
nextnano

Software for semiconductor nanodevices

nextnanomat Documentation

Release 5.6.4.6

nextnano GmbH

Jan 06, 2026

GETTING STARTED

1	GUI tabs	3
1.1	Input	3
1.1.1	General features of the Input tab	3
1.1.2	Example and New Input Files	3
1.1.3	Create new Input File with predefined Structure	4
1.1.4	Autocomplete feature	4
1.2	Template	6
1.2.1	Variable definition in input file (required)	6
1.2.2	Sweep over a variable + optional post-processing	6
1.2.3	Post-processing of existing sweep	8
1.3	Template (Beta)	8
1.3.1	Introduction	8
1.3.2	The input file	8
1.3.3	How to create a sweep	10
1.3.4	Save input files and run a sweep	10
1.4	Simulation	12
1.5	Output	13
1.5.1	Navigation	14
1.5.2	Visualization	15
1.5.3	Features	16
1.5.4	Export functionality	22
1.5.5	Further information	22
2	Main menu functions	25
2.1	File	25
2.1.1	Create New Input File	25
2.2	Edit	25
2.2.1	Replace existing Tabs with Spaces	25
2.2.2	Replace IF with WHEN	25
2.2.3	Open Keyword Tree, Syntax Help	25
2.3	Run	29
2.3.1	Add Directory Tree to Batch List	29
2.3.2	Softkill Selected Simulation	29
2.3.3	Check Simulations for Termination (beta)	29
2.4	View	30
2.5	Tools	30
2.5.1	Activate license	30
2.5.2	Convert nextnano ³ Input File to nextnano++	30
2.5.3	Convert xml Input File to negf Input File	31
2.5.4	Clean Up Simulation Output Folder	31
2.5.5	Generate nextnanopy config file	31
2.5.6	Generate System Snapshot for Troubleshooting	33
2.6	Help	33

3	Settings	35
3.1	Workspace	35
3.1.1	Simulation	35
3.1.2	Executable for Windows and Linux	35
3.1.3	Material database	36
3.1.4	License	36
3.2	Preferences	36
3.2.1	Editor/Input	36
3.2.2	View	36
3.2.3	Gnuplot export	36
3.2.4	Data export	39
3.3	Advanced	42
3.3.1	Performance	42
3.3.2	Expert/Debugging	42
3.3.3	Custom executable	42
4	Color maps	45
4.1	Introduction	45
4.2	Implementation	45
4.3	References	45
5	Export Functionalities	47
5.1	Exporting to ParaView	47
5.1.1	Setting Up nextnanomat	47
5.1.2	Direct Export	48
5.1.3	Exporting 2D Outputs with Python Scripts	49
5.2	Gnuplot export	63
5.2.1	How to export 1D plot files with Gnuplot	63
5.2.2	How to export 2D or 3D plot files with Gnuplot	64
5.2.3	How to optimize the looks of a Gnuplot graph	64
5.3	Export via python scripts	68
5.3.1	How to call custom python scripts directly from nextnanomat	68
5.4	Other	68
6	HTCondor	71
6.1	HTCondor on nextnanomat	71
6.2	Recommended Installation Process	71
6.2.1	Summary of settings (Example)	73
6.2.2	Config file	73
6.2.3	Configuring a pool without a domain	74
6.3	Submitting jobs to HTCondor pool with nextnanomat	77
6.4	Useful HTCondor commands for the Command Prompt	78
6.5	HTCondor Pool - Managing Slots	78
6.5.1	Dynamic slots	79
6.6	Machine states	80
6.7	Machine activities	80
6.8	Configuration options for the Central Manager computer	80
6.9	FAQ	80
6.10	Problems with HTCondor	81
6.10.1	Error: communication error	81
6.10.2	Error: condor_store_cred add failed with Operation failed. Make sure your ALLOW_WRITE setting include this host.	81
6.10.3	Error? Check the Log files	81
6.11	Known bugs	82
6.12	Run your custom executable on HTCondor with nextnanomat	82
6.12.1	Input file identifier	82
6.12.2	Settings for Hello World (HW)	82
6.12.3	Settings for Quantum ESPRESSO (QE)	82
6.12.4	Settings for ABINIT	83

7	Release Notes	85
7.1	5.6.4.6 (2025-12-17)	85
7.2	5.2.13.0 (2025-06-20)	86
7.3	5.2.9.0 (2025-05-30)	86
7.4	4.3.2.15 (2024-03-28)	86
7.5	4.3.2.8 (2023-08-07)	87
7.6	4.3.1.8 (2022-08-05)	88
7.7	4.2.8.13 (2021-07-05)	89
7.8	4.2.8.7 (2021-02-05)	89
7.9	4.2.8.6 (2020-12-09)	90
8	FAQs	91
8.1	Features and Functionalities	91
8.1.1	How do I produce 1D slices through the 2D plots in the GUI?	91
8.1.2	Is there a way to produce a 1D (or 2D) plot of some result, for example the probability density with the conduction band edge superimposed? In other words, can the GUI show multiple plots at once?	91
8.1.3	What is the difference between “List view” and “Tree view”?	91
8.1.4	How to prevent fonts from changing back to Calibri in the input tab?	92
8.1.5	How can I add additional debug information to the LOG file?	92
8.1.6	Can I pass additional command line arguments to the executables of simulation tools?	92
8.2	Error Messages	93
8.2.1	Error: File format is not valid	93
8.2.2	Error: The specified executable is not a valid application for this OS platform.	95
8.2.3	<i>Licensing.dll</i> cannot be found	95
8.2.4	Error during license activation	95
8.2.5	Unverified license	95
8.3	Problems and Limitations	96
8.3.1	The Window of the graphical user interface gets (too) small	96
8.3.2	My simulations are still shown as running, although they are finished	96

The *nextnanomat* GUI is a convenient graphical user interface for *nextnano++*, *nextnano*³, and *nextnano.NEGF*. It allows for 1D, 2D and 3D visualizations of computed results. This workflow manager is designed in a general way allowing to run any executables and overview generated output files.

GUI TABS

1.1 Input

- *General features of the Input tab*
- *Example and New Input Files*
- *Create new Input File with predefined Structure*
- *Autocomplete feature*

Within the Input tab, you can visualize and edit the input file required for starting a simulation. Both ASCII and XML format are supported. The input file specifies all information required for the simulation tool. E.g. the material, the structure, which types of calculations and boundaries, which output files and their format. . .

The syntax and the supported features depend on the tool used for calculations. For detailed information see related documentation:

- `nextnano++`
- `nextnano3`
- `nextnano.MSB`
- `nextnano.NEGF`

1.1.1 General features of the Input tab

- Autocomplete
- Context-sensitive online help
- Editing
- Multiple input tabs
- Printing
- Search in File
- Syntax highlighting
- Use of variables

1.1.2 Example and New Input Files

Each installation of the nextnano software comes with an extensive set of example input files for each nextnano tool of that installed package. In most cases these example input files are used to learn about the software and the syntax. Small adjustments to the example files will suffice to adapt them to your simulation requirements and device's structure. The structure definition and material database offer high flexibility and customization options and will certainly cover your desired device.

Instead of adapting an example input file, advanced users can write their file from scratch, generate it with *nextnanopy* or use our set of template input files.

1.1.3 Create new Input File with predefined Structure

By using our set of template input files, a new and clean input file will be created - with a simple predefined structure. This is the perfect starting point for a highly customized input file for setting up the simulation twin of your device and research. The basic structure of the template input file contains all **required** blocks for the simulation, but no additional details like our example input files do. So adjustments will be easier. An Example is shown in [Figure 1.1.3.2](#). When creating such a new file you can select the nextnano tool, the crystal structure and the dimensionality of your new simulation, see [Figure 1.1.3.1](#).

Note

The template input files are searched for in a directory relative to the executable of the defined nextnano tool. Thus, this feature only works if the paths to the executables are specified within Tools - Options. This way the syntax of the newly created input file is up to date with the executable version used for the simulation.

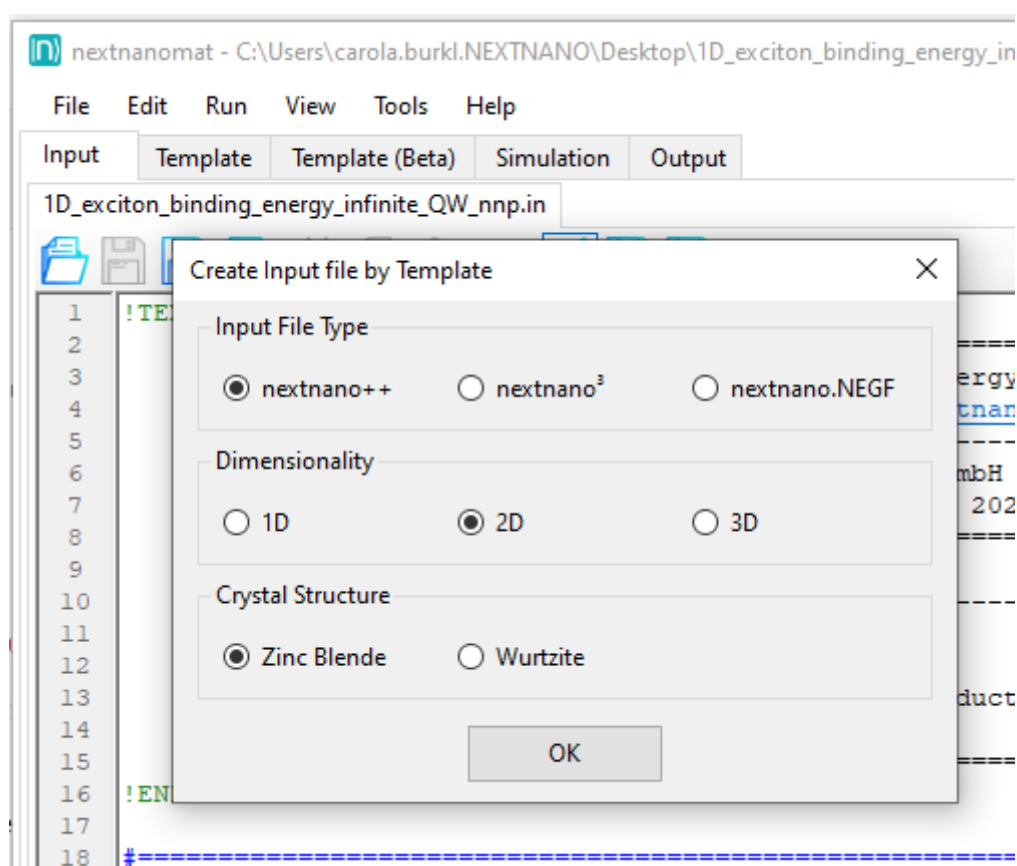


Figure 1.1.3.1: Dialog for the selection of the input file type and requirements.

1.1.4 Autocomplete feature

By using the key combination 'ctrl' plus 'space' the autocomplete menu will pop up. It will show available keywords for the current section as well as the names of defined variables.

Last update: 2025/07/09

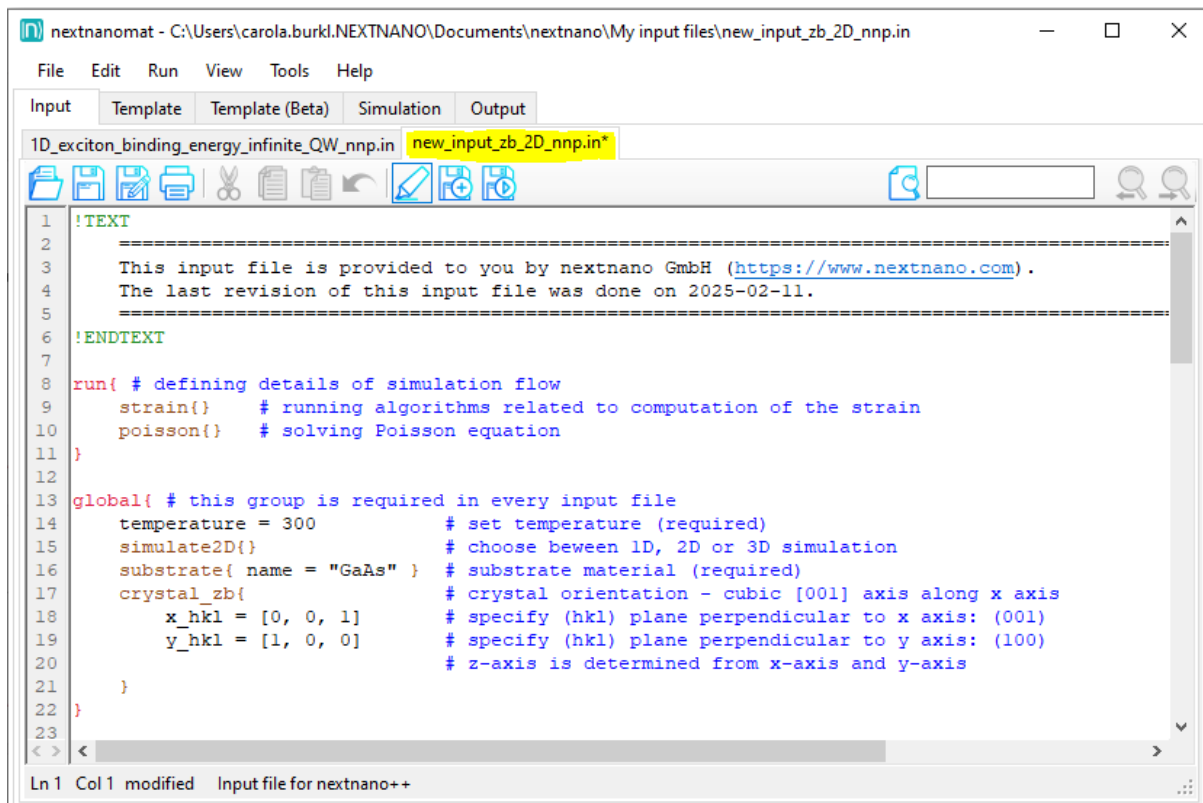


Figure 1.1.3.2: Newly created Input file with a predefined structure for the chosen input file type and requirements. In this example a nextnano++ input file for a 2D simulation of a zincblende material.

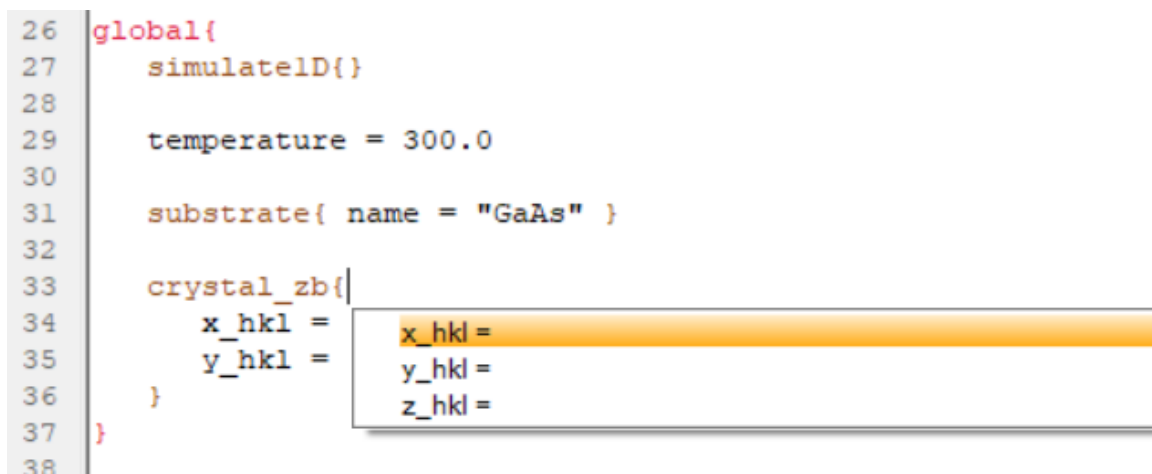


Figure 1.1.4.1: The *nextnanomat* autocomplete menu.

1.2 Template

This web page describes the template feature of *nextnanomat*.

In the Template tab you can **overview all variables** which are defined in the input file. You can **sweep** a chosen variable by a list of values or a range of values. Additionally, there is the option to compare the sweep results by the **post-processing feature**: It collects specified data from sweep files and stores them in a separate file, so that e.g. interband transition energy can be visualized as a function of well width.

1.2.1 Variable definition in input file (required)

```
<Variable> = <Value> <Comment>
```

with the following syntax definition:

Variable (required)

A <Variable> starts with a \$ or % sign (% should be preferred), can contain the characters A-Z, a-z, 0-9 and the underscore (_). It is case-sensitive. See documentation of Input Syntax in respective tools for further reference.

Value (required)

Any string after the = sign without # and ! characters is considered as <Value>. It is used as default value for the Template and *Template (Beta)* user interface.

Comment (optional)

Starts with a # or ! sign and is used as a description of the variable in the Template user interface. The comment may also include the following keywords (including brackets):

- # (DoNotShowInUserInterface)
This variable is not shown to the user in the Template user interface.
- # (DisplayUnit:<Unit>)
A string to show the unit for the variable (optional)
- # (ListOfValues:<default>)
Default values for List of values input control
- # (RangeOfValues:From=1,To=10,Step=1)
Default values for Range of values input control
- # (HighlightInUserInterface)
Highlights variable in Template user interface in yellow

Note

Every line that starts with a \$ (not nn3) or % (after stripping leading spaces) and includes a = is considered a variable definition.

Example

```
$well_width = 6 # Variable for quantum well width. (DisplayUnit:nm)↵
↪(RangeOfValues:From=6,To=18,Step=2) (HighlightInUserInterface)
```

1.2.2 Sweep over a variable + optional post-processing

If an input file includes at least one variable definition, it is possible to automatically sweep over it. The Template Tab of *nextnanomat* can be seen in Figure 1.2.2.1, already pre-filled for a variable sweep. To reproduce, please follow these steps:

1. The input file has to be chosen as the template file. Thus all correct variable definitions are loaded into the list.
2. Choose a sweeping variable and complete the required fields, e.g. List of values for QW_Separation.
3. The input files can be created by clicking the Create input files button.
4. They will be added to the batch list. To start the simulations you have to switch to the Simulation-tab, see Figure 1.2.2.2.

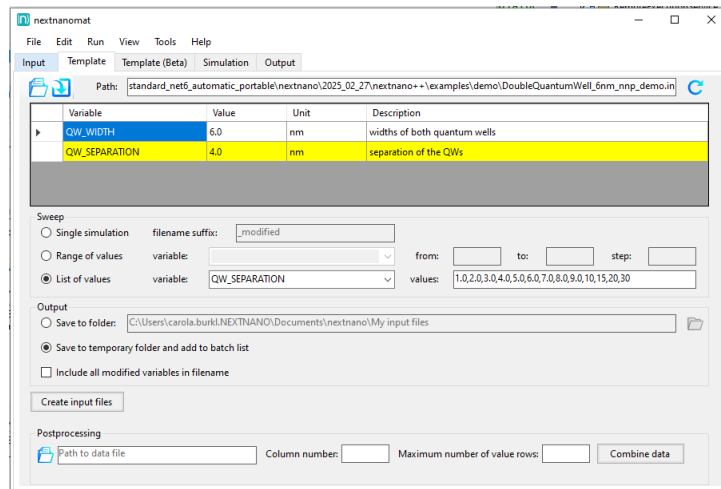


Figure 1.2.2.1: Template tab of *nextnanomat* with double quantum well input file loaded.

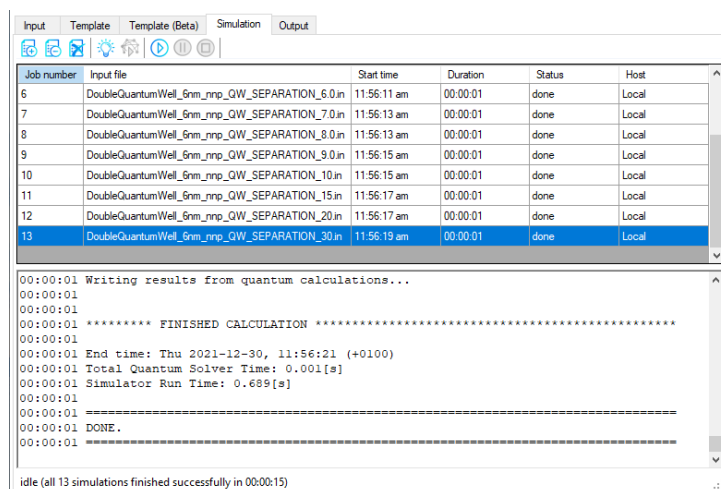


Figure 1.2.2.2: Batchlist of successfully simulated template sweep.

The sweep of a variable is now completed.

When the simulations are done, you have the possibility to use the post-processing feature. Its input fields are also located in the Template tab and can be seen in Figure 1.2.2.3. To use post-processing,

5. Choose the output file (*.dat) and the column number of the variable you want to compare, e.g. the file *energy_spectrum_quantum_region_Gamma.dat* displayed in Figure 1.2.2.4 and Figure 1.2.2.5
6. Additionally, state the maximum number of values (rows) which will be compared. (It is not allowed to exceed the existing number of rows.)
7. Now push the button Combine data and
8. Visualize your results in the Output-tab, see Figure 1.2.2.6.

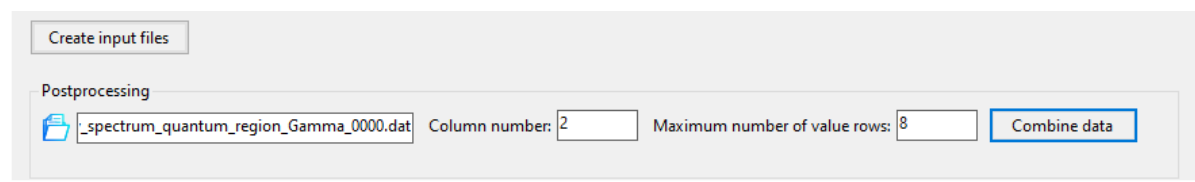


Figure 1.2.2.3: Additional fields to post-process a variable sweep.

For the steps 5 and 6 it is necessary to check the data structure of the simulation result you are interested in. Go to Output tab, select the data file and toggle **Text view**.

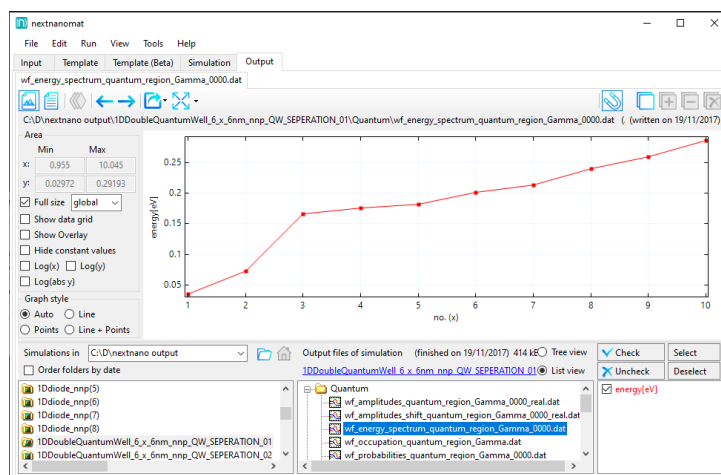


Figure 1.2.2.4: Energy eigenvalues of the first four electron states. Visualized as curves.

The result plots the chosen data in relation of the sweeping variable, see Figure 1.2.2.6.

1.2.3 Post-processing of existing sweep

If you have already done some sweeps in the past, you can also use the post-processing feature of *nextnanomat* retroactively. Note, additional to the post-processing steps described above, you need to load the correct template file and select and fill in the same variable values which have been used for the sweep.

1.3 Template (Beta)

1.3.1 Introduction

The Template (Beta) tab has a similar functionality to the Template tab. It allows convenient sweeps of an input file through the use of variables. In comparison to the standard Template feature, the Beta version allows nested sweeps of multiple variables. Its use is intended for experienced users who need to investigate their structure in complex dependency of more than one variable.

1.3.2 The input file

Any input file, which contains variables, can be used. At least two variables should be defined to use the full functionality of Template beta. Different options are provided to load an input file. It can be imported from the currently active input tab, searched within the file explorer or by inserting the path. Once opened, the defined variables should be shown on the right hand side. If changes to the file have been made by/with an external application, it can easily be updated through the reload button.

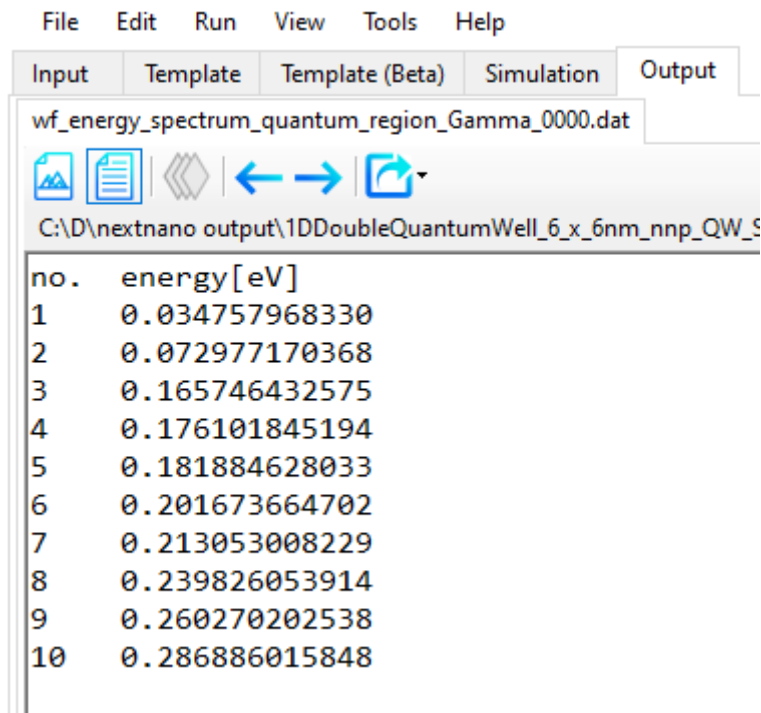


Figure 1.2.2.5: Text view of same output file.

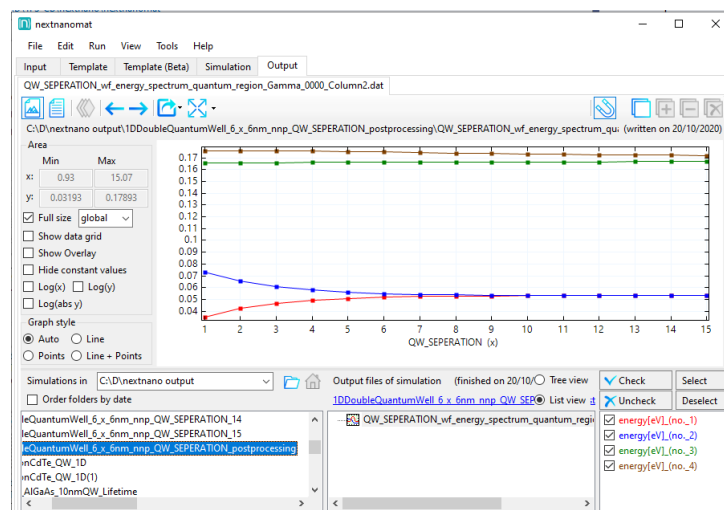


Figure 1.2.2.6: Result of post-processing. Electron energy eigenvalue in relation of quantum well separation.

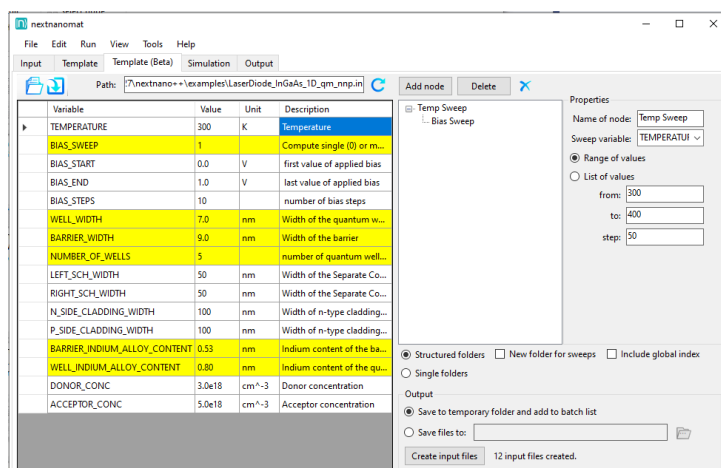


Figure 1.3.1.1: Full view of the Template (Beta) tab.

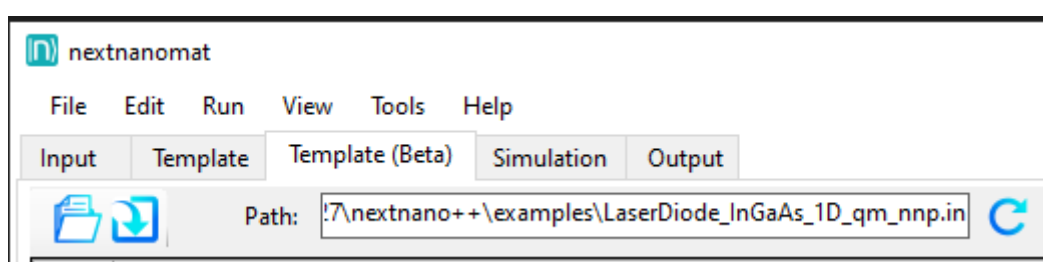


Figure 1.3.2.1: Menu to load and update the template file.

1.3.3 How to create a sweep

Follow the next steps:

1. (If the white panel is empty, click on the **Add node** button.)
2. Select the node **New Sweep**.
3. In the *property panel* you can define the parameters for this sweep:
 1. Choose a variable (e.g. \$TEMPERATURE).
 2. Set a correlating name for the sweep (e.g. “TempSweep”, [Figure 1.3.3.1](#)).
 3. Select **List of values** or **Range of values** and define the values for the sweep.
4. To add a second, third etc. sweep, repeat the steps above.
 1. To create a nested sweep, select a parent node before clicking on the **Add node** button. The new node will be nested below the selected one. (e.g. “BiasSweep”, [Figure 1.3.3.2](#))
 2. If no node is selected, the new sweep will be on the same level and they will not be correlated to each other.

1.3.4 Save input files and run a sweep

In this example we use three different temperature values and a nested bias sweep with four different bias values for each temperature (see [Figure 1.3.3.1](#) and [Figure 1.3.3.2](#)). Thus $3 \times 4 = 12$ different input files are to be created. Each input file receives the name and the values of both sweep parameters for identification.

Before running the simulation, there are different options for the simulation output:

Either the option **Single folders** or **Structured folders** can be selected. The first option generates one unique simulation output folder for each input file (ergo, for this example separate 12 folders, see [Figure 1.3.4.2](#)), whereas the **structured** option generates nested output folders matching the sweep structure. For the structured option a

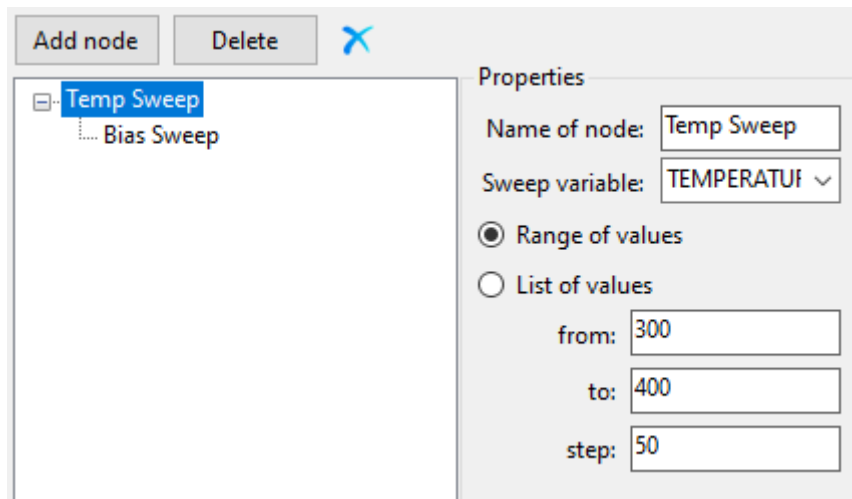


Figure 1.3.3.1: Parent sweep

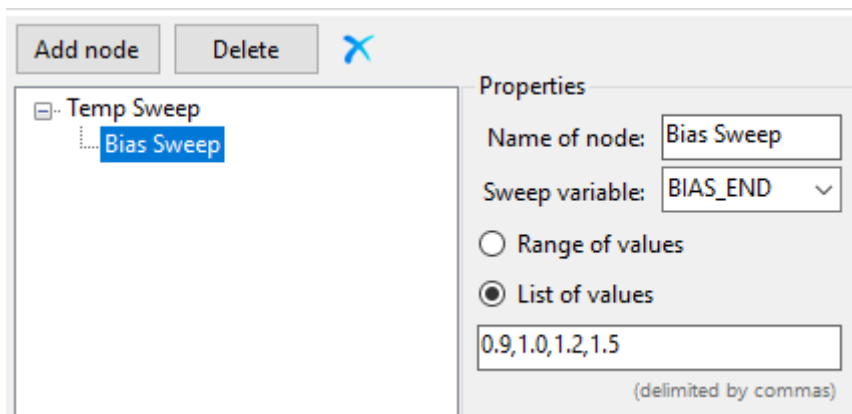


Figure 1.3.3.2: Nested sweep

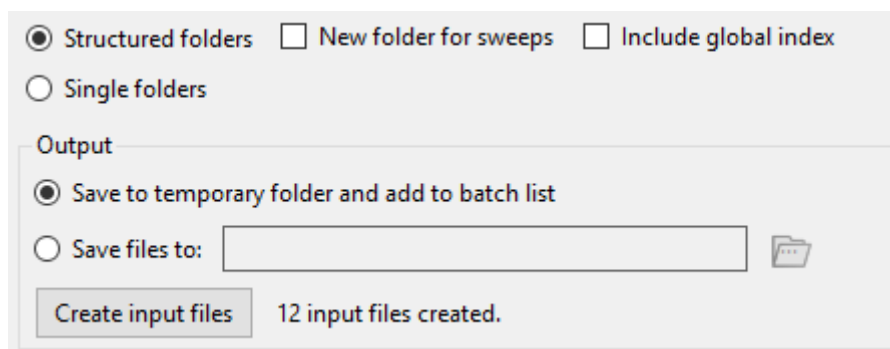


Figure 1.3.4.1: Output options

global index can be included which enumerates each sweep (Figure 1.3.4.3). And/Or a parent folder - for each sweep - can be added (Figure 1.3.4.4).

