron temperature. (The “effective electronic temperature” is calculated in each iteration and this value is used as the screening temperature). For model 1 and 3 the screening temperature  $T_{\text{eff}}$

does not necessarily match the calculated average electron temperature. The “effective electronic temperature” is written to the file `Current-miscellaneous.txt`.

- model 1:  $T_{\text{eff}} = T + T_{\text{offset}} \exp(-\frac{T}{T_{\text{offset}}})$  with  $T_{\text{offset}}$  specified as `<Temperature_Offset_parameter>`  
We found that 140 K is a reasonable, empirically found value that is suited for both MIR and THz QCLs.

```
<Temperature_Offset_parameter> 140 </Temperature_Offset_parameter>
<!-- T_offset (model #1 only) -->
```

- model 2: self-consistent calculation until the effective temperature converges below the desired accuracy (requires several iterations of the calculation). The accuracy is specified by `<Accuracy_Self_consistent_Electron_Temperature>`.

```
<Accuracy_Self_consistent_Electron_Temperature> 0.05
</Accuracy_Self_consistent_Electron_Temperature>
<!-- self-consistent calculation (model #2 only) -->
```

- model 3: The effective temperature  $T_{\text{eff}}$  is directly specified by `<Electron_Temperature_for_Screening>`.

```
<Electron_Temperature_for_Screening> 300
</Electron_Temperature_for_Screening>
<!-- (model #3 only) -->
```

```
<ImpurityScattering_Strength> 1.0 </ImpurityScattering_Strength>
```

For the impurity scattering strength, 1.0 is the normal physical calculation. Other values may be used for testing the importance of impurity scattering. It should be 1.0, unless one wants to test the influence of a reduced ( $< 1.0$ ) or enhanced ( $> 1.0$ ) impurity scattering strength.

## Alloy scattering

Alloy scattering can be considered using the following command.

```
<Alloy_scattering>yes</Alloy_scattering>
```

Note that alloy scattering is only considered for the material specified by `<Material_for_scattering_parameters>`.

By default the following alloy squared matrix element for a zinc-blende ternary  $A_xB_{1-x}C$  is considered: 
$$\left| \langle \alpha, k_0 | H_{\text{alloy}} | \beta, k_0 + k \rangle \right|^2 = \int dz \frac{x(1-x) a^3 (\Delta E_c)^2}{4} |\phi_{\alpha}(z) \phi_{\beta}(z)|^2$$
 where  $\Delta E_c$  is the conduction band offset between the binary compounds  $AC$  and  $BC$ :  $\Delta E_c = E_c(AB) - E_c(BC)$ .

This squared matrix element can be tuned with respect to the above formula by using the following command (tuning the squared matrix element is equivalent of tuning the scattering rate):

```
<Alloy_scattering>yes</Alloy_scattering>
<Tune_Alloy_scattering>0.5</Tune_Alloy_scattering>
```

## Electron-electron scattering

Electron-electron scattering processes are considered in the elastic approximation.

```
<Electron_Electron_Scattering>yes</Electron_Electron_Scattering>
```

Optionally, one can tune the strength of the electron-electron scattering mechanism using the additional commands

```
<Tune_Elect_Elect_Scattering_Strength>yes</Tune_Elect_Elect_Scattering_Stren
gth>
<Elect_Elect_Scattering_Strength>1.5</Elect_Elect_Scattering_Strength>
```

A value of 1.0 gives the normal calculation.

```
</Scattering>
```

## Poisson equation

The electrostatic mean-field interactions (electron-electron and electron-impurities interactions) can be switched on/off by using yes or no in the `<Poisson>` command.

```
<Poisson> yes </Poisson>
```

## Lateral motion in the directions perpendicular to the transport

The parameters from the lateral motion (i.e. the two-dimensional in-plane motion in the directions perpendicular to the growth axis) are assumed to be homogeneous along the structure. Hence the parameters for the lateral motion have to be taken from a fixed material. This material is indicated by `<Material_for_lateral_motion>`.

Then the discretization energy for this lateral motion has to be specified with `<Value unit="meV">`.

```
<Lateral_motion>
  <Material_for_lateral_motion>well</Material_for_lateral_motion>
  <Value unit="meV">5</Value>
</Lateral_motion>
```

This former command sets the energy spacing between the ground and first excited state for the in-plane motion. Note there is a further parameter for the in-plane motion `<Energy_Range_Lateral>`, in the `<Simulation_Parameter>` section described below, which sets the cut-off energy (i.e. the energy range) for the subband dispersion.

