# Java implementation of two-dimensional input device editor for nextnano³

Alexander Hersonski

Fakultät für Informatik
Technische Universität München

# Contents

next**nano³** is a simulator for the calculation of the electronic properties of one, two and three-dimensional nanostructers in non-equilibrium.

### Contact person

Prof. Dr. Peter Vogl

**Walter Schottky Institut**
**TU München**
**Am Coulombwall 3**
**D-85748 Garching**
Tel. +49-89-289-12750
Fax +49-89-289-12737
vogl@wsi.tu-muenchen.de

Stefan Birner

**Walter Schottky Institut**
**TU München**
**Am Coulombwall 3**
**D-85748 Garching**
Tel. +49-89-289-12752
Fax +49-89-289-12737
stefan.birner@nextnano.de

### nextnano³ details

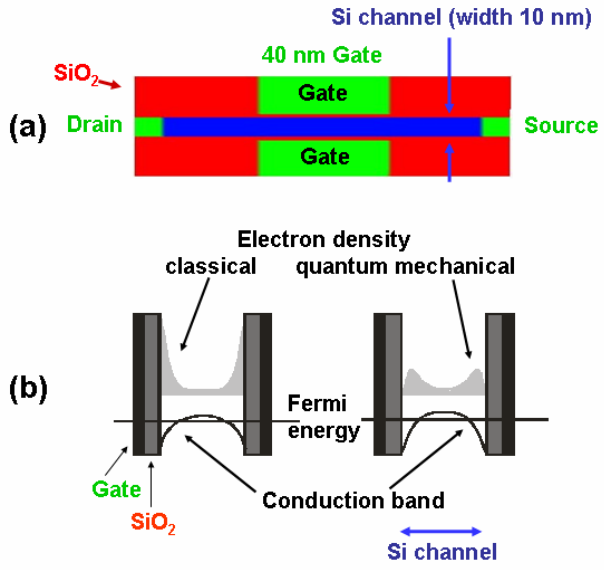
next**nano³** – a state-of-the-art simulation tool for 3D quantum nanodevices

next**nano³** is a simulator for calculating, in a consistent manner, the realistic electronic structure of three-dimensional heterostructure quantum devices under bias and its current density close to equilibrium. The electronic structure is calculated fully quantum mechanically, whereas the current is determined by employing a semiclassical concept of local Fermi levels that are calculated self-consistently.

This code allows one to solve the 8-band-k.p-Schrödinger-Poisson equation for arbitrarily shaped 3D heterostructure device geometries, and for any (III-V and Si/Ge) combination of materials and alloys (including ternaries and lattice matched quaternaries, as well as nitrides in the zincblende or wurtzite structure) oriented along any chosen crystallographic growth direction. The method includes band offsets of the minimal and higher band edges, absolute deformation potentials, total elastic strain energy that is minimized for the whole device, the long-range Hartree potential induced by charged impurity distributions, voltage induced charge redistribution, piezo- and pyroelectric charges, as well as surface charges, in a fully self-consistent manner. In addition, magnetic fields can be included. The charge density is calculated for a given applied voltage by assuming the carriers to be in a local equilibrium that is characterized by energy-band dependent local quasi-Fermi levels. These local quasi-Fermi levels are determined by global current conservation, where the current is assumed to be proportional to the density and to the gradient of the quasi-Fermi level (associated with each band) exactly as in the semiclassical limit.

In the calculation of the current, recombination and generation processes can be included. Furthermore, our method automatically includes tunneling via the globally

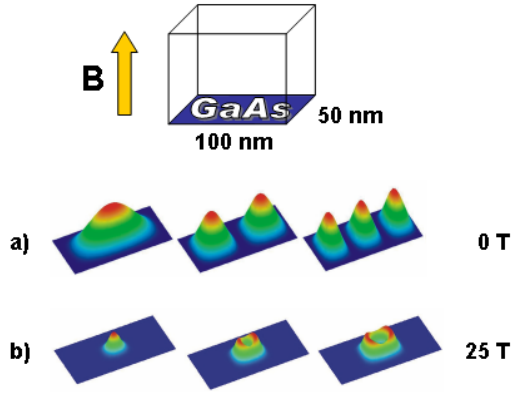
calculated electronic states, and yields optical transition energies and optical matrix elements.




*(a) Schematic plot of a Double Gate MOSFET*
*(b) Cut through the 10 nm Si channel: Comparison of classical and quantum mechanical electron density and conduction band edge profile across the Si channel at room temperature. The quantum mechanical simulation gives a smaller current than a classical drift-diffusion calculation (by ~30% for a gate voltage of 0.4 V and drain voltage of 0.2 V).*

Fig.1a: Schematic plot of a Double Gate MOSFET
Fig 1b: Cut through the 10 nm Si channel

For a given nanostructure, the computations start by globally minimizing the total elastic energy using a conjugate gradient method. This yields the local strain tensor which in turn determines the piezoelectric polarization charges, the deformation potentials and band offsets. Subsequently, the multi-band-Schrödinger, Poisson, and current continuity equations are solved iteratively. All equations are discretized according to the finite difference method invoking the box integration scheme. The irregular rectilinear mesh is kept fixed during the calculations.




*(a) Electron density associated with the three lowest eigenstates of a GaAs 2D electron gas confined by high rectangular potential barriers.*
*(b) Effect of a high magnetic field perpendicular to the GaAs plane on these states. All 3 states shown belong to the lowest Landau level.*

Fig.2a: Electron density by high rectangular potential barriers
Fig 2b: Effect of a high magnetic field

The main iteration scheme itself consists of two parts. In the fist part, the wave functions and potential are kept fixed and the quasi-Fermi levels are calculated self-consistently from the current continuity equations, employing a conjugate gradient method and a simple relaxation scheme.

In the second part, the quasi-Fermi levels are kept constant, and the density and the potential are calculated self-consistently from the Schrödinger and Poisson equation. The discrete 8-band Schrödinger equation represents a huge sparse matrix (typically of dimension $10^5$ for 3D structures) and is diagonalized using the Jacobi-Davidson method that yields the required inner eigenvalues and eigenfunctions close to the energy gap. To reduce the number of necessary diagonalizations, we employ an efficient predictor-corrector approach to calculate the potential from the nonlinear Poisson equation. In this approach, the wave functions are kept fixed within one iteration and the density is calculated perturbatively from the wave functions of the previous iteration. The nonlinear Poisson equation is solved using a modified Newton method, employing a conjugate gradient method and line minimizations. The code is written in Fortran 90 and consists of some 180.000 lines by now.

For 1D simulations a web based input file generator is available that guides the user through all steps necessary for creating input files. Extensive online documentation as well as several tutorial files are available on the next**nano³** website at http://www.wsi.tu-muenchen.de/next**nano³** (restricted by login and password). Executables and the source code (tested on Windows, Linux and Unix) are available for download. The material parameters in the database can be adjusted manually. Output files (band structure, densities, wave functions, strain, current, …) can be visualized using standard graphics tools like Origin, AVS, Gsharp or MATLAB. Examples that were treated so far include quantum dots, HEMTs and Double Gate MOSFETs.

### Background

Building up on an already existing 1D input file generator which permits the specification of simple heterostructures by means of Perl/HTML, a Java 2D interface was programmed which shows an improved functionality and allows the editing of two-dimensional geometries.
The interface permits to design a 2D semiconductor device geometry and to process it with the next**nano**³executable by means of a generated input file. Any change in the device geometry made by the Device Editor will instantaneously alter the generated input file.
x, y coordinates of the device regions are given in a next**nano³** readable format, together with further information which are necessary as a input parameters for the simulation program.

This application allows the creation of ASCII input files for next**nano³** by means of a convenient graphics editor. The graphical Device Editor files can be saved and reloaded to be modified at a later stage. The device editor uses modern technologies of programming, among other things XML, Java Swing and JavaMailAPI (see http://java.sun.com und http://java.sun.com/xml)

It is also possible to save the project schemes on hard disk to allow patterns to be read in at later times for adjusting purposes. In addition, a mail module is inserted in the application which allows sending input files via mail. To use this feature the user should know the name of the corresponding SMTP server, login and password for the authentification on this server.

The application was written in form of Java applets. The advantage is that the user can run the device editor independently of platform either as a Java applet on-line or as a Java application off-line. The tests were carried out on Windows 9x, 2000, XP and Linux.

### System requirements

- Supported operating systems: Linux, Windows ME/9x/2000/XP, Mac OS X
- 3 MB HDD space and 128 MB RAM
- Sun JDK/JRE 1.3 or newer.