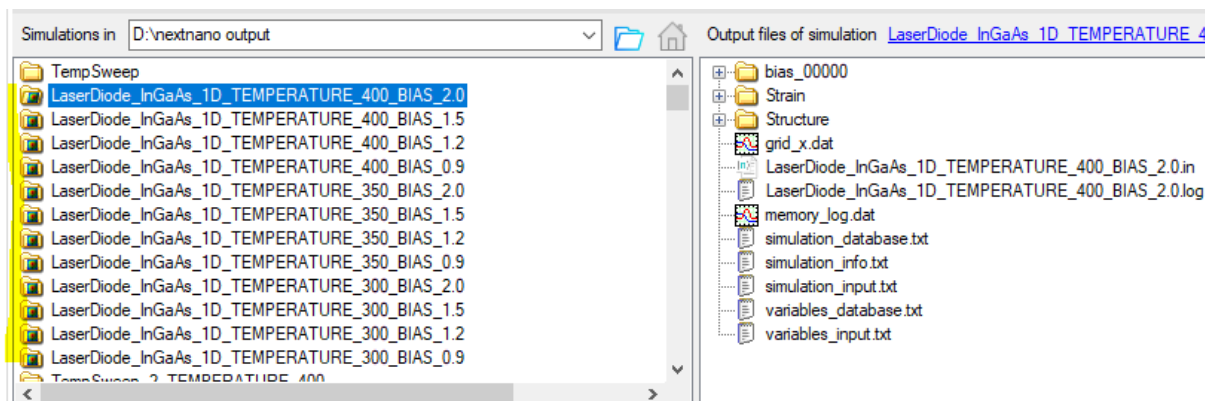


Figure 1.3.4.2: Output option a - Single folders.

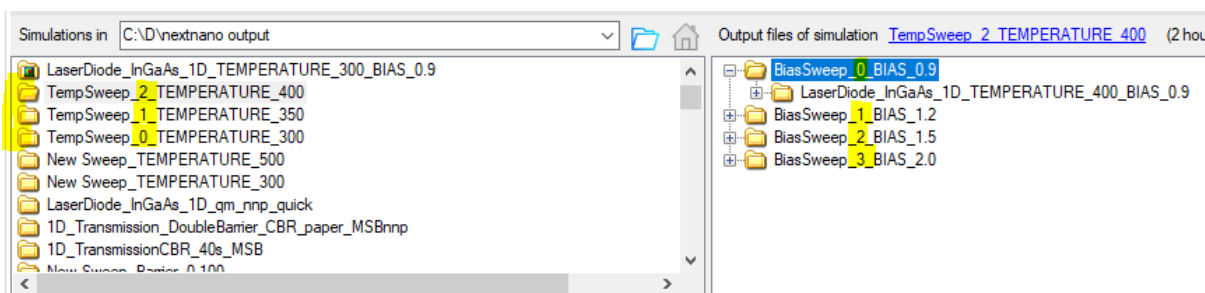


Figure 1.3.4.3: Output option b - Structured folders plus global index.

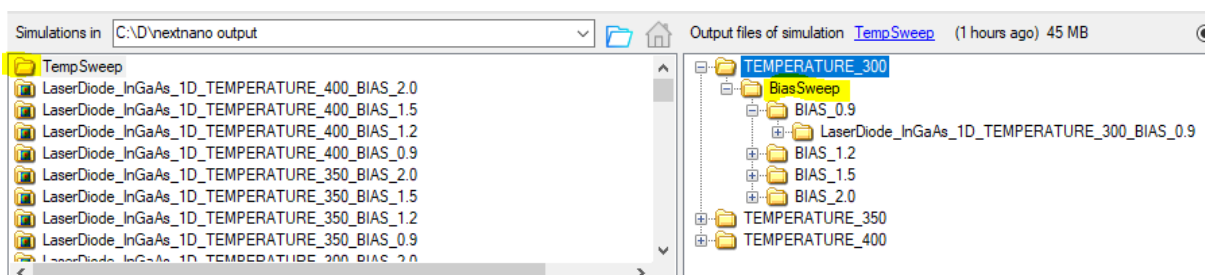


Figure 1.3.4.4: Output option c - Structured folders plus parent folder.

Lastly you can choose whether you want to create the input files only temporary to directly execute them or if you want to store them on the disk for later use.

After setting all your output preferences, click on the **Create Input files** button. You should see a confirmation

Create input files 12 input files created.

after your input files are created successfully:

Next, go to the **Simulation** tab to find the created input files and execute them.

1.4 Simulation

The processing of simulations is done here:

- The simulations can be started, paused and stopped.
- There is a **batch list**, where input files can be added or removed. It is also possible to add complete directories plus sub-directories by the option **Add Directory Tree to Batch List** of the **Simulation**- menu tab.

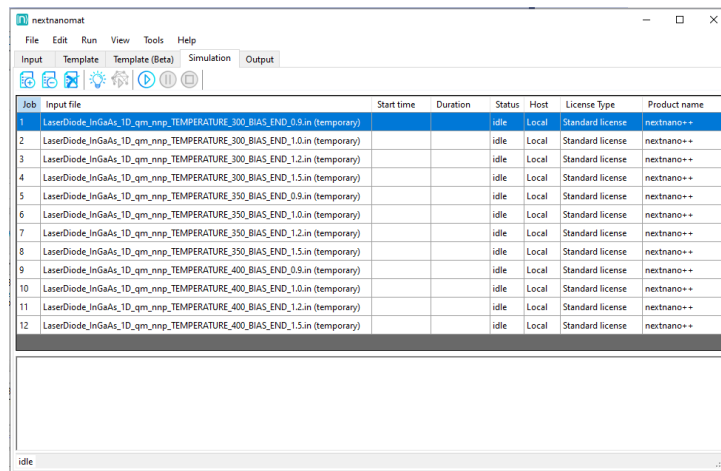


Figure 1.3.4.5: Created input files in simulation tab.

- Information of the processed simulation is provided. Such as Start time, Duration and Status.
- The **.log-file** of the currently running simulation is written below. This file, with information on the calculation, is also stored in the output folder.

1.5 Output

- *Navigation*
 - *Menu*
 - *Panels for 1D, 2D and 3D*
- *Visualization*
 - *Supported Output Formats*
 - *Text view*
 - *1D View*
 - *2D View*
 - *3D View*
- *Features*
 - *Overlay*
 - *Hide constant values*
 - *Show differences*
 - *Snap to gridpoints*
 - *Special fullsize*
 - *Auto select color map*
 - *Fix middle color to specific value*
- *Export functionality*
- *Further information*
 - *View settings*

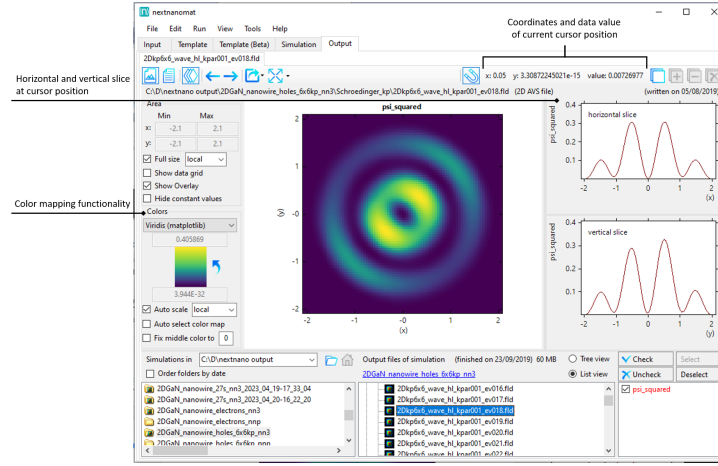


Figure 1.5.1.3: Panels for a 2D data file

1.5.2 Visualization

Supported Output Formats

The workflow manager *nextnanomat* can display the following output formats:

.txt

The file is displayed as a **text**.

.dat

The file is displayed as a **graph**, e.g. scalar field $f(x)$ or vector field $\mathbf{F}(x)$

x_1	$f(x_1)$
x_2	$f(x_2)$
...	
x_n	$f(x_n)$

or

x_1	$f_1(x_1)$	$f_2(x_1)$	...	$f_m(x_1)$
x_2	$f_1(x_2)$	$f_2(x_2)$	...	$f_m(x_2)$
...				
x_n	$f_1(x_n)$	$f_2(x_n)$	...	$f_m(x_n)$

.mtx or .mat

Matrix format for e.g. a matrix, a table, a $f(x,y)$ graph. The x and y axes are labeled with integer numbers.

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \dots & & & & \\ A_{m1} & A_{m2} & A_{m3} & \dots & A_{mn} \end{bmatrix}$$

.vtr - 2D/3D

Scalar field $f(x,y,z)$ or vector field $\mathbf{F}(x,y,z)$. They can be viewed using the ParaView software which is a full 3D visualization software while *nextnanomat* only displays 2D slices of 3D data files.

.fld - 1D/2D/3D

Scalar field $f(x,y,z)$ or vector field $\mathbf{F}(x,y,z)$

If a file extension is unknown it is treated as if it were a .txt file.

Text view

If `text view` is toggled, the content of the selected file is displayed. This view is read-only. For editing, the file can be exported to a custom texteditor. (We do not recommend to edit data files!)

1D View

Within graph view, one-dimensional data is displayed as curves. Whether data values are represented as points or lines is up to the user. Antialiasing can be switched on or off via [View settings](#). Additionally, the line thickness, background color or the usage of a grid can be customized.

If a data file contains multiple columns, each column is represented by its own curve. The list of available columns is shown within the `column view` panel. Only checked curves will be displayed. Selected curves are highlighted to allow fast recognition of related data.

Helpful features for one-dimensional data visualization are:

- [Overlay](#)
- [Hide constant values](#)
- [Show differences](#)
- [Snap to gridpoints](#)
- [Special fullsize](#)
- [Gnuplot export](#)

2D View

Two-dimensional data is displayed as a heat map, also known as pseudo-coloring. Each value of the two-dimensional grid is mapped to a distinct color. The color difference perceived allows the visual interpretation of value differences. Therefore the change of the color gradient needs to be perceived linearly by the human eye. Some color maps, like the rainbow map, introduce artefacts and have misleading visual perception, which increases the risk for misinterpretation of scientific data. Find more information about [Color maps](#).

Helpful features for two-dimensional data visualization are:

- [Auto select color map](#)
- [Fix middle color to specific value](#)
- [Gnuplot export](#) of surface plots

3D View

Real three-dimensional plots are not supported by *nextnanomat*. Instead 2D planes of the 3D data are visualized through heat maps, same as for 2D data. An additional panel allows the selection of the displayed plane (xy, xz or yz) and the position of the slice. This panel can be seen in [Figure 1.5.3.10](#). If three-dimensional visualization is aimed for, we recommend exporting data to ParaView.

Helpful features for three-dimensional data visualization are:

- All of the 2D features
- [Exporting to ParaView](#)

1.5.3 Features

Overlay

For the best experience when visually analyzing the results of the simulation, it is sometimes necessary to look at different files at the same time. We call this the Overlay feature.

1. Select the plots you want to memorize in the `column view` panel.
2. Click the `Add to Overlay` button, see [Figure 1.5.3.1](#). Alternatively, use the keys `a` (add) or `+` (Numpad only).

3. Select another output file.
4. (optional) add plots of multiple files to the overlay list.

⇒ Your memorized plots will be displayed in gray on top of the currently selected file.

You can check and edit the content of your overlay in the `overlay list` panel, this panel can be shown or hidden by clicking the `Show list of overlays` button. Remove selected curves with the `Remove from Overlay` button, shown in [Figure 1.5.3.1](#), or use the `Del/Entf` or `d` (delete) key on your keyboard.

You can also export the overlay graphs to one combined image file, for further information refer to [Gnuplot export](#).

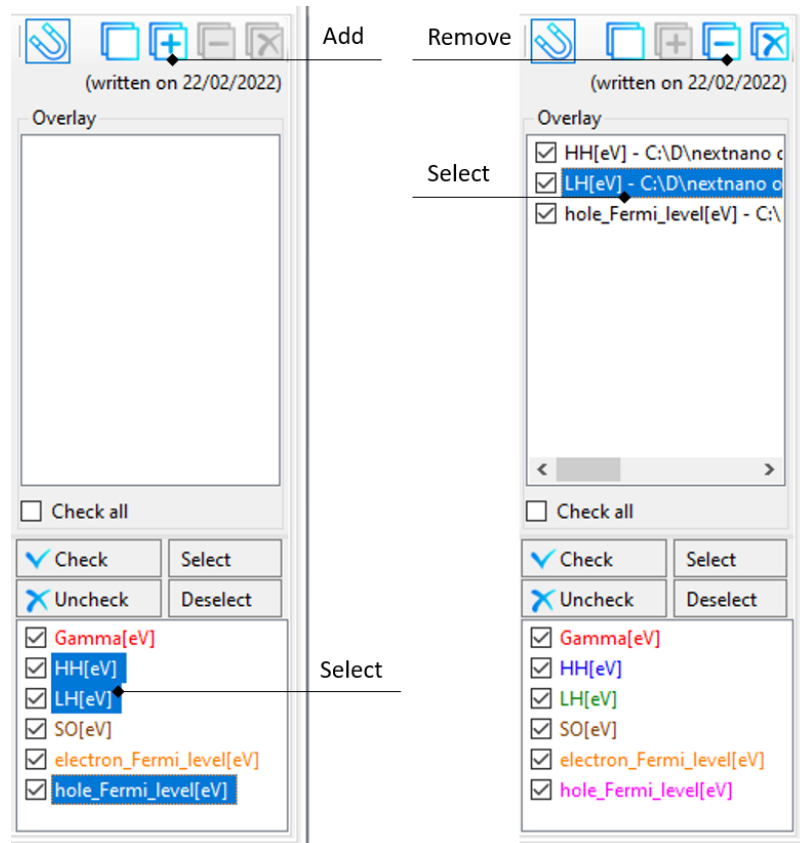


Figure 1.5.3.1: Workflow to add or remove files from Overlay.

Hide constant values

Note

Be careful when using this feature. We recommend to always disable it after usage.

This feature hides any parts of the curves where the value does not change within 5 grid points (two to the left and right). Its purpose is for visualizing wave functions and probability densities on top of band edges. The constant part which represents the energy level will be hidden. This allows to focus on the changing parts as well as for the underlying band edge to be seen. The influence of this feature can be seen in [Figure 1.5.3.2](#) and [Figure 1.5.3.3](#).

Show differences

Whenever exactly two columns are selected in the `column view` panel, the value difference between these two curves is shown while the mouse is hovered over the data, see [Figure 1.5.3.4](#).

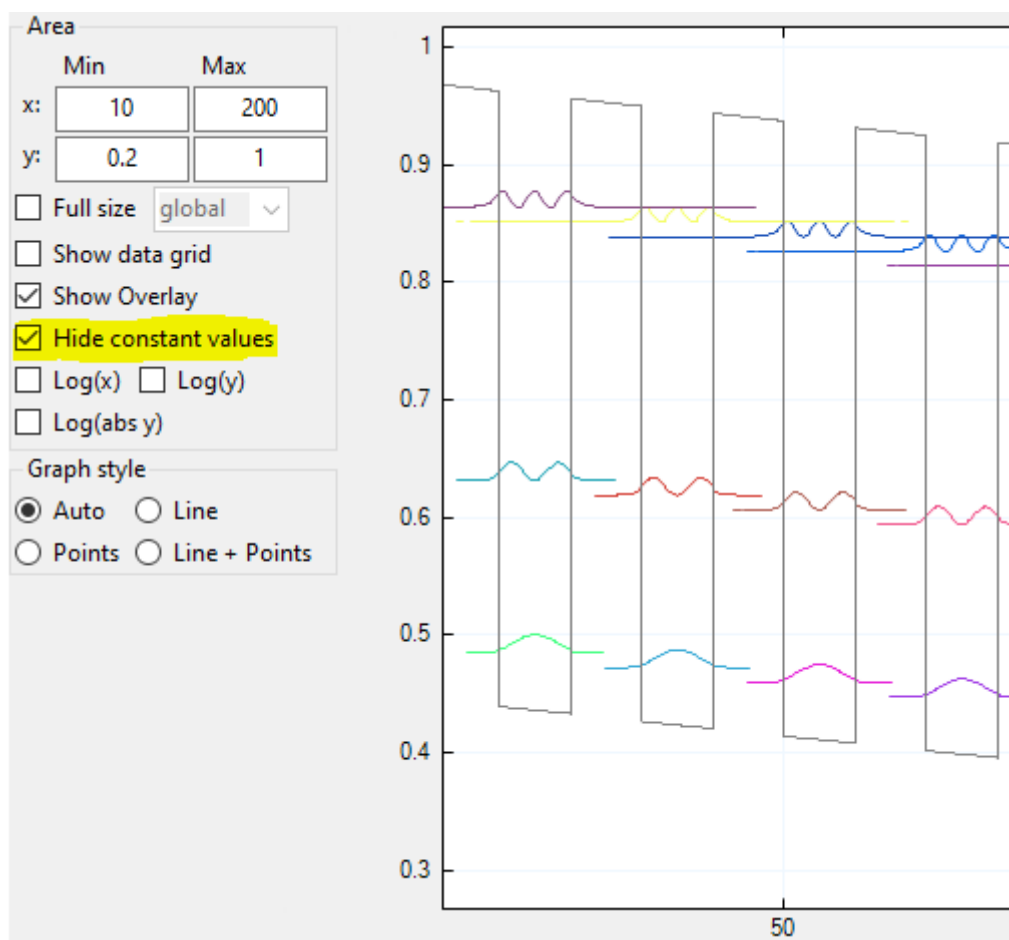


Figure 1.5.3.2: Hide constant values on.

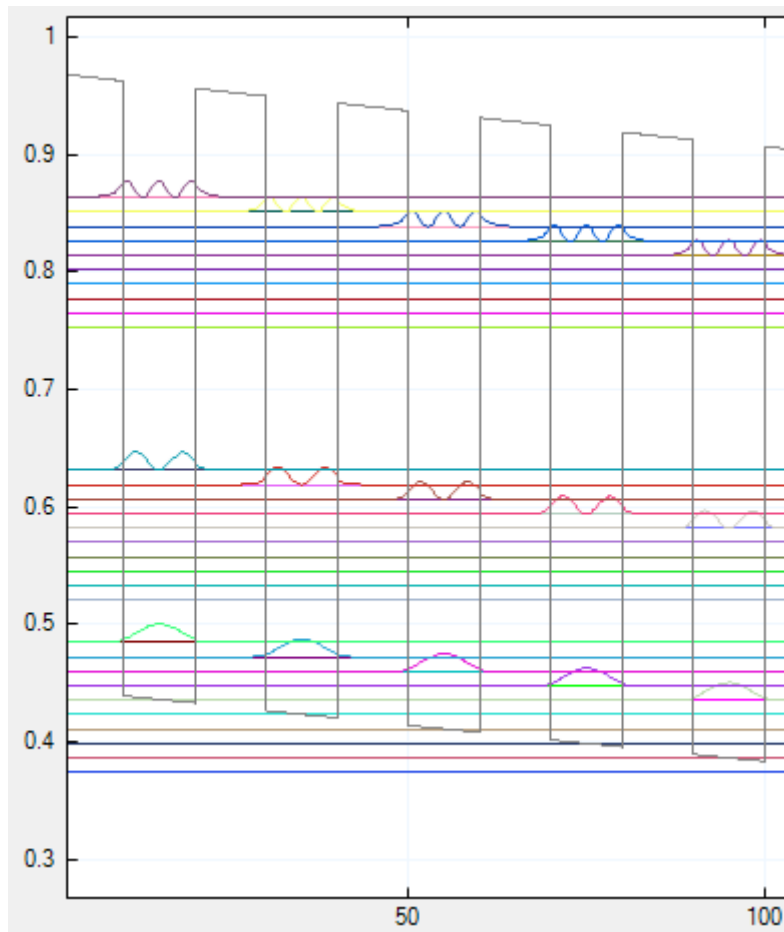


Figure 1.5.3.3: Hide constant values off.

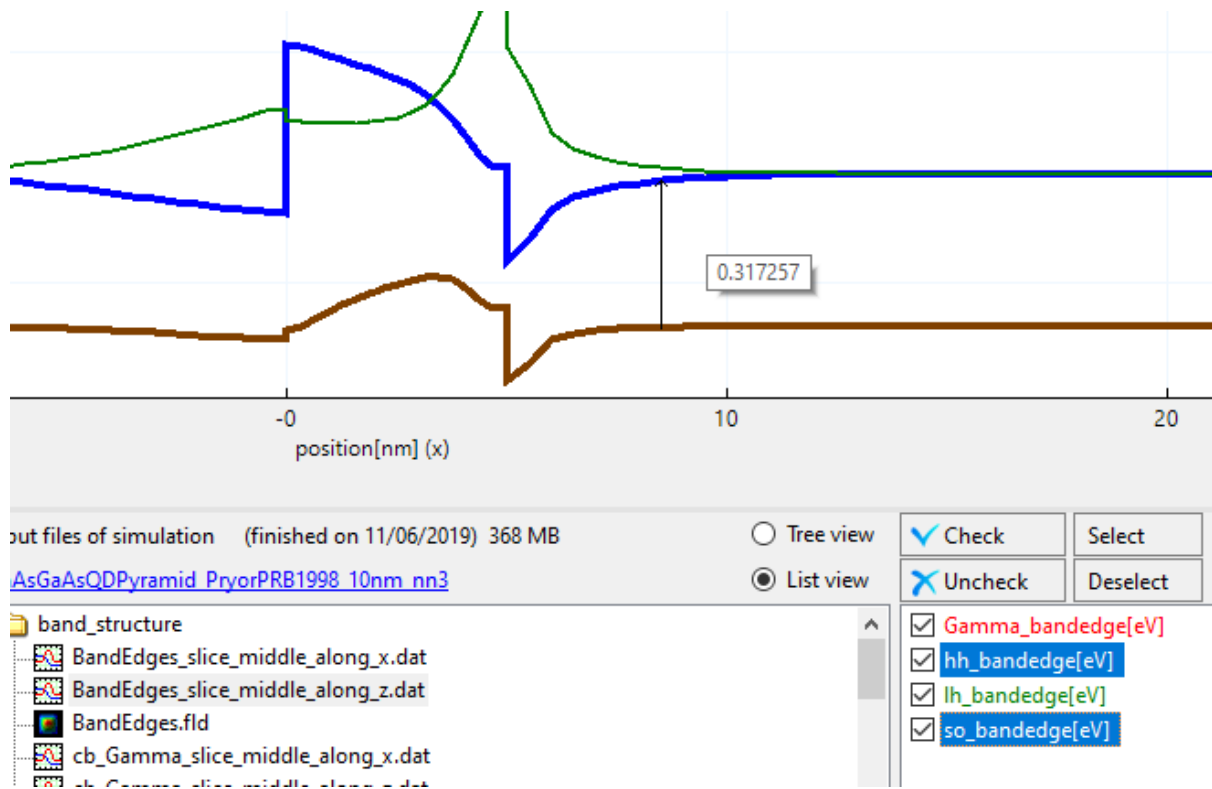


Figure 1.5.3.4: Arrow showing value difference of two data curves.

Snap to gridpoints

If snap to grid points button is toggled (default: on) a small cursor cross follows the nearest data curve while the mouse cursor is hovered over the data. The coordinates and value displayed in the upper right corner of *nextnanomat* match the data values of the curve, see Figure 1.5.3.5.

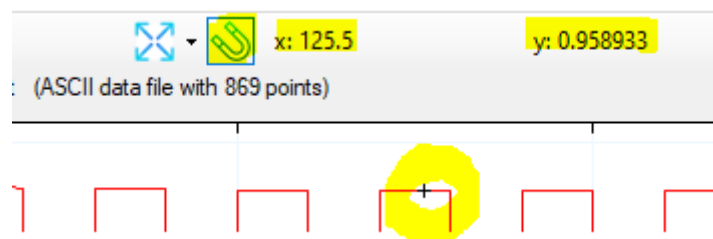


Figure 1.5.3.5: Snap to grid points feature enabled.

Special fullsize

By using this feature, the view of the output diagram will be matched perfectly to a single graph. Either the selected graph of the column view, or it is iterating through all available columns of the currently displayed file. This feature is useful for example if a file contains multiple columns with a highly different range of values. Displaying all graphs at the same time (normal fullsize mode) can lead to certain curves appearing to be zero. By selecting such curve and clicking the special fullsize button, the y-axis limits are set to its respective minimum and maximum and the curve is displayed correctly. An example for this use-case can be seen in Figure 1.5.3.6.

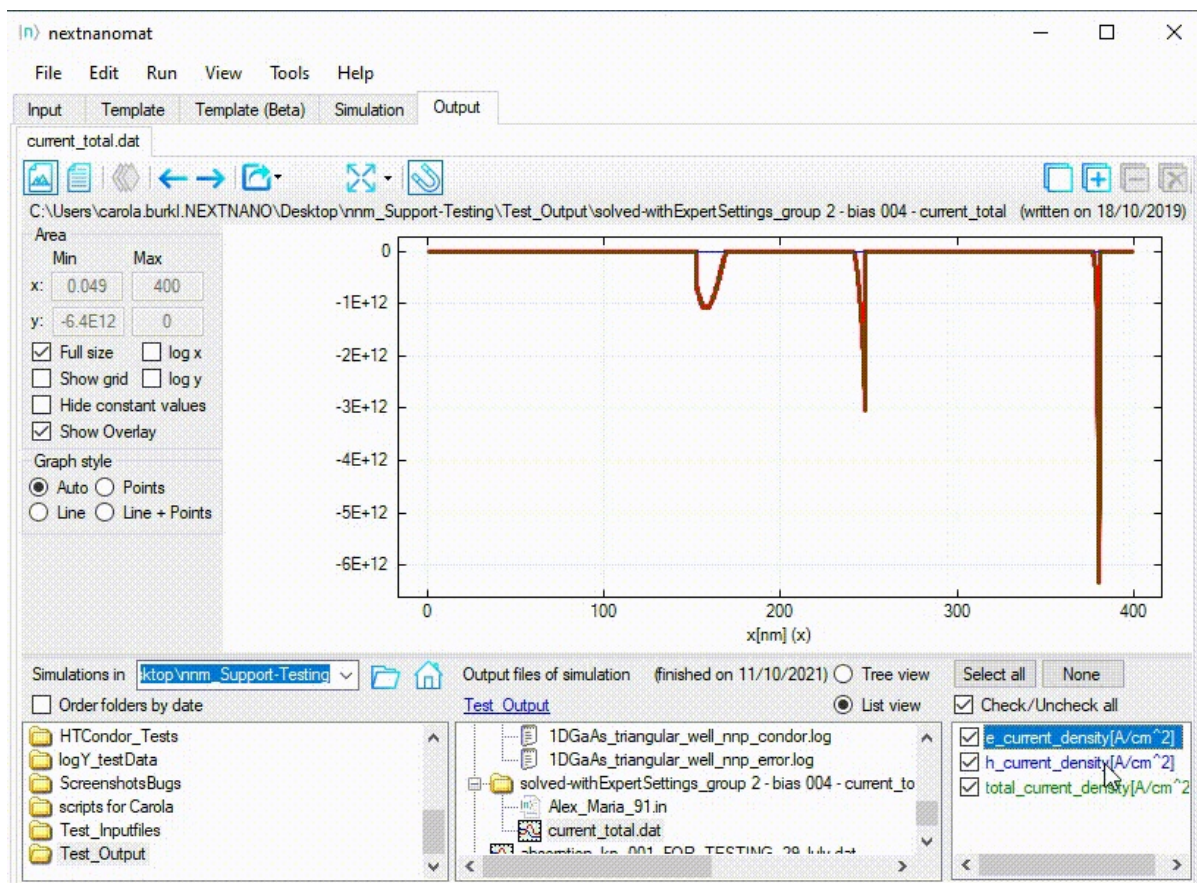


Figure 1.5.3.6: Special fullsize for a diverging range of values.

Furthermore this feature is also useful for any other file containing more than one column. Iterating through the columns and displaying each curve in its optimal frame, allows for fast and convenient data evaluation, see Figure 1.5.3.7. .. example one high number of curves

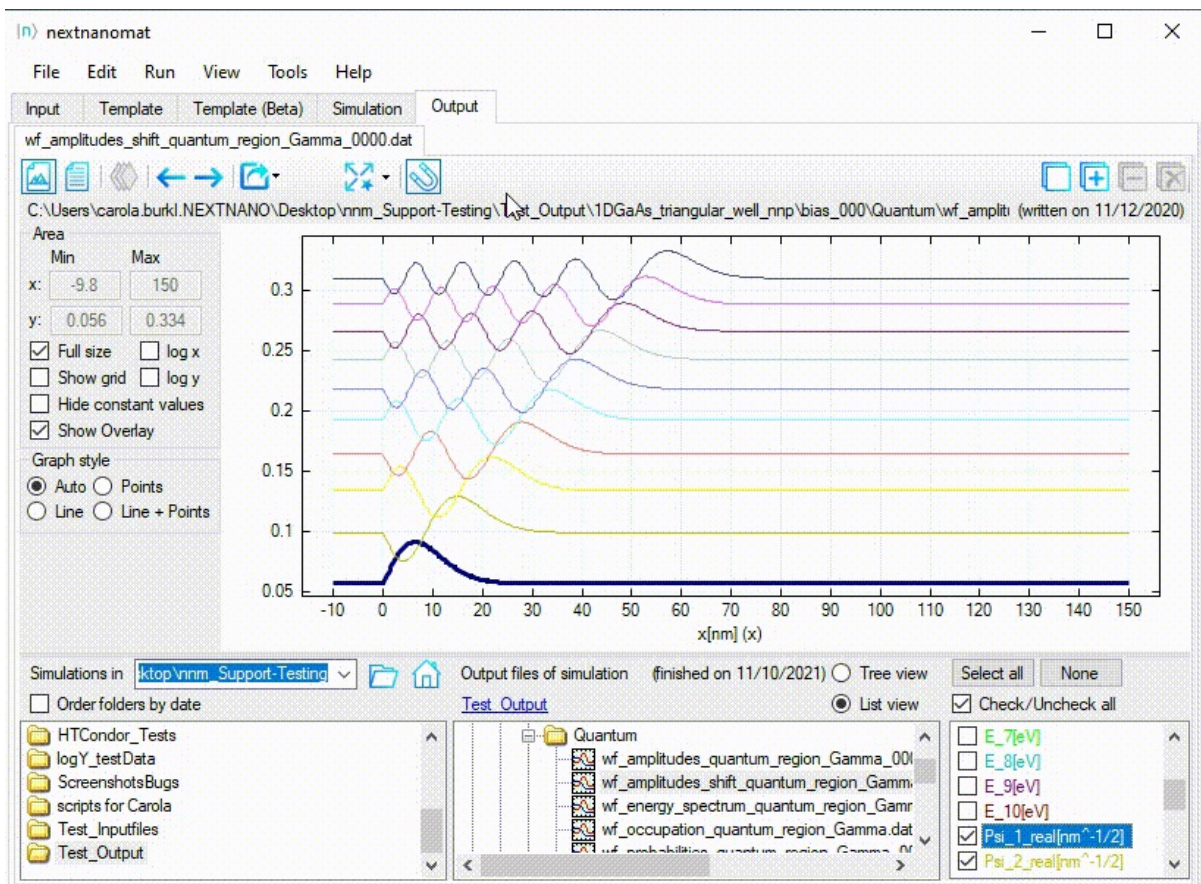


Figure 1.5.3.7: Special fullsize for a high number of curves.

Auto select color map

If auto selection of a color map is active, a diverging color map is selected for data files spanning from negative to positive values. Else a linear color map is chosen.

Fix middle color to specific value

This feature is aimed for the visualization of diverging data in combination with a diverging color map. It fixes the mapping of the neutral middle color to a specific value and thus ensures a symmetric color representation of diverging data.

Note

To be able to support custom color maps, this feature is not limited to predefined diverging color maps. To avoid unintentional usage we recommend to enable this feature on demand only.

The feature is enabled by checking the corresponding check box. If enabled, it is active whenever a diverging color map is assumed (error prone to enable flexibility) **and** the data file contains the specified value. Else it is inactive and grayed out. The visual feedback on the activity of the feature can be seen in [Figure 1.5.3.8](#) and [Figure 1.5.3.9](#)

Example usage of the feature can be seen in [Figure 1.5.3.10](#).

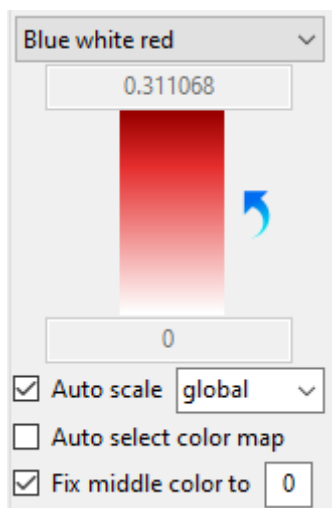


Figure 1.5.3.8: Enabled and active

1.5.4 Export functionality

Most of the export options are available from the context menu (right-click on the visualization). Some specific options can be accessed from the output menu button **Export** and **Open in specific Format**. For the latter option the custom defined paths to installed applications are used (with exception of the Gnuplot application). These paths need to be defined in the [Data export settings](#).

You can find a detailed description of the various export mechanism in the section [Export Functionalities](#)

1.5.5 Further information

References, you might find helpful...