## Simulation parameters

```
<Simulation_Parameter>
  <Coherence_length_in_Periods> 1 </Coherence_length_in_Periods>
```

<Coherence\_length\_in\_Periods> corresponds to the accounted coherence length of electrons in numbers of period. 1 should be enough for typical QCLs.

- 1  $\Rightarrow$  coherent transport from one period to the next
- N  $\Rightarrow$  coherent transport between N+1 periods

In almost all existing QCL designs, 1 is enough. The user should not change this value. (Larger number are needed for superlattices.

The coherence length for which self-energies are considered can be further limit by using the following command:

```
<Simulation_Parameter>
<Coherence_length_in_nm>50.0</Coherence_length_in_nm>
...
</Simulation_Parameter>
```

Such limitation will speed up the calculation of the self-energies. On the other hand, this can reduce the accuracy of the scattering processes if this length is below the actual coherence length in the simulated device. It is recommended to asses the accuracy of such approximation by comparing to the results for different values of this coherence length to the full calculation.

## Basis state calculation

These parameters only affect the basis state calculation time. They are not critical for the total calculation time.

The spatial grid defines the number of grid points per period.

```
<Spatial_grid_spacing unit="nm"> 0.2 </Spatial_grid_spacing>
```

The number of periods used for basis state calculation only: corresponds to  $2N+1$  k points per miniband

```
<Number_of_lateral_periods_for_band_structure> 4
</Number_of_lateral_periods_for_band_structure>
```

This is the number of periods in the Schrödinger equation. The Schrödinger equation is solved in the beginning to determine the modes that are input to the NEGF core algorithm.) number of grid points/period \*  $(2N+1)$  should stay below  $\sim 10^4$  for fast calculation time. A larger number of periods, <Number\_of\_lateral\_periods\_for\_band\_structure>, can be used for calculating accurately the electronic band structure (energy levels and wave functions in band edge profile) that is plotted in the file Wannier-Stark\_levels.dat.

## Energy grid

The energy grid spacing (which is always a homogeneously spaced grid) is set using the following command:

```
<Energy_grid_spacing unit="meV"> 2 </Energy_grid_spacing>
```

Note that this choice is critical for the NEGF calculation time: the energy grid needs to be fine enough to resolve the peaks of the Green's functions. However, a finer energy grid will increase the computation time.

## Cut-off energies

Two different cut-off energies need to be defined in the input file.

The axial cut-off energy is related to the motion along the heterostructure axis (i.e.  $z$  direction, growth direction). The command

```
<Energy_Range_Axial unit="meV"> 500 </Energy_Range_Axial>
```

sets a cut-off energy which determines how many states/subbands are accounted. This cut-off energy is measured from the lowest energy state (i.e. the lowest miniband for periodic structures).

The lateral (i.e. in-plane) energy cut-off is related to the description of the in-plane dispersion of the subbands (i.e.  $(x, y)$  directions). The command

```
<Energy_Range_Lateral unit="meV"> 300 </Energy_Range_Lateral>
```

sets the range of the subband dispersion which are accounted, measured from the bottom of each subband. This range has to be large enough to describe the elastic scattering processes. In a QCL, it has to be larger than the lasing transition energy.

The two cut-off energies are critical in both the accuracy of the results and in the computation time.

## Adjusting the energy range of the Green's functions

The energy range of the Green's functions are set automatically using the following default values:  $E_{\text{min}} = E_{\text{(lowest electronic state)}} - E_{\text{cut-off}}$   $E_{\text{max}} = E_{\text{(highest electronic state)}} + E_{\text{cut-off}}$  where  $E_{\text{cut-off}}$  is the maximum of axial and lateral energy cut-offs.

However, if needed, this default energy range can be modified by using the commands `<Emin_shift>` and/or `<Emax_shift>`. To modify the minimum energy, the following command can be used

```
<Emin_shift unit="meV"> 0 </Emin_shift>
```

where 0 is the default value - a negative value increases the energy range of the Green's functions

towards lower energies. This energy shift is made with respect to the value  $E_{\text{min}}$   $\Rightarrow$   $E_{\text{min}} + \text{<Emin\_shift unit="meV">}$

For shifting the maximum energy limit, the following command can be used

```
<Emax_shift unit="meV"> 0 </Emax_shift>
```

where is the default value - a positive value increases the energy range of the Green's functions towards higher energies. This energy shift is made with respect to the value  $E_{\text{max}}$   $\Rightarrow$   $E_{\text{max}} + \text{<Emax\_shift unit="meV">}$