### Getting started with nextnano³ device editor

A Java Runtime Environment must be installed. If you have not installed Java, you can download it here: http://java.sun.com/j2se/1.4/download.html

An installation program does not exist yet. To start the application proceed as follows:

1. The downloaded ZIP file must be unpacked into the desired directory.

   - additional libraries (**mail.jar, activation.jar, JSX2.0.9.5.jar**) for the running of the mail and Load/Save module of the applets must be copied into the following directory: **$JAVA-HOME/j2se.../lib/ext/**
2. Policy file ".java.policy" must be copied into your HOME directory.
   e.g. on Windows XP “C:\Documents and Settings\username”
   (only needed for the applet, not needed for 3b)

3.

a. For starting the applets a double click on index.html is enough.
b. Starting from the Command Prompt
   On Windows: “Start”→”Run…”→”cmd”
   **`java -jar ($YOUR_PATH)\nanoproject.jar`**
   or
   **`javaw –jar ($YOUR_PATH)\nanoproject.jar`**

   Here one must consider that PATH for to Java binary directory is set accordingly.
   For Windows i.e..**:** set PATH=C:\PathToJava\bin;%PATH%

After starting two windows appear: a browser window and an applet window . In the browser window all Softkeys are listed (3a). For 3b, only the applet window will open.

There is the possibility to open a new simulation area (scheme) or to load an existing pattern. In the popup-menu "Simulator" there is also the option "Delete policy file", if the user wants to disable read/write access of the applet to the hard disc.

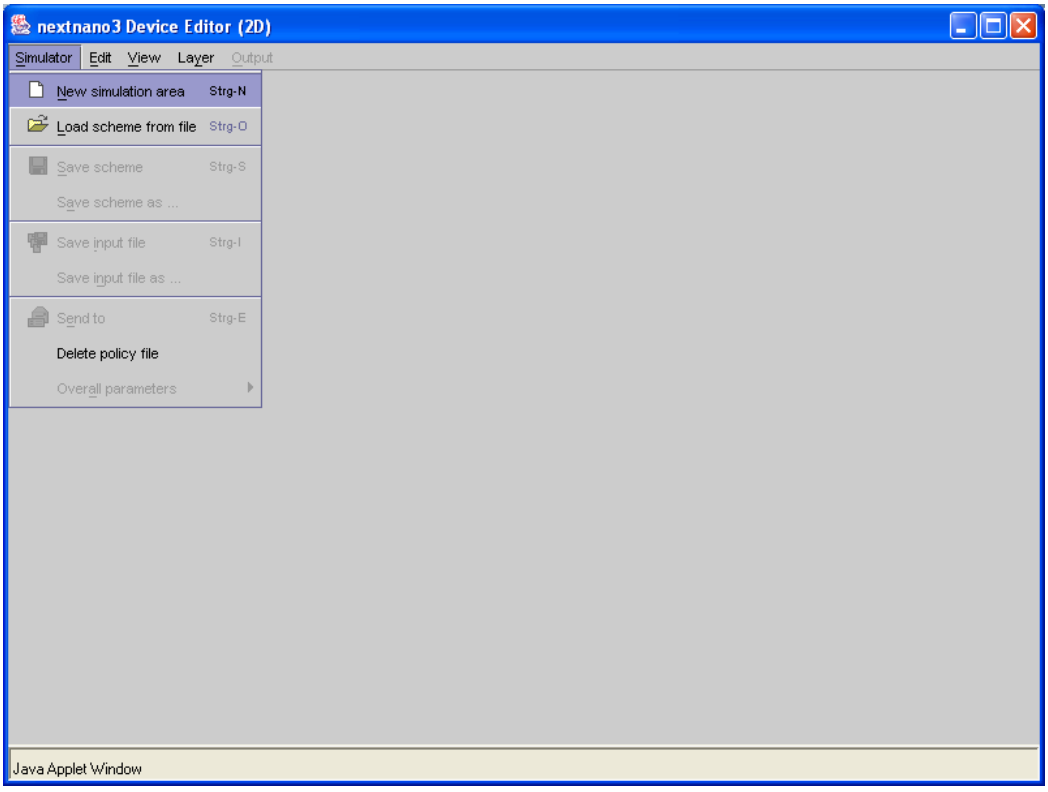



Fig .3: The device editor view → Start a new edit area

Choose the **Domain type**, i..e. 110 (x,y) if the device geometry should lie in a (x,y) coordinate system (recommended), (101) for (x,z) orientation or (011) for (y,z) orientation. **Maximum X and Y** coordinates determine the maximum extension in x and y direction respectively of the device one want to simulate (units: nm). **Division Size along axis** is the zoom factor on a suitable scale (can be changed during editing the device).

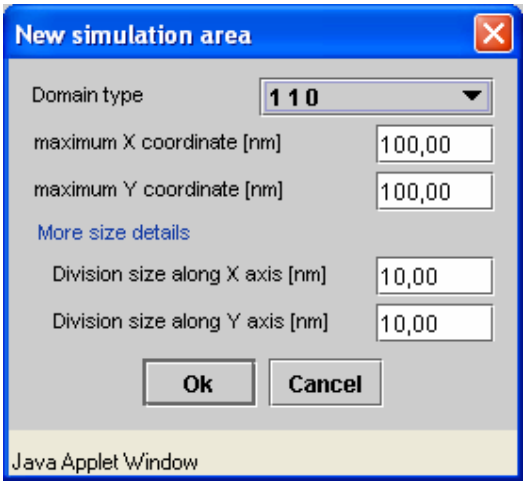



Fig. 4: The device editor view → New simulation area

In "Composite" view the graphical objects are shown and in "Input file" view the input file which is constructed at run time is shown as a text file in next**nano³** syntax including keywords and specifiers.

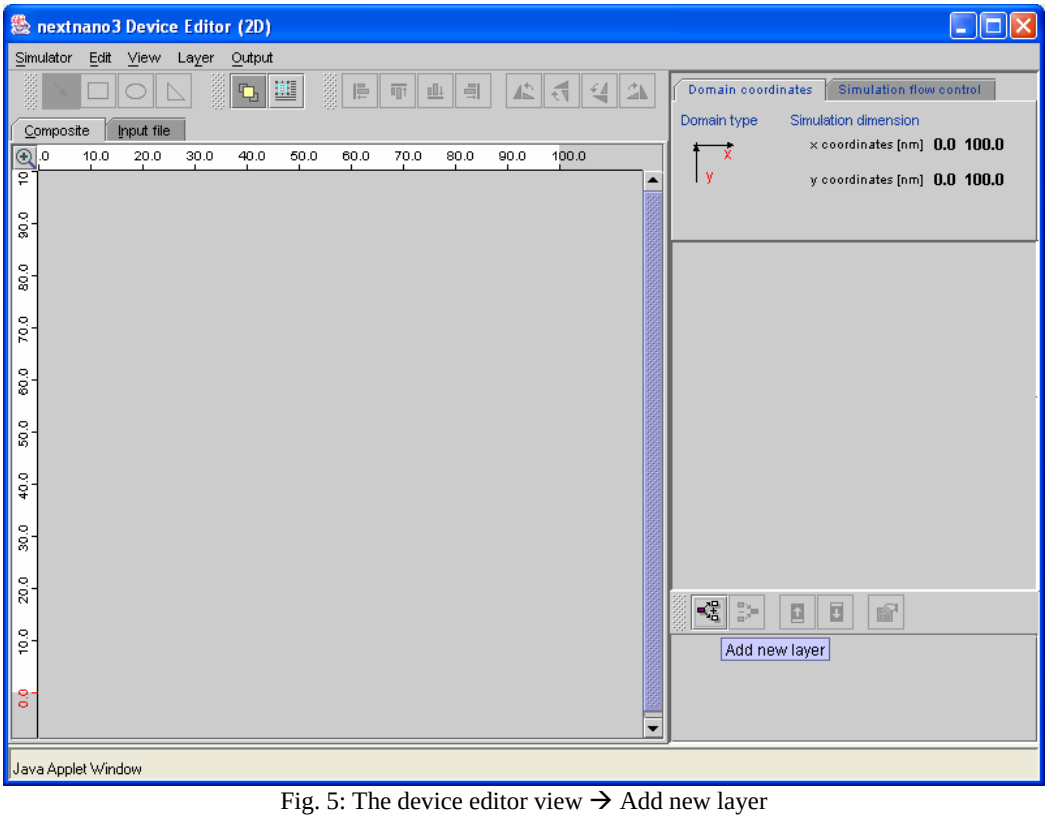



Fig. 5: The device editor view → Add new layer

A new layer is constructed by clicking on the buttton "Add new Layer". The principle of layers is similar to the well-known Photoshop© layers. A layer contains the elements belonging to a material and thus forms a cluster.

In the Pop-up window one can choose between material, doping and quantum regions. “Default material" determines the material which has the lowest priority (i.e. `region-priority=1`).


