SimuLAMP™

Software for Optical and Thermal Design and
Optimization of LED Lamps and Arrays

Graphical User Interface Manual

Version 2.0

User Support: SimuLED-support@str-soft.com
Software Sales: STR-sales@str-soft.com
Phone: +7 812 320 4390 Fax: +7 812 326 6194
STR Group Ltd. www.str-soft.com
Engels av. 27, P.O. Box 89, 194156, St. Petersburg, Russia

Table of contents

1 Introduction

SimuLAMP software package is designed for modeling of LED lamp operation. It has user-friendly graphical user interface (GUI) allowing the user to specify the geometry and physical parameters of LED lamps, run the computations and visualize the results. To simplify input of the geometry, the whole lamp is considered as an axisymmetric object, while the submount with single or several LED dice is simulated in a 3D way. So, the user needs to specify 2D side view of the lamp and 2D layout of the submount.

SimuLAMP can be used as a stand-alone tool or coupled with **SpeCLED™**[1] and **RATRO™**[2] software package which provides the chip I-V characteristics as a function of temperature.

SimuLAMP provides solution of the following tasks:

- Solution of coupled optical/thermal problem in a complex package geometry accounting for heat release in the LED chip and heat release in an encapsulant due to light absorption and Stokes shift
- Advanced model of light conversion in individual phosphor and phosphor mixtures (for white-light LED lamps)
- Support of single- and multichip package configurations including RGB LEDs
- Simulations of the electrical circuit used in operation of multi-pixel LED array
- Analysis of package operating in DC/AC/Quasi-CW modes

The user should specify the following lamp parameters:

- Lamp design (lens, PCB, heatsink, etc.)
- Optical characteristics of the lamp units (absorption/refraction indices of the lamp materials, reflection and transmission coefficients of the interfaces, etc.)
- Top view of the lamp with positioning the LED die (dice) placed on the submount
- LED emission spectrum (for each LED in the lamp)
- Emission pattern (for each LED in the lamp)

[1] RATRO™ — **RA**y-**TR**acing Simulat**O**r of Light Propagation http://www.str-soft.com/products/RATRO/

[2] SpecLED™ - **Spre**ading of **C**urrent in **L**ight **E**mitting **D**iodes http://www.str-soft.com/products/SpeCLED/

- I-V and L-I characteristics of each LED
- Current magnitude for each LED or an electric circuit used for power supply applied to LED array

Among all characteristics of the LED lamp provided by **SimuLAMP** the following are the most important:

- Temperature distribution over the LED package, thermal resistance
- Near-field and far-field radiation patterns
- Output light spectrum, color uniformity
- Optical losses in the package
- CRI, CCT and other characteristics of white-light LEDs

2 Getting Started

The main window of the **SimuLAMP** program contains the following tab windows:

- **Lamp** – specification of the lamp design. It includes the following of the second level:
  - ✓ **Geometry** – drawing of the contours specifying the 2D side view of the lamp geometry in axisymmetric approximation. The user can draw new elements, such as lines, arcs or splines, either by clicking the mouse or by explicit specification of the object coordinates, as well as transform, move or copy the existing elements. The lamp geometry should be represented as a number of closed contours without dangling, intersecting or overlapping lines. All closed contours are automatically identified by the program as separate blocks. The user should draw the contours representing the submount, case, heat sink, and epoxy. The LED chip is considered as internal zero-thickness surface on the submount top and should not be created as an individual block.
  - ✓ **Blocks** – assigning the materials to the closed contours and assigning the boundary conditions to the external boundaries and block interfaces. Prior to assigning operation, the materials should be created on the **Materials** tab. Besides, special zones such as active region, submount and conversion medium (zone filled with phosphor) should be identified by the user manually.

- ✓ **Grid** – generation of the computational grid in all blocks. The user should assign the fragmentation of the boundaries, while 2D grid inside the blocks is built automatically. Triangular and structured (in quadrilateral blocks only) grid is supported.

- **Dice** – specification of 2D layout of LED dice on the circular submount. Like **Lamp**, **Dice** tab includes several tabs of the second level:
  - ✓ **Geometry** – drawing of the submount circle and contours of one or several LED chips inside.
  - ✓ **Blocks** – assigning the LED chips with given I-V characteristics to the drawn contours. Prior to assigning operation, the chips should be created on the **LED Characteristics** tab.
  - ✓ **Grid** – generation of the computational grid in all blocks.
- **Materials** – specification of the thermal and optical materials properties. Manual assignment and import from other project files are supported.
- **LED Characteristics** – specification of the LED chips, including assignment of the chip emission spectrum at different temperatures and currents, I-V and I-WPE characteristics at different temperatures, and emission pattern in the far field or near field (with the use of RATRO simulations files). The LED characteristics can either be assigned manually or imported from the computations made by SpeCLED. The spectra can either be assigned using the analytical approximations with temperature and current dependent parameters or by user-specified functions of 2 types: wavelength - temperature dependent or wavelength – temperature – current dependent. The emission pattern in the far field can either be assigned using the list of conventional patterns, or by user-specified functions. To specify the near field the user can import RATRO simulations file (ray file).
- **Phosphor** – assigning properties of the phosphors. The user can specify a list of the phosphors and characterize each of them by their particle parameters (concentration, size distribution, refractive index), optical properties (emission spectrum, absorption coefficient spectrum, internal quantum efficiency) and scattering properties (the model used for scattering/absorption pattern and scattering pattern calculations).
- **Circuit** – specification of the current flowing through the LED dice. DC, AC and Quasi-CW mode of LED operation are supported. Two modes of the current specification are available:
  - ✓ Direct assignment of current flowing through the particular chip