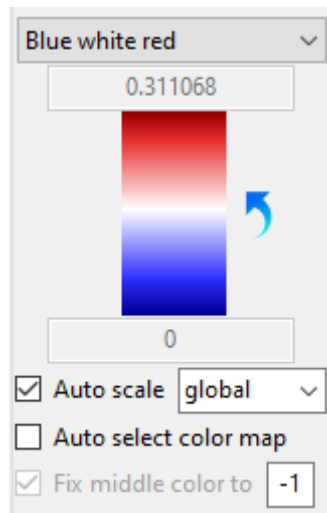


Figure 1.5.3.9: Enabled but inactive

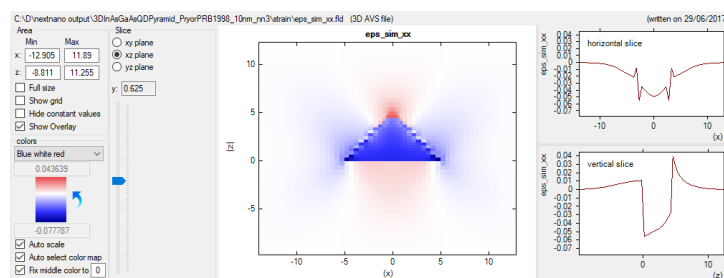


Figure 1.5.3.10: Visualization of diverging data with fix middle color feature enabled and active.

View settings

View

Data export settings

Data export

Clean up output folder

Clean Up Simulation Output Folder

Color maps

Color maps

MAIN MENU FUNCTIONS

Drop down functions of main menu. Most functions are identical to the menu buttons of each tab page. Those which do not clone a button functionality will be described here.

- *File*
- *Edit*
- *Run*
- *View*
- *Tools*
- *Help*

2.1 File

2.1.1 Create New Input File

This feature is part of the *Input tab* functionality and described there, see *Create new Input File with predefined Structure*.

2.2 Edit

2.2.1 Replace existing Tabs with Spaces

This function will replace all Tabs with Spaces in the currently active input file. The number of spaces is defined in Editor Options, *Editor/Input*.

2.2.2 Replace IF with WHEN

In newer versions of the *nextnano++* tool, the conditional specifier *if* is replaced by *when*. This function facilitates the corresponding syntax update of existing and custom input files, by replacing all instances of the outdated syntax with the newer syntax.

2.2.3 Open Keyword Tree, Syntax Help

By clicking this menu function a .html keyword tree will be displayed using your default browser. The keyword tree interactively visualizes the syntax hierarchy and keywords of the current input file type. An example for the *nextnano++* tool can be seen in :numref::fig-nnmt_input_keytree

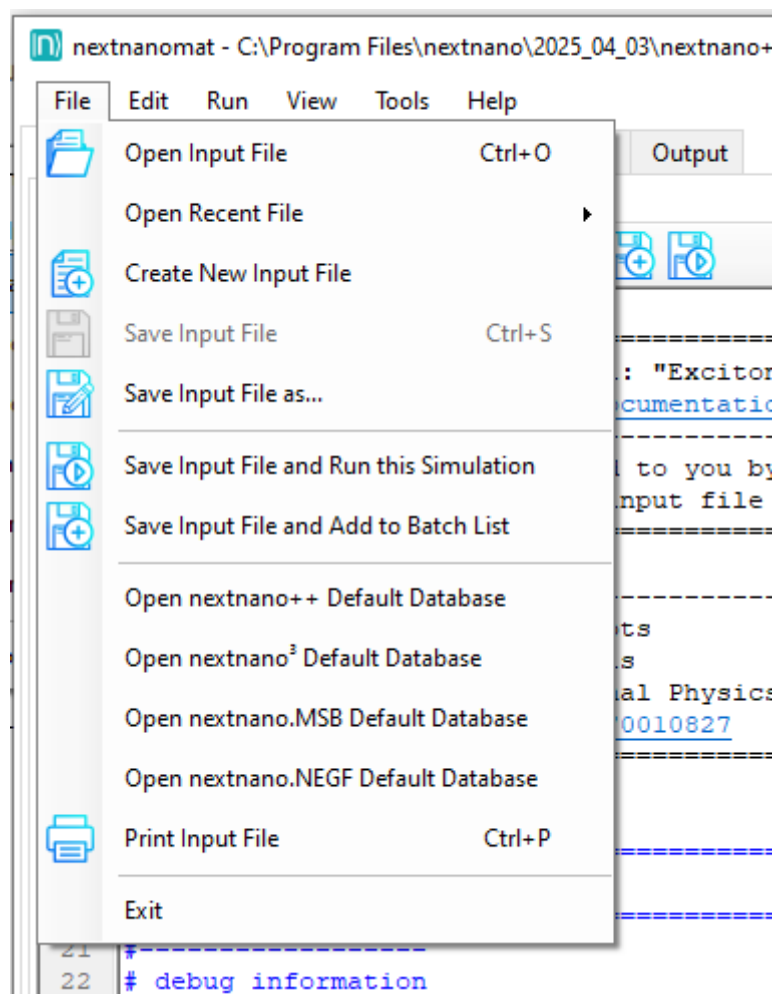


Figure 2.1.1: File main menu functions

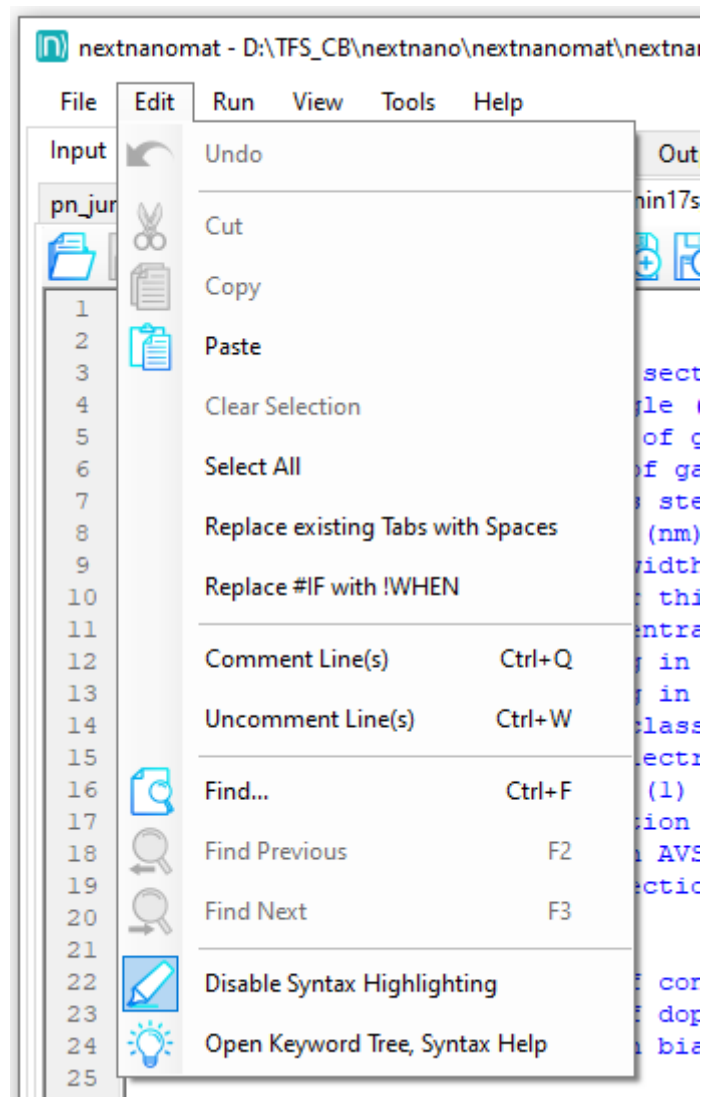


Figure 2.2.1: Edit main menu functions

- ☒ nextnanoplus
 - ☒ global
 - ☐ simulate1D
 - ☐ simulate2D
 - ☐ simulate3D
 - ☐ crystal_zb
 - ☐ crystal_wz
 - ☐ substrate
 - ☐ temperature
 - ☐ temperature_dependent_bandgap
 - ☐ temperature_dependent_lattice
 - ☐ magnetic_field
 - ☐ periodic
 - ☐ grid
 - ☒ impurities
 - ☐ donor
 - ☐ acceptor
 - ☐ charge
 - ☐ contacts
 - ☐ structure
 - ☐ strain
 - ☐ classical
 - ☐ poisson
 - ☐ currents
 - ☐ quantum
 - ☐ optics
 - ☐ import
 - ☐ output
 - ☐ run
 - ☐ postprocessor
 - ☐ database

Figure 2.2.3.1: Example keyword tree for a version of the *nextnano++* tool.

2.3 Run

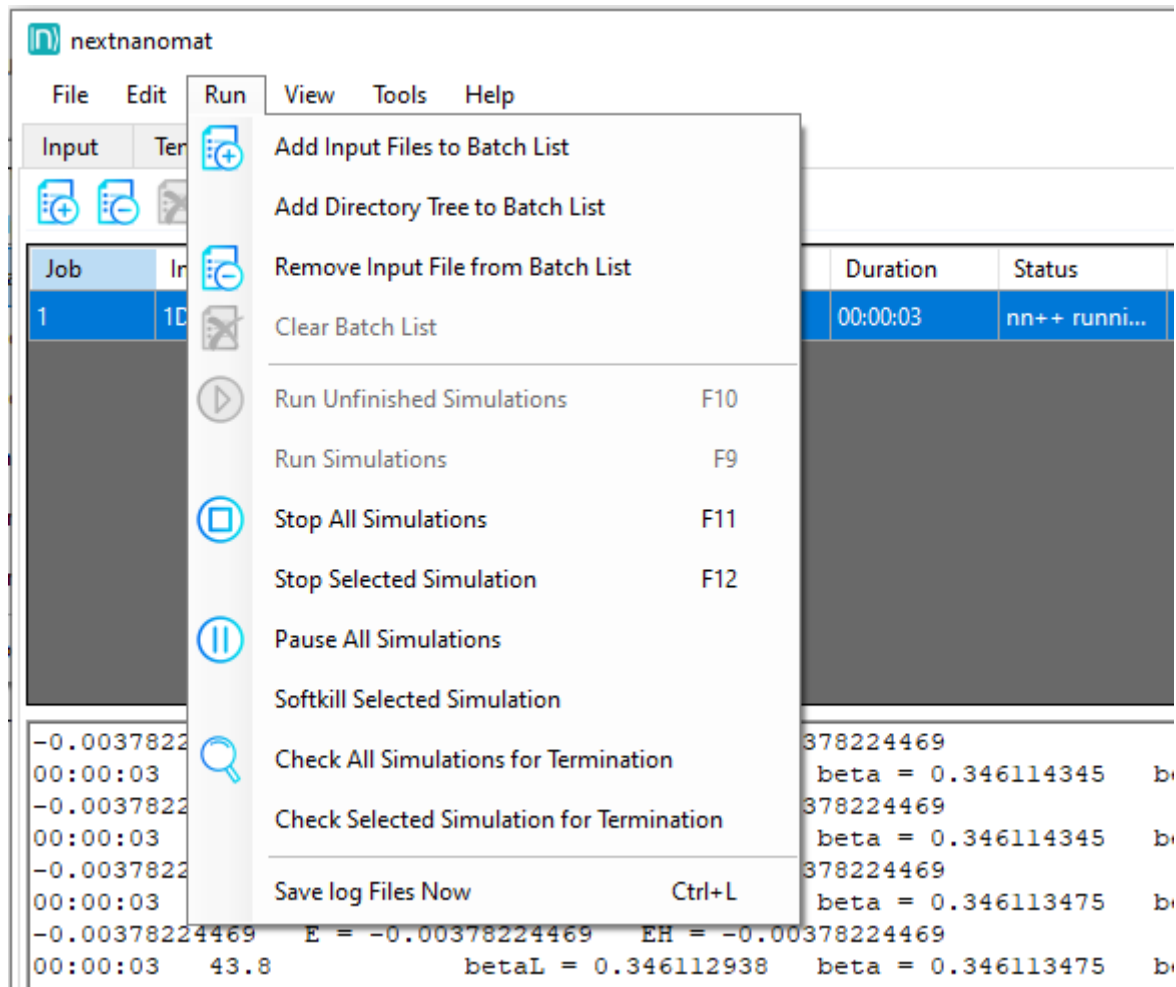


Figure 2.3.1: Run main menu functions

2.3.1 Add Directory Tree to Batch List

By clicking this main menu item a folder selection dialog will pop up. All input files of the selected directory and its sub-directories will be added to the batch list. No need to manually select each file.

2.3.2 Softkill Selected Simulation

Instead of killing the process directly, this softkill function tells the running nextnano tool to terminate after the current iteration loop has finished. This option is recommended, if you want to evaluate the simulation results of that simulation. It is supported by the *nextnano++* and *nextnano³* tool.

2.3.3 Check Simulations for Termination (beta)

It rarely occurs that *nextnanomat* is unable to detect the completion of the simulation job. While you can verify in the Task Manager that the nextnano tool has already finished its calculation, the job is still displayed as running within the workflow manager *nextnanomat*. This main menu function call, is for that use-case only. If the option is clicked, a query to the simulation process is sent. In case the process has exited, the simulation will be finished and its output processed.

As this is still a beta feature, feedback on that functionality is appreciated. Contact our support team at [nextnano Help Center](#).

2.4 View

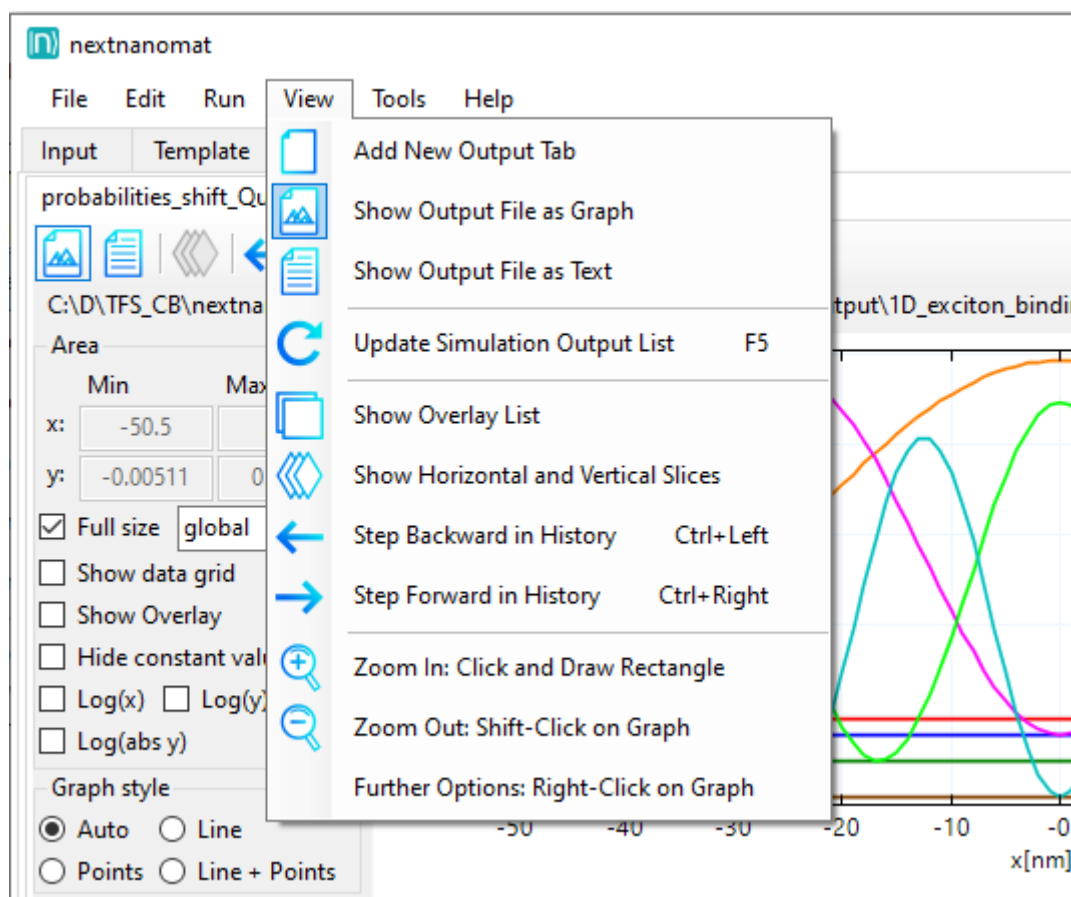


Figure 2.4.1: View main menu functions

All *View* main menu functions are either self-explanatory or duplicate a button's function of the *Output* tab. Please refer to the corresponding documentation.

2.5 Tools

2.5.1 Activate license

By clicking *Activate License* in the *Tools* menu, a license activation dialog will appear. If you have troubles finding your previously downloaded license files on your PC, you can just re-activate them. No additional licenses will be generated, you will just receive another copy of your previously generated files. The license activation procedure is documented here: [nextnano software](#)

2.5.2 Convert nextnano³ Input File to nextnano++

How to use this automatic conversion:

In the menu select *Tools* ==> *Convert nextnano³ input file to nextnano++*

When you use the automatic conversion of *nextnano³* input file into *nextnano++*, you will find that it probably does not work completely.

If you save and run the *nextnano++* input file that has been converted and that has the suffix `_nnp.in`, very likely some errors appear indicating which line(s) to change. Then some manual adjustments are needed, but the rough structure should help a lot for the conversion.

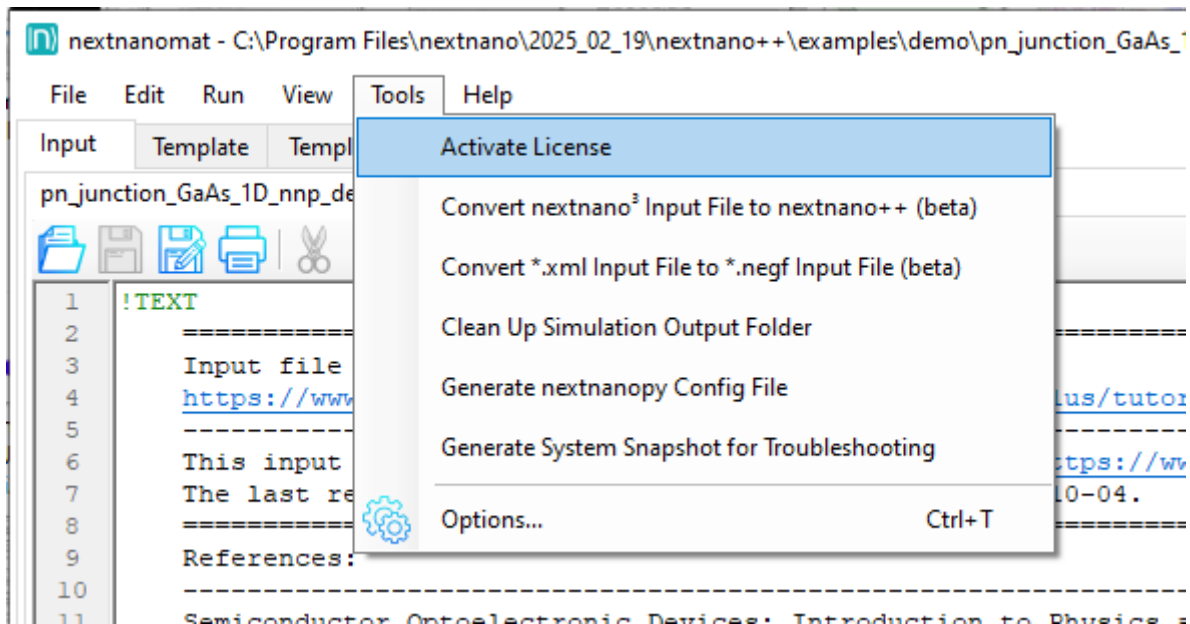


Figure 2.5.1: Tools main menu functions

2.5.3 Convert xml Input File to negf Input File

How to use this automatic conversion:

In the menu select Tools ==> Convert *.xml Input File to .negf Input File*

This conversion allows easy adjustment of the older input file structure to the current format. Some manual adjustments may be needed, but the rough structure should help a lot for the conversion.

2.5.4 Clean Up Simulation Output Folder

After some months of simulating, the output folder can get quite huge. There is a feature to conveniently clean up the output folder with additional options, e.g. creating a backup of the input files. In the main menu, click on Tools and select Clean Up Simulation Output Folder. The dialog can be seen in [Figure 2.5.4.1](#). To check multiple files at the same time, you need to select multiple files first and click on one checkbox afterwards.

2.5.5 Generate nextnanopy config file

If you want to execute a nextnano product using *nextnanopy*, a config file is needed. This file contains all relevant information and paths to the licenses, nextnano executables and material database.

To make it more comfortable for you to use both *nextnanomat* and *nextnanopy* or switch to *nextnanopy*, you can export the settings already stored within *nextnanomat* into the format needed by *nextnanopy*.

Taken from the *nextnanopy* documentation on https://github.com/nextnanopy/nextnanopy/blob/master/docs/examples/Example0_Set_up_the_configuration.rst:

Per default the config file needs to be located at your home directory (usually C:\Users\Your_User\.nextnanopy-config). When you import *nextnanopy* for the first time, it will automatically generate the configuration file with few default parameters. If this file already exists, it will not modify it. You can set up this configuration file only once and you do not need to worry about it anymore, except when you renew your license or update the nextnano version.

Thus, if you want to have your custom paths as specified in *nextnanomat*, just export a config file using this function and save it into your home directory. Then *nextnanopy* will not overwrite that file. After a new installation of a nextnano update, you can easily update the paths for *nextnanopy* as well, by using this function again.

Example config file:

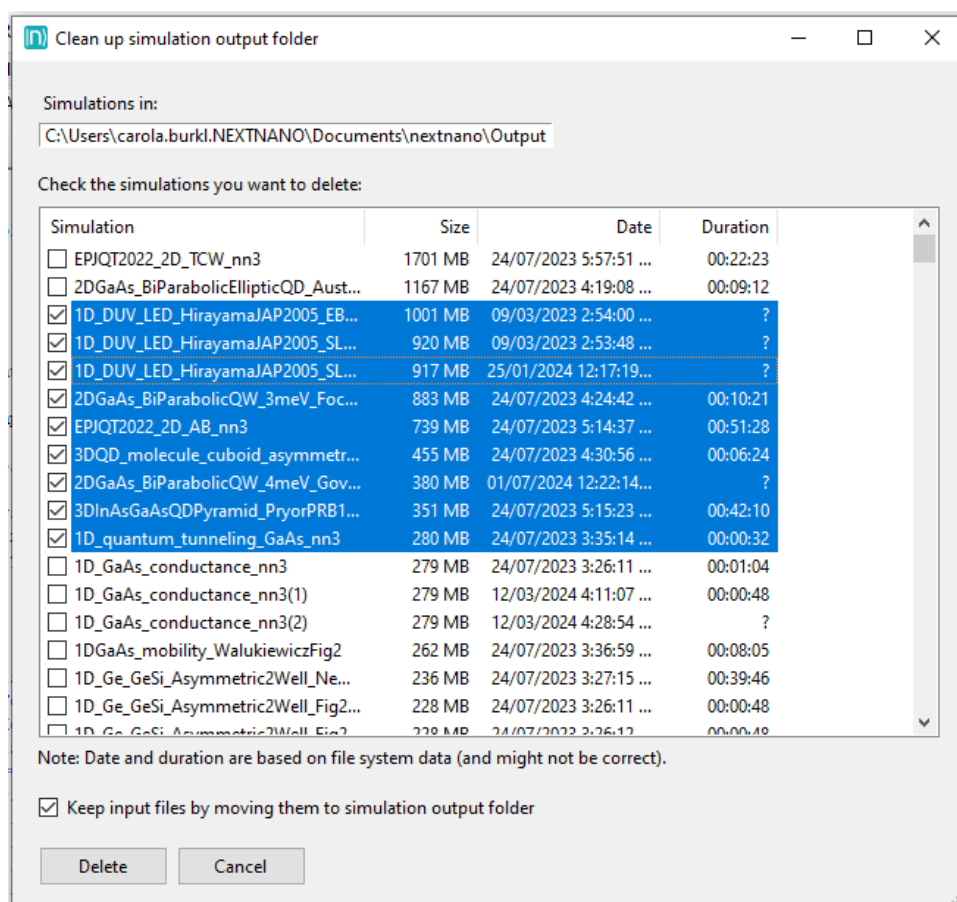


Figure 2.5.4.1: Dialog to help clean up the simulation output folder.

```
[nextnano++]
exe = C:\Program Files\nextnano\2021_12_24\nextnano++\bin 64bit\nextnano++_Intel_
↪64bit.exe
license = C:\Users\homer.simpson\Documents\nextnano\License\License_nnp.lic
database = C:\Program Files\nextnano\2021_12_24\nextnano++\Syntax\database_nnp.in
outputdirectory = D:\nextnano output
threads = 3

[nextnano3]
exe = C:\Program Files\nextnano\2021_12_24\nextnano3\Intel 64bit\nextnano3_Intel_
↪64bit.exe
license = C:\Users\homer.simpson\Documents\nextnano\License\License_nnp.lic
database = C:\Program Files\nextnano\2021_12_24\nextnano3\Syntax\database_nn3.in
outputdirectory = D:\nextnano output
threads = 0
debuglevel = -1
cancel = -1
softkill = -1

[nextnano.NEGF]
exe = C:\Program Files\nextnano\2021_12_24\nextnanoNEGF_2020_06_22\nextnano.NEGF\
↪nextnano.NEGF.exe
license = C:\Users\homer.simpson\Documents\nextnano\License\License_nnQCL.lic
database = C:\Program Files\nextnano\2021_12_24\nextnanoNEGF_2020_06_22\nextnano.NEGF\
↪Material_Database.xml
outputdirectory = D:\nextnano output
threads = 3

[nextnano.MSB]
exe = C:\Program Files\nextnano\2021_12_24\nextnano.MSB\nextnano.MSB\x64\nextnano.MSB.
↪exe
license = C:\Users\homer.simpson\Documents\nextnano\License\License_nnMSB.lic
database = C:\Program Files\nextnano\2021_12_24\nextnano.MSB\nextnano.MSB\Materials.
↪xml
outputdirectory = D:\nextnano output
debug = 0
```

2.5.6 Generate System Snapshot for Troubleshooting

In the menu, you can either find it under ‘Tools’ -> ‘Generate System Snapshot for Troubleshooting’ or under ‘Help’ -> ‘Generate Troubleshooting Snapshot’.

By clicking this function, a text file - containing most relevant information for debugging - will be created. Save this file and send it to the nextnano Support Team to assist you.

2.6 Help

In the *Help* menu, information **about** the software and direct links to the nextnano online documentation of each simulation tool can be found. Furthermore, there are a couple of contact possibilities listed, from our customer support portal *nextnano Help Center* to our social media accounts. If you plan to contact support, the menu function *Generate System Snapshot for Troubleshooting* generates a text file which you can attach to your request.

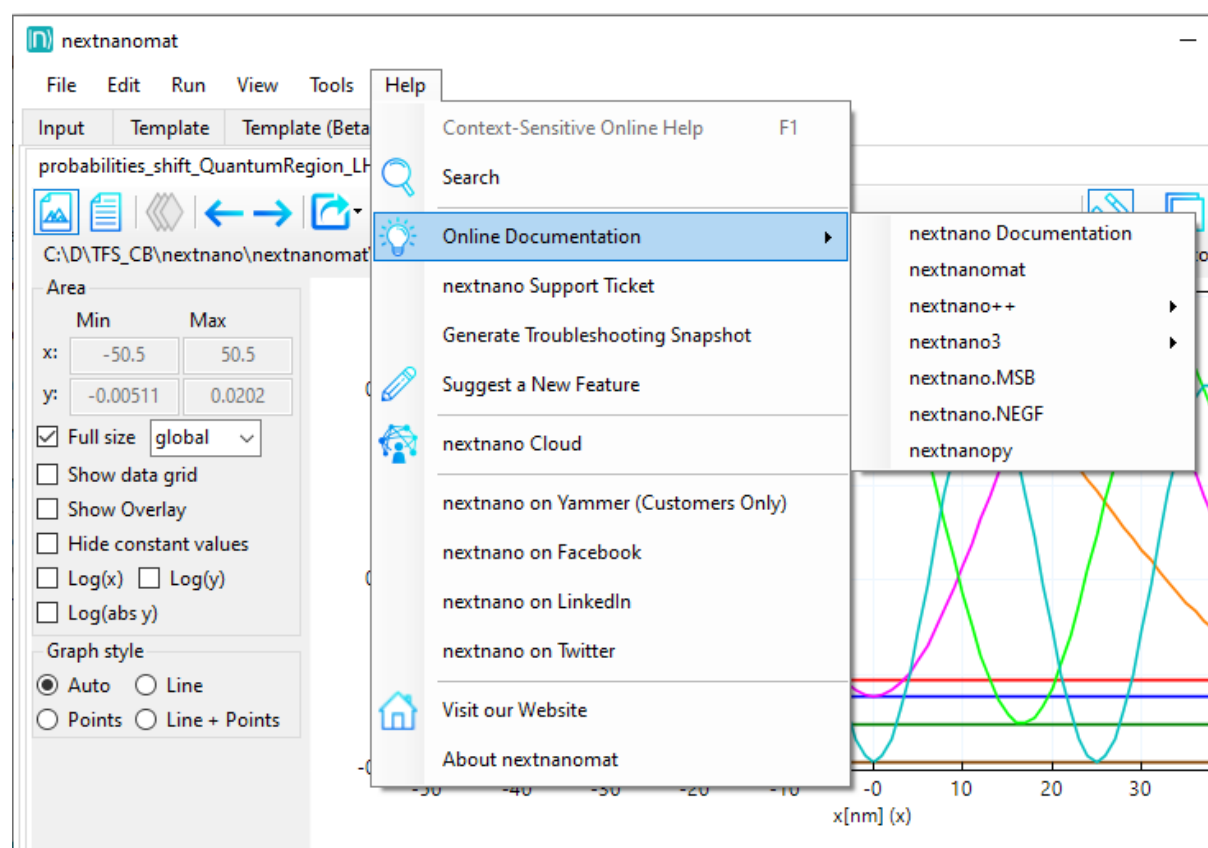


Figure 2.6.1: Help main menu functions

referred to and found in menu under **Tools -> Options**

3.1 Workspace

3.1.1 Simulation

Warning

Including memory usage in the log file notably extends time of simulation. Therefore, this option should be used only when necessary.

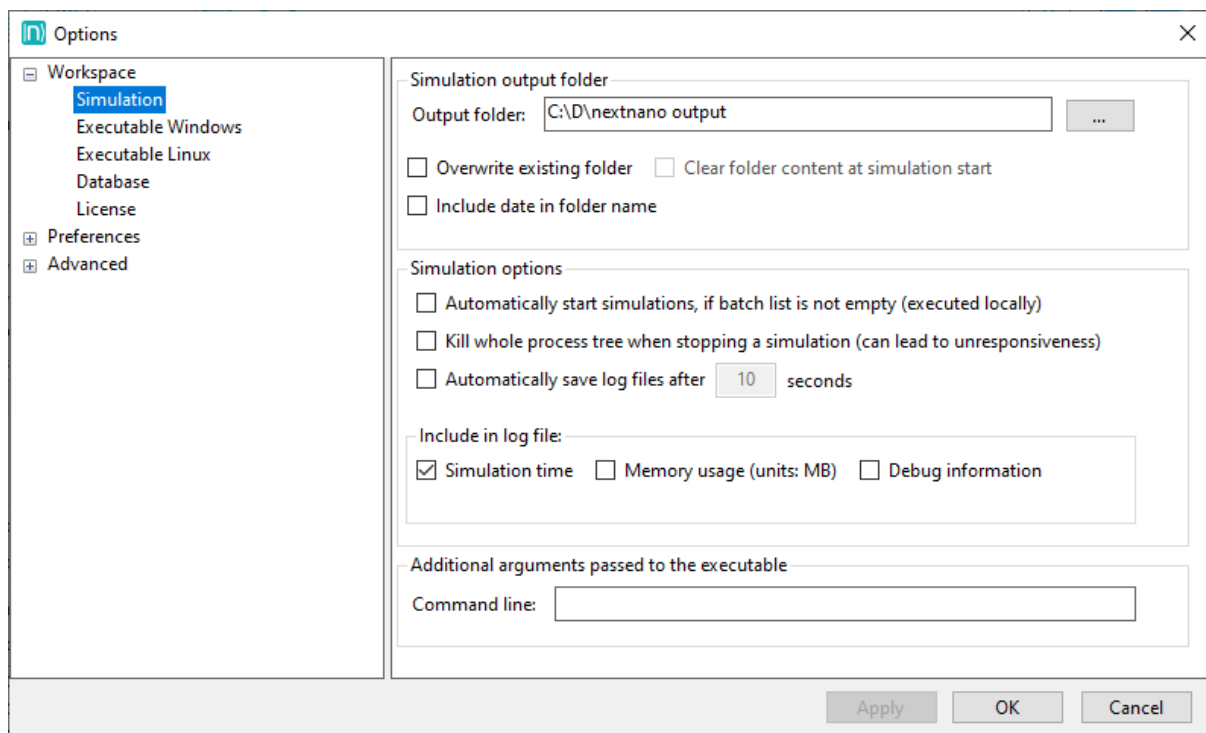


Figure 3.1.1.1: Settings for simulation and simulation output.

3.1.2 Executable for Windows and Linux

The paths to the nextnano tools can be specified here. Note, there are two tabs, one for the Windows and one for the Linux executables. When installing a package, the paths are automatically set at program start. If a specific version of a nextnano tool is downloaded, the path can be specified here. Please verify that you have the correct nextnano tool downloaded and specified, matching your license.