Fig. 6: The device editor view → Add new layer

### Using nextnano³ device editor

To draw suitable geometrical figures (rectangles, semi-ellipses and triangles) one should select the appropriate shape buttons (1) and draw the object by using the mouse. The position and the size of the object can also be changed by using the keyboard (2).

Green circle: This check box shows whether the chosen layer is visible

Yellow circle: This check box shows which layer is active, i.e. can currently be edited.

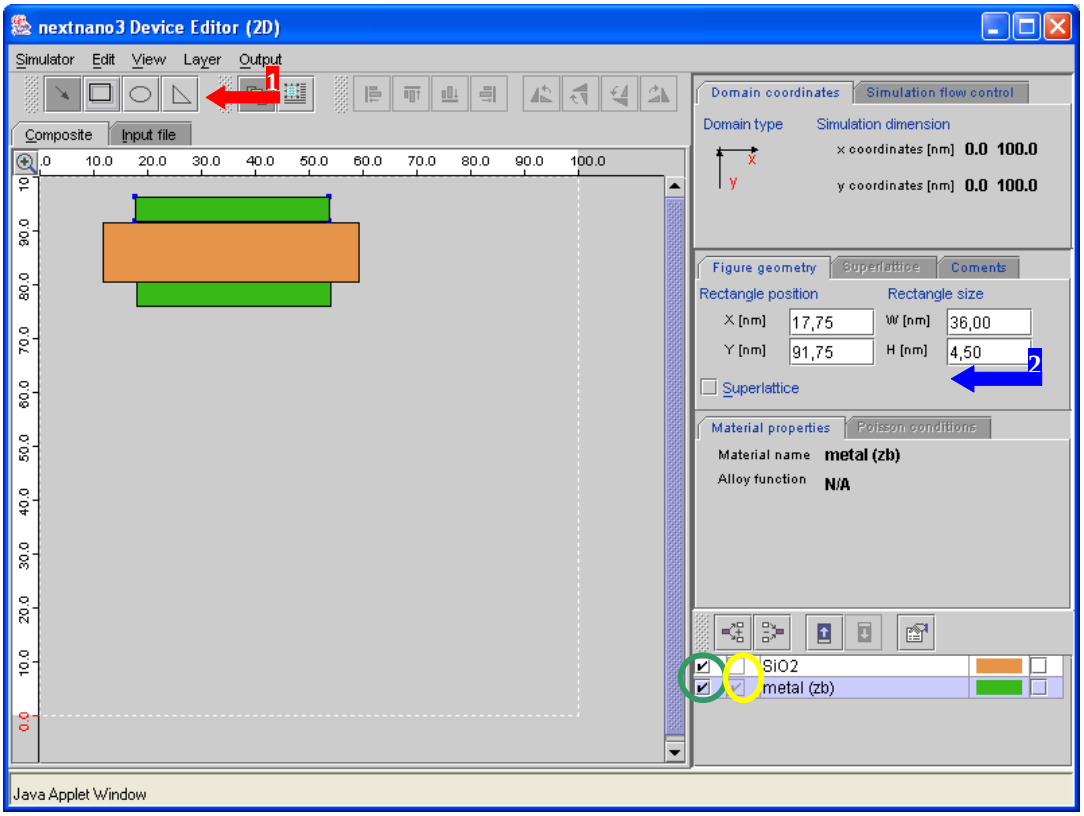



Fig. 7: Using device editor → Drawing

**Creating a doping region**

Device editor permits to specify different types of clusters. For drawing doping regions, do the following: Click on button "Add new layer" → "doping". The geometrical objects generated get transparent colors.

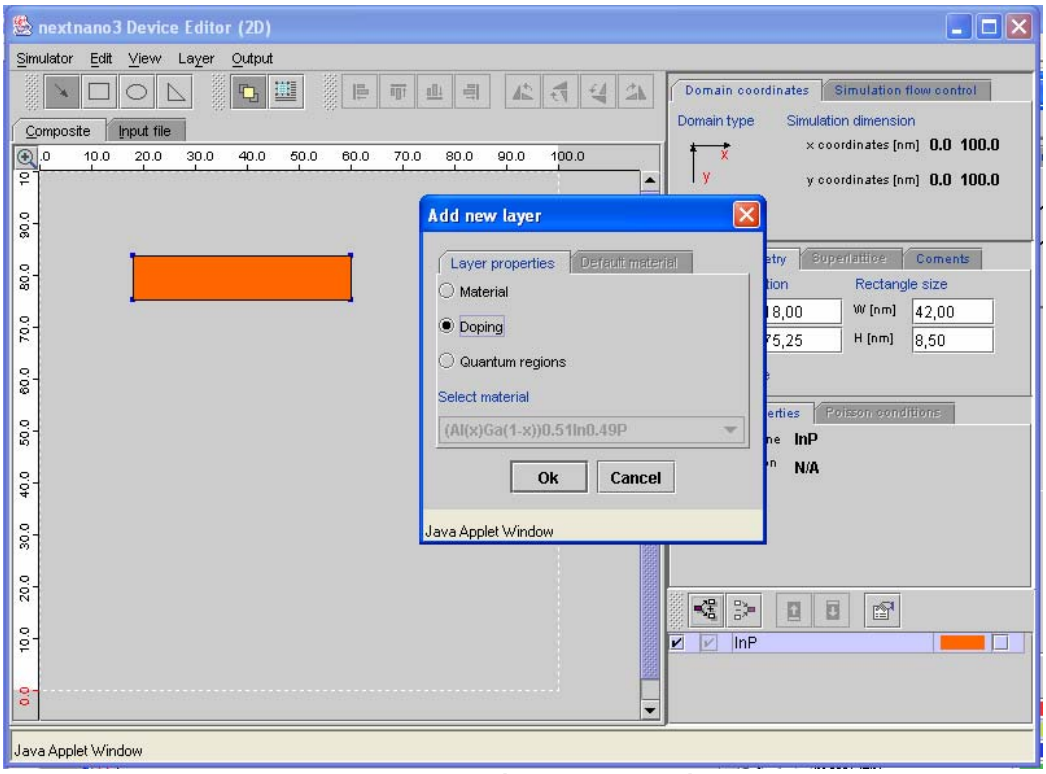



Fig. 8: Using device editor → Add new layer → Add doping

With the icon "Layer properties" one can change the type of doping (marked in the picture)