- ✓ Specification of electric circuit used for power supply applied to LED array. In this mode, the user draws the electric circuit and specifies the voltage applied to the whole circuit and passive elements if necessary.

➢ **Solver** – specification of the computational settings, running the solver, and visualization of the results. Three modes of computations are available:

- ✓ **Isothermal Ray Tracing** – an isothermal light emission problem. The user should assign some computational parameters and run the program. Results of the computations are presented in the **Results** tab opened on the **Solver** tab after the computation is completed. The results include the distribution of the color and intensity of the extracted light in the across the far-field sphere. Color coordinates, color temperature and color rendering index are calculated both for each point of the far-field distribution and for the lamp as a whole.
- ✓ **Heat Transfer** – running a heat transfer problem ignoring the heat release in the light absorption. The user should assign some computational parameters and run the program. Results of the computations are stored in *.cgs files which are visualized by **SimuLEDView** tool supplied with the SimuLAMP package.
- ✓ **Coupled Heat Transfer and Ray Tracing** – running a heat transfer problem accounting for the heat release in the light absorption. The user should assign the computational parameters of both heat transfer and ray tracing and run the program. The light emission results are presented in the **Results** tabs after the computation is completed. The heat transfer results are visualized by **SimuLEDView** tool.

➢ **Functions** – scripts describing complex dependencies of all properties used in SimuLAMP on various parameters. Five groups of functions are available:

- ✓ **Wavelength Dependencies**. Functions of wavelength can be used to specify complex spectra of the LED emission, phosphor emission and excitation, phosphor scattering and absorption cross sections spectra, and of any wavelength-dependent optical parameters of the lamp materials. The user can also assign functions with additional temperature and/or current variables, which can be used to specify temperature-current dependent emission spectra of the LED.
- ✓ **Temperature Dependencies**. Functions of temperature can be used to specify temperature and current dependent parameters in the analytical approximations of the

LED emission spectrums, LED I-V and I-L curves, phosphor emission and excitation, internal quantum efficiency, etc.

✓ **Coordinate Dependencies**. Functions of coordinates can be used to specify approximations of the parameters of the lamp materials as a function of local coordinates. Additional temperature variable can also be used in this type of functions.

✓ **Angular Dependencies**. Functions of polar angle and wavelength or polar angle and asymmetry factor can be used to specify scattering pattern of the phosphor particles (asymmetry factor can also be wavelength and temperature dependent) and emission pattern of the LED chip (asymmetry factor is replaced by g-parameter here).

✓ **Radial Dependencies.** Functions of radial coordinate can be used to specify size distribution of the phosphor particles.

Projects of **SimuLAMP** with the lamp and dice geometry, the LEDs and phosphors properties, the functions assignment and the results of simulations are stored in files with extension *.lmp.

In addition to the built-in visualization of light emission, the results are stored in ASCII files with the extension *.cgs to be viewed by **SimuLEDView** visualization tool, which provides more detailed presentation of the computations.

Starting **SimuLAMP** graphical user interface requires registration of the program. **SimuLAMP** is supplied on terms of single-node or network license using dongle key protection. The program is delivered with a SenseLock dongle key that should be plugged in a USB port on the PC where **SimuLAMP** will be used.

3 Lamp Tab Window

The **Lamp** tab window is designed to provide an easy-to-use specification of the lamp design. It includes the following tabs of the second level: **Geometry**, **Blocks** and **Grid**. The measurement units used within **Lamp** tab are mm.