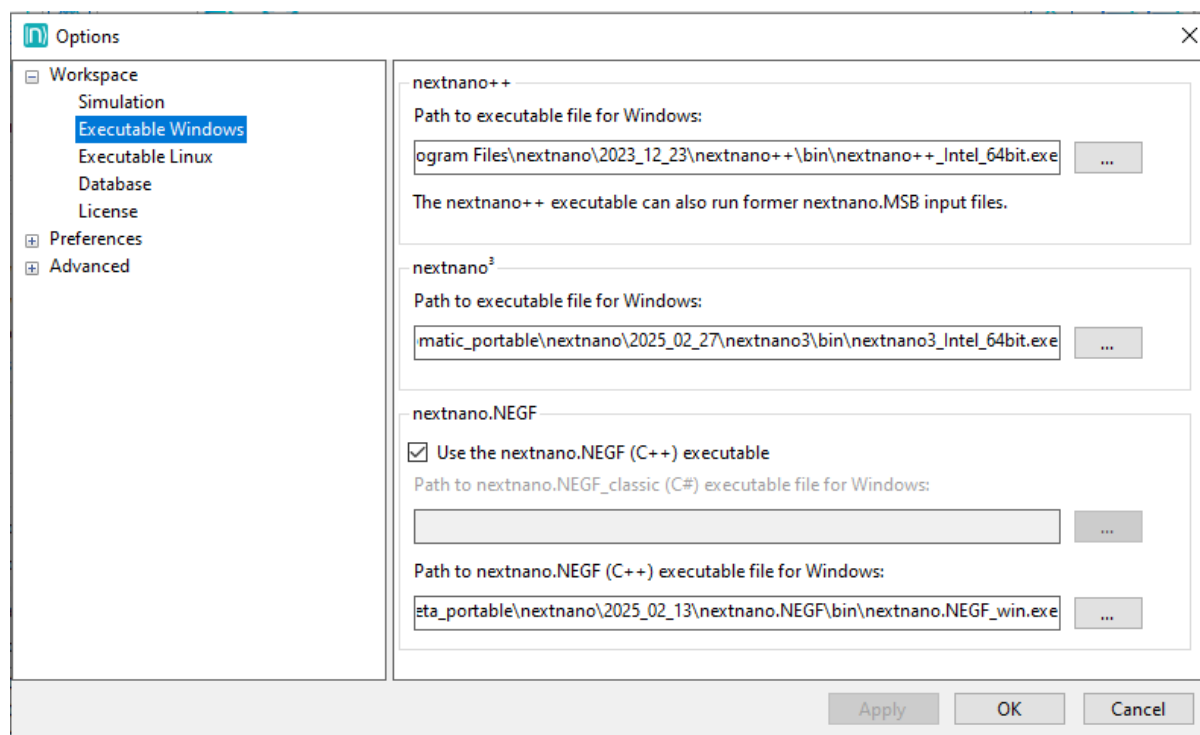


Figure 3.1.2.1: Settings for the Windows executables of the nextnano tools.

3.1.3 Material database

Path to the databases are prefilled at app start. If specific versions of the nextnano tools are used and downloaded separately from nextnanomat, then paths need to be adjusted manually.

3.1.4 License

3.2 Preferences

3.2.1 Editor/Input

In the editor settings, you can customize the syntax highlighting or font of the input tab. “Tab stop distance” defines the number of spaces a single Tab will be replaced with. Automatic replacement can be switched on by setting the check box accordingly. Additionally, Tab replacement can be triggered via the “Edit” main menu function for a specific input file.

3.2.2 View

3.2.3 Gnuplot export

Gnuplot can be used to export and save nextnano simulation results, see documentation: [Gnuplot export](#).

The default settings for the Gnuplot export can be adjusted in this options tab. For example, you can specify the line thickness, position of the legend (called key in Gnuplot) and many more.

If you want to use a **stylesheet** to incorporate advanced settings - like the specification of color sequence or line styles, you can either link an existing one or create one from the default nextnano Gnuplot settings. If you create a new one from the nextnano settings, the path to the stylesheet will be automatically updated. It is recommended to open that file with a texteditor to get familiar with the syntax. You can then easily adapt it to your needs.

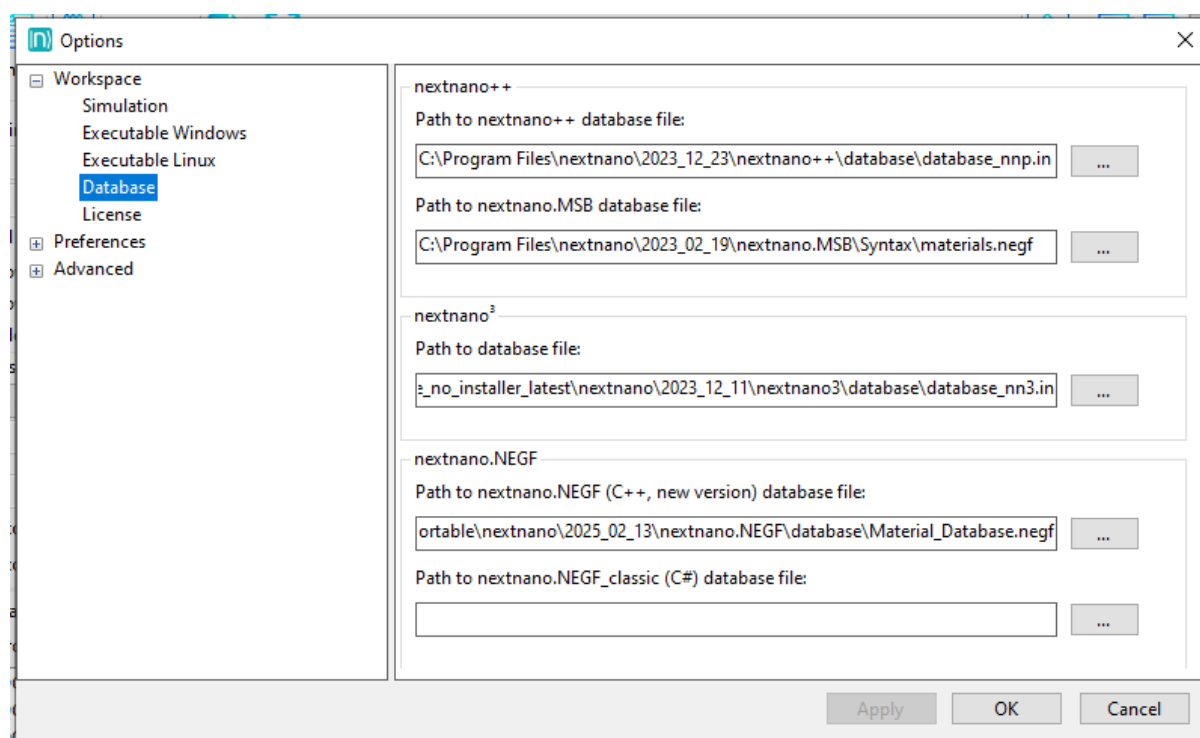


Figure 3.1.3.1: Settings for material databases.

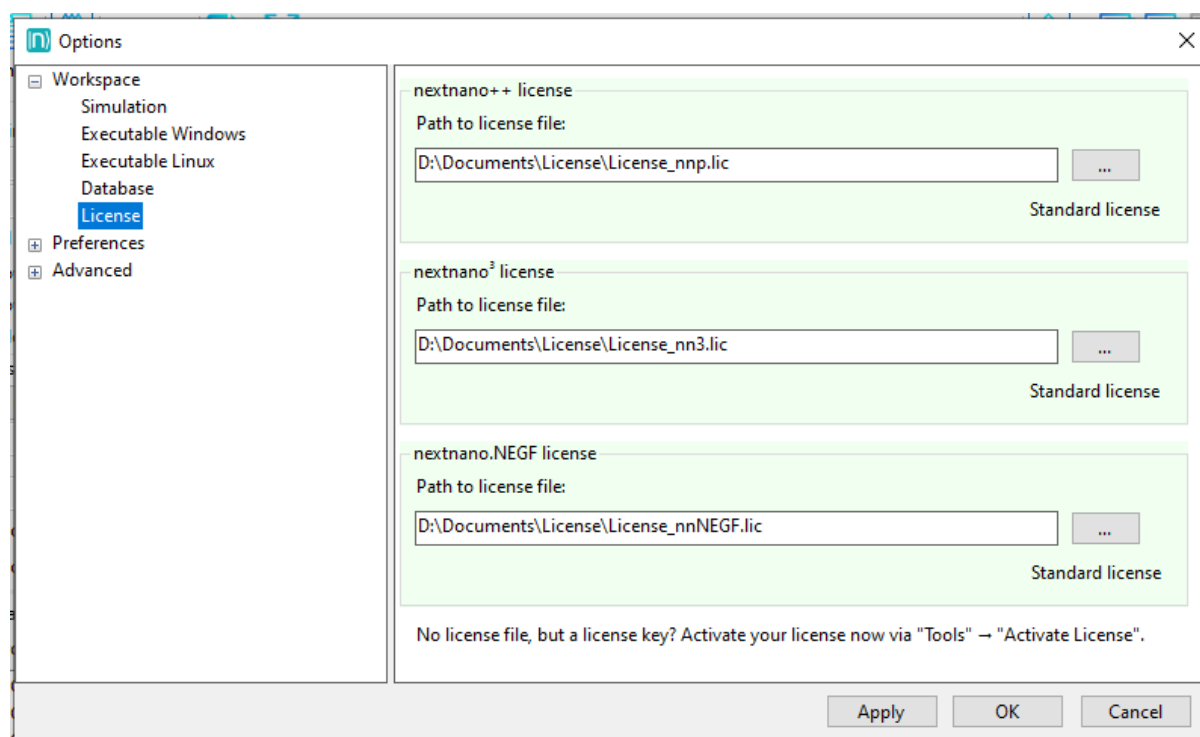


Figure 3.1.4.1: Settings for nextnano License(s).

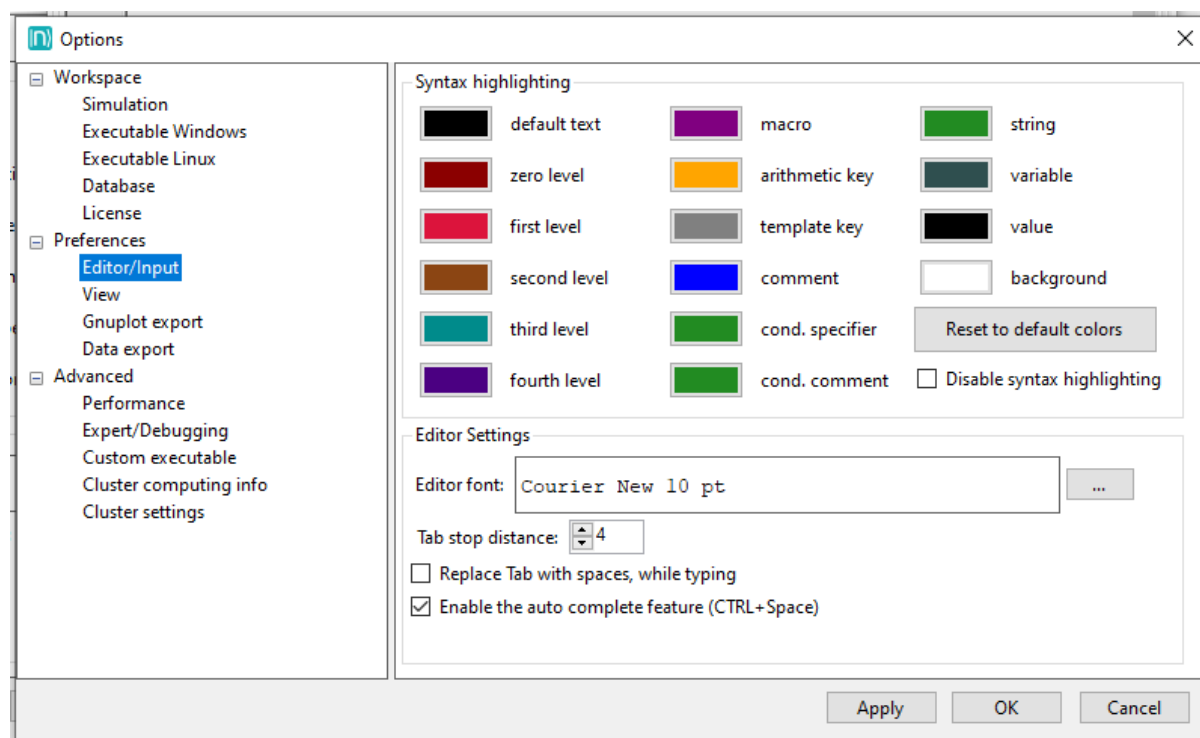


Figure 3.2.1.1: Settings for input tab.

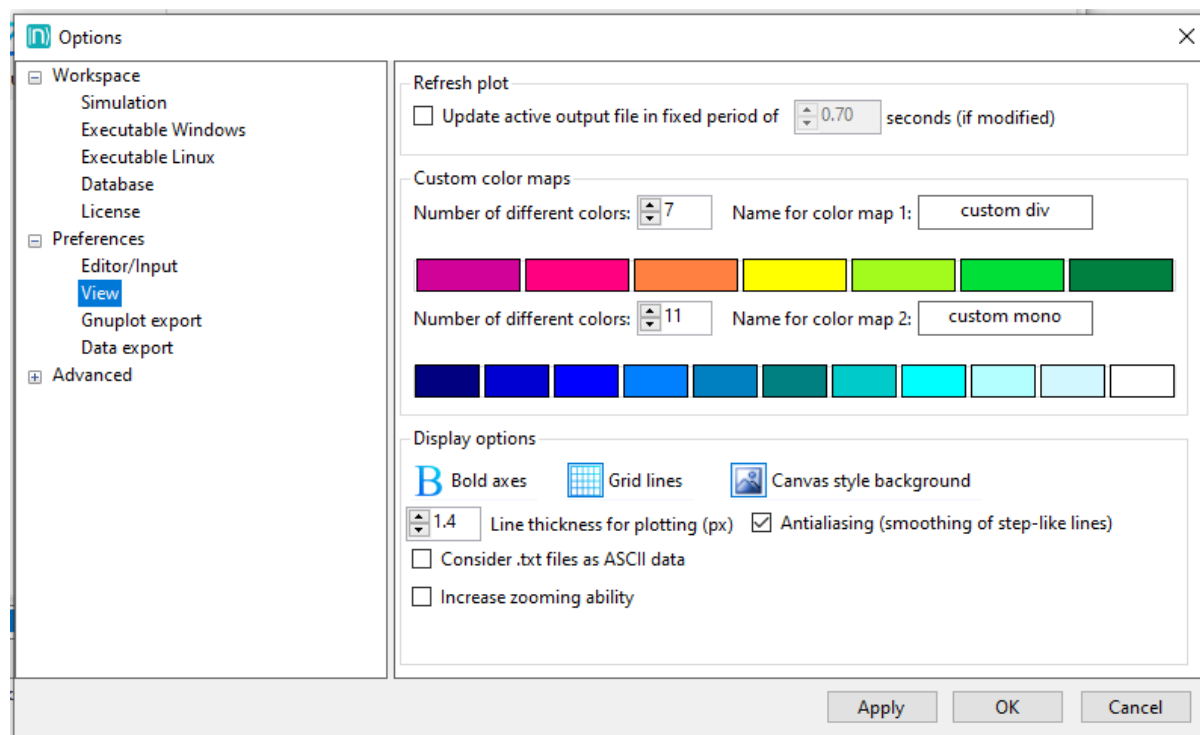


Figure 3.2.2.1: Settings for output tab and graph style.

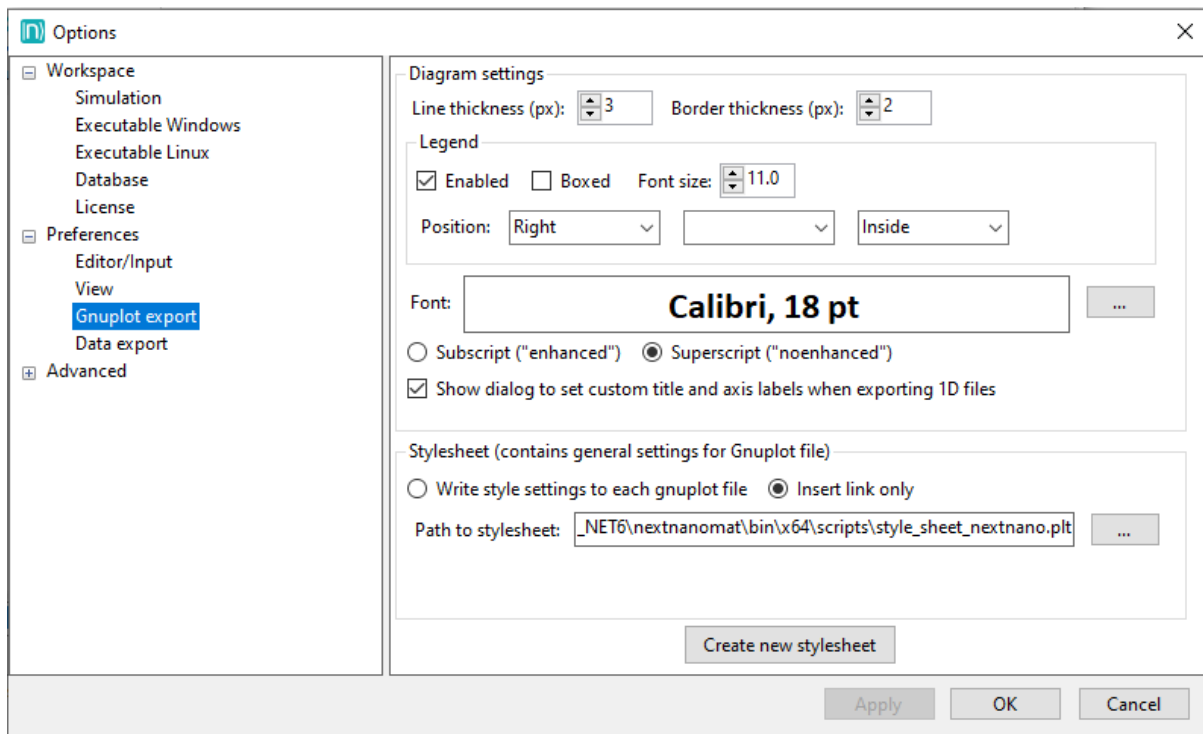


Figure 3.2.3.1: Settings for Gnuplot export and graph style.

Advantages of using a stylesheet

- Central, single file with your settings for plotting Gnuplot plots
- Use of links and variables (linking to the stylesheet) within the plot files
- Adjust settings at one place and automatically get the changes for all your plots
- Great for publications to achieve uniform style with possibility for last minute changes

To use a central stylesheet, select the checkbox *Insert link only*.

Note

You can create multiple stylesheets for different purposes. Set the path within Gnuplot settings to the one you want to use for new plots generated by *nextnanomat*.

3.2.4 Data export

write the path to the ParaView executable in your computer,

e.g. *C:\Program Files\ParaView 5.6.0-Windows-msvc2015-64bit\bin\paraview.exe*

These customizable data export software can be used to conveniently export data from the output tab. How to use: Click on the export option in the output tab and the currently displayed file will be opened with the specified software.

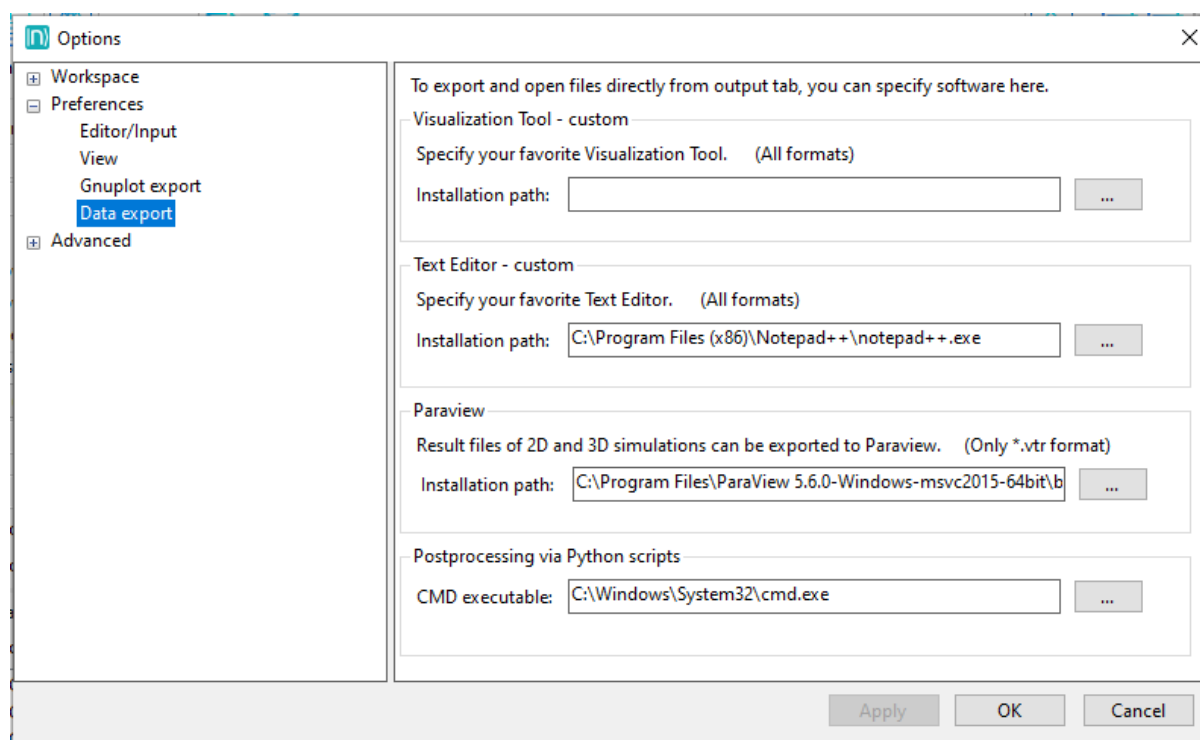


Figure 3.2.4.1: Settings for data export.

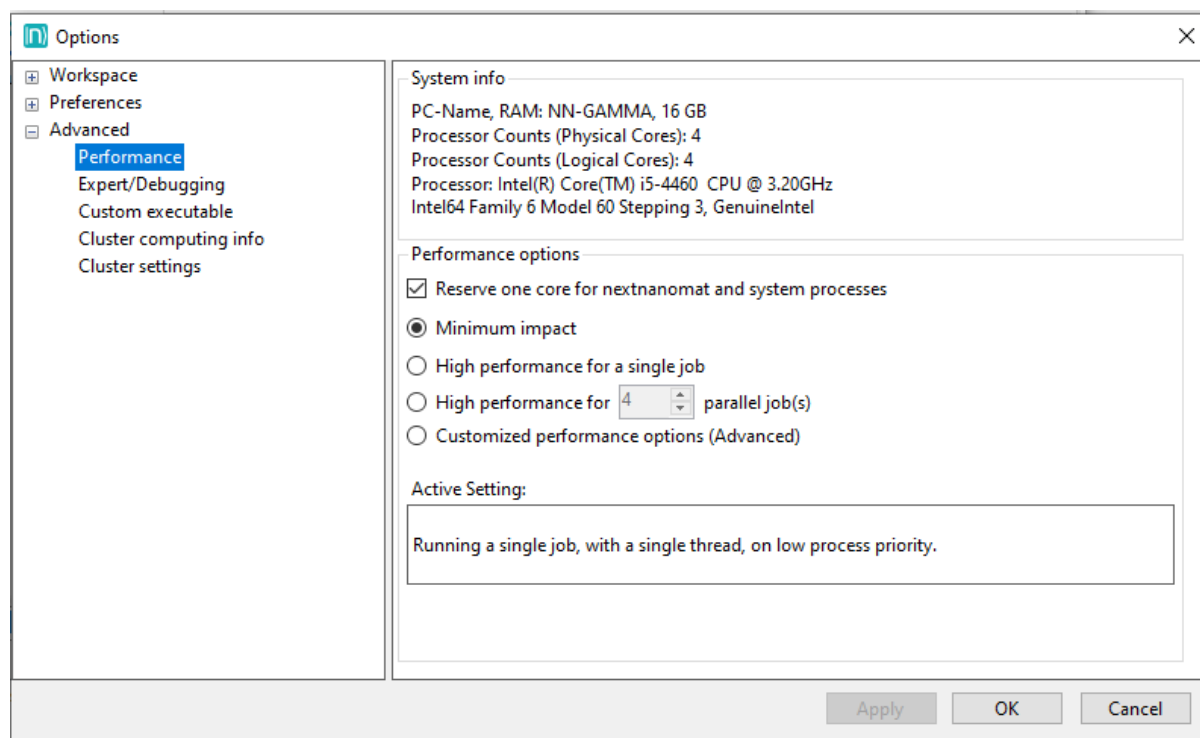


Figure 3.3.1.1: Settings for simulation performance.

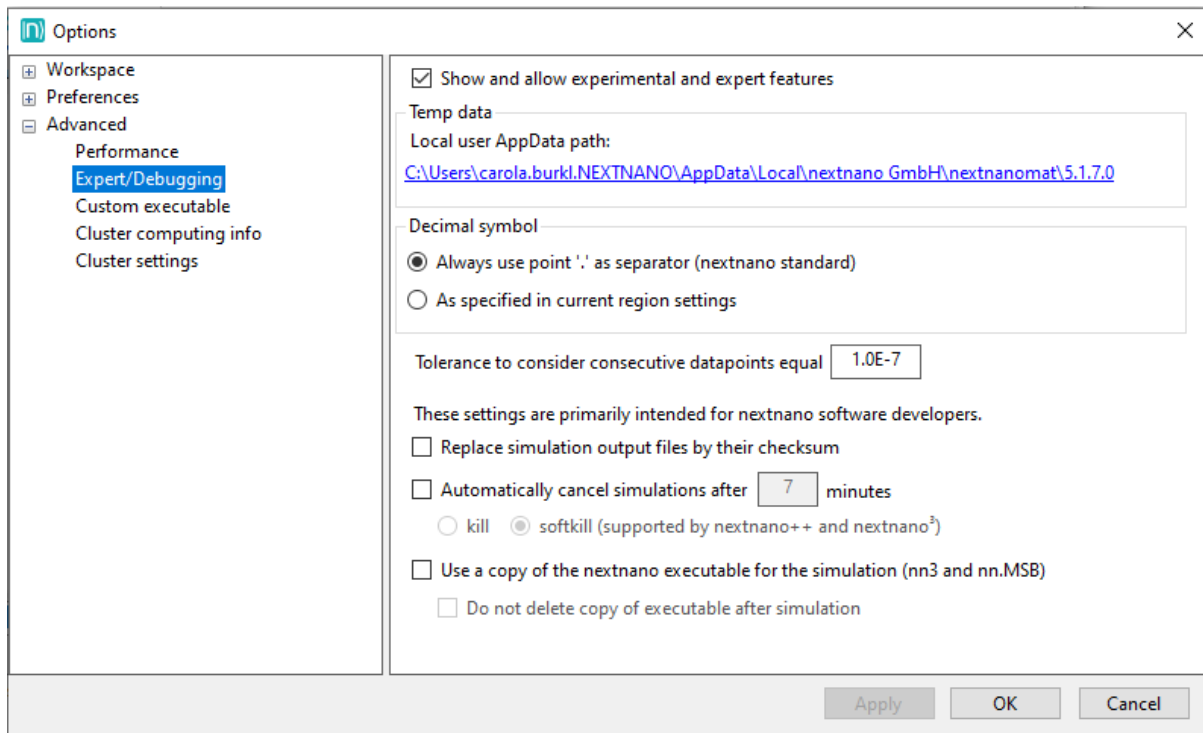


Figure 3.3.2.1: Settings for advanced user experience and debugging.

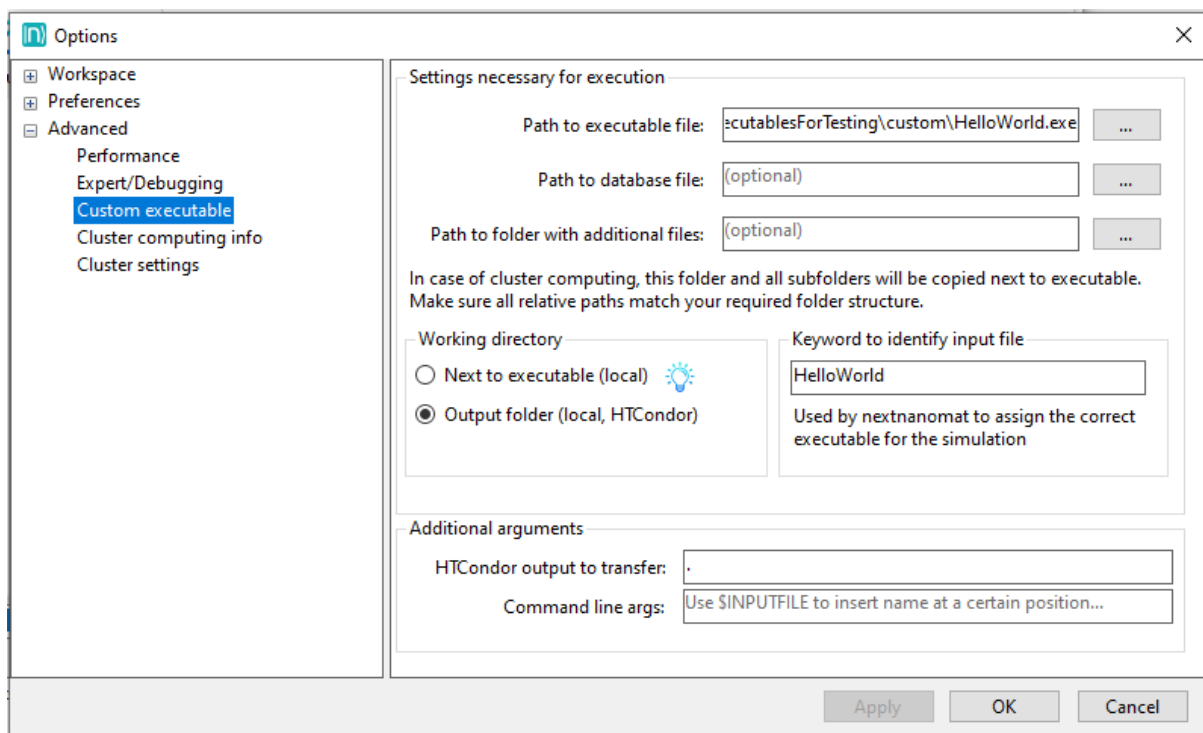


Figure 3.3.3.1: Settings for custom executable.

3.3 Advanced

3.3.1 Performance

3.3.2 Expert/Debugging

3.3.3 Custom executable

In the settings shown in [Figure 3.3.3.1](#) a custom executable can be defined, to be executed by the workflow manager *nextnanomat*.

Note

This is no default use-case for *nextnanomat* and should only be used by advanced users. Additionally, as it is a nice-to-have feature only, support regarding this feature is limited.

Most executables which can be started by command line, can be started by *nextnanomat*. This allows the usage of the batch list for polling and job execution control, as well as the visualization of output results (if the format is supported) or comfortable editing of input files. In many cases a combination with our cluster computing feature HTCCondor is possible.

After defining the path to the executable as well as the working directory, a keyword identifier is necessary.

In case your executable needs input files:

You need to find a common string within these files, thus *nextnanomat* can choose the correct executable for each job execution.

In case your executable does not need input files:

You still need to create some dummy files as placeholder or job submission files. The keyword can be arbitrary.

For example the *HelloWorld* executable (see [Figure 3.3.3.1](#)), which is executing a fixed script independent of input variables, can be started by submission of a text file containing *HelloWorld*. This enables the workflow manager *nextnanomat* to start and monitor the execution and to collect the results.

Optionally a path to a folder structure or the database can be defined, if the executable needs additional files during execution. Specific command line arguments can be added at the bottom of this settings tab. The variable `$INPUTFILE` will insert the filename of the actual input file at the specified position of the command line arguments.

If it does not work as wished for, we recommend to switch the flag `output additional debug info` on (Expert Settings). The log then contains the command line arguments *nextnanomat* uses to start the application. Thus allows you to debug easily.

Example: How to run TiberCAD from *nextnanomat*

To run TiberCAD you need to finish its installation and put your TiberCAD license file into the respective folder. Move the example or your personal input files to a folder with write permission. Choose the following settings in the custom executable tab, [Figure 3.3.3.2](#):

Note

A special folder has to be created, which contains all input *.tib*, geometry *.geo* and mesh *.msh* files on the same hierarchy level. The path to that folder needs to be specified in the settings for the custom executable.

This enables the editing - including syntax highlighting - of the input files within *nextnanomat*. And furthermore the parallel execution of input files (see [Figure 3.3.3.3](#)) and visualization of the output.

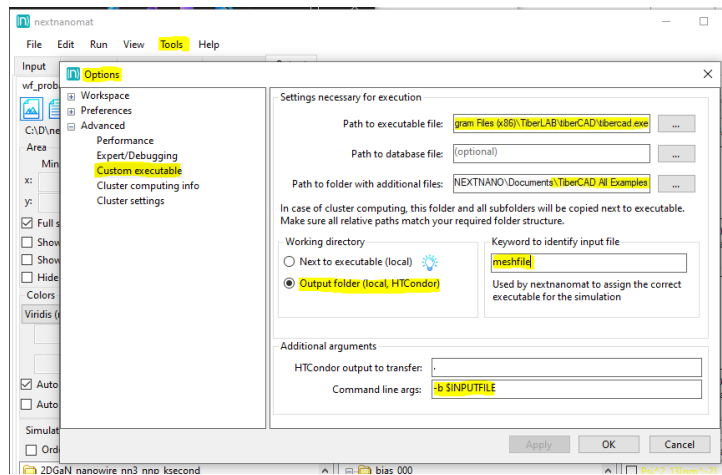


Figure 3.3.3.2: Settings to run TiberCAD as a custom executable.

File Edit Run View Tools Help					
Input Template Template (Beta) Simulation Output					
Job number	Input file	Start time	Duration	Status	Host
1	quantum_poisson.tif	19:20:17	00:00:05	done	Local
2	bit.tif	19:36:46	00:00:06	running (1)	Local
3	bulk.tif	19:36:47	00:00:01	done	Local
4	bulk_nIPD.tif	19:36:47	00:00:01	ERROR	Local
5	diode_ID.tif	19:36:47	00:00:05	running (4)	Local
6	DSC.tif	19:36:48	00:00:04	running (3)	Local
7	InAs_qw.tif	19:36:49	00:00:02	ERROR	Local
8	InGaAs_1D.tif	19:36:51	00:00:01	running (2)	Local
9	ITO_P3HT.tif			in queue	Local

Figure 3.3.3.3: Parallel execution of TiberCAD files by the *nextnanomat* Job Manager.

COLOR MAPS

On this page you will find some background information about scientific visualization of data. Additionally, the color map choices and customization options of the [nextnano software](#) are explained. The references and sources used to generate these maps can be found in the last section.

4.1 Introduction

Color maps for scientific software need to intuitively represent data without visual distortion.

4.2 Implementation

4.3 References

EXPORT FUNCTIONALITIES

5.1 Exporting to ParaView

ParaView is a free graphing utility specialized for 2D and 3D visualization. It can be downloaded from here: [ParaView](#). To be able to export results in *nextnanomat* directly to ParaView specify the Executable file in Tools - Options:

5.1.1 Setting Up nextnanomat

For the exercises we will require two different settings in *nextnanomat* (our user interface).