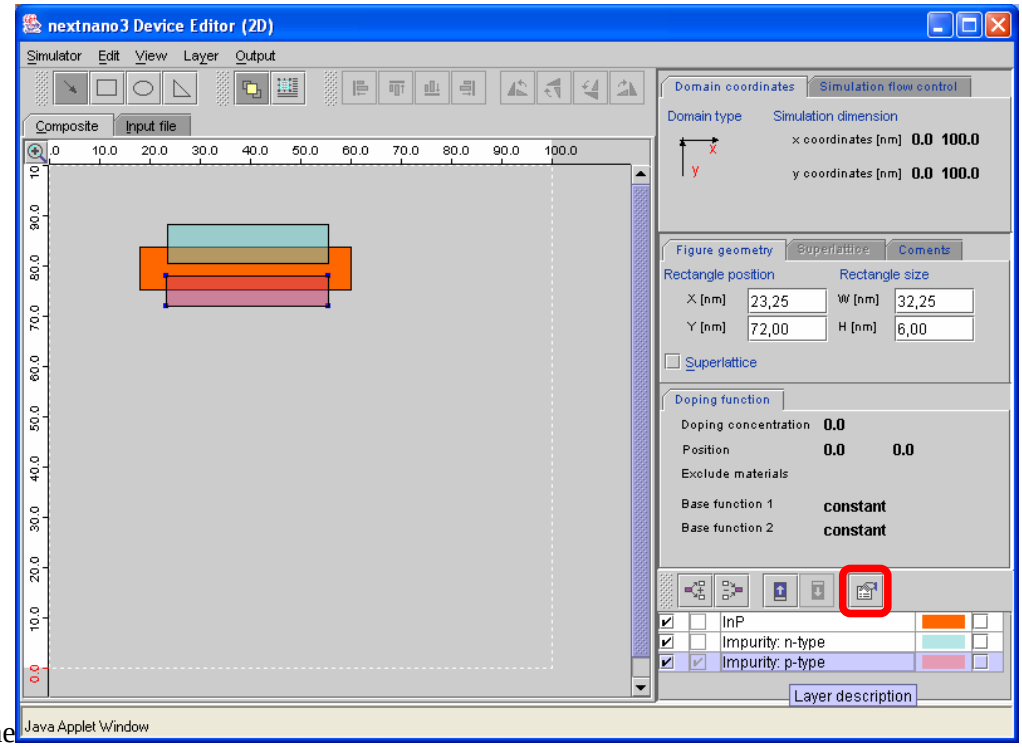



Fig. 9: Using device editor → Create the doping → Layer properties

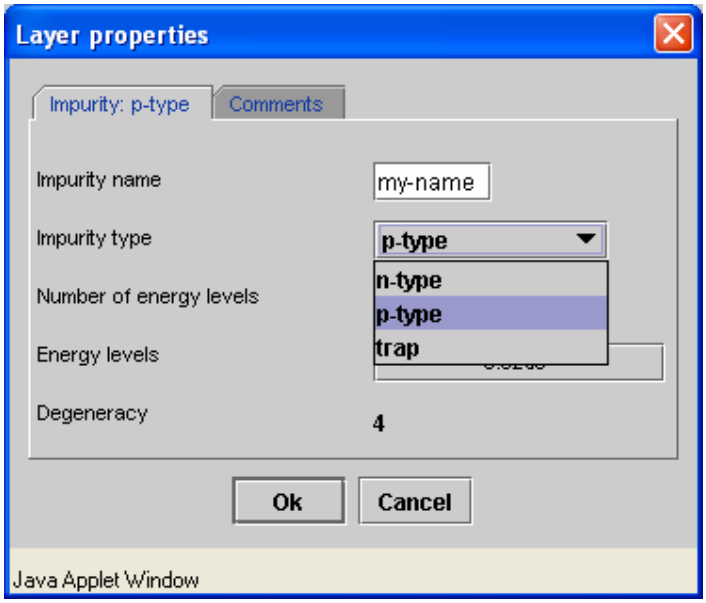



Fig. 10: Doping → Layer properties

Other doping functions and their properties are accessible with a right mouse click on the left pane.

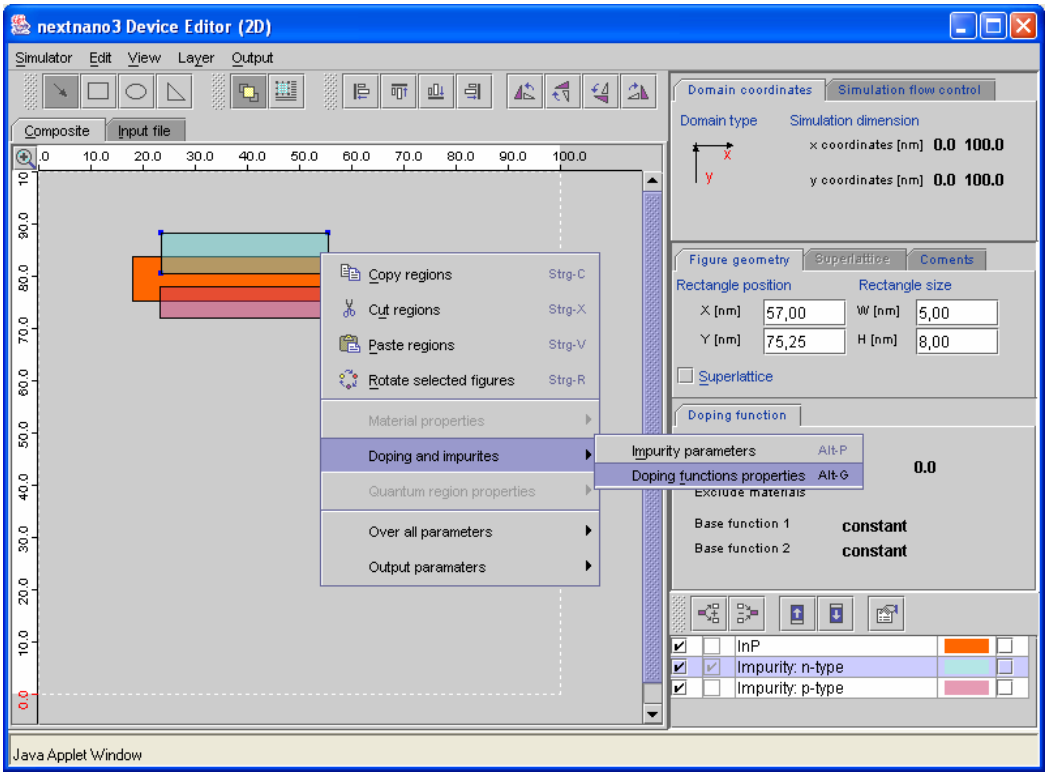



Fig. 11: Doping → Doping function properties

### Creating an quantum region

Quantum regions can also be drawn using the icon "Add new Layer". Diagrammatically, a quantum region has the same properties as doping regions, thus the objects have transparent colors.


Fig. 12: Using device editor → Add new layer → add quantum regions (1)

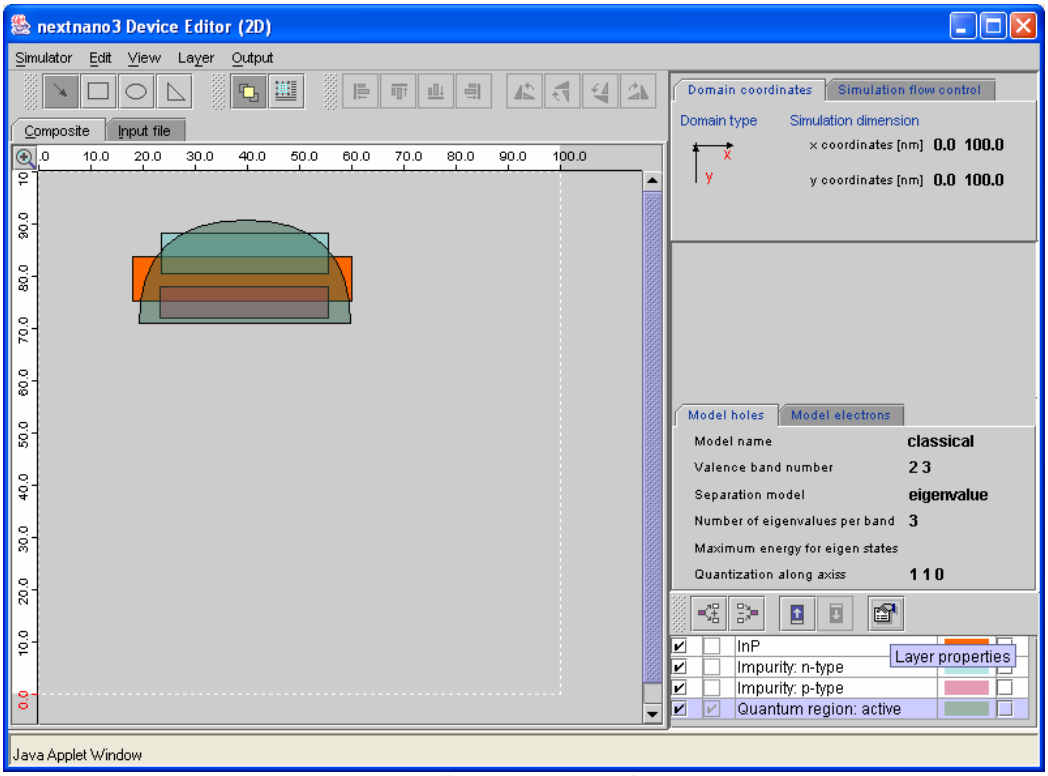



Fig. 13: Using device editor → Add new layer → add quantum regions (2)

Via the icon "Layer properties", one gets to the properties of the quantum regions.