To know whether changing the default setting of the energy range of the Green's functions can be done or is needed, one has to look at the output of the spectral functions (see the example in the [FAQ](#))

## Convergence factors for the Green's functions

The NEGF self-consistent loop ends successfully if the following 2 convergence factors are reached for the lesser Green's function (GF) and the current (relative difference between two consecutive iterations).

```
<Convergence_Value_GF>      1e-4 </Convergence_Value_GF>
<Convergence_Value_Current> 1e-4 </Convergence_Value_Current>
```

A number of maximum iterations is specified in case the above convergence values are not reached.

```
<N_max_iterations> 300 </N_max_iterations>
```

Small values for the convergence values, together with sufficiently large value for the maximum number of iteration, will give the more accurate results.

```
</Simulation_Parameter>
```

## Output

In order to output 2D energy resolved graphs for the local density of states and the energy resolved electron density some flags are necessary. Also, the energy resolved gain for a specified photon energy has to be specified.

```
<Output>
  <EnergyResolvedPlots> yes </EnergyResolvedPlots>
  <EnergyResolvedPlots_MinimumEnergyGridSpacing> 0.5
</EnergyResolvedPlots_MinimumEnergyGridSpacing>

  <EnergyResolved_Gain> yes </EnergyResolved_Gain>
  <EnergyResolved_Gain_PhotonEnergy unit="meV"> 15
</EnergyResolved_Gain_PhotonEnergy>
```

</Output>

## Gain (optional)

Choose a method how the gain should be calculated:

- -1 no gain calculation
- gain without self-consistency only
- 1 gain with self-consistency only
- 2 gain with and without self-consistency

```
<Gain>
  <GainMethod> 1 </GainMethod>
```

Once the calculation for a particular bias point has converged, the gain is calculated. The difference between self-consistent and nonself-consistent calculations is discussed in the papers of A. Wacker et al., Phys. Rev. B 66, 085326 (2002), and APL 86, 04110 (2005). The self-consistent gain calculation is needed when intrasubband scattering processes are important, which is the case in THz QCLs. In mid-infrared QCLs, the gain calculation without self-consistency is found to be sufficient, and is much less time consuming.

The energy spacing between two photon energies is given by dE\_Phot and dE\_Phot\_Self\_Consistent.

without self-consistency:

```
<dE_Phot          unit="meV"> 2.0 </dE_Phot>
<Ephoton_Min      unit="meV"> 140 </Ephoton_Min>
<Ephoton_Max      unit="meV"> 170 </Ephoton_Max>
```

with self-consistency:

```
<dE_Phot_Self_Consistent unit="meV"> 2.0 </dE_Phot_Self_Consistent>
<Ephoton_Min_Self_Consistent unit="meV"> 140
</Ephoton_Min_Self_Consistent>
<Ephoton_Max_Self_Consistent unit="meV"> 170
</Ephoton_Max_Self_Consistent>

<MaxNumber_SelfConsistent_Iterations> 50
</MaxNumber_SelfConsistent_Iterations>
<ConvergenceFactor_Gain_SelfConsistent> 0.001
</ConvergenceFactor_Gain_SelfConsistent>
```

Calculation of gain only between the following values of potential drop per period (in order to save CPU time)

```
<Vmin unit="mV"> 160 </Vmin>
<Vmax unit="mV"> 400 </Vmax>
</Gain>
```

## Threshold

Within this gain section, a threshold option is possible in the following way:

```
<Gain>
...
<Cavity_Losses unit="cm^{-1}">5</Cavity_Losses>
<Threshold>yes</Threshold>
...
</Gain>
```

The cavity losses need to be specified by the `<Cavity_Losses>` command (unit  $\text{cm}^{-1}$ ). In this case, if the `<Threshold>yes</Threshold>` option is activated, the voltage sweep is performed until the threshold condition is fulfilled (i.e. the maximum gain matches the cavity losses). The threshold bias and threshold current are then interpolated based on the calculated bias points.

In the case of a combined temperature-voltage sweep, the value of the last calculated bias value below threshold bias is used as a starting point of the following bias sweep at the next temperature.

From:

<https://nextnano-docu.northeurope.cloudapp.azure.com/dokuwiki/> - **nextnano.NEGF - Software for Quantum Transport**

Permanent link:

[https://nextnano-docu.northeurope.cloudapp.azure.com/dokuwiki/doku.php?id=qcl:input\\_file](https://nextnano-docu.northeurope.cloudapp.azure.com/dokuwiki/doku.php?id=qcl:input_file)



Last update: **2023/01/05 10:07**