Direct Exporting

The first setting is the simplest one and it is used to open the selected .vtr file in *nextnanomat* into ParaView. It will require to follow the next steps:

1. In the menu bar of *nextnanomat*, click on Tools $\rightarrow$ Options $\rightarrow$ Data Export
2. Provide the path of the executable of ParaView in your computer in the field ParaView installation path (e.g. *C:\Program Files\ParaView 5.11.1\bin\paraview.exe*).
3. Click OK
 - In *Data export*, write the path to the ParaView executable in your computer, e.g. *C:\Program Files\ParaView 5.6.0-Windows-msvc2015-64bit\bin\paraview.exe*

Exporting with Python Scripts

The second setting is used for launching ParaView through Python scripts. As we will see in the next examples, this is very useful for post-processing several input files, including setting up overlayed plots. For this setting it will be required to follow the same steps as before, but not including the path of the executable of a terminal window (*cmd.exe* for Windows, as example) in the field Python executable file.

Figure 5.1.1.1 presents the final result provided the required information.

Installing the Python scripts in your Computer

Copy the three python scripts shown above in a folder and create an empty file called *__init__.py*. This last file is required in order *Gui5.py* be identified as a package by Python.

A more powerful resource in the integration of *nextnano* and ParaView is the possibility to incorporate Python scripts in the post-processing of the simulation data. For performing the next exercises, we will use the second setting described above, where ParaView will be called from a command line in a terminal application (for example, *C:\Program Files\cmd.exe* in Windows).

Actually, we will make use of three scripts:

- *Paraview_postprocessing.py*, that contains all post-processing tasks to be performed on each file exported

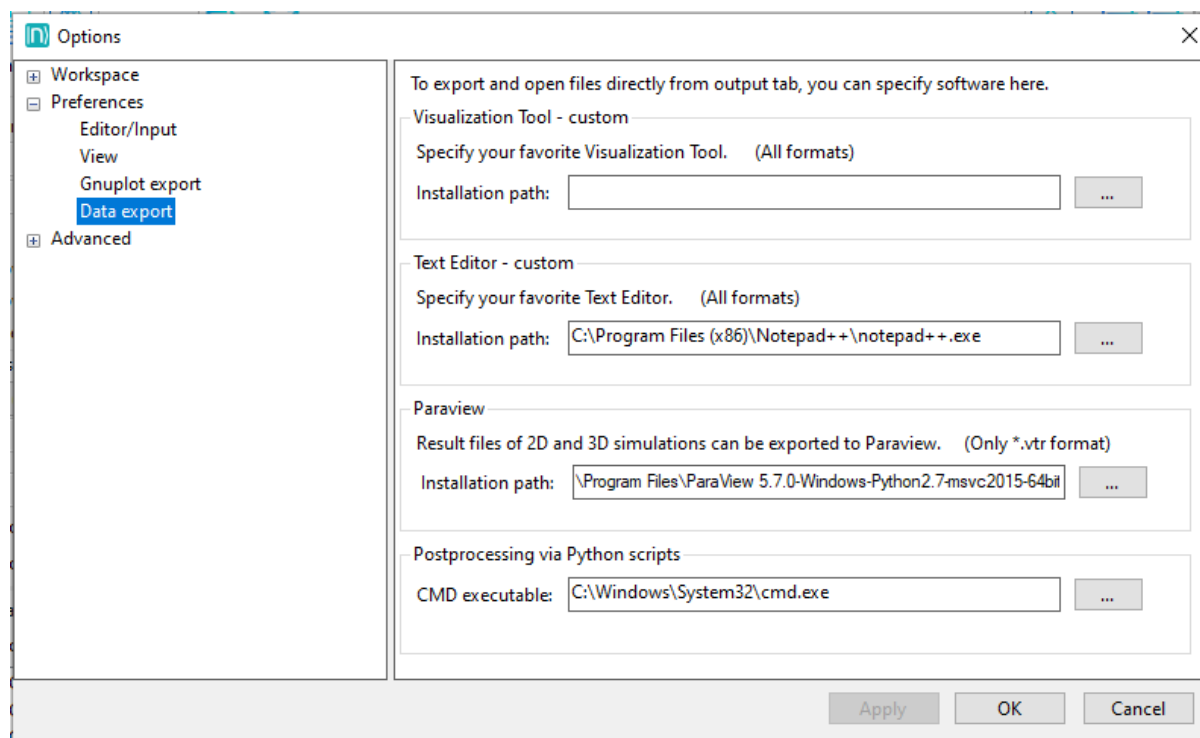


Figure 5.1.1.1: Required settings for exporting .vtr files from *nextnanomat* to ParaView in this tutorial. This menu is accessed clicking in Tools :math:`\rightarrow` Options :math:`\rightarrow` Data Export, within *nextnanomat*.

- *dialog_window.py*, that will capture information of *nextnanomat* and to launch a new window for selecting files to be exported, and predefine some global properties of the plots. It is the actually the one that will launch ParaView and the post-processing script.
- *Gui_5.py*, is a simple script for generating the window for selecting files, described before. In order to be recognized as a package to be imported by the previous script, it is required to create an empty file called `__init__.py` in the same folder where this script will be stored.

Copy them in on your computer. All scripts shall be stored in the same directory.

5.1.2 Direct Export

1. Specify path to ParaView in Tools - Options if you have not done so before
2. **Select a 3D .vtr file of the simulation output folder and click on the Export and open in specific format button.**
Choose Open File with ParaView (ParaView will open automatically.)
3. Within ParaView:
 - a. (The selected file should be highlighted automatically.) Click on Apply.
 - b. Other settings:
 - I. Representation: Surface
 - II. Cell/Point Array Status: Choose the array to be displayed. For example, one can display the file `bandedges.vtr` and choose the array `Gamma` that corresponds to the Gamma conduction band. Click on Apply.
 - III. Coloring: Choose the array to be displayed. (Gamma, for example)

Now you can play with the tool. Rotating, changing opacity and adding filters. ParaView is a very rich tool, hence it requires some time to explore all its capabilities. We recommend investing some time to learn about its filters:

- First, the file should be highlighted and then click on **Filter** $\Rightarrow$ **Alphabetical**.

Some interesting ones are:

- Edit cells by region
- Calculator
- Contour

Using **File** $\Rightarrow$ **Save State**, the values of all variables can be stored on your PC. These states can be reloaded, which is useful for the generation of scripts.

For further information, we recommend a good tutorial from [TACC](#).

When to use: This single-click-procedure is very convenient when we want to load a single file in the Pipeline Browser of ParaView. Having already experience with this tool it will require some basic steps to display the data on the screen of this platform.

5.1.3 Exporting 2D Outputs with Python Scripts

Input Files:

- *2DQuantumCorral_nnp.nnp* from the tutorial about **Electron wave functions in a cylindrical well (2D Quantum Corral)** of [nextnano++](#).

Scope of the tutorial:

- Export 2D vtr files from *nextnanomat* to ParaView
- Choosing predefined filters in ParaView
- Preset of some parameters for the plots
- Defining overlay files
- Dealing with files with mixed units

Introduced Keywords:

- `output{ format2D }`
- `output{ format3D }`

Relevant output Files:

- *\bias_00000\bandedges.vtr*
- *\bias_00000\Quantum\probabilities_quantum_region_Gamma_.vtr*
- *\bias_00000\Quantum\probabilities_shift_quantum_region_Gamma.vtr*

Scripts:

- *nn_ParaView_Integration.py*
- *nn_ParaView_GUI.py*
- *nn_ParaView_Plotter.py*

ParaView supports the use of Python scripts for loading and displaying objects in its Pipeline Browser. In the next tutorials we will present how to perform this integration using nextnano.

Our scripts come to assist not only to load the results of the simulations in this platform, but also allow to preset the application of some filters and some global settings even before launching the program.

- *Setting up nextnanomat for exporting .vtr files using Python scripts*
- *Scripts and files of this tutorial*

- *Single plot*
 - *In nextnanomat*
 - *In the Integration GUI*
 - *In ParaView*
- *Plot with overlay*
 - *In nextnanomat*
 - *In Integration GUI*
 - *In ParaView*
 - *Changing Scale Factor of warped files of shifted probability density*
 - *Changing Scale Factor of warped files of the band edges overlayed with shifted probability density*

Setting up nextnanomat for exporting .vtr files using Python scripts

In order to load files in ParaView through Python scripts from *nextnanomat* it will require the following settings:

1. In the menu bar of *nextnanomat*, click on Tools >> Options >> Data Export
2. Provide the path of the executable of ParaView in your computer in the field Paraview installation path (e.g. *C:\Program Files\ParaView 5.11.1\bin\paraview.exe*).
3. Provide the path of the executable of a terminal window in the field Python executable file (e.g. *C:\Windows\System32\cmd.exe*).
4. Click on OK

Figure 5.1.3.1 presents the final result provided the required information.

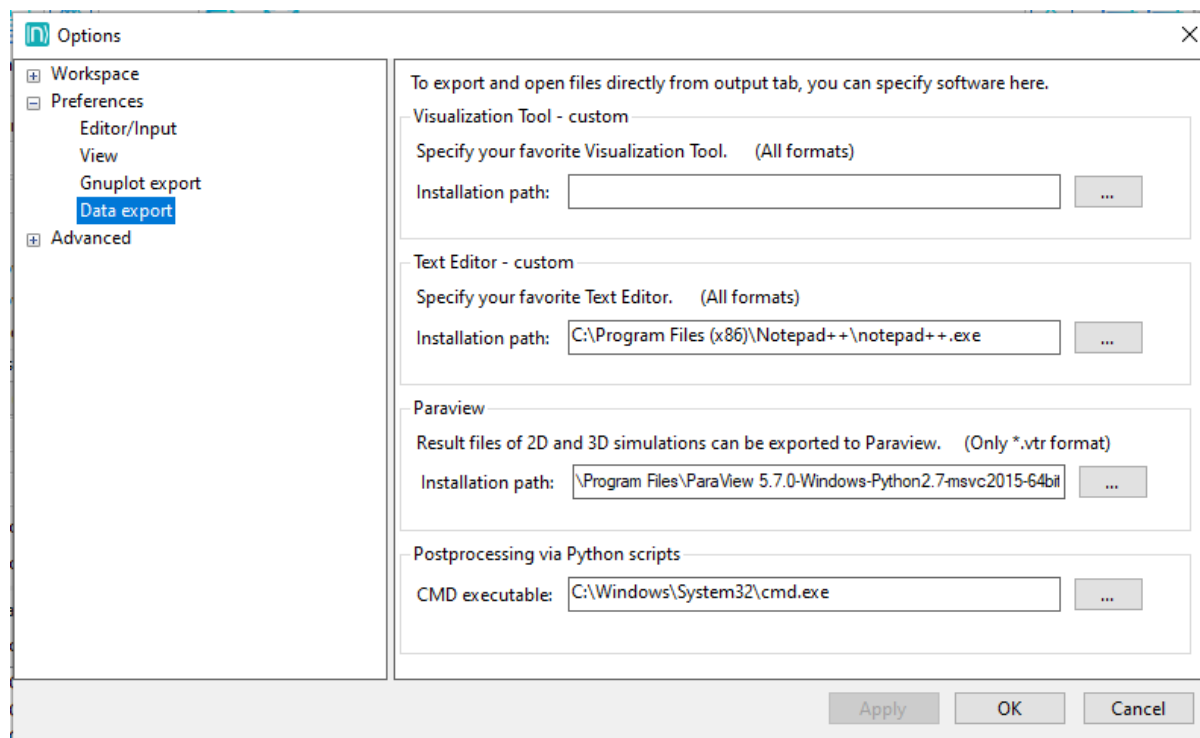


Figure 5.1.3.1: Settings in *nextnanomat* for launching ParaView using Python scripts

These scripts are very useful for post-processing several data files, specially for setting up overlayed plots as we will see below.

Scripts and files of this tutorial

In the nextnano package you will find all necessary files we will use in this tutorial. The scripts for the integration are in the folder *Scripts\ParaView*.

The first (*nn_ParaView_Integration.py*) captures information of *nextnanomat* and launches the graphical user interface shown in Figure 5.1.3.2 using the script *nn_ParaView_GUI.py*. After pressing the Launch ParaView button, the script *nn_ParaView_Plotter* is copied to the simulation folder and modified with all information captured in the user interface. Automatically *nextnanomat* will launch ParaView.

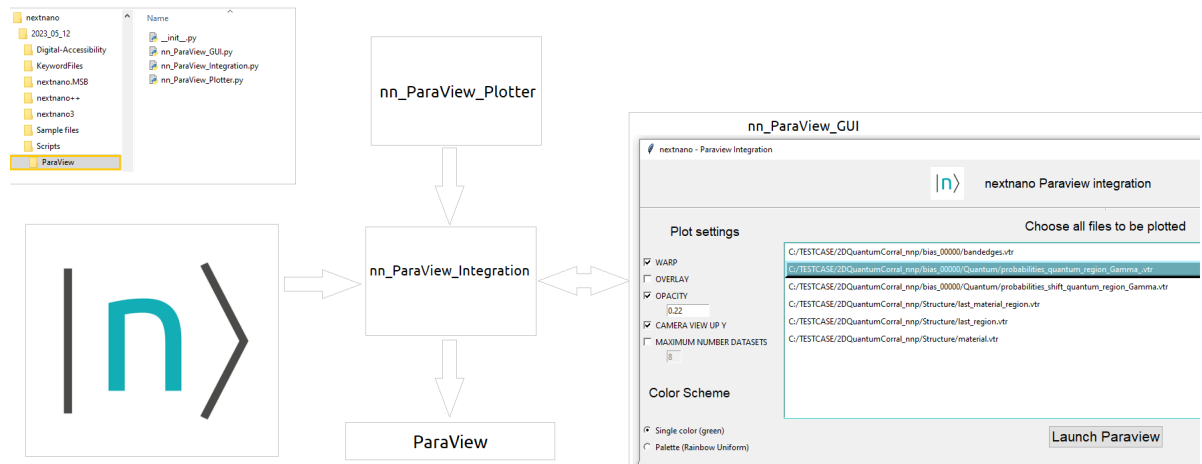


Figure 5.1.3.2: Scripts for integration of *nextnanomat* and ParaView and the interactions between the components.

ParaView supports the use of Python scripts for loading and displaying objects in its Pipeline Browser. In the next topics, we will illustrate some examples how this can be done.

The input file *2DQuantumCorral_nnp.in* used in this tutorial can be found in the folder *\Sample files\nextnano++ sample Files* within the nextnano package. This corresponds to a two-dimensional simulation and, for now, we are interested in the results of the quantum computations.

In the section `output{ }` of the input file we can specify the format of these files whose default is `AvsBinary_one_file`, for the case of the 2D and 3D plots, because being less memory demanding than the other available options. Nevertheless, ParaView does not support this format. In this way, for exporting files from nextnano into ParaView it is required that the results be coded as VTK file. For this reason it is important to specify in the input file

```
``format2D = VTKAscii (for 2D plots)``
```

or

```
``format3D = VTKAscii (for 3D plots)``
```

as we can observe in the example.

For making our discussion more interesting we will modify the input file *2DQuantumCorral_nnp.in* reducing some dimensions, and adding more information about the band edges, as illustrated below.

```

8      $RADIUS_CORE = 4.0                                     # Core_
↪radius
9      $RADIUS_SHELL = 6.0                                    #_
↪Shell radius

15     $GRID_SPACING1 = 0.5                                    #_
↪(DoNotShowInUserInterface)
16     $GRID_SPACING2 = 0.5                                    #_

```

(continues on next page)

(continued from previous page)

```

17  ↪(DoNotShowInUserInterface)
    $GRID_SPACING3 = 0.5 / 3                                     # ↪
18  ↪(DoNotShowInUserInterface)
    #-----#
19
20  #-----#
21  # derived parameters                                         #
22  #-----#
23  $START_x = $CENTER_x - $RADIUS_SHELL - 1.0                  # ↪
    ↪(DoNotShowInUserInterface)
24  $START_y = $CENTER_y - $RADIUS_SHELL - 1.0                  # ↪
    ↪(DoNotShowInUserInterface)
25  $END_x   = $CENTER_x + $RADIUS_SHELL + 1.0                  # ↪
    ↪(DoNotShowInUserInterface)
26  $END_y   = $CENTER_y + $RADIUS_SHELL + 1.0                  # ↪
    ↪(DoNotShowInUserInterface)

159  classical{
160      Gamma{}
161      HH{}
162      LH{}
163      SO{}
164      output_bandedges{}
165  }

```

Let us simulate the modified input file using *nextnanomat* and observe the results in the output folder. Suppose that we are interested to plot the file *probabilities_quantum_region_Gamma.vtr* present in the *Quantum* folder. It represents the probability density of several states of our device. Selecting the plots in the menu at the bottom right corner of *nextnanomat* we can display the beautiful set of patterns for this geometry. Let us choose one of them – the plot Ψ^2_{14} , as example.

After that, the whole file (not only the selected plot) will be loaded in the Pipeline Browser of ParaView, but no other task will be performed. This is what we called before as the direct method. Let us see how we can improve this integration.

Now select the file to be exported (a .vtr file), click in the icon **Export and Open in specific Format** and select **Open File with Paraview**, as shown in [Figure 5.1.3.3](#).

After that the whole file (not only the selected plot) will be loaded in the Pipeline Browser of Paraview, but no other task will be performed. This is what we called before as the direct method. Let us see how we can improve this integration.

Single plot

Now let us check how the integration is performed using these scripts.

In nextnanomat

Let us return to *nextnanomat* and select some .vtr file as we did it before (*Quantum\probabilities_quantum_region_Gamma.vtr*). Click on the icon **Export and Open in specific Format** and select **Call Python Postprocessing scripts**, as shown in [Figure 5.1.3.4](#). Within the *\Scripts\ParaView* folder in the *nextnano* package select the script *nn_ParaView_Integration.py*.

In the Integration GUI

A new panel will be opened with a list of all .vtr files of the simulation folder. By default, the selected item in the list of files to be plotted corresponds to the file that was initially selected in *nextnanomat* when launching the Python script. For deselecting it, click once on its file name. At least one file must be chosen in order to open ParaView. For selecting more than one file, just click on it (not need to press the CTRL key).

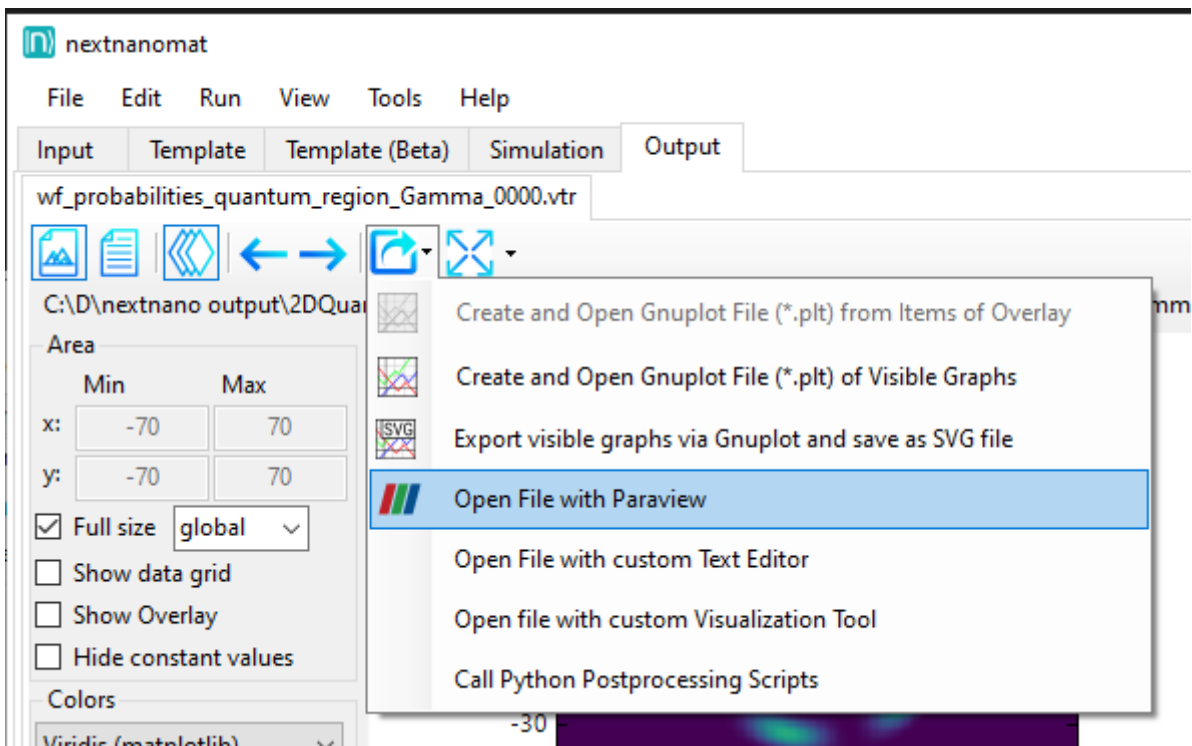


Figure 5.1.3.3: Button for exporting datafiles to ParaView using the direct method.

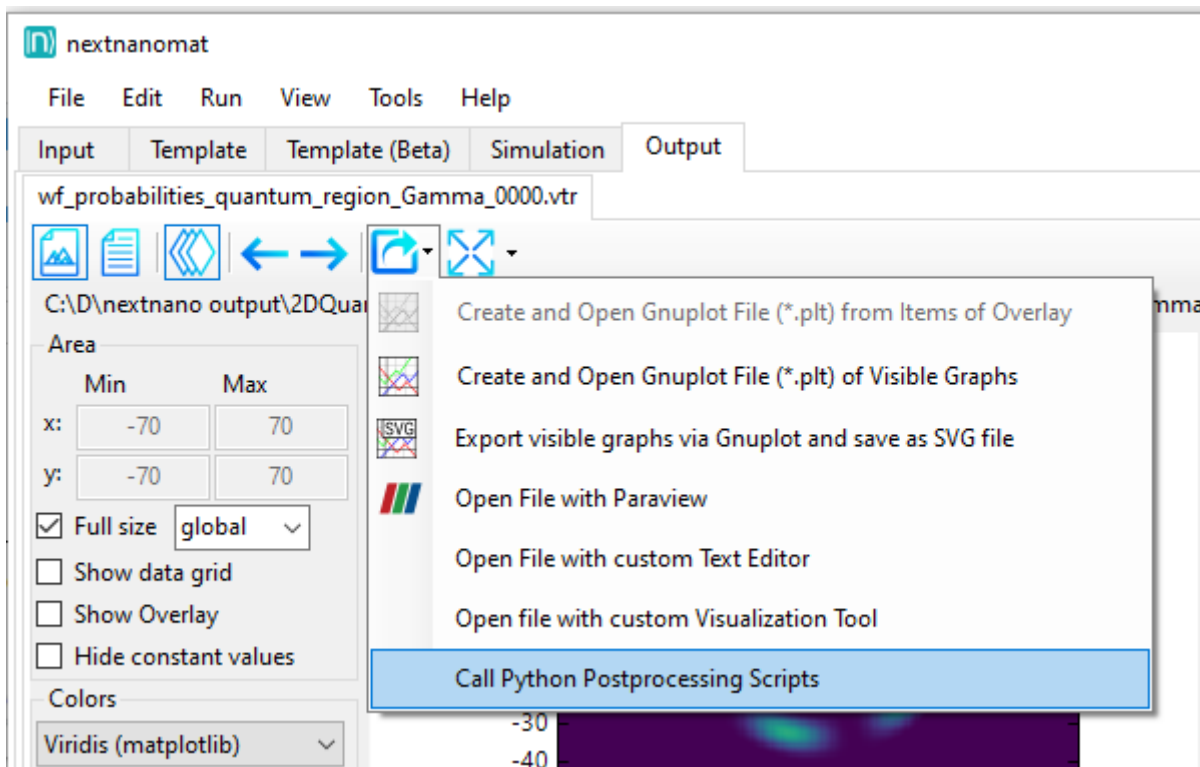


Figure 5.1.3.4: Button in nextnanomat for exporting datafiles to ParaView using Python Scripts.

At the left of this window you can find some plot settings that can be chosen, and they correspond to general settings. In another words: all selected files to be plotted will be affected by these general settings.

Instead of describing all these settings one by one in this section of the tutorial, it is more convenient to introduce them with practical examples. Let us start with a simple transformation (in ParaView usually called *Filter*) on the original 2D plots. We will select **WARP** in the menu for creating a 3D plot, where the third axis corresponds to probability density at each (x,y) position of the plane. Additionally, we will select **OPACITY** and **Single color** as color scheme. Click on the button **Launch ParaView**. The final result of applying these settings is shown in [Figure 5.1.3.5](#).

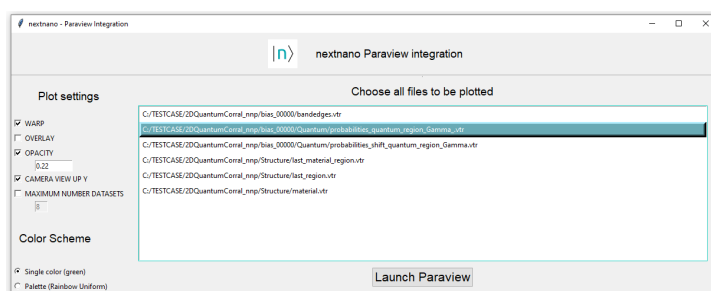


Figure 5.1.3.5: Settings in Graphical user interface for integration of *nextnanomat* and ParaView for the case of single plot.

In ParaView

Now we can observe a larger number of plots is presented in the **Pipeline Browser**. The first object has the same name of the absolute path of the file plotted. Below it, each column (dataset) of the file is plotted with corresponding sequence of post-processing tasks and settings defined previously in the interface.

As we can see in the next images, they were plotted in a single color and semi-transparent.

The [Figure 5.1.3.6](#) we can observe that for this example, where the option **WARP** was selected, the original 2D representation for this probability density function ($\Psi^2_{1}(nm^{-2})$, $\Psi^2_{2}(nm^{-2})$,...) will be presented in colors, while its “warped” version ($\Psi^2_{1_warped}$, $\Psi^2_{2_warped}$,...) will be displayed as semi-transparent surfaces.

The warped objects are the result of applying the **WarpByScalar** filter of ParaView to each component, that converts 2D functions in a 3D representation, where the third axis, in this case, corresponds to the value of probability density at each (x,y) in the plane.

The **Scale Factor** used for the warped plot is estimated by the script, and it can be dynamically changed within ParaView, moving the corresponding slider in the **Properties** tab or filling a numeric value in front of it. Let us try Choosing in ParaView some of the warped files, write the number **300** after the slider, and press the button **Apply**. Looking at the other warped plots of this file, you will observe that this change was applied also to them. This is an important feature implemented by our script and only applies to the **Scale Factor**. [Figure 5.1.3.7](#) shows one example how the interconnection of the components react when the **Scale Factor** of any component is changed.

Looking at the **Information** tab under the **Pipeline Browser** of ParaView ([Figure 5.1.3.8](#)) see a larger number of datasets was actually exported, but only 8 are displayed. This is because we limited to a maximum of datasets to be displayed through the variable **MAXIMUM_NUMBER_DATASETS** as default in the *nn_ParaView_Plotter* script. This limitation is used to avoid data explosion in ParaView. Nevertheless, this number can be changed by modifying the value of **MAXIMUM_NUMBER_DATASETS** in the user interface for nextnano-ParaView integration, as we will do in the next section.

Hint

The presets captured in the user interface for integration ([Figure 5.1.3.5](#)) aims to automate the configuration of each object to be displayed in the **Pipeline Browser**. Nevertheless they can be dynamically changed in

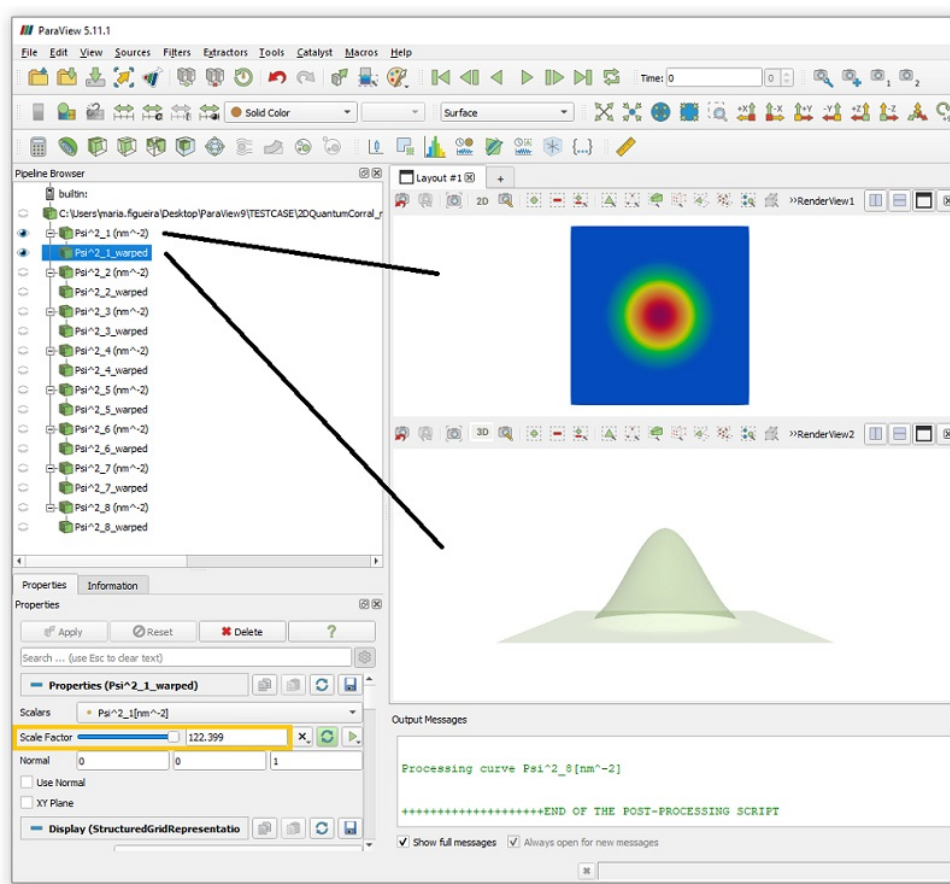


Figure 5.1.3.6: Organization of the files in the Pipeline Browser of ParaView. The imported datafile is registered with the absolute path of the file. For each component of the datafile, the 2D map and the warped version are presented. Highlighted in orange is the value estimated for the Scale Factor for the WarpByScalar filter.

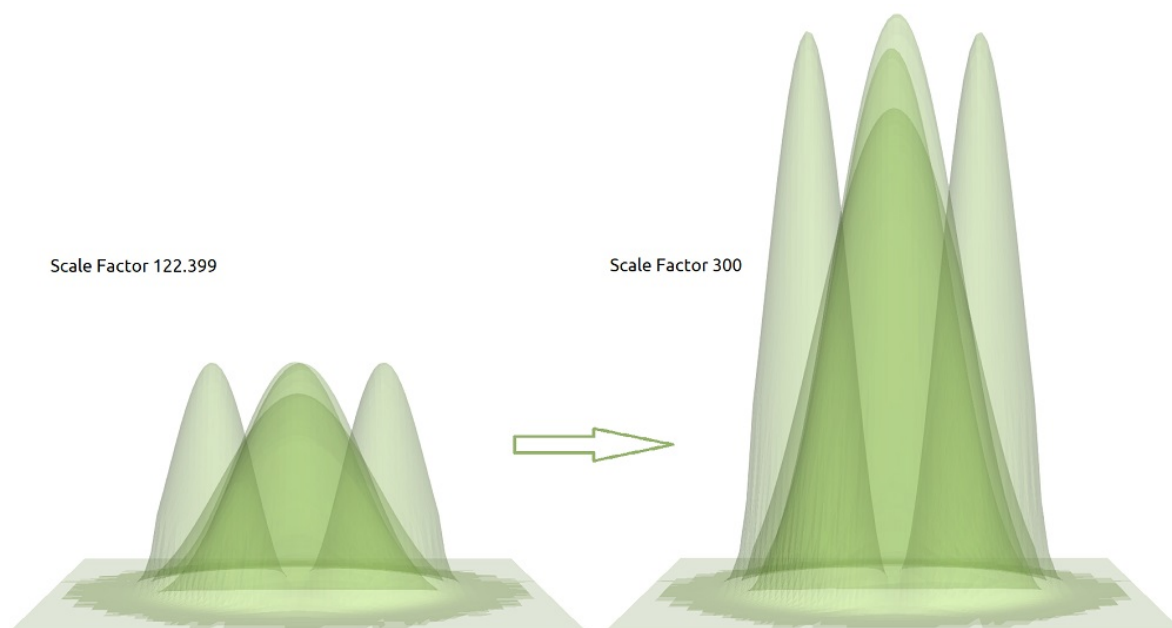


Figure 5.1.3.7: The first three lower probability density functions of electrons in the Gamma band using two different scale factors in the WarpByScalar filter in ParaView. The scale factors of the objects of the same datafile are connected: changing the value in one component will automatically update the Scale Factor of the other warped plots.

this platform. [Figure 5.1.3.8](#) shows two examples of plot controls that can be used to active and deactivate plots (the eyeball) and to adjust the camera for a plot with perfect alignment of the plot to the z-axis.

Clicking on the eyeballs in ParaView, each component can be activated or hidden individually. In [Figure 5.1.3.10](#) the eight lowest probability functions for the electrons in the Gamma band are displayed.

Hint

Use semi-transparent surfaces for displaying superposition of two or more states, as presented in [Figure 5.1.3.11](#)

When to use the single plot method: This method is ideal for the case you want to export a single plot, using the standard parameters and/or simple filters of the script. In this example we need only one click for requesting the WarpByScalar filter and another to launch the tool. When not changing any plot setting, no other clicks are necessary.

Plot with overlay

Displaying overlaid images represents a powerful feature in *nextnanomat*. Our script for integration is capable to setup all ressources to reproduce overlay of warped-3D plots also in ParaView. We will demonstrate how to do it plotting the file `\Quantum\probabilities_shift_quantum_region_Gamma.vtr` overlayed with the *bandedges.vtr*.