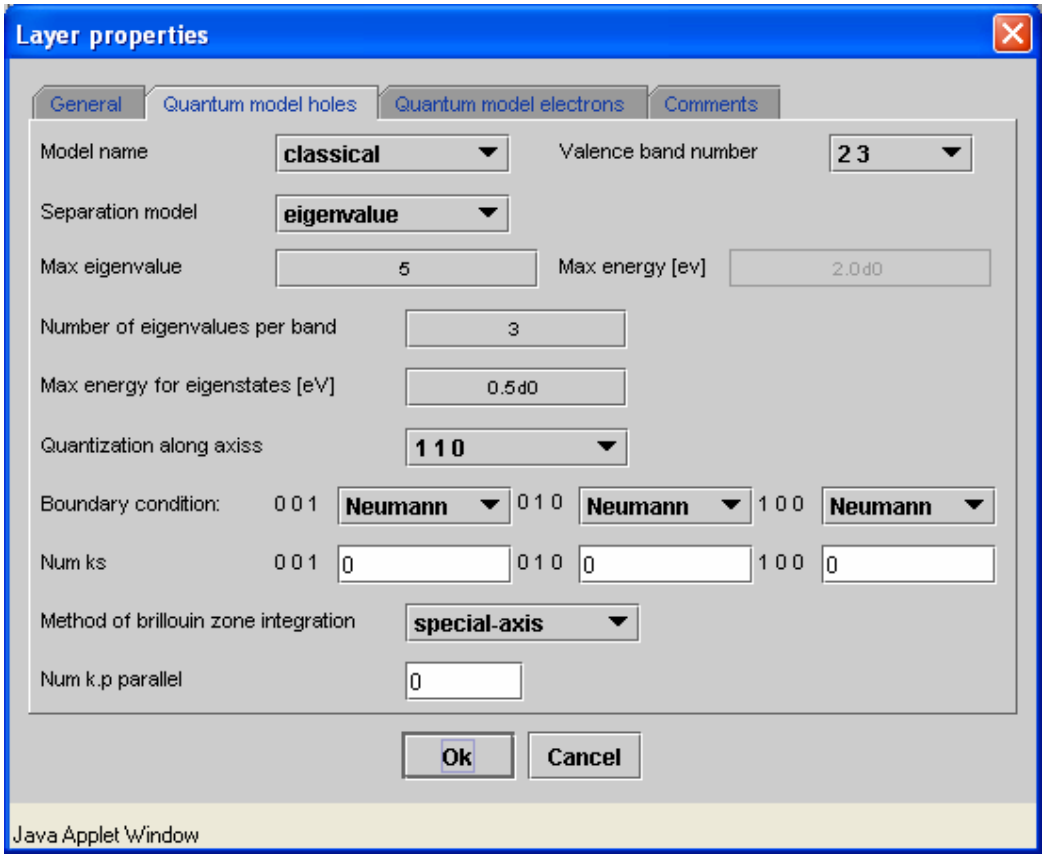



Fig. 14: Using device editor → Add new layer → add quantum regions → layer properties

### Editing

#### Layer

The following functions have an intuitive meaning

### Add new layer

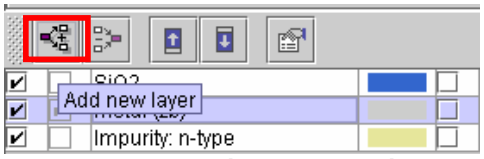



Fig. 15: Using device editor → Editing layer→ add new layer

### Remove selected layer

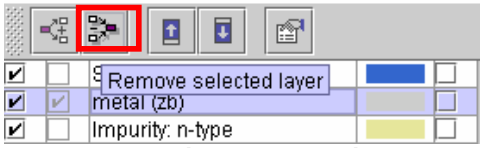



Fig. 16: Using device editor → Editing layer → remove selected layer

### Move up, move down selected layer

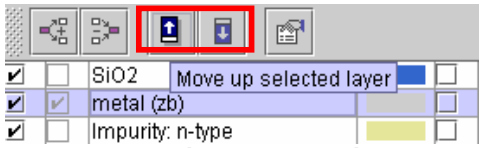



Fig. 17: Using device editor → Editing layer → move up, move down

### Layer properties

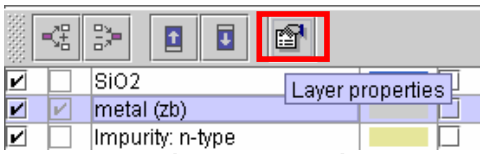



Fig. 18: Using device editor → Editing layer → add quantum regions (2)

### Choosing a layer color

The application generates a random color for any newly provided layer. With a double click on the layer color, one can change it.

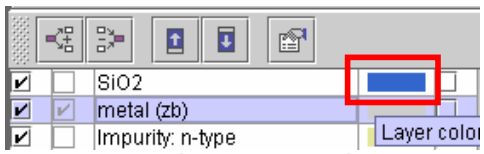



Fig. 19: Using device editor → Editing layer → chooseing a layer color

### Rotate

With a right mouse click on the selected region, the window "Rotate selected figures" pops up.

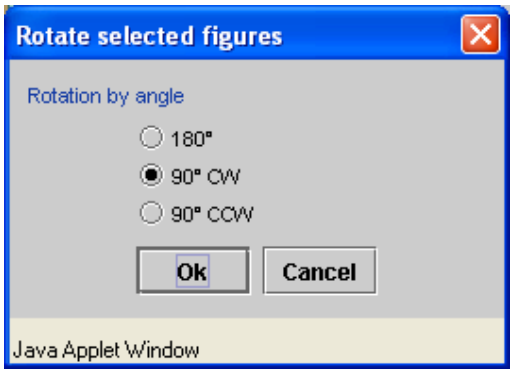



Fig. 20 Editing region→ Rotate selected figures

### Move a region

One can move one or several regions either by using the mouse, the keyboard (CTRL + mouse), or by directly specifying the amount in the menu dialog “figure geometry" (only for one region).

### Copy a region

One can copy one or several regions either by the right mouse click and selecting "Copy regions" or by the key combination CTRL + C.

### Cut a region

One can cut one or several regions either by the right mouse click and selecting "Cut regions" or by the key combination CTRL + X.

### Paste

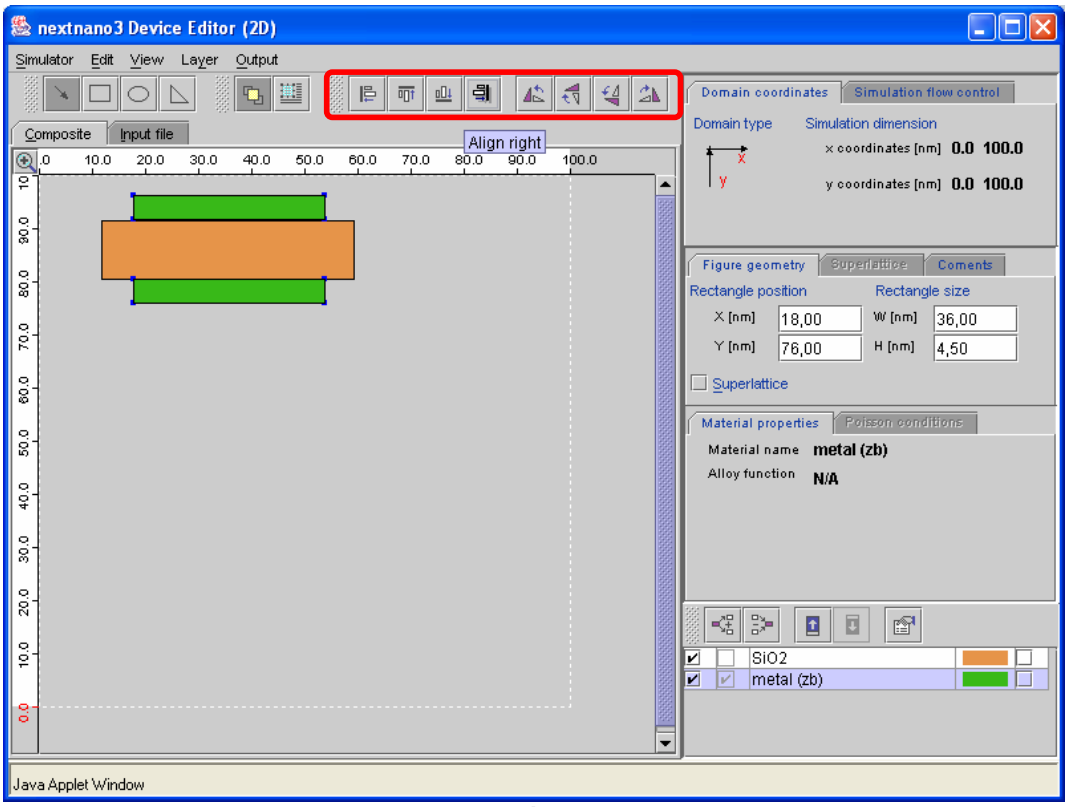
One can paste one or several regions either by the right mouse click and selecting "Paste regions" or by the key combination CTRL + V.

### Clone region

This function is accessible through the menu „EDIT → Clone regions“ or by the key combination CTRL+B

### Aligning and concatenating of objects

Two or several objects can be selected with the combination of SHIFT+mouse. With the help of align and concatenate buttons (highlighted in the picture) the objects can be aligned or concatenated.




Fug. 21: Using input device editor → align und concatenate of objects

In the following picture an example structure (Double Gate MOSFET) is shown which consists of different clusters.