3.1 Geometry

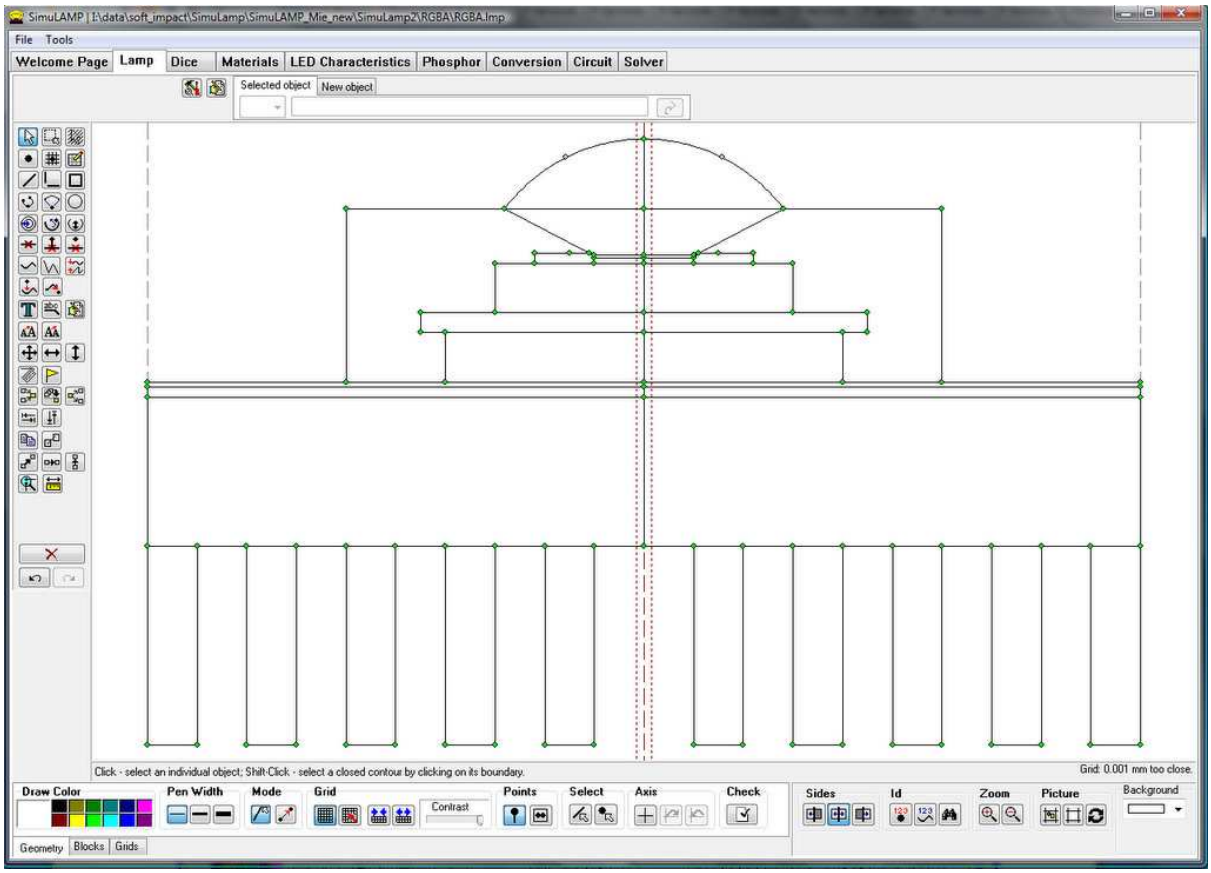


Fig. 1

Geometry tab is designed for drawing the contours specifying the side view of the lamp geometry in axisymmetric approximation. It includes the following elements:

- **Graphics Window** providing the visual lamp presentation
- **Object Editor** on top of the window providing explicit specification of the coordinates of the lamp elements
- **Drawing** toolbar on the left-hand side providing the tools to create or modify the lamp elements (see Appendix 2: Tools for Geometry Specification for details)

- **Viewing** toolbar at the bottom providing access to some options of geometry specification (see Appendix 3: Options of Geometry Visualization for details)

The **Object Editor** located on top of the **Geometry** tab provides explicit modification of the coordinates of boundary end points (see **Fig. 2**). It includes tabs **Selected Object** and **New Object** allowing the user to edit coordinates of a selected existing object and to create a new object by specifying its coordinates, respectively. Each tab contains the following items:

- **Object Type** combo box specifying the type of the edited object, which can be **Point**, **Line**, **Arc**, **Circle**, Polyline or **Spline**. For a selected object, this parameter can not be changed. For a new object, it should be selected by the user prior to specification of the object coordinates.
- The list of the parameters to be specified:
  - ✓ For a **Point**: X, Y – point coordinates.
  - ✓ For a **Line**: X1, Y1, X2, Y2 – coordinates of first and last ending points.
  - ✓ For an **Arc**: Xc, Yc, R, A1, A2 – center coordinates, radius, and angles of the arc ending points.
  - ✓ For a **Circle**: Xc, Yc, R – circle center and radius.
  - ✓ For a **Polyline** or **Spline**: X1, Y1, X2, Y2, X3, Y3, etc – coordinates of all points.
- A text field with the parameters. Text field contains both name of the parameters, which are added automatically, and their values, which are entered by the user. Switching the cursor position between values of different parameters can be made by pressing **Tab** key (go to the next parameter) and **Shift-Tab** (go to the previous parameter).
- Button  **Apply**.

Fig. 2

Some options of the geometry visualization are assigned in **Options** dialog window which is available by **Options** button on the top toolbar (see Appendix 4: Options Dialog WindowAppendix 4: Options Dialog Window for details).

3.2 Blocks