In nextnanomat

We will repeat the same procedure as before following the next steps:

- select a .vtr file of the simulation within *nextnanomat*
- click on the icon **Export** and **Open in specific Format** and select **Call Python Postprocessing scripts**

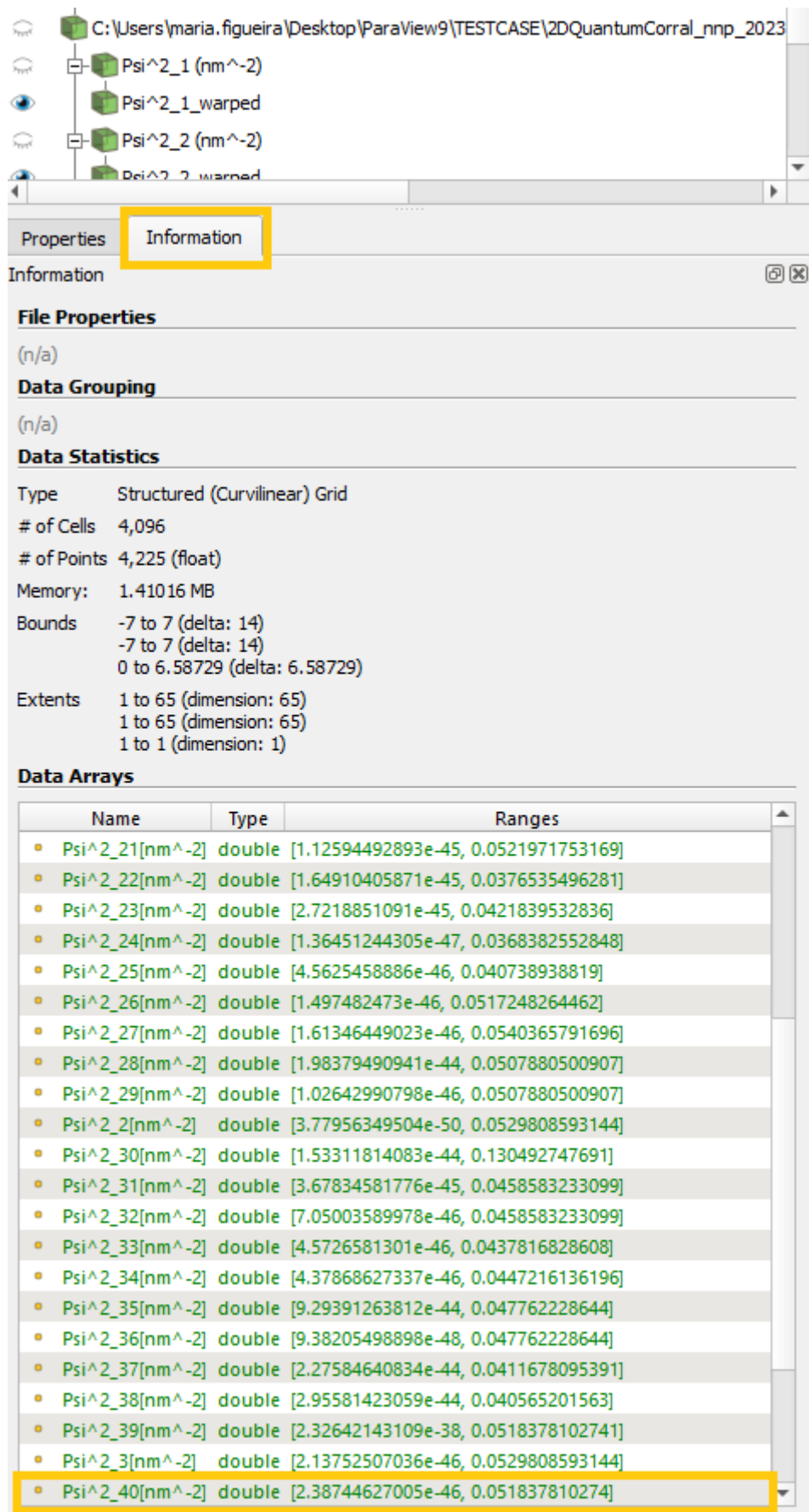


Figure 5.1.3.8: Information tab in ParaView showing the all sets within the datafile. For avoiding data explosion, only a reduced number were actually processed setting the variable `MAXIMUM_NUMBER_DATASETS` to 8 in the Python scripts.

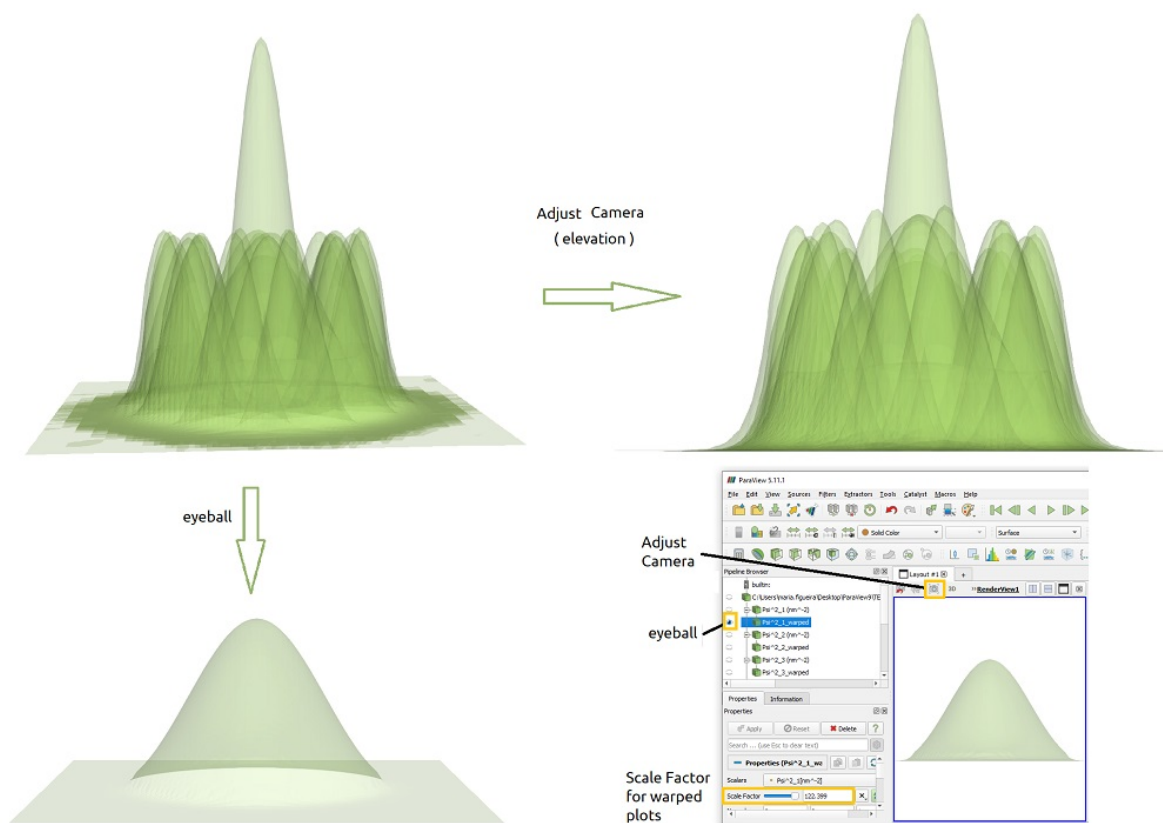


Figure 5.1.3.9: Plot controls in ParaView for selection of the component to be displayed (eyeball) and adjust of the camera. Moving the slider of Scale Factor the height of the peaks can be modified.

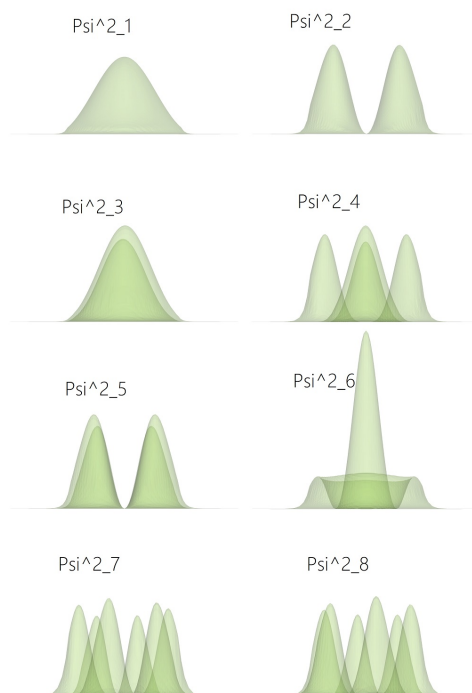


Figure 5.1.3.10: Probability density function of the first lower states of the electrons in the Gamma band.

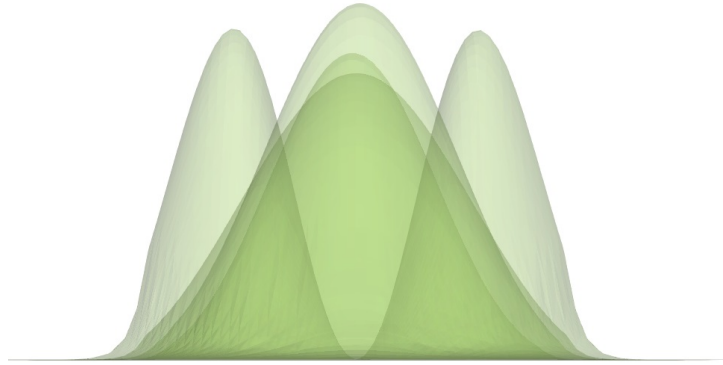


Figure 5.1.3.11: Use of semi-transparent surfaces is advantageous for comparing two functions, as in this example.

- select the script *nm_ParaView_integration.py*
- select the file: *\Quantum\probabilities_shift_quantum_region_Gamma.vtr*

In Integration GUI

This time we will make the following selections:

- **MAXIMUM_NUMBER_DATASETS** and the number 5 (The box shall be checked in order to capture this number)
- **WARP**
- **OVERLAY**
- Color scheme: **Palette**

Once the option **OVERLAY** is checked, a list of .vtr files is displayed. Choose, for this example, the file *\bias_00000\bandedges.vtr*. In this current implementation only one file shall be selected as overlay. Press the button **Launch ParaView**. Figure 5.1.3.12 shows the final configuration.

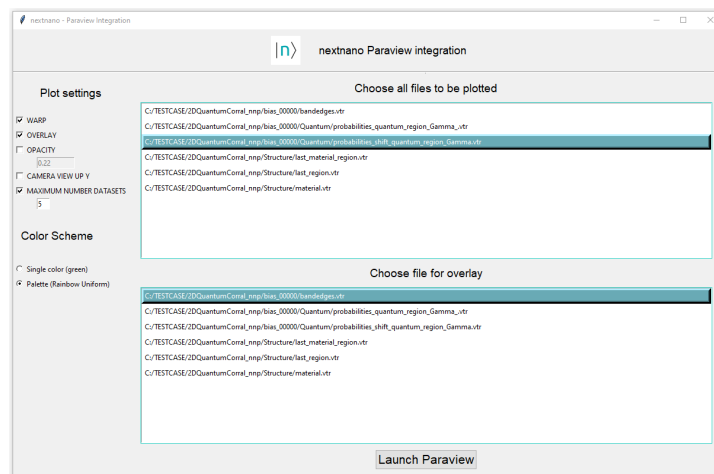


Figure 5.1.3.12: Settings in Graphical user interface for integration of *nextnanomat* and ParaView for the case of overlaid plots.

In ParaView

After the message **END OF THE POST-PROCESSING SCRIPT** is displayed in ParaView, we will observe that two group of files are presented: the one corresponding to the overlay (the band edges) and the other the main plot (shifted probabilities). Now only 5 components are shown for each file.

Once again the 2D plots were warped (option `WARP`). Now they are presented using the Rainbow Uniform palette and they are not longer with the Z-axis aligned vertically. With a simple click in ParaView, they can be realigned vertically as we can see in Figure 5.1.3.13.

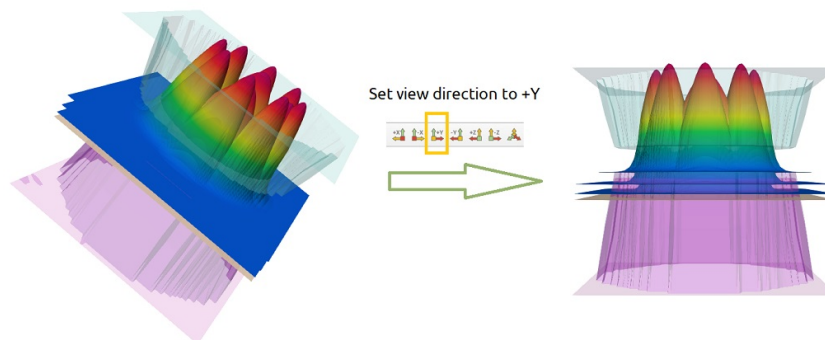


Figure 5.1.3.13: Plot of the first five lowest probability density functions. Clicking on the +Y button in ParaView they can be realigned vertically

The warped plots of the overlay files are always shown at the top of the pipeline as a semi-transparent surface using a different palette. The only exception opaque surface for the overlay corresponds to the electron and hole Fermi levels (see Figure 5.1.3.14).

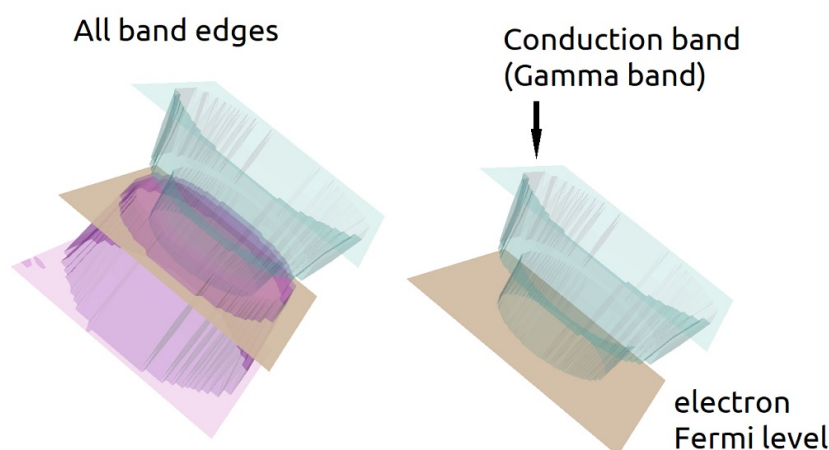


Figure 5.1.3.14: Overlay files are always plotted using a different palette and using semi-transparent surfaces. The Fermi levels are plotted as opaque surface.

Now let us discuss in more detail the second group of plots concerning the “shifted” wave functions (actually probability densities). Looking at the tab `Information` under the `Pipeline Browser` we will observe that actually we have in total 40 columns about the eigenvalues and 40 columns about the shifted wave functions. Nevertheless, in the pipeline we observe that only the 5 shifted wave functions are plotted. In another words, the eigenvalues are not considered as dataset for this specific kind of file.

Changing Scale Factor of warped files of shifted probability density

The file related with the shifted probability densities requires an special care, because each component correspond to the combination of two different informations: the probability density (in unit of nm^{-2} in the case of 2D simulations) and the shift of this function by the corresponding eigenvalue (in unit of eV), as shown in Figure 5.1.3.15. When overlayed with any file whose unit is energy, these components shall be plotted in the same scale as the overlay plot, independent of the scalar factor in the warp transformation of part corresponding to the probability density.

Similar to the case of a single plot, all components of this datafile present the `Scale Factor` for the warped file interconnected. Then, changing the `Scale Factor` of one component of the datafile (`probabilities_shift`), will

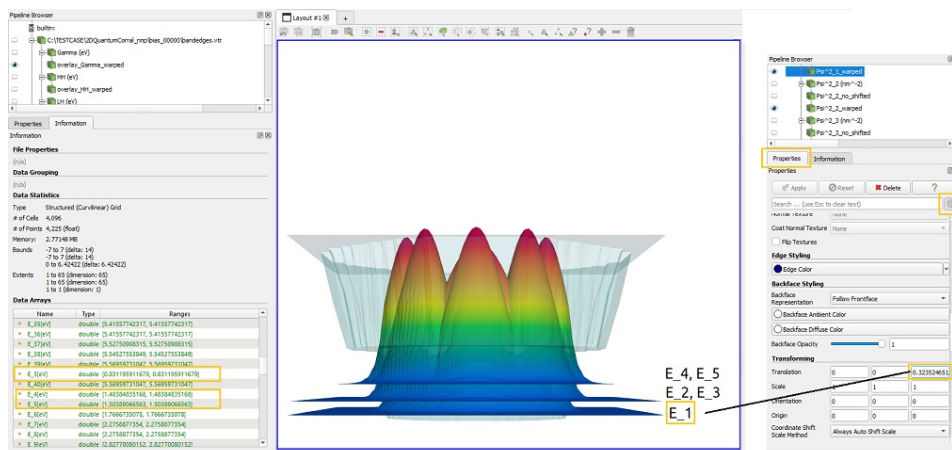


Figure 5.1.3.15: Probability density functions shifted corresponds to the sum of the probability function and its corresponding eigenvalue. For representing correctly when overlayed with the band edges, the warped version shall translate the basis of this plot by the corresponding eigenvalue.

change the height of each function, but will not shift the eigenvalue energy (the base of the plot), as shown in the Figure 5.1.3.16

As demonstration, click on the component `Psi^2_1_warped`, and change the Scale Factor. As we mentioned, you will observe that the bases does not change.

Changing Scale Factor of warped files of the band edges overlayed with shifted probability density

Nevertheless, changing the Scale Factor of the overlay file (the band edges in this case), whose by default is 1.0, does not affect the plot of the shifted probability densities. This is expected because components of different datafiles are not interconnected. In this case, it is necessary to change the energy scale used in the plot of the shifted probabilities. This is done multiplying all the values of translation in the warped file of the shifted probabilities by the new factor of the overlay Scale Factor.

As example, let us change the Scale Factor of the `overlay_bandedge_gamma_warped` by 0.1. Then for the correct overlay with the shifted probabilities we will require to multiply the translation value by 0.1, for `Psi^2_1_warped`, `Psi^2_2_warped`, and so on. This is illustrated in Figure 5.1.3.17.

There are a couple of another situations where a similar procedure is necessary and not all of the possible combinations are implemented in our script. ParaView provide several other resources to make the interconnection among the objects of the Pipeline Browser. Use the current script as an example how to implement this kind of associations, and feel free to explore another possibilities when creating your own script.

Hint

The warped plots uses a general setting of the camera that it is not universal, and may not be ideal for all plots. This can be easily fixed clicking on the Reset Camera button within ParaView or interacting with the plot in this platform.

Warning

As discussed above, files with mixed units will require special attention when overlayed with another files. Be aware to adjust the components properly.

When to use the overlayed plot method: This method is ideal for the case you want to superpose information of two files with different information. In the case of plotting `probabilities_shift` files overlayed with `bandedges` the

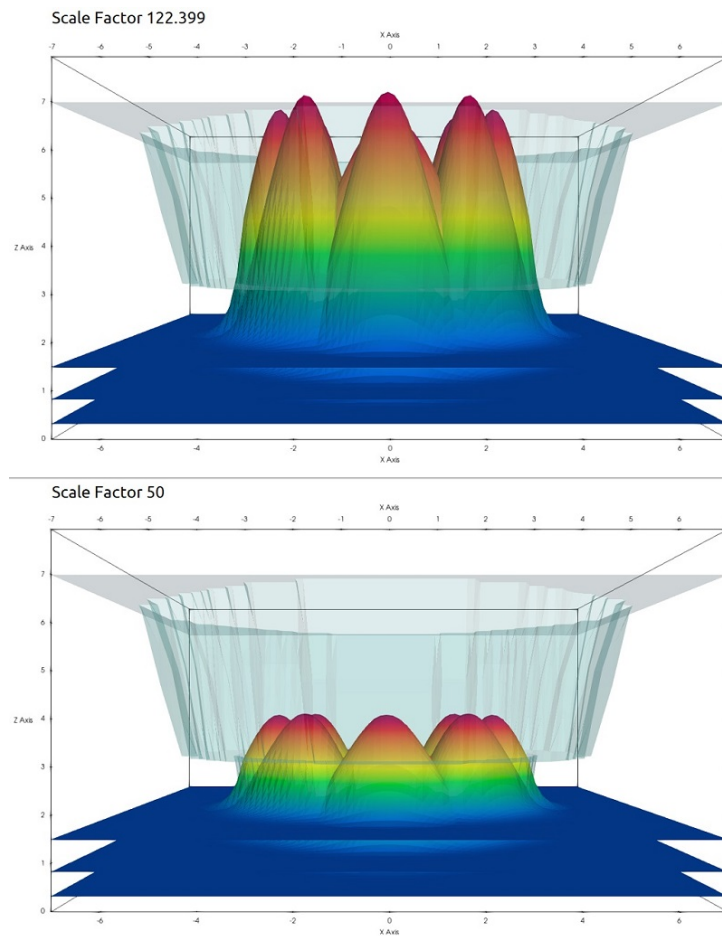


Figure 5.1.3.16: Probability density functions shifted by their corresponding eigenvalues. Changing the Scale Function of the warped plots of one component will shift the height of the plots, but not the base of them.

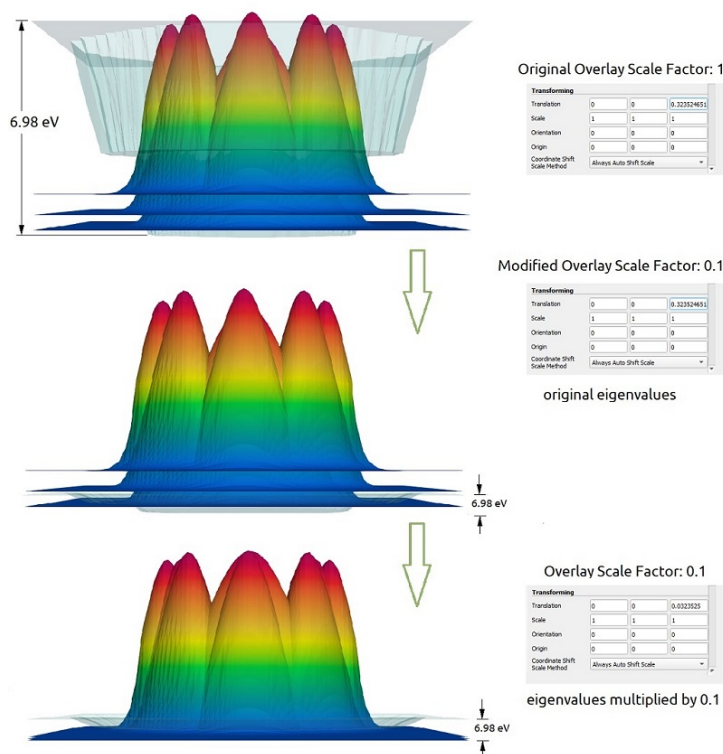


Figure 5.1.3.17: Probability density functions shifted by their corresponding eigenvalues. Changing the Scale Function of the warped plots of one component will shift the height of the plots of the other components of the same datafile, but not the base of them.

script align the two set of data in the same reference frame.

5.2 Gnuplot export

Gnuplot is a free graphing utility, which allows scientists to visualize data interactively. If installed it can be used to export 1D, 2D & 3D nextnano results.

5.2.1 How to export 1D plot files with Gnuplot

- *Moving plot-files to another device*

Either the currently selected file can be exported (by usage of the context menu) or the contents of the overlay list can be combined (1D) and exported (by usage of the output menu button). An example of such a combined file can be seen in Figure 5.2.1.1.

Moving plot-files to another device

1D-plots are linked to the original .dat file(s). So if you want to move your plot to another device, you can either save your plot as .pdf/.svg/.png file directly in gnuplot (recommended), or if you want to move the original .plt-file you also have to move all necessary .dat files (the paths can be adjusted when opening the .plt file with a text editor).



Warning

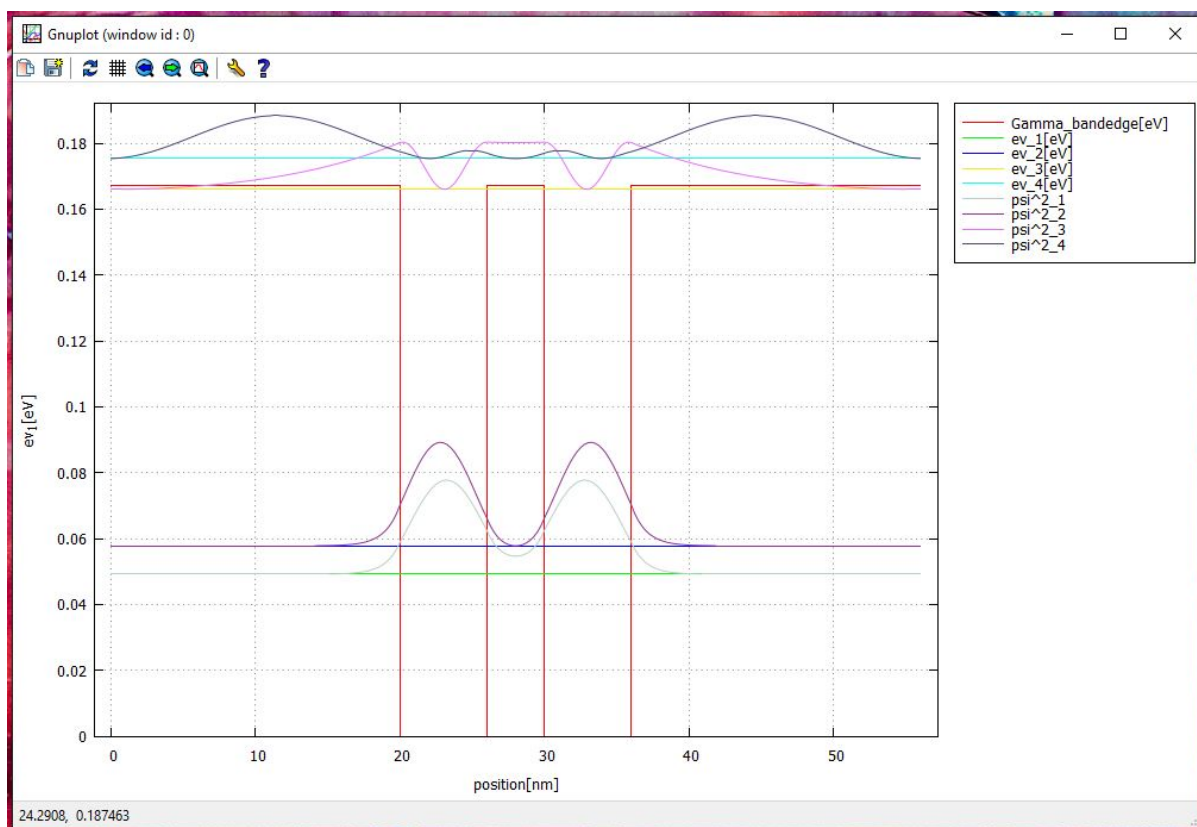


Figure 5.2.1.1: One-dimensional Gnuplot export of band edge plus probability densities.

If you move the 1D plot file without data files or without adjusting the paths, it will be broken. (Will not open when file is double-clicked.)

5.2.2 How to export 2D or 3D plot files with Gnuplot

In 2D you have some additional options for your gnuplot file compared to the 1D export, displayed in [Figure 5.2.2.1](#). The plot can be displayed as a color map, analog to the implementation of nextnano, or as a surface plot, which is a pseudo three-dimensional plot ([Figure 5.2.2.2](#)). Title and labels are optional and if they are not specified they will be taken directly from the file (if provided). If `remember settings` is checked, the next time the panel will be pre-filled with these settings. If you always use the same settings and do not want this dialog to be displayed each time, you can choose the `Create Gnuplot file - last used (*.plt)` option in the context menu.

Gnuplots are interactive, see [Figure 5.2.2.3](#), which makes them suitable to create animations e.g. for presentations. Furthermore they can be saved as vector graphics. To understand the dependency between plot files and their raw data files, please refer to the next section.

5.2.3 How to optimize the looks of a Gnuplot graph

- *Optimize your Gnuplot Graph manually*
- *Generate high quality graphs*

The plot style for all Gnuplot exports can be customized in [Gnuplot export](#). This is convenient to achieve uniform style of the exported graphs e.g. for a presentation. A style-sheet can be used which can be shared within the whole work-group.

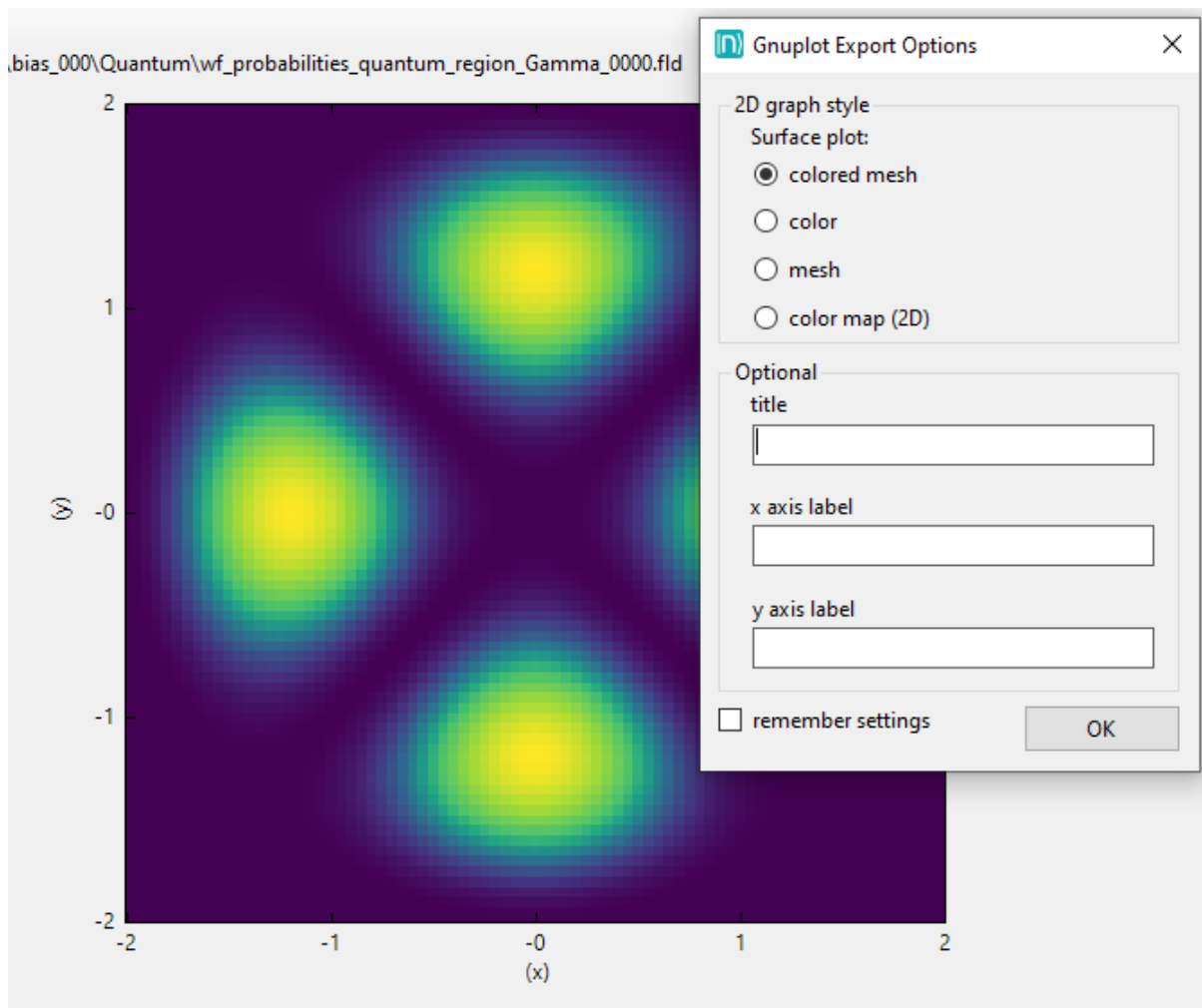


Figure 5.2.2.1: Additional options for more-dimensional Gnuplot export.

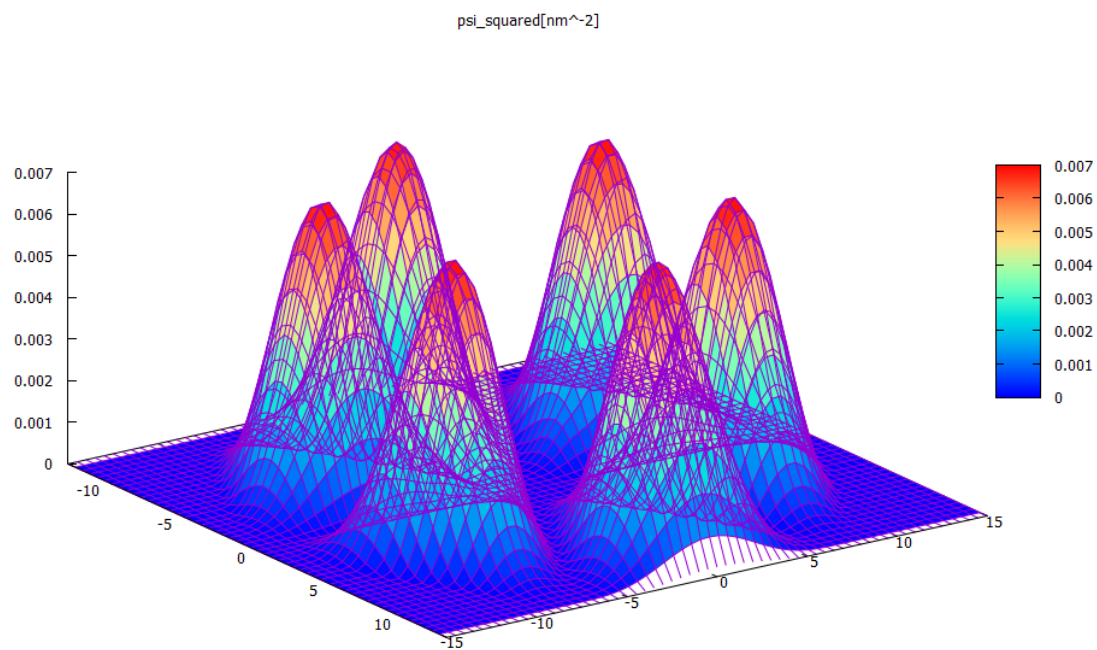


Figure 5.2.2.2: Example of Gnuplot surface plot option colored mesh (Probability density of the 10th wave function in a hexagonal structure.)