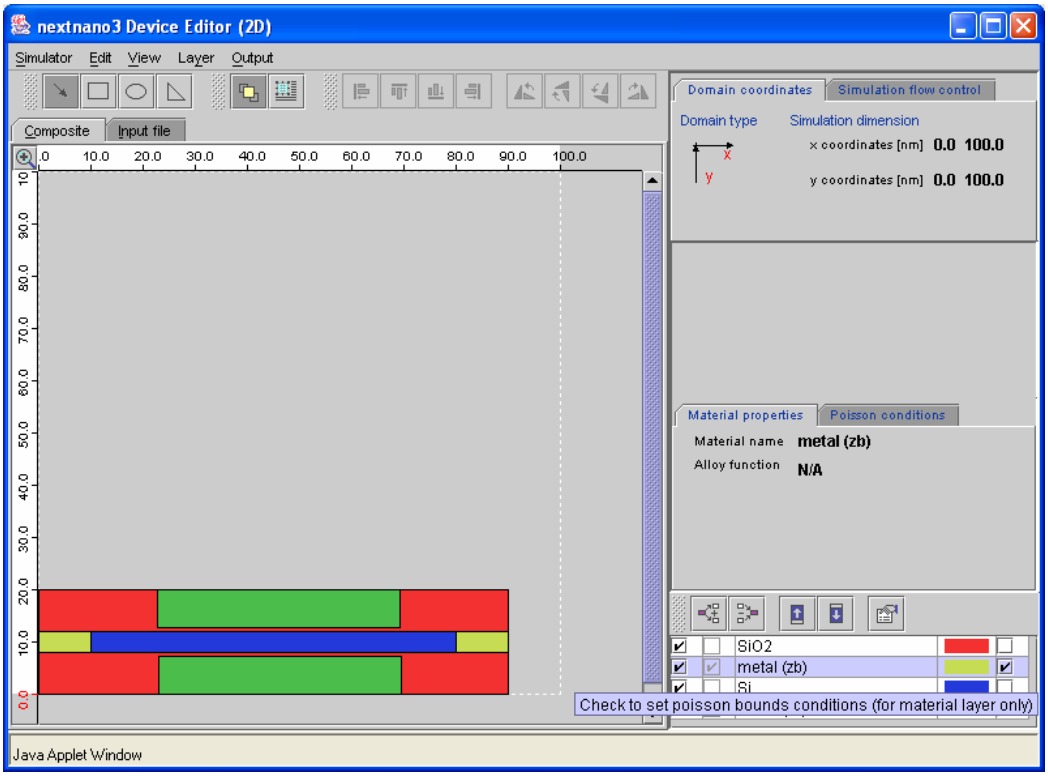



Fig. 22: Using the Device Editor → Editing , Poisson boundary conditions

Red circle: If the material is a contact, a Poisson boundary condition can be set.

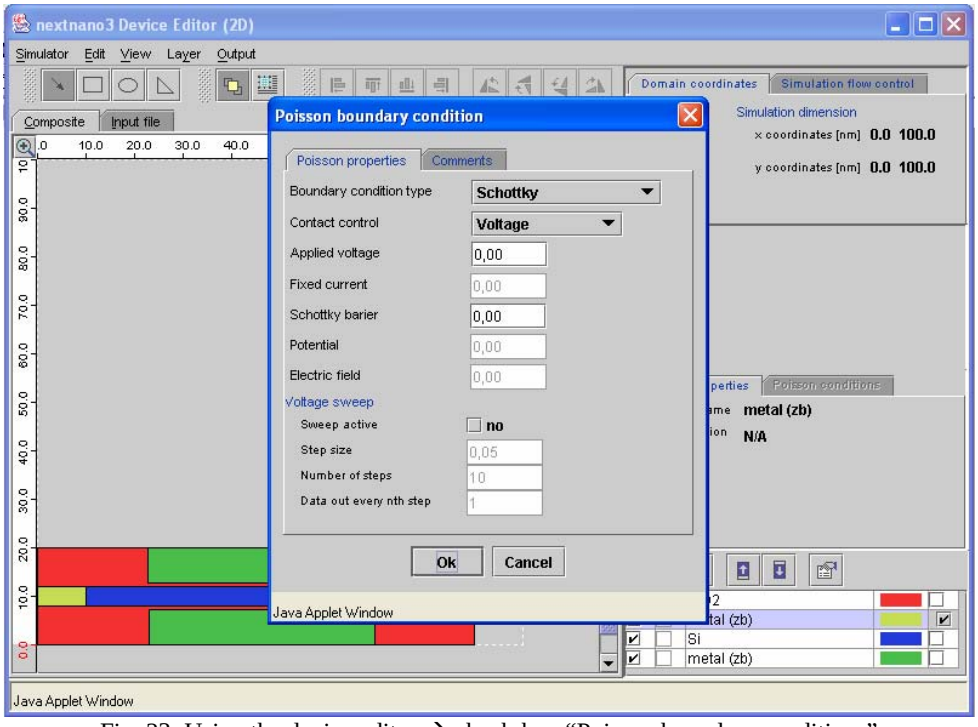



Fig. 23: Using the device editor → check-box “Poisson boundary conditions”

**Grid**

The grid can be visualized by clicking on the "GRID" button which is marked in the picture by a red square.

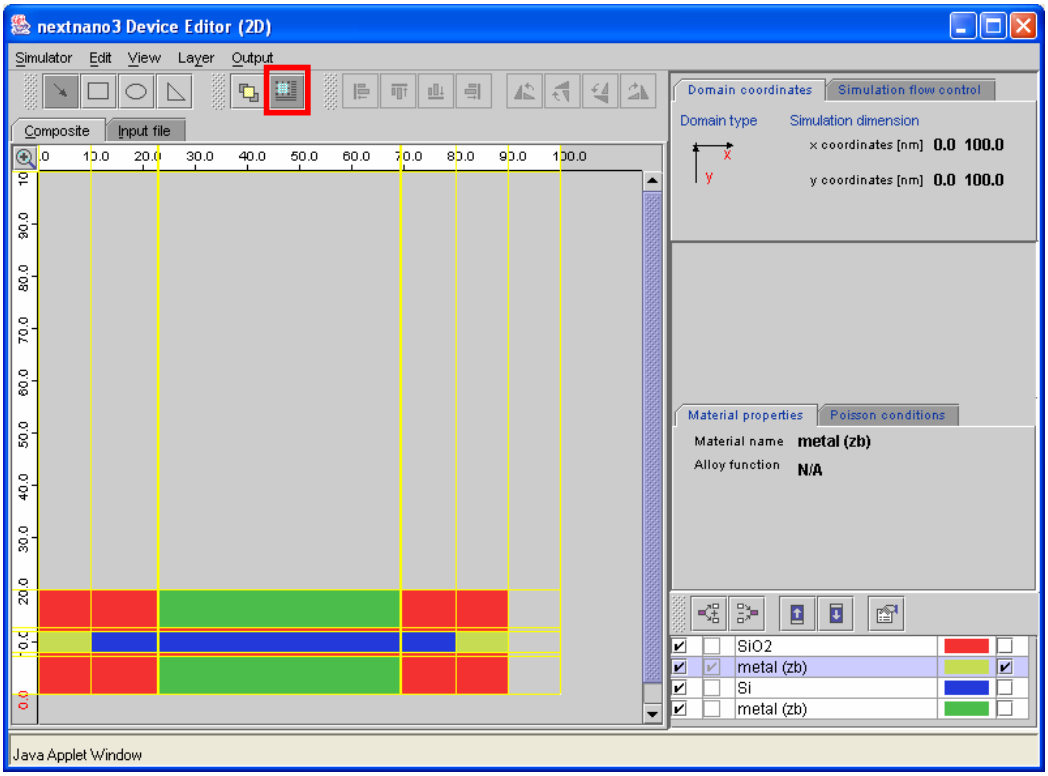



Fig. 24: Grid → Using the grid option

### Grid properties

For performing accurate next**nano³** calculations it is very important to add additional grid lines. This can be done by inserting additional nodes between the yellow grid lines. If one clicks on a specific column or row in the axes area, the manual input of "number of nodes" and "grid factor" (default should be 1.0) is possible. In the picture the input fields are marked by a blue rectangle. Then the whole column or row is active (marked with a transparent blue rectangle). A double-click on the chosen column or row makes it possible to open a window “grid properties” to manually input additional grid nodes.

