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, e.g. THz_QCL_GaAs_AlGaAs_Fathololoumi_OptExpress2012_10K.xml, are described in the following.

General syntax

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

```
<?xml version="1.0" encoding="utf-8"?>

<nextnano.QCL 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.QCL code of Thomas Grange.
   [YuSST2010]
   Injector doping level-dependent continous-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</u> <!-- (hkl) are the Miller indices of the plane
perpendicular to the y direction. -->
    <k>1</v>
    <l>0</w>
   <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 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 `<OverwriteMaterial>`.

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 repect 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 [cm^{-2}]
- 1 = 3D doping density in the doped region in [cm^{-3}]
- 2 = Averaged 3D doping density over the whole structure in [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 cm^{-2} instead of 0.1 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>
```

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.

```
<Acoustic_Phonon_Scattering> no </Acoustic_Phonon_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 in more detail. Then one can artificially tune the coupling strength due to LO phonons. The microscopic value of the LO phonon coupling strength 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 -
->
```

Charged impurities

There are 3 models available for the effective temperature of the electrons involved in electrostatic screeening, 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 $T_{\rm eff}$ does not necessarily match the calculated average electron temperature. The "effective electronic temperature" is written to the file Current-miscellaneous.txt.

- model 1: $T_{\rm eff} = T + T_{\rm offset} \exp(-\frac{T}{T_{\rm offset}})$ with $T_{\rm offset}$ 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 convergences 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 $T_{\rm eff}$ is directly specified by <Electron_Temperature_for_Screening>.

```
  <Electron_Temperature_for_Screening> 300
</Electron_Temperature_for_Screening>
```

```
    <!-- (model #3 only) -->

    <Phenomenological_Electron_Temperature> no
</Phenomenological_Electron_Temperature>

    <Self_consistent_Electron_Temperature> no
</Self_consistent_Electron_Temperature>
```

```
    <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>.

```
    </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)
```

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 free 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">`.

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

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 =⇒ coherent transport from one period to the next
- N =⇒ 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.

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 ~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 is critical for the calculation time. It is a homogeneously spaced grid. It holds for higher temperatures: More broadening, i.e. less energy grid points are sufficient.

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

Adjusting the energy range of the Green's functions

(`<Emin_shift>`, `<Emax_shift>`)

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

0 is the default value - a negative value increases the energy range of the Green's functions towards low energies. This energy shift is with respect to $E_{\rm min}$ = Energy(lower Wannier-Stark state) - `Energy_Range_Axial`.

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

0 is the default value - a positive value increases the energy range of the Green's functions towards high energies. This energy shift is with respect to $E_{\rm max}$ = Energy(higher Wannier-Stark state) + Energy(higher lateral state).

The FAQ contains an example of different choices of `<Emin_shift>` and `<Emax_shift>`.

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

xy direction, evaluated from lowest of one state

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

z direction, evaluated from lowest state/miniband of one period.

The 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>      0.0005 </Convergence_Value_GF>
<Convergence_Value_Current> 0.0005 </Convergence_Value_Current>
```

```
<N_max_iterations> 400 </N_max_iterations>
```

Number of maximum iterations if the above convergence values are not reached. Higher values give more accurate results.

```
<Convergence_Minimum> 0.01 </Convergence_Minimum>
```

Minimum convergence factor for the Green's function to be considered as a valid result (if
`N_max_iterations` is reached without satisfying `Convergence_Value_GF`). Above the output for
the current will be -1.

The number of threads used during the simulation can be limited by the following input. For an
automatic setting, use 0.

```
<Maximum_Number_of_Threads>7</Maximum_Number_of_Threads>
```

```
</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> 0 </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>
```