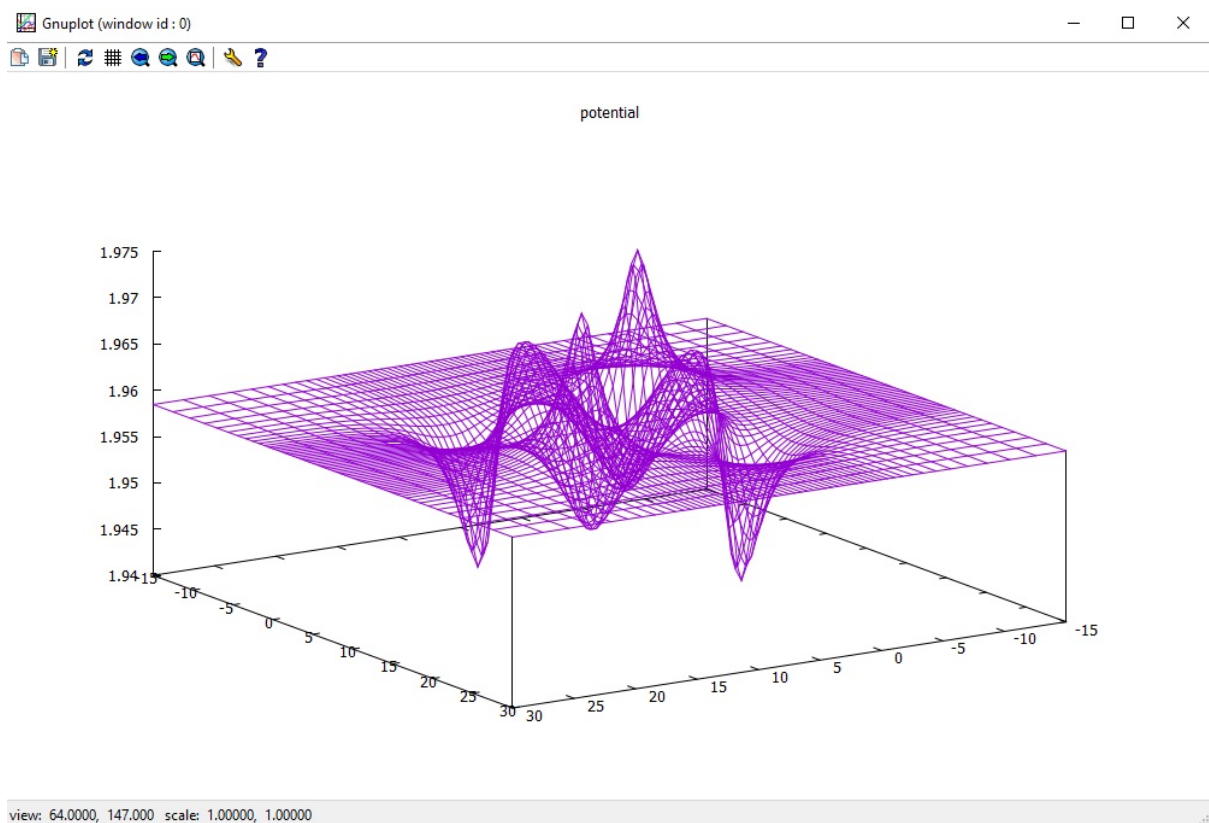


Figure 5.2.2.3: Interactive surface plot.

Optimize your Gnuplot Graph manually

Alternatively or additionally to the style-sheet you can customize each Gnuplot file on its own. Just open the file with a text editor of your choice and change, add or remove commands.

Collection of some useful gnuplot commands.

Semi-log plot

```
set logscale x
set logscale y
```

Change the line thickness (lw 4)

```
plot 'D:\bandedges.dat' linetype rgb "#FF0000" pt 5 ...
plot 'D:\bandedges.dat' linetype rgb "#FF0000" pt 5 lw 4 ...
```

Change font size of the x axis (20)

```
set xlabel "position (nm)" font "sans - serif"
set xlabel "position (nm)" font "sans - serif,20"
```

Use subscript and superscript (enhanced vs. noenhanced)

```
plot 'D:\density_hole.dat' using 1:2 title "p (10^{18} cm^{-3})"
↪ "enhanced ..."
plot 'D:\density_hole.dat' using 1:2 title "p (10^18 cm^-3)" noenhanced ↪
↪ ...
```

Change range [x_{min} , x_{max}] and [y_{min} , y_{max}] of the graph

```
set xrange [-0.3:100.3]
set xrange [0:100]
set yrange [-1.5:2.5]
set yrange [-1.0:0.5]
```

Set/Remove grid

```
set grid
unset grid
```

Change thickness of the border (lw)

```
set border lw 2
set border lw 3
```

Set/Remove legend

```
set key on
set key off
```

Set/Remove box around legend (box)

```
set key on ... box
set key on ... nobox
```

Increase font size in legend

```
set key on ... font "sans - serif,14"
set key on ... font "sans - serif,18"
```

Specify the location of legend

```
set key left top inside ...
set key right bottom outside ...
```

Remove line from legend (notitle)

```
plot 'D:\bandedges.dat' using 1:2 title "E_c " ...
plot 'D:\bandedges.dat' using 1:2 notitle "E_c " ...
```

Add Greek letter to line in legend (" Γ " enhanced)

```
using 1:2 title "{/Symbol G} [eV]" enhanced
```

produces Γ .

e, l, m, q produce ϵ , λ , μ , and θ , respectively.

Add a label to the point (x, y) in the plot

```
set label "label" at 0.5,1.5
```

Add an arrow

```
set arrow from 1.5,0.3 to 4,2
```

Graph Title

```
set title "title" font "sans - serif,18"
```

Generate high quality graphs

1. In gnuplot window: Click on **Export plot to file**
2. Save as SVG files (.svg)
3. Open the saved .svg file with [Inkscape](#)
4. File $\Rightarrow$ Export .PNG Image... $\Rightarrow$ Select Drawing $\Rightarrow$ Export

5.3 Export via python scripts

5.3.1 How to call custom python scripts directly from nextnanomat

Custom written python scripts can be called directly from the Output tab. The parent folder of the python script, Path to the currently displayed output file as well as the path to the simulation folder are transferred as system arguments, see [Figure 5.3.1.1](#). (If you need other specific information to be exported, just mail your request to the nextnano support team or use the widget on this website.) By using those system arguments in your python script, individual post-processing of specific output files is reduced to a one-click effort.

Related: [Exporting 2D Outputs with Python Scripts](#)

5.4 Other

In [Data export](#) custom paths to installed applications (e.g. ParaView, notepad++) can be set, to allow convenient export of output files for the purpose of visualization or post-processing.

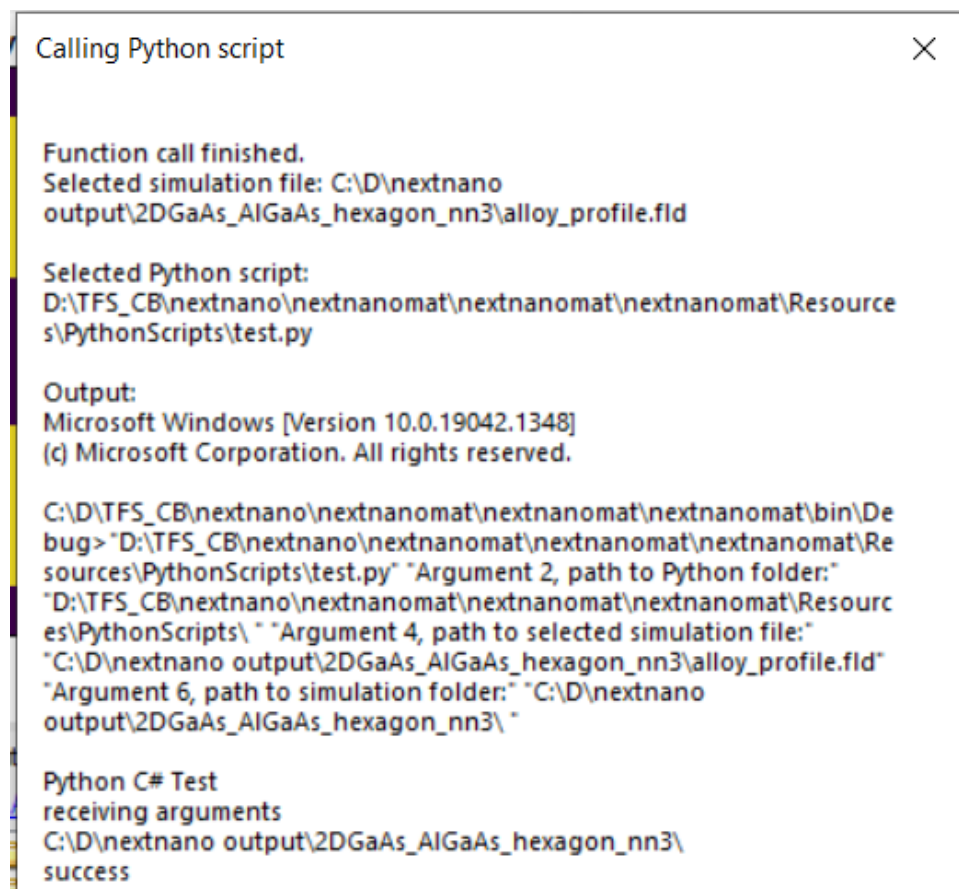


Figure 5.3.1.1: Example of transferred system arguments when calling a python script.

HTCONDOR

You can use HTCondor to run the [nextnano software](#) on your local computer infrastructure (“on-premise”). Essentially, *nextnanomat* submits the job either locally or on the “HTCondor” cluster. In both cases, the results of the calculations are located on your local computer.

This feature is only supported with our new license system.

6.1 HTCondor on nextnanomat

The following shows a screenshot from *nextnanomat*. 6 computers are connected to the HTCondor pool called **e25nn**. 120 slots are configured, 44 are currently available. Computers 2, 3, 4 and 6 are selected to accept jobs. Computers 2 and 6 are currently not available as they are in use.

6.2 Recommended Installation Process

Download HTCondor installer from [HTCondor](#).

1. On the webpage, click on Download and go to Current Stable Release of UW Madison (as of September 24 2020, HTCondor 8.8.10).
2. We recommend the file for Windows in Native Packages. The filenames look similar to this one:
 - `condor-8.8.10-513586-Windows-x64.msi` (Version 8.8.10)
3. Select the file, agree to the license agreement and download the .msi file. When you download it, you can optionally enter your name, email address and institution and subscribe to the HTCondor newsletter.

Install HTCondor.

1. Start installer
2. Click Next and then accept License Agreement
3. Then there are two options. There will be one special computer that manages all HTCondor jobs (Central Manager), and other computers as submit/execute nodes. If there is no Central Manager yet, we have to create a new pool.
 1. If you **are** on the Central Manager, choose **Create a new HTCondor Pool** and fill in the name of the Pool, e.g. `nextnanoHTCondorPool`. This is a unique name for your pool of machines.
 2. If you **are not** the Central Manager, choose **Join an existing HTCondor Pool** and fill in the host-name of the central manager, e.g. `computername` where `nextnanoHTCondorPool` has been created.
4. Tick **Submit jobs to HTCondorPool** and choose **Always run jobs and never suspend them**. (Alternative: If you do not want other people to run jobs on your machine at all, select **Do not run jobs on this machine** or if you do not want other people to run jobs on your machine while you are working, select **When keyboard has been idle for 15 minutes..** You can of course modify these settings later.)
5. Fill in your domain name (Example: Your Windows domain, e.g. `yourcompanyname.com` (without `www`).) All PCs of your network should get the same domain name, this does not necessarily have to be your Windows domain.

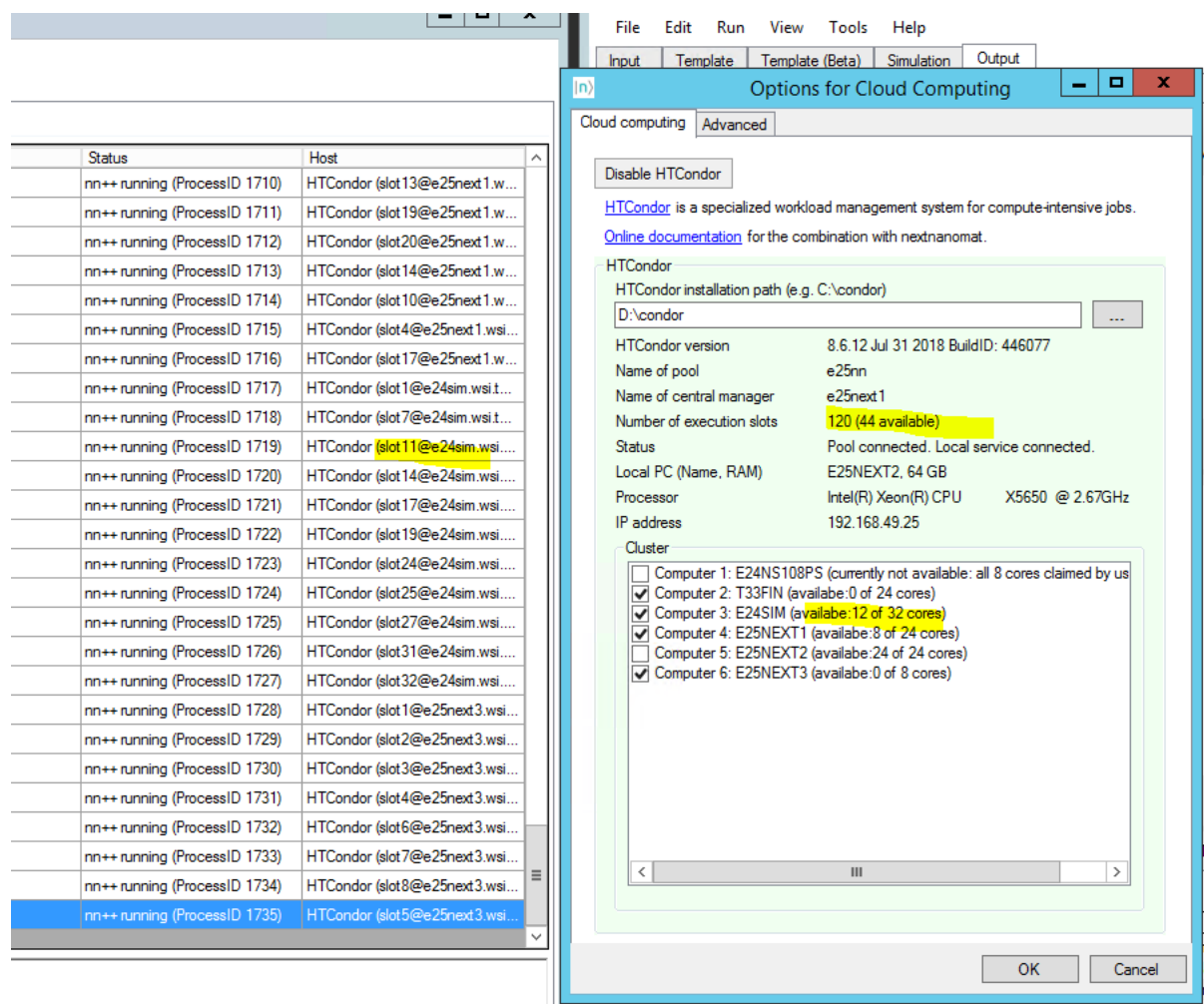


Figure 6.1.1: Screenshot taken from nextnanomat with integrated HTCondor feature.

6. Hostname of SMTP Server and email address of administrator (not needed currently, leave it blank)
7. Path to Java Virtual Machine (not needed currently, leave it blank)
8. Host with Read access: *
9. Host with Write access: \$(CONDOR_HOST), \$(IP_ADDRESS), *.yourdomainname.com, 192.168.178.*; (**Replace** *.cs.wisc.edu with your domain name and **add** your local IP subnet e.g. 192.168.178.*). On Windows you can find your IP subnet by opening the Command Prompt cmd.exe and typing in ipconfig.
10. Host with Administrator access * (or \$(IP_ADDRESS))
11. Enable VM Universe No
12. Choose an installation directory and press next (e.g. C:\condor\). The directory Program Files is problematic due to write permissions, so we do not recommend using it.
13. Press Install and type in the Administrator password of your PC. (You need Administrator rights.)
14. Once installed, please restart the computer. Then your new pool or pool member should be up and running.

A few more setups

1. To be able to submit jobs from *nextnanomat* to HTCondor, you have to store your credentials once. Open a command shell and type the following command: `condor_store_cred add`
 - Enter your password and you are ready to submit your first HTCondor job.
 - If this does not work, try to enter `condor_store_cred add -debug` for more output information on the error.
2. Please make sure that *nextnanomat* has successfully found the HTCondor pool. In *nextnanomat* go to Tools -> Options -> Cloud computing. If everything is correctly set up, you will find the “HTCondor” section highlighted with green color, and the available computers show up in “Cluster”. If this is not the case, maybe you have not installed HTCondor on the computer where you are running *nextnanomat*. Please also check that the HTCondor installation path is correctly set within *nextnanomat*, e.g. the default path C:\condor might not be the one where you installed HTCondor.

6.2.1 Summary of settings (Example)

```

Hostname (for HTCondor pool): computername.yourcompanyname.com
Policy: "Always run jobs"
Accounting domain: yourcompanyname.com
Read access: *
Write access: $(CONDOR_HOST), $(IP_ADDRESS), *.yourcompanyname.com, 192.168.178.*
Administrator: $(IP_ADDRESS)

```

6.2.2 Config file

You can find your HTCondor config settings in the file C:\condor\condor_config. Let us look at an example below.

- Your company is called Simpson.
- Your Windows domain is called `simpson.com`.
- Your HTCondor pool shall have the name `TheSimpsonsCondorPool`.
- The HTCondor host that manages the HTCondor jobs has the computer name `homer.simpson.com`.
- Your computer is called `lisa.simpson.com`.
- The computers in your network have the IP range `192.168.188.*`. (or `2001:db8:2042::*` in IPv6)

```

RELEASE_DIR = C:\condor
LOCAL_CONFIG_FILE = $(LOCAL_DIR)\condor_config.local
REQUIRE_LOCAL_CONFIG_FILE = FALSE
LOCAL_CONFIG_DIR = $(LOCAL_DIR)\config
use SECURITY : HOST_BASED
#CONDOR_HOST: $(FULL_HOSTNAME)           # on computer called homer
CONDOR_HOST: homer                       # on computer called lisa
COLLECTOR_NAME = TheSimpsonsCondorPool  # only on computer called homer
#UID_DOMAIN =                           # empty if you do not have a domain
UID_DOMAIN = simpson.com
SOFT_UID_DOMAIN=TRUE                     # entry is missing if you do not have a domain
FILESYSTEM_DOMAIN = simpson.com         # entry is missing if you do not have a domain
CONDOR_ADMIN =
SMTP_SERVER =
ALLOW_READ = *
ALLOW_WRITE = $(CONDOR_HOST), $(IP_ADDRESS), *.simpson.com, 192.168.188.*,
↪ 2001:db8:2042::*
ALLOW_ADMINISTRATOR = $(IP_ADDRESS)
use POLICY : ALWAYS_RUN_JOBS
#use POLICY : DESKTOP
WANT_VACATE = FALSE
WANT_SUSPEND = TRUE
#DAEMON_LIST = MASTER SCHEDD COLLECTOR NEGOTIATOR STARTD # on computer called homer
#DAEMON_LIST = MASTER SCHEDD STARTD KBDD                 # on computer called lisa if
↪ keyboard idle 15 minutes option was chosen
DAEMON_LIST = MASTER SCHEDD STARTD                       # on computer called lisa

```

6.2.3 Configuring a pool without a domain

It is also possible to set up a HTCondor pool with computers that are not on a domain by using the **NO_DNS** option in the compig file. This also provides an easy way to setup a mixed pool with Windows and Linux machines. Below are example configurations for central manager and submit/execute nodes for Windows and Linux, respectively. The user passwords on Windows machines need to be stored by `condor_store_cred` add command as above, on Linux machines no password is needed.

Central Manager on Windows:

```

RELEASE_DIR = C:\condor
LOCAL_DIR = $(RELEASE_DIR)
LOCAL_CONFIG_FILE = $(LOCAL_DIR)\condor_config.local
REQUIRE_LOCAL_CONFIG_FILE = FALSE

use SECURITY : HOST_BASED

CONDOR_HOST = $(FULL_HOSTNAME)
COLLECTOR_NAME = EXAMPLE_POOL

NO_DNS = True
DEFAULT_DOMAIN_NAME = MY_DOMAIN

ALLOW_CONFIG = $(IP_ADDRESS)
ALLOW_ADMINISTRATOR = $(CONDOR_HOST), $(IP_ADDRESS), condor_pool@$(UID_DOMAIN),
↪ EXAMPLE_POOL

ALLOW_READ = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_WRITE = $(IP_ADDRESS)/8, $(CONDOR_HOST)

```

(continues on next page)

(continued from previous page)

```

ALLOW_NEGOTIATOR = $(CONDOR_HOST), condor_pool@$(UID_DOMAIN)

ALLOW_ADVERTISE_MASTER = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_ADVERTISE_STARTD = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_ADVERTISE_SCHEDD = $(IP_ADDRESS)/8, $(CONDOR_HOST)

ALLOW_DAEMON = condor_pool@$(UID_DOMAIN), condor@$(UID_DOMAIN), $(IP_ADDRESS)
DAEMON_LIST = MASTER STARTD SCHEDD NEGOTIATOR COLLECTOR

use POLICY : ALWAYS_RUN_JOBS
WANT_VACATE = FALSE
WANT_SUSPEND = TRUE

```

Submit/Execute Node on Windows:

```

RELEASE_DIR = C:\condor
LOCAL_DIR = $(RELEASE_DIR)
LOCAL_CONFIG_FILE = $(LOCAL_DIR)\condor_config.local
REQUIRE_LOCAL_CONFIG_FILE = FALSE

use SECURITY : HOST_BASED

CONDOR_HOST = 192.168.188.xy #substitute with the ip address of the central manager
COLLECTOR_HOST = $(CONDOR_HOST)

NO_DNS = True
DEFAULT_DOMAIN_NAME = MY_DOMAIN

ALLOW_CONFIG = $(IP_ADDRESS), $(CONDOR_HOST)
ALLOW_ADMINISTRATOR = $(CONDOR_HOST), $(IP_ADDRESS), condor_pool@$(UID_DOMAIN),
↳EXAMPLE_POOL

ALLOW_READ = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_WRITE = $(IP_ADDRESS)/8, $(CONDOR_HOST)

ALLOW_ADVERTISE_MASTER = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_ADVERTISE_STARTD = $(IP_ADDRESS)/8, $(CONDOR_HOST)
ALLOW_ADVERTISE_SCHEDD = $(IP_ADDRESS)/8, $(CONDOR_HOST)

ALLOW_DAEMON = condor_pool@$(UID_DOMAIN), condor@$(UID_DOMAIN), $(IP_ADDRESS),
↳$(CONDOR_HOST)
DAEMON_LIST = MASTER STARTD SCHEDD

use POLICY : ALWAYS_RUN_JOBS
WANT_VACATE = FALSE
WANT_SUSPEND = TRUE

```

Central Manager on Linux:

```

RELEASE_DIR = /usr
LOCAL_DIR = /var
LOCAL_CONFIG_FILE = /etc/condor/condor_config.local
REQUIRE_LOCAL_CONFIG_FILE = false
LOCAL_CONFIG_DIR = /etc/condor/config.d

## Pathnames
RUN      = $(LOCAL_DIR)/run/condor

```

(continues on next page)

(continued from previous page)

```

LOG      = $(LOCAL_DIR)/log/condor
LOCK     = $(LOCAL_DIR)/lock/condor
SPOOL    = $(LOCAL_DIR)/spool/condor
EXECUTE  = $(LOCAL_DIR)/lib/condor/execute
BIN      = $(RELEASE_DIR)/bin
LIB      = $(RELEASE_DIR)/lib/condor
INCLUDE  = $(RELEASE_DIR)/include/condor
SBIN     = $(RELEASE_DIR)/sbin
LIBEXEC  = $(RELEASE_DIR)/lib/condor/libexec
SHARE    = $(RELEASE_DIR)/share/condor
GANGLIA_LIB64_PATH = /lib,/usr/lib,/usr/local/lib
PROCD_ADDRESS = $(RUN)/procd_pipe

use SECURITY : HOST_BASED

CONDOR_HOST = $(FULL_HOSTNAME)
COLLECTOR_NAME = EXAMPLE_POOL

NO_DNS = True
DEFAULT_DOMAIN_NAME = MY_DOMAIN

ALLOW_CONFIG = $(IP_ADDRESS)
ALLOW_ADMINISTRATOR = $(IP_ADDRESS), $(CONDOR_HOST), condor_pool@$(UID_DOMAIN), ↵
↵EXAMPLE_POOL

ALLOW_WRITE = $(CONDOR_HOST), $(IP_ADDRESS)/8
ALLOW_READ = $(CONDOR_HOST), $(IP_ADDRESS)/8

ALLOW_NEGOTIATOR = $(CONDOR_HOST), condor_pool@$(UID_DOMAIN)

ALLOW_ADVERTISE_MASTER = $(IP_ADDRESS)/8
ALLOW_ADVERTISE_STARTD = $(IP_ADDRESS)/8
ALLOW_ADVERTISE_SCHEDD = $(IP_ADDRESS)/8

ALLOW_DAEMON = condor_pool@$(UID_DOMAIN), condor@$(UID_DOMAIN), $(IP_ADDRESS)
DAEMON_LIST = MASTER, STARTD, SCHEDD, NEGOTIATOR, COLLECTOR

# Do not phone home
CONDOR_DEVELOPERS = NONE
CONDOR_DEVELOPERS_COLLECTOR = NONE

SSH_TO_JOB_SSHD_CONFIG_TEMPLATE = /etc/condor/condor_ssh_to_job_sshd_config_template

```

Submit/Execute Node on Linux:

```

RELEASE_DIR = /usr
LOCAL_DIR = /var
LOCAL_CONFIG_FILE = /etc/condor/condor_config.local
REQUIRE_LOCAL_CONFIG_FILE = false
LOCAL_CONFIG_DIR = /etc/condor/config.d

## Pathnames
RUN      = $(LOCAL_DIR)/run/condor
LOG      = $(LOCAL_DIR)/log/condor
LOCK     = $(LOCAL_DIR)/lock/condor
SPOOL    = $(LOCAL_DIR)/spool/condor

```

(continues on next page)

(continued from previous page)

```

EXECUTE = $(LOCAL_DIR)/lib/condor/execute
BIN      = $(RELEASE_DIR)/bin
LIB      = $(RELEASE_DIR)/lib/condor
INCLUDE  = $(RELEASE_DIR)/include/condor
SBIN     = $(RELEASE_DIR)/sbin
LIBEXEC  = $(RELEASE_DIR)/lib/condor/libexec
SHARE    = $(RELEASE_DIR)/share/condor
GANGLIA_LIB64_PATH = /lib,/usr/lib,/usr/local/lib
PROCD_ADDRESS = $(RUN)/procd_pipe

use SECURITY : HOST_BASED

CONDOR_HOST = 192.168.188.xy #substitute with the ip address of the central manager
COLLECTOR_HOST = $(CONDOR_HOST)

NO_DNS = True
DEFAULT_DOMAIN_NAME = MY_DOMAIN

ALLOW_CONFIG = $(IP_ADDRESS)
ALLOW_ADMINISTRATOR = $(IP_ADDRESS), $(CONDOR_HOST), condor_pool@$(UID_DOMAIN),
↳EXAMPLE_POOL

ALLOW_WRITE = $(CONDOR_HOST), $(IP_ADDRESS)/8
ALLOW_READ = $(CONDOR_HOST), $(IP_ADDRESS)/8

ALLOW_ADVERTISE_MASTER = $(IP_ADDRESS)/8
ALLOW_ADVERTISE_STARTD = $(IP_ADDRESS)/8
ALLOW_ADVERTISE_SCHEDD = $(IP_ADDRESS)/8

ALLOW_DAEMON = condor_pool@$(UID_DOMAIN), condor@$(UID_DOMAIN), $(IP_ADDRESS)
DAEMON_LIST = MASTER, STARTD, SCHEDD

# Do not phone home
CONDOR_DEVELOPERS = NONE
CONDOR_DEVELOPERS_COLLECTOR = NONE

SSH_TO_JOB_SSHD_CONFIG_TEMPLATE = /etc/condor/condor_ssh_to_job_sshd_config_template

```

6.3 Submitting jobs to HTCondor pool with nextnanomat

Submit job

1. Add a job to the Batch list in the **Run** tab.
2. Click on the **Run in HTCondor Cluster** button (button with triangle and network).

Show information on HTCondor cluster

1. Click on **Show Additional Info for Cluster Simulation**.
2. Press the **Refresh** button on the right.
3. The results of the `condor_status` command are shown, i.e. the number of compute slots are displayed.
4. You can select another HTCondor command such as `condor_q` to show the status of your submitted jobs, i.e. select `condor_q`, and then press the **Refresh** button.
 - You can type in any command in the line **System command:**, e.g. `dir`.
 - The button **Open Documentation** opens the online documentation (this website).

Results of HTCondor simulations

- Once your HTCondor jobs are finished, the results are automatically copied back to your simulation output folder <nextnano simulation output folder>\<name of input file>\.
- For debugging purposes regarding the HTCondor job, you can analyze the generated log file, <input file name>.log.

6.4 Useful HTCondor commands for the Command Prompt

- `condor_submit <filename>.sub` Submit a job to the pool.
- `condor_q` Shows current state of own jobs in the queue.
 - `condor_q -nobatch -global -allusers` Shows state of all jobs in the cluster. Of all users.
 - `condor_q -goodput -global -allusers` Shows state and occupied CPU of all jobs in the cluster.
 - `condor_q -allusers -global -analyze` Detailed information for every job in the cluster.
 - `condor_q -global -allusers -hold` Shows why jobs are in hold state.
- `condor_status` Shows state of all available resources.
- `condor_status -long` Shows state of all available resources and many other information.
- `condor_status -debug` Shows state of all available resources and some additional information, e.g. *WARNING: Saw slow DNS query, which may impact entire system: getaddrinfo(<Computername>) took 11.083566 seconds.*
- `condor_rm` Remove jobs from a queue:
 - `condor_rm -all` Removes all jobs from a queue.
 - `condor_rm <cluster>.<id>` Removes jobs on cluster <cluster> with id <id> (It seems <cluster> can be omitted, and id is the JOB_IDS number.)
- `condor_release -all` If any jobs are in state hold, use this command to restart them.
- `condor_restart` Restart all HTCondor daemons/services after changes in config file. It might also resolve problems when no computer is found in the network.
- `condor_version` Returns the version number of HTCondor
- `condor_store_cred query` Returns info about the credentials stored for HTCondor jobs
- `condor_history` Lists the recently submitted jobs. If for a specific job ID the status has the value ST=C, then this job has been completed (C) successfully.
- `condor_status -master:` returns Name, HTCondor Version, CPU and Memory of central manager
- Open Command Prompt `cmd.exe` as Administrator. Type in: `net start condor`. This has the same effect as restarting your computer, i.e. the networking service `condor` is started. This is useful if you have changed your local `condor_config` file.

6.5 HTCondor Pool - Managing Slots

- Each PC runs one `condor_startd` daemon. By default, the `condor_startd` will automatically divide the machine into slots, placing one core in each slot. E.g. a 6-core computer with hyperthreading has 12 logical processors. Alternatively, the number of cores (or logical processors) can be distributed to the slots as follows.

```
SLOT_TYPE_1 = cpus=4
SLOT_TYPE_2 = cpus=4
SLOT_TYPE_3 = cpus=2
SLOT_TYPE_4 = cpus=1
```

(continues on next page)

(continued from previous page)

```

SLOT_TYPE_5 = cpus=1
SLOT_TYPE_1_PARTITIONABLE = TRUE
SLOT_TYPE_2_PARTITIONABLE = TRUE
SLOT_TYPE_3_PARTITIONABLE = TRUE
SLOT_TYPE_4_PARTITIONABLE = TRUE
SLOT_TYPE_5_PARTITIONABLE = TRUE
NUM_SLOTS_TYPE_1 = 1
NUM_SLOTS_TYPE_2 = 1
NUM_SLOTS_TYPE_3 = 1
NUM_SLOTS_TYPE_4 = 1
NUM_SLOTS_TYPE_5 = 1

```

- **PartitionableSlot:** For SMP (symmetric multiprocessing) machines, a boolean value identifying that this slot may be partitioned.
- **DynamicSlot:** For SMP machines that allow dynamic partitioning of a slot, this boolean value identifies that this dynamic slot may be partitioned.
- **SlotID:** For SMP machines, the integer that identifies the slot.
- A useful command might be: `condor_status -af Name TotalCpus DynamicSlot PartitionableSlot SlotID`. It returns the requested properties of each slot:
 - Name
 - TotalCpus
 - DynamicSlot
 - PartitionableSlot
 - SlotID

6.5.1 Dynamic slots

In our pool we have chosen dynamic partitioning which gives full flexibility. For instance, a quad-core CPU that is dynamically partitioned can accept

- 4 single-threaded jobs (`request_cpus = 1`)
- 2 jobs with 2 threads each (`request_cpus = 2`)
- 2 jobs of which one is single-threaded (`request_cpus = 1`) and the other uses 3 threads (`request_cpus = 3`)
- 1 job with 4 threads (`request_cpus = 4`).

```

#####
# Dynamic partitioning
# We use HTCondors dynamic partitioning mechanism.
# Each PC has one partitionable whole machine slot.
# (It seems that hyperthreading is not taken into account.)
#####
NUM_SLOTS = 1
NUM_SLOTS_TYPE_1 = 1
SLOT_TYPE_1 = 100%
SLOT_TYPE_1_PARTITIONABLE = true
SlotWeight = Cpus

```

6.6 Machine states

A machine is in any of the following 6 states. The most important one are Owner, Unclaimed, Claimed.