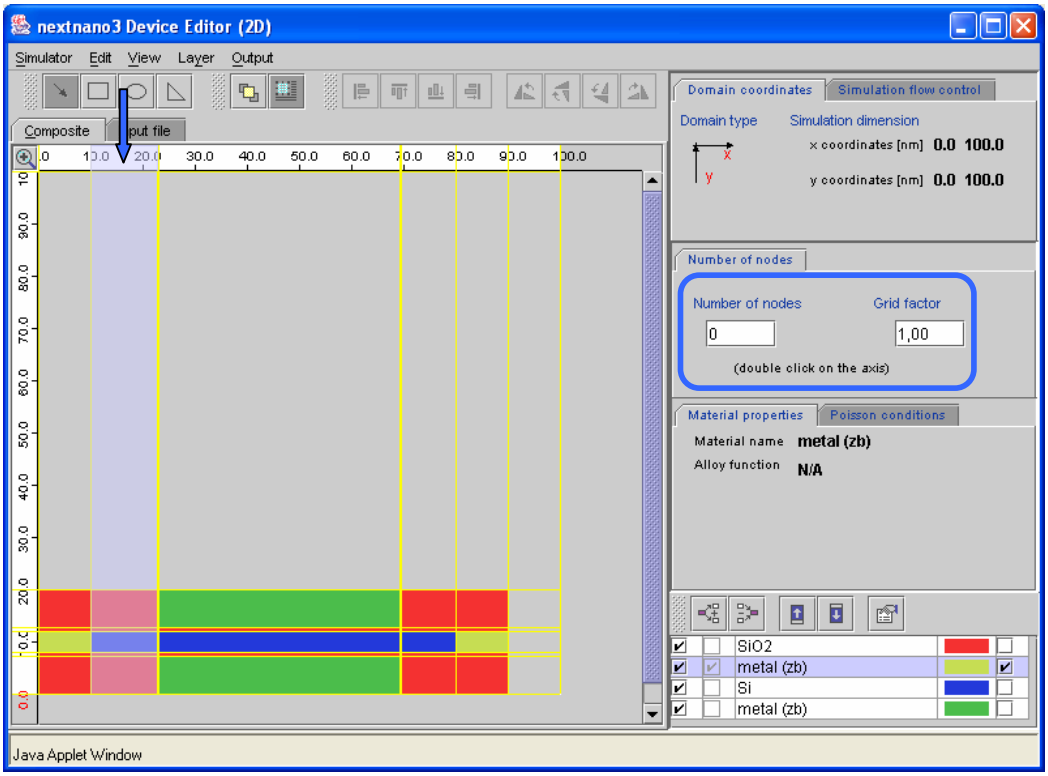



Fig. 25: Grid → Grid properties: Choice of the active column

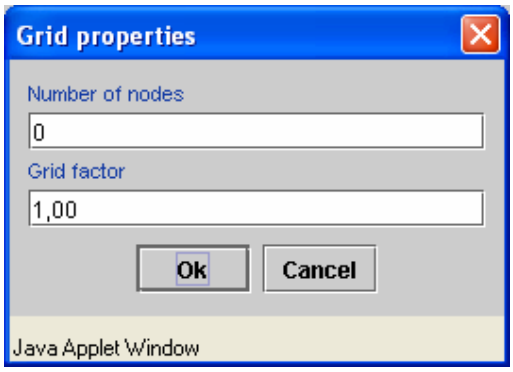



Fig. 26: Grid → Grid properties: Number of nodes, grid factor

**Change editor scale (Zoom)**

For an explicit examination or additional editing of the objects, a zoom function is included in the Device Editor. The purpose of this function is to increase or to reduce the scale (zoom factor). The button "Change editor scale" can be found at the “origin” position of the x and y coordinate axes.

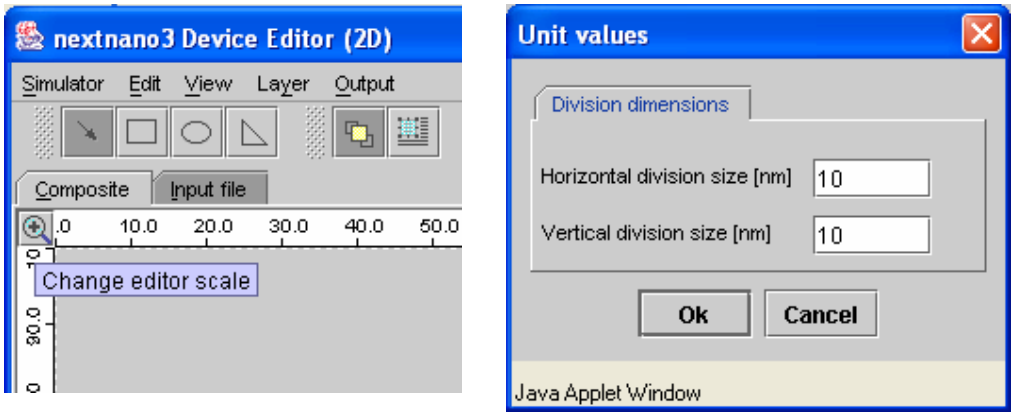



Fig. 27: Change editor scale: Position of zoom function, unit values

### Input file

A click on "input file" [1] opens the view to the input file. The area is divided into two windows. In window [2] the input file is shown which does all changes in the device geometry at run time (by means of an XML engine). The input file will be parsed in frame [3]. This function is very useful to inspect the necessary steps.

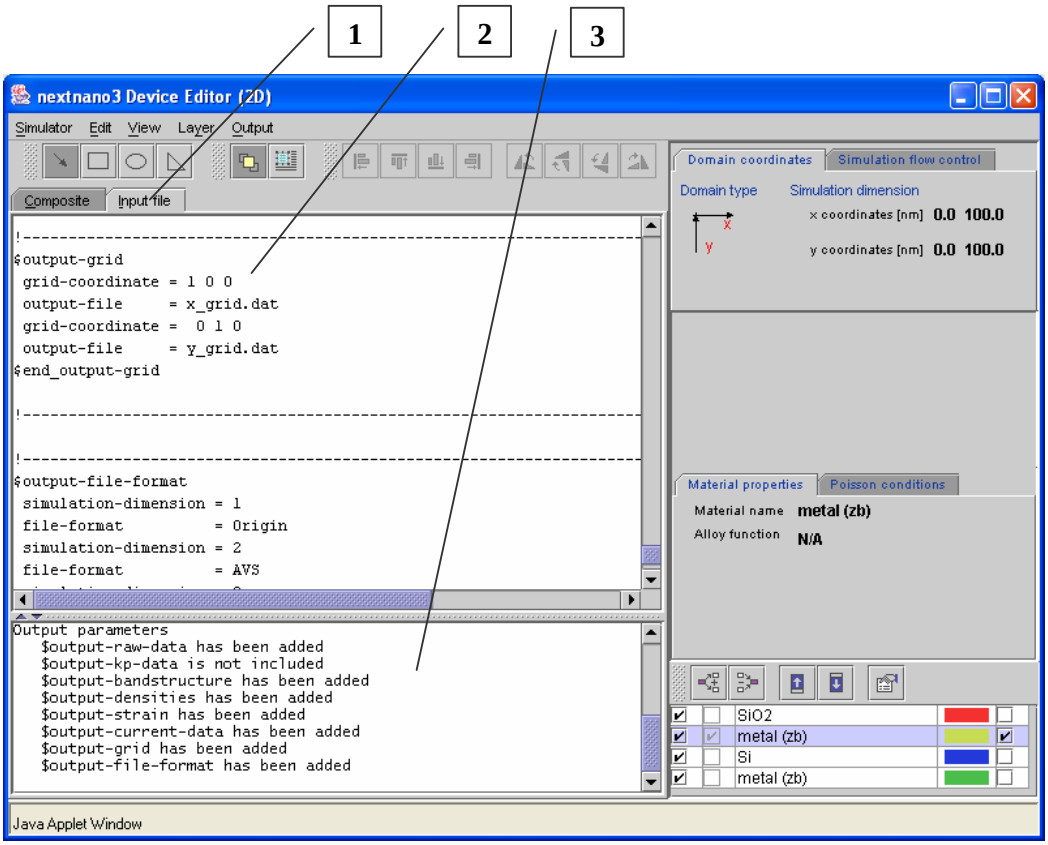



Fig.28: View of input file

**Sending input file via email:**

A mail module is included in the device editor which allows sending input files via email. By choosing from the menu "Simulator → Send to" the window "Edit SMTP server properties" opens. In the window the server name, login name and password has to be filled in for access to the server. If the authentification at the serverwas successful, the dialog window "Send to email" opens in which the user can enter one or several email addresses. If for some reason the authentification at the server failed, the input mask appears again where one can modify the name of the server, login and password.

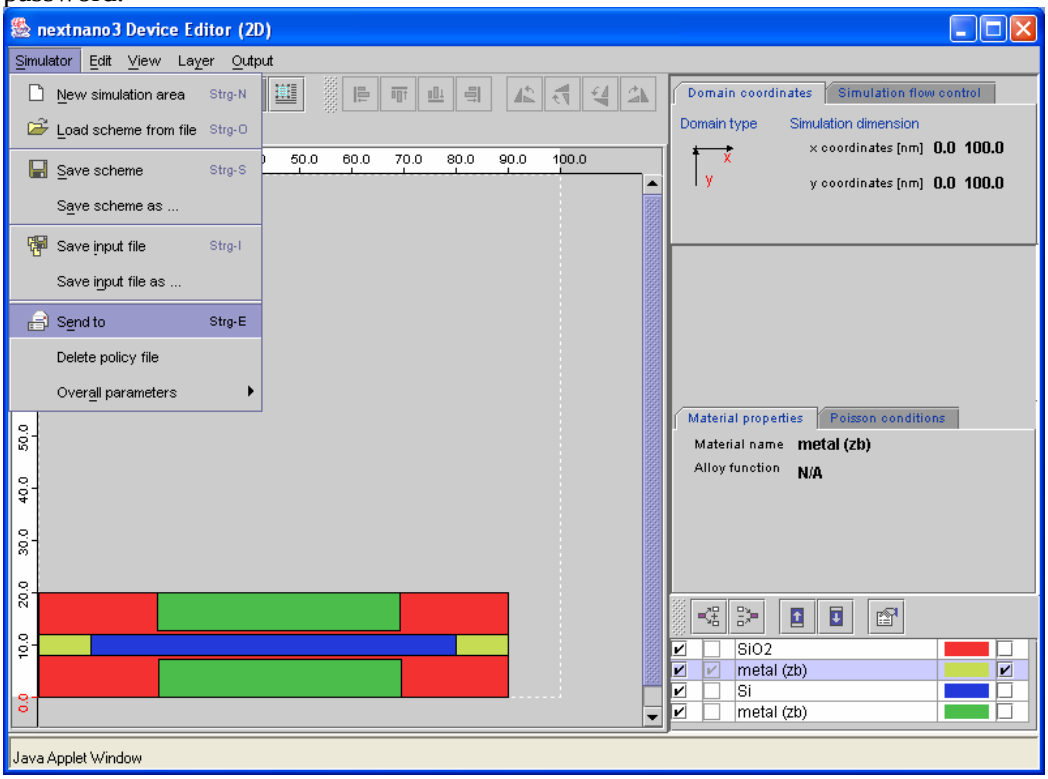


Fig. 29: Sending email

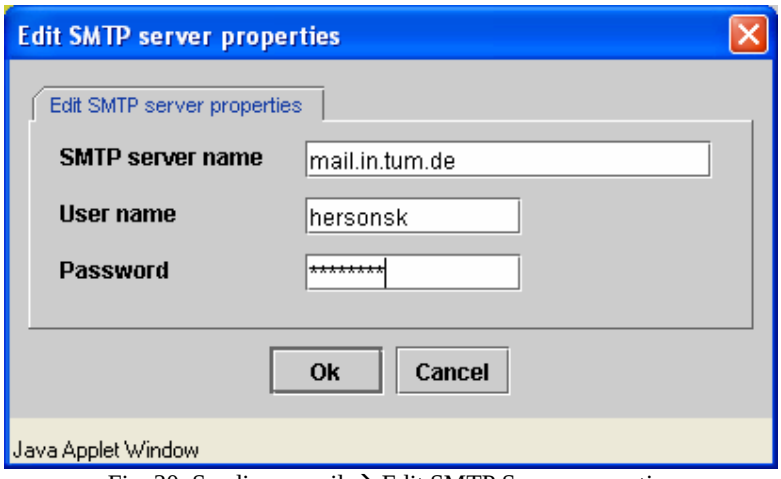


Fig. 30: Sending e-mail → Edit SMTP Server properties

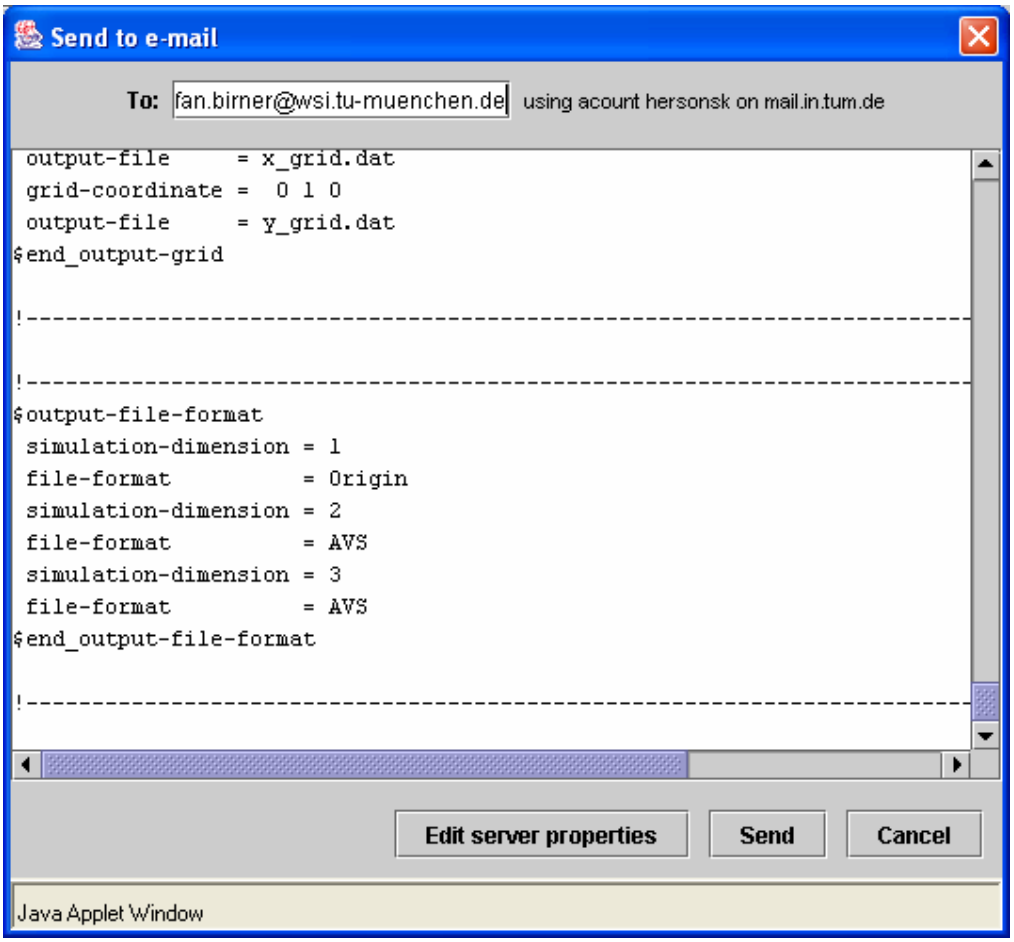



Fig. 31: Sending email → send to

## Shortkeys

| | | | | | |
|---|---|---|---|---|---|
| **edit** | increase the size and the width of the figure | shift + **[right]** | | new simulation area | strg+N |
| | decrease the size and the width of the figure | shift + **[left]** | | | |
| | decrease the size and the height of the figure | shift + **[up]** | **scheme** | load scheme from file | strg+O |
| | increase the size and the height of the figure | shift + **[down]** | | save scheme to file | strg+S |
| | | | | | |
| | move right | ctrl + **[right]** | **input file** | save input file | strg+L |
| | move left | ctrl + **[left]** | | send input file via e-mail | strg+E |
| | move up | ctrl + **[up]** | **layer:** | Add new layer | alt+A |
| | move down | ctrl + **[down]** | | remove layer | alt+X |
| | | | | move up selected layer | alt+U |
| | undo operation | strg+Z | | move down selected layer | alt+D |
| | redo operation | strg+Y | | edit selected layer | alt+F12 |
| | copy regions | strg+C | **material properties** | Material name | alt+N |
| | cut regions | strg X | | Alloy function | alt+F |
| | paste regions | strrg+V | **PBC** | set poisson boundary condition | strg+B |
| | clone regions | strg+B | | view poisson boundary condition | alt+E |
| | rotate object | strg+R | | | |
| | | | **doping und impurites:** | inpurity parameters | alt+P |
| **2 objects** | alighn left | shift+L | | doping funktions properties | alt+G |
| | align top | shift+T | **quantum region properties:** | deactivate quantum cluster | alt+Q |
| | align bottom | shift B | | quantum model holes | alt+H |
| | align right | shift R | | quantum model electrons | alt+S |
| | | | | | |
| | concatenate left | strg+L | **view:** | compisite view | alt+C |
| | concatenate top | strg+T | | view inputfile | alt+I |
| | concatenate bottom | strg+B | | edit modus | alt+F2 |
| | concatenate right | strg+R | | show grid und regions | alt+F3 |
| **output** | output 1-band-Schroedinger | shift+F2 | | change simulation area zoom | alt+Z |
| | output k.p. data | shift+F3 | | | |
| | output band structure | shift+F4 | **overall parameters** | domain coordinates | strg+D |
| | output densities | shift+F5 | | Lattice temperature | Strg+T |
| | output strain | shift+F6 | | simulation flow control | strg+F |
| | output current data | shift+F7 | | magnetic filed | strg+M |
| | output grid | shift+F8 | | default material name | strg+A |

**Pictures**