- **Backfill:** (not relevant for us)
- **Owner:** The machine is being used by the machine owner, and/or is not available to run HTCondor jobs. When the machine first starts up, it begins in this state.
- **Unclaimed:** The machine is available to run HTCondor jobs, but it is not currently doing so.
- **Matched:** The machine is available to run jobs, and it has been matched by the negotiator with a specific schedd. That schedd just has not yet claimed this machine. In this state, the machine is unavailable for further matches.
- **Claimed:** The machine has been claimed by a schedd.
- **Preempting:** The machine was claimed by a schedd, but is now preempting that claim for one of the following reasons.

- The owner of the machine came back
- Another user **with** higher priority has jobs waiting to run.
- Another request that this resource would rather serve was found.

6.7 Machine activities

Each machine state can have different activities. The machine state Claimed can have one out of these four activities.

- Idle:
- Busy:
- Suspended:
- Retiring:

6.8 Configuration options for the Central Manager computer

With this option in the `condor.config` file on the central manager, one can set a policy that the jobs are spread out over several machines rather than filling all slots of one computer before filling the slots of the other computers.

```
##-----nn: SPREAD JOBS BREADTH-FIRST OVER SERVERS
##-- Jobs are "spread out" as much as possible,
##   so that each machine is running the fewest number of jobs.
NEGOTIATOR_PRE_JOB_RANK = isUndefined(RemoteOwner) * (- SlotId)
```

6.9 FAQ

Q: I submitted a job to HTCondor, but nothing happens. The *nextnanomat* GUI says “transmitted”.

A: It could be that *nextnanomat* has not read in all required settings. You can try the command `condor_restart`. Please make sure that you entered your credentials using `condor_store_cred add -debug`. You should then restart *nextnanomat*.

Q: I submitted a job to HTCondor, but the Batch line of *nextnanomat* is stuck with preparing. What is wrong?

A1: Did you store your credentials after the installation of HTCondor? If not, enter `condor_store_cred add` into the command prompt to add your password, see above (Recommended Installation Process).

A2: Did you change your password recently? If yes you have to reenter your credentials for HTCondor. Enter `condor_store_cred add` into the command prompt to add your password, see above (Recommended Installation

Process). If this does not work, try to enter `condor_store_cred add -debug` for more output information on the error.

Q: I specified target machines in Tools - Options. Afterwards every submitted job to HTCCondor is stuck with transmitting. What is wrong?

A: The value for `UID_DOMAIN` within the `condor_config` file needs to be the same for every computer of your cluster. (You can easily test it in a command prompt with `condor_status -af uiddomain`) If it's not the same value, no matching computer will be found and the job will not be transmitted successfully.

6.10 Problems with HTCCondor

6.10.1 Error: communication error

If you receive the following error when you type in `condor_status`

```
C:\Users\"<your user name>">condor_status
Error: communication error
CEDAR:6001:Failed to connect to <123.456.789.123>
```

you can check whether the computer associated with this IP address is your HTCCondor computer using the following command.

```
nslookup 123.456.789.123
```

It is also a good idea to type in

```
nslookup
```

This will return the name of the Default Server that resolves DNS names. If it is not the expected computer, you can open a Command Prompt as **Administrator** and type in `ipconfig /flushdns` to flush the DNS Resolver Cache.

```
C:\Users\"<your user name>">ipconfig /flushdns
```

If the DNS address cannot be resolved correctly it could be related to a VPN connection that has configured a different default server for Domain Name to IP address mapping. E.g. if your Windows Domain is called `contoso.com` (which is only visible within your own network and your own HTCCondor pool) but your DNS is resolved to `www.contoso.com` (which might be outside your local HTCCondor pool).

6.10.2 Error: condor_store_cred add failed with Operation failed. Make sure your ALLOW_WRITE setting include this host.

Solution: Edit `condor_config` file and add host, i.e. local computer name (here: `nn-delta`).

```
ALLOW_WRITE = $(CONDOR_HOST), $(IP_ADDRESS)
==> ALLOW_WRITE = $(CONDOR_HOST), $(IP_ADDRESS), nn-delta
```

6.10.3 Error? Check the Log files

If you encounter any strange errors, you can find some hints in the history or Log files generated by HTCCondor. You can find them here:

C:\condor\spool

- history

C:\condor\log

- CollectorLog
- MasterLog

- MatchLog
- NegotiatorLog
- ProcLog
- SchedLog
- ShadowLog
- SharedPortLog
- StarterLog
- StartLog

More details can be found here: [Logging in HTCondor](#)

6.11 Known bugs

- HTCondor < 8.9.5 works with all nextnano executables
- HTCondor >= 8.9.5 works with nextnano executables newer than 2020-Jan

6.12 Run your custom executable on HTCondor with nextnanomat

You can even run your own executable with *nextnanomat* locally or on HTCondor! We tested the following programs:

- HelloWorld.exe
- Quantum ESPRESSO (pw.exe)
- ABINIT (abinit.exe)

6.12.1 Input file identifier

An input file identifier is a special string in the input file that signals to *nextnanomat* whether the input file is an input file for the *nextnano++*, *nextnano³*, *nextnano.NEGF* or *nextnano.MSB* tools, or for a custom executable.

6.12.2 Settings for Hello World (HW)

In *nextnanomat*, we need the following settings:

- Path to executable file: e.g. D:\HW\HelloWorld.exe
- Input file identifier: e.g. HelloWorld
- Working directory: Select ‘Simulation output folder’
- HTCondor: Output folder and files (transfer_output_files = ...): .

Open input file `input_file_for_HelloWorld.in` (or any other input file that contains the string HelloWorld) and run the simulation either locally or on HTCondor.

6.12.3 Settings for Quantum ESPRESSO (QE)

Our folder structure is

- D:\QE\inputfile\My_QE_inputfile.in (QE input file)
- D:\QE\input\pseudo\C.UPF (pseudopotential file for atom species ‘C’ as specified in input file)
- D:\QE\exe\pw.exe (QE executable file)
- D:\QE\exe*.dll (all dll files needed by pw.exe)
- D:\QE\working_directory\QE_nextnanomat_HTCondor.bat (batch file)

In *nextnanomat*, we need the following settings:

- Path to executable file: e.g. D:\QE\working_directory\QE_nextnanomat_HTCCondor.bat
- Path to folder with additional files: D:\QE\
- Input file identifier: e.g. &control
- Working directory: Select 'Simulation output folder'
- HTCCondor: Output folder and files (transfer_output_files = ...): .
- (Additional arguments passed to the executable: \$INPUTFILE)

The batch file (*.bat) contains the following content:

```
.\exe\pw.exe -in .\inputfile\My_QE_inputfile.in
```

This means that relative to the working directory, pw.exe is started, and the specified input file is read in. In this input file, the following quantities are specified:

- C.UPF: name of pseudopotential file
- ./input/pseudo/: path to pseudopotential file C.UPF

Open input file My_QE_inputfile.in and run the simulation either locally or on HTCCondor.

Things that could be improved:

- Write all files into output folder created by *nextnanomat*. In particular, the folder output/ should be moved.
- condor_exec.exe is deleted (better: do not copy it back)
- all *.dll files should be deleted (better: do not copy them back)
- Do not copy back *.exe and *.dll files (both HTCCondor and local)

6.12.4 Settings for ABINIT

Our folder structure is

- D:\abinit\inputfile\t30.in (ABINIT input file)
- D:\abinit\input* (input files needed by ABINIT)
- D:\abinit\exe\abinit.exe (ABINIT executable file)
- D:\abinit\exe*.dll (all dll files needed by abinit.exe)
- D:\abinit\working_directory\abinit_nextnanomat.bat (batch file)
- Path to executable file: e.g. D:\abinit\working_directory\abinit_nextnanomat.bat
- Path to folder with additional files: D:\abinit\
- Input file identifier: e.g. acell
- Working directory: Select 'Simulation output folder'
- HTCCondor: Output folder and files (transfer_output_files = ...): .
- Additional arguments passed to the executable: (empty)

The batch file (*.bat) contains the following content:

```
.\exe\abinit.exe < .\input\ab_nextnanomat_HTCCondor.files
```

This means that relative to the working directory, abinit.exe is started, and the specified input file is read in. In this input file, the following quantities are specified:

- .\inputfile\t30.in: name of input file
- .\input\14si.pspnc:

Open input file `t30.in` and run the simulation either locally or on HTCondor.

Notes

- `condor_exec.exe` is deleted (better: do not copy it back)
- all `*.dll` files should be deleted (better: do not copy them back)
- Do not copy back `*.exe` and `*.dll` files (both HTCondor and local)

Last update: 08/01/2025

RELEASE NOTES

7.1 5.6.4.6 (2025-12-17)

New features

- Overall: support dedicated input file extensions for different nextnano tools
- **Overall: show notification banner for connected licenses**
 - supports free, evaluation and professional licenses
 - reminder before expiry
 - email draft for quotation request
- Options: user setting to add nextnano tool name to output folder
- Editor tab: add context sensitive online help for nextnano.NEGF
- Template beta tab: store and load sweep data in XML file
- Template beta tab: user option to store sweep data in generated input file
- Simulation: support running nested folders for simulation

Updates & Improvements

- Runtime framework updated to .NET 8
- Options: allow increasing the number of threads during a running simulation
- Editor tab: prioritize syntax keyword file next to nextnano tool executable
- Main menu: update help menu functions

Bugfixes

- Output tab: update timestamp of simulation folder without switching views
- Output tab: DPI aware scaling of slices control borders
- Output tab: default space allocation for overlay and 3D slider, DPI aware
- Output tab: fix sorting error for simulation output folder with numeric folder names
- Simulation: fix HTCondor sim timer counter
- Simulation: fix display of log file visible in wrong tabs
- Simulation: fix incomplete log file in case log is saved automatically in fixed time interval
- Postprocessing: fix wrong pop-up message after error
- Template beta: include check for max. output path length
- Overall: handle error if input file is removed or deleted outside of nextnanomat

7.2 5.2.13.0 (2025-06-20)

Updates & Improvements

- Output tab: support space character in .fld file names

Bugfixes

- Output tab - 2D & 3D: overflow error in slices view in case of constant data
 - Output tab: value cut off at $\sim E28$
 - Output tab: graph control filewatcher buffer overflow
 - Output tab: axis tick format for logarithmic scales
-

7.3 5.2.9.0 (2025-05-30)

New features

- DPI-aware graphical user interface supporting all screen resolutions
- Simulation tab: query to check simulation process for termination
- Editor tab: replace tabs with spaces

Updates & Improvements

- Runtime framework changed to .NET 6
- **New layout for options control, introducing new settings**
 - New settings for performance options
 - Merged cluster settings to default options control within Advanced group
 - Start where you left - load last used tab page

Bugfixes

- On high resolution screens Windows App Size is not re-set to 100% after running a simulation
-

7.4 4.3.2.15 (2024-03-28)

New features

- Added option to kill whole process tree instead of process when stopping a simulation
- Added support of new nextnanoLicenseActivator executable (former ClientSideActivator and ClientSideServerActivation are no longer needed)
- Editor tab: support of “!Data” syntax for *nextnano++*
- Editor tab: feature ‘Create new input file by template’ is implemented for *nextnano++* input files
- Output tab: option to display $\text{Log}(\text{abs}(y))$

Updates & Improvements

- Improved layout for license activation message feed
- Input file conversion for *nextnano*³ -> *nextnano++* is now a standalone executable, can be called by command line
- Output tab: small viusalization changes, e.g. fonts

- Change of default performance settings
- Input tab: more accurate naming for context menu functions
- Improved folder structure for installation and portable packages
- Improve selection of matching database for different nextnano packages

Bugfixes

- Compatibility fix for new license check
- Cluster computing feature for *nextnano.NEGF* simulations
- HTCondor, change default OS target for cluster computing
- Update existing jobs in Batch List after license changes

Remaining known Bugs

- In installed versions running a simulation reverts Windows App Scaling back to 100%
-

7.5 4.3.2.8 (2023-08-07)

New features

- Added specific licenses for nextnano³ and evaluation
- Added support for free licenses
- Distinguish license types, display type during simulations and in settings
- Added nextnano documentation (pdf) to installed package
- Editor tab: added html keywordtree for nextnano++ to help with input file syntax
- Editor tab: convert xml input files to negf input files (for nextnano.MSB and nextnano.NEGF)
- Editor tab: added html keyword trees for all products to help with input file syntax
- Settings: verify licenses directly within settings
- Settings: added preview possibility for Editor and Output settings
- Settings: added support for two nextnano.NEGF tool versions (C# & C++)
- Output tab: added “local” and “global” full-size differentiation

Updates & Improvements

- Editor tab: updating links to online documentation for context help menu
- Editor tab: converted input file (nextnano³ to nextnano++) is opened in new tab
- Simulation tab: update queued simulations after change in settings
- Settings: functionality to disable syntax highlighting
- Settings: restructured Options Form
- Settings: added customizable tolerance for comparison of constant values (expert features)
- Output tab: autosize for pointer labels
- Output tab: changed numeric format for pointer labels
- Output tab: button position of full-size button
- Output tab: performance improvement in 1D output visualization during curve(s) selection
- Output tab: separate, re-order and re-name check/select buttons
- Output tab: improved spacing between data visualization and diagram axis

- Output tab: advanced python and ParaView export routines

Bugfixes

- Event handling, custom export function
- License activation prompt, appearing at every program start
- Editor tab: syntax highlighting for conditional statements
- Editor tab: input file conversion: added error handling for files located in program files
- Editor tab: Warning message for deprecated syntax “#IF to !WHEN”
- Simulation tab: running a cloud nextnano.NEGF simulation does no longer change the input file extension
- Simulation tab: command line arguments for starting nextnano.NEGF simulations
- Simulation tab: fix performance issues for simulating a batch of jobs
- Settings: display of Pool names for HTCondor
- Settings: opening cloud options form
- Output tab: color scale initialization
- Output tab: adjust reference of message box to new menu location of export feature
- Output tab: visualization of txt licenses
- Support free licenses to be located in program files

Remaining known Bugs

- In installed versions running a simulation reverts Windows App Scaling back to 100%
-

7.6 4.3.1.8 (2022-08-05)

New features

- standard selection of color maps for 2D and 3D
- feature to customize color maps
- couple of features related to color maps (e.g. invert scale, fix middle color etc. -> refer to online documentation)
- syntax support for new input file format *.negf
- customizable syntax highlighting
- support for new NEGF licenses
- feature to run batch list automatically when new files are added
- added accessibility evaluation and statement for the [nextnano software](#)
- allow integration for tiberCAD input files and output
- output, show value difference for two selected curves

Updates & Improvements

- 1D Export to Gnuplot of overlay items includes currently displayed file
- add or remove items to overlay list by keyboard only
- improved syntax highlighting
- options to specify whether and how output should be overwritten

- set online help references to the new nextnano documentation

Bugfixes

- prevent crash at application start due to color-map initialization
- prevent application crash due to IPv6 IP addresses
- modified file star
- prevent duplicate opening of the same input file
- increase functionality and stability of Template(Beta) tab

Remaining known Bugs

- input file shows in simulation tab during CPU heavy simulations
-

7.7 4.2.8.13 (2021-07-05)

New features (New functionality added)

- enable to set an OS-Target for remote execution pool
- implement logging functionality to help customer support in case of errors
- enable closing of last input tab

Updates (Existing features improved)

- sort simulation output by date feature is now taking lastWriteTime instead of LastAccessTime
- reduce message boxes in error handling of missing files during a simulation

Bugfixes (Bugs which got fixed)

- prevent unicode characters in license path to fail license check
- handle output filewatcher errors in case of access restricted folders or deleted files

Known bugs (Bugs which are still present, but known)

- input file shows in simulation tab during CPU heavy simulations

7.8 4.2.8.7 (2021-02-05)

New features

- Output: New nextnano_temp folder for HTCondor simulations
- Input: Added optical feedback (*) for unsaved files
- Tools: New functionality to generate nextnanopy config file from within nextnanomat

Updates

- Error handling in case of inaccessible Licensing.dll (GUI is no longer closed)
- Functionality to copy sample files at first program start is removed (Can cause confusion due to syntax updates in newer program versions. -> Working examples are always found in corresponding installation folder.)
- Batch list: In case of non-working simulation, prompt same error message only once

Bugfixes

- HTCondor .log-file transfer
- Update notification: Fix version comparison (10<9)

Known bugs

- Licensing: local simulations don't start with licenses of old model (*.txt files) Input files
- New intersubband absorption input and tutorial files

7.9 4.2.8.6 (2020-12-09)

New features

- Specify Operating system via command line arguments when starting nextnanomat
- Waiting batch list, which automatically starts simulation whenever a new item is added (enable per Tools-Options)
- functionality to import template file from current input tab
- Expert feature to activate license per fingerprint-file (Enables license activation for other computer than own. E.g. colleague has troubles due to his OS or anti-virus software)

Updates

- Template tab: Added user warning in case input file might be overwritten
- HTCondor: Disable functionality if no Pool is selected

Bugfixes

- Set Output text view to read-only
- Culture conflicts due to multi-threading (This led to a buggy output visualization on PCs with many cores)
- Enable nextnano.NEGF to be run with HTCondor

Known bugs

- HTCondor: .log-file transfer (overwriting in case of simultaneously running simulations with same name)

Last update: 2025-10-16

8.1 Features and Functionalities

- *How do I produce 1D slices through the 2D plots in the GUI?*
- *Is there a way to produce a 1D (or 2D) plot of some result, for example the probability density with the conduction band edge superimposed? In other words, can the GUI show multiple plots at once?*
- *What is the difference between “List view” and “Tree view”?*
- *How to prevent fonts from changing back to Calibri in the input tab?*
- *How can I add additional debug information to the LOG file?*
- *Can I pass additional command line arguments to the executables of simulation tools?*

8.1.1 How do I produce 1D slices through the 2D plots in the GUI?

1. Visualize horizontal and vertical slices:

Menu View ==> Show Horizontal and Vertical Slices (or click on appropriate button)

2. Export horizontal or vertical slices:

Right-click on 2D graph, Export horizontal/vertical slice as data (.dat) or as image (.png).

8.1.2 Is there a way to produce a 1D (or 2D) plot of some result, for example the probability density with the conduction band edge superimposed? In other words, can the GUI show multiple plots at once?

Yes, this is possible. Multiple output files can be shown at once, see [Overlay feature documentation](#).

8.1.3 What is the difference between “List view” and “Tree view”?

In/For the Output tab of *nextnanomat*, you can switch between these options using: Tools ==> Options ==> View ==> Output folder browser

List view is the default and shows a list of the top level output folders. Tree view additionally offers the possibility of showing the subfolders within the output folder panel.

If you want to sort your output folders by date, List view is better suited.

8.1.4 How to prevent fonts from changing back to Calibri in the input tab?

Problem

As our text editor is based on **Rich Text Box** WinForms its behavior may be sensitive to the choice of your **Keyboard Language**. Certain input languages for your keyboard, e.g., **English (Germany)**, may cause the editor to write in **Calibri** font instead of your selected editor font, which you specified in **Tools/Options/Editor**.

Note

Calibri is a default font of Windows.

Your desired font is restored upon saving the input file which updates the syntax highlighting.

Solutions

If you have other language settings already installed, you can consider switching between them to see if one of them is supported. If so, then you can use this setting to write codes in the input file with *nextnanomat*. You can do it using a shortcut **Start + Space**.

If you are lacking supported language settings for your keyboard, then go to **Settings / Time & Language / Language** and **Add a language**.

Some known supported language settings for keyboards are:

- English (United States)
- English (United Kingdom)
- German (Germany)
- Polish

Known not supported language settings for keyboards are:

- English (Germany)
- Japanese

8.1.5 How can I add additional debug information to the LOG file?

Within the *nextnanomat* options *Expert/Debugging* some flags can be found which add additional information to the log file. Please check the following flags and re-run the simulation so the log file will include this information:

1. 'Include simulation time in log file'
2. 'Include memory usage in log file (units: MB)'
3. 'Include additional debug information in log file'

8.1.6 Can I pass additional command line arguments to the executables of simulation tools?

Yes, this is possible. Go to **Tools ==> Options ==> Expert settings ==> Command line**

For *nextnano³*, one could use for instance: `-database "D:\My folder\nextnano3\Syntax\my_database_nn3.in`

`-threads 4`

Last update: 25/10/2024

8.2 Error Messages

- *Error: File format is not valid*
- *Error: The specified executable is not a valid application for this OS platform.*
- *Licensing.dll cannot be found*
- *Error during license activation*
- *Unverified license*

8.2.1 Error: File format is not valid

A beta feature in a recent Windows 10 version (1803) causes the following error message when starting *nextnanomat*:

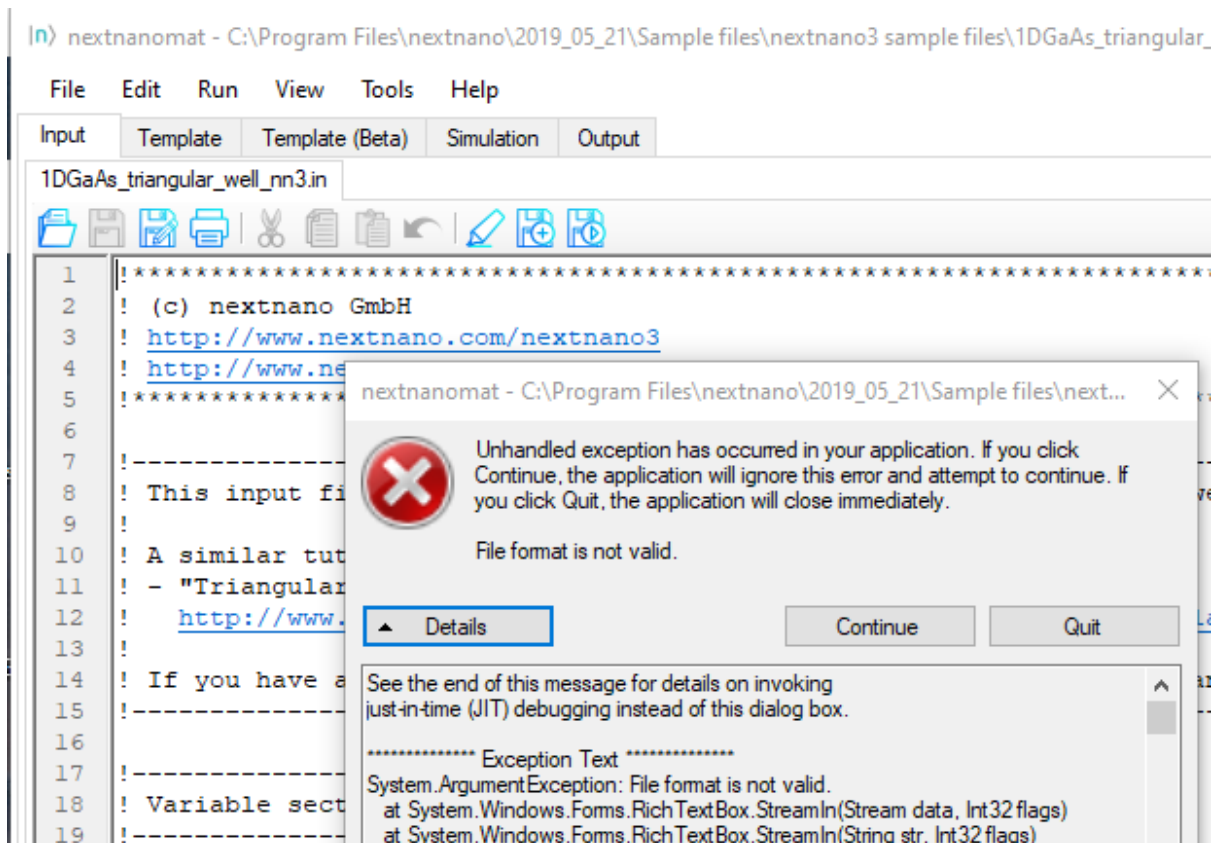


Figure 8.2.1.1: Input file format exception.

You can disable this feature in the **Region's settings** of Windows (screenshot see Figure 8.2.1.2).

1. Select Additional date, time, & regional setting
2. Then Change date, time, or number formats
3. Go to the Administrative Tab
4. Click Change system locale...
5. And finally you can un-select that Beta-Feature.

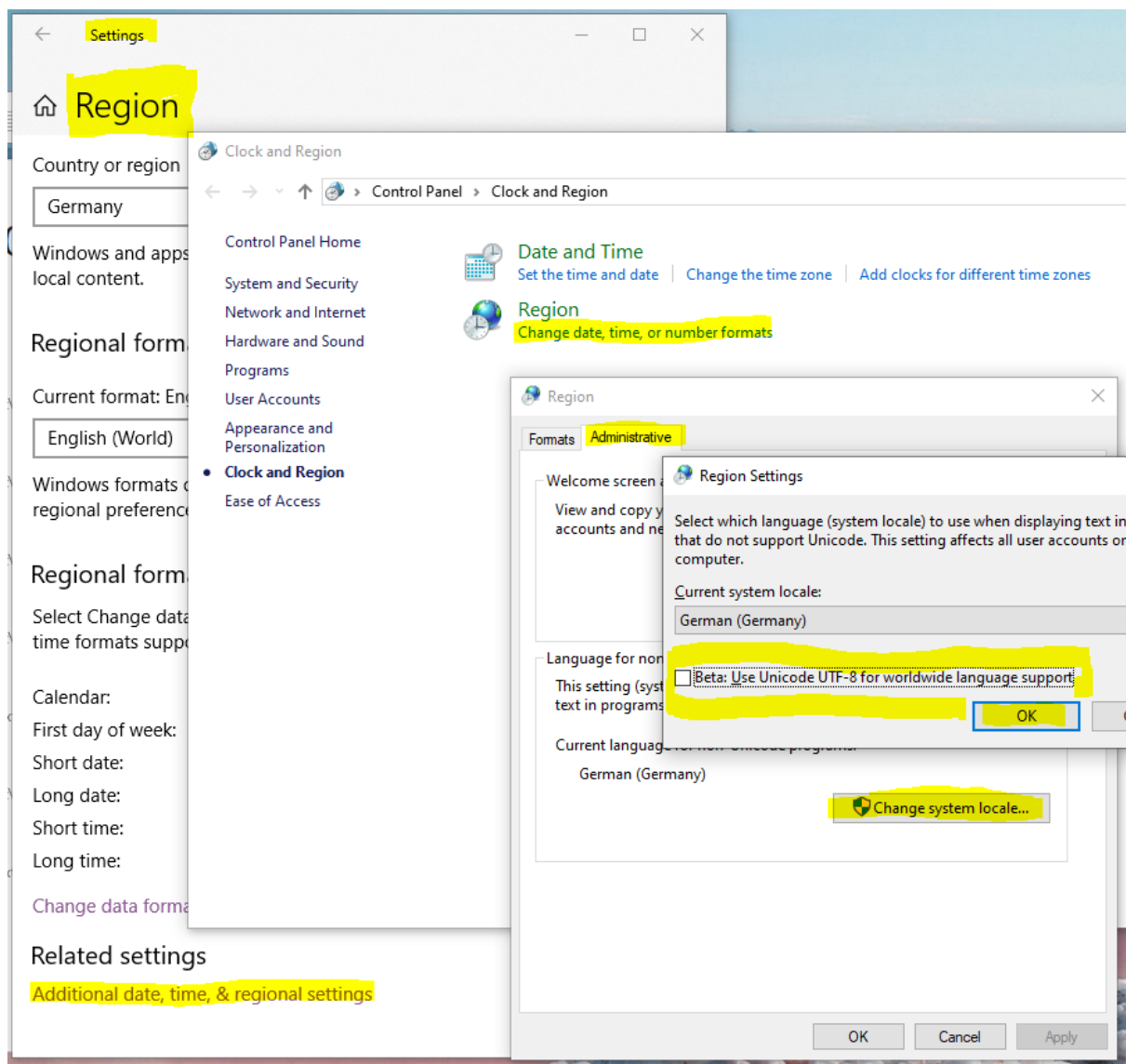


Figure 8.2.1.2: How to change Windows 10 region settings.

8.2.2 Error: The specified executable is not a valid application for this OS platform.

If you get this error, you are executing an executable that is not suited for this operating system. There are two possible reasons:

- You are executing a 64-bit version on a 32-bit operating system. Solution: Specify the location of the 32-bit *nextnano++* or *nextnano³* executables. Tools ==> Options... ==> Simulation ==> nextnano executable file ==> 32bit.exe
- You are executing the Intel or Microsoft 64-bit executable that has not been compiled for your rather old CPU. Our 64-bit executables use more advanced CPU instructions compared to our 32-bit executables. Solution: Use the 32-bit *nextnano++* executable that has been compiled with the Microsoft compiler. Tools ==> Options... ==> Simulation ==> nextnano++ executable file ==> nextnano++_Microsoft_32bit.exe

Sometimes the displayed error message does not contain any content apart from ERROR.

You can share information about your CPU with us using the menu Help ==> Generate System Snapshot for Support and then send us the generated file nextnano_SystemSnapshot.txt.

8.2.3 Licensing.dll cannot be found

This error prevents the license check within *nextnanomat*. It is not a critical error as local simulations can still be run without any problems, hence you can choose to ignore it. (Each product performs its own license check).

We recommend installing the [Microsoft Visual C++ Redistributable version](#). Choose the corresponding version matching your operation system architecture (most likely X64) from the section *Latest Microsoft Visual C++ Redistributable Version*.

This solution is assuming the .NET framework version is newer than 4.7. In case of uncertainty it can be looked up within the *Troubleshooting Snapshot text file* which is documented here: [Generate System Snapshot for Troubleshooting](#). If .NET framework is below 4.7 a more recent version should be installed.

8.2.4 Error during license activation

There are multiple possible error sources...

Connection error

It can help to retry a couple of minutes or an hour later.

We are currently using the port 8001 to establish a server connection. If this port is blocked by your IT department, a connection cannot be established. Either you can open the port or use the offline activation.

User cannot be found

Check for spelling errors in your license key. Also, make sure you are using the license key of this year and not your previous one. If it cannot be resolved, contact support.

No licenses have been saved

This means a connection to the server has been established and a valid user has been found. But still no license could be saved. It could be that the number of licenses exceeded, the xml reading process of the license failed or the directory cannot be accessed. Please contact support to figure out the source of this error.

If you are unable to resolve the error, please make a screenshot of the error message and contact nextnano support.

8.2.5 Unverified license

Problem

The error *Unverified license* in the *License Type* column of the simulation tab displayed by *nextnanomat* after running a simulation.

Solution

We recommend installing the [Microsoft Visual C++ Redistributable](#). Choose the corresponding version

matching your operation system architecture (most likely X64) from the section *Latest Microsoft Visual C++ Redistributable Version*.

Last update: 2025/07/09

8.3 Problems and Limitations

- *The Window of the graphical user interface gets (too) small*
 - *My simulations are still shown as running, although they are finished*
-

8.3.1 The Window of the graphical user interface gets (too) small

On a high resolution screen with a Windows app scaling greater than 100%, it seems the Window size of *nextnanomat* can be reset to 100% e.g. when starting a simulation. As a result, everything is quite small and unreadable.

To avoid this issue:

1. Navigate to 'nextnanomat.exe' within Windows file explorer
2. Right click and select application properties
3. Go to 'Compatibility' tab
4. Change/Override high DPI setting
5. Select 'System (Enhanced)'

8.3.2 My simulations are still shown as running, although they are finished

Sometimes the ending of a simulation process is not detected correctly. Please use this *Run menu* function to manually trigger the inquiry, *Check Simulations for Termination (beta)*

Last update: 03/04/2025

Last update: 25/10/2024

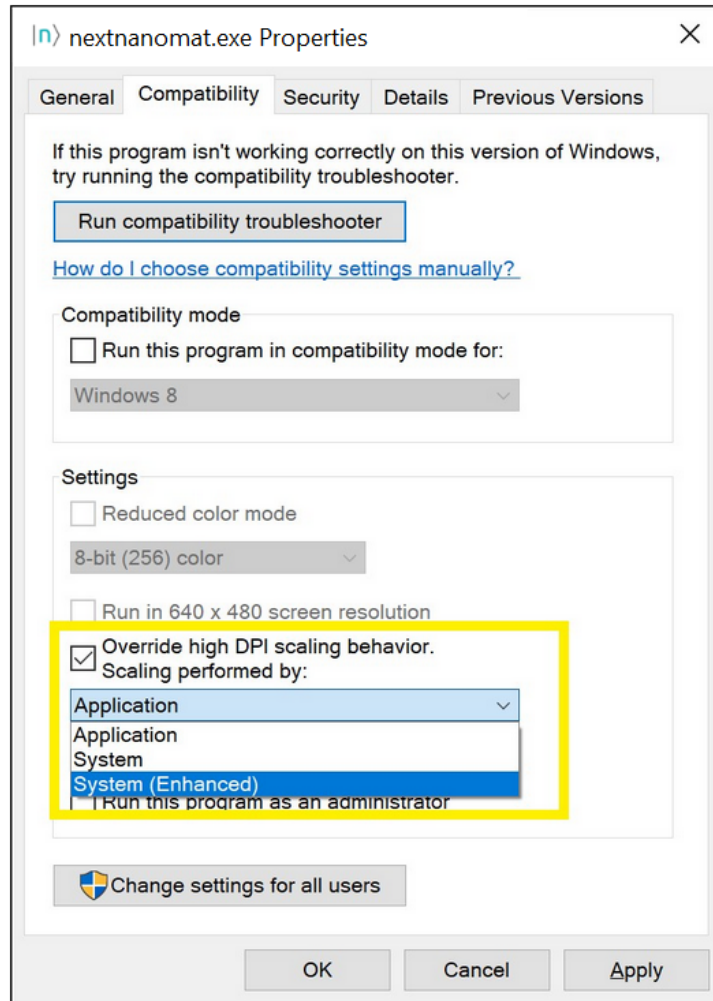


Figure 8.3.1.1: Screenshot of application properties dialog.

Note

From version number 5.0.0.0 onwards, the workflow manager and graphical user interface *nextnanomat* is DPI aware. If you experienced this issue, please upgrade your nextnano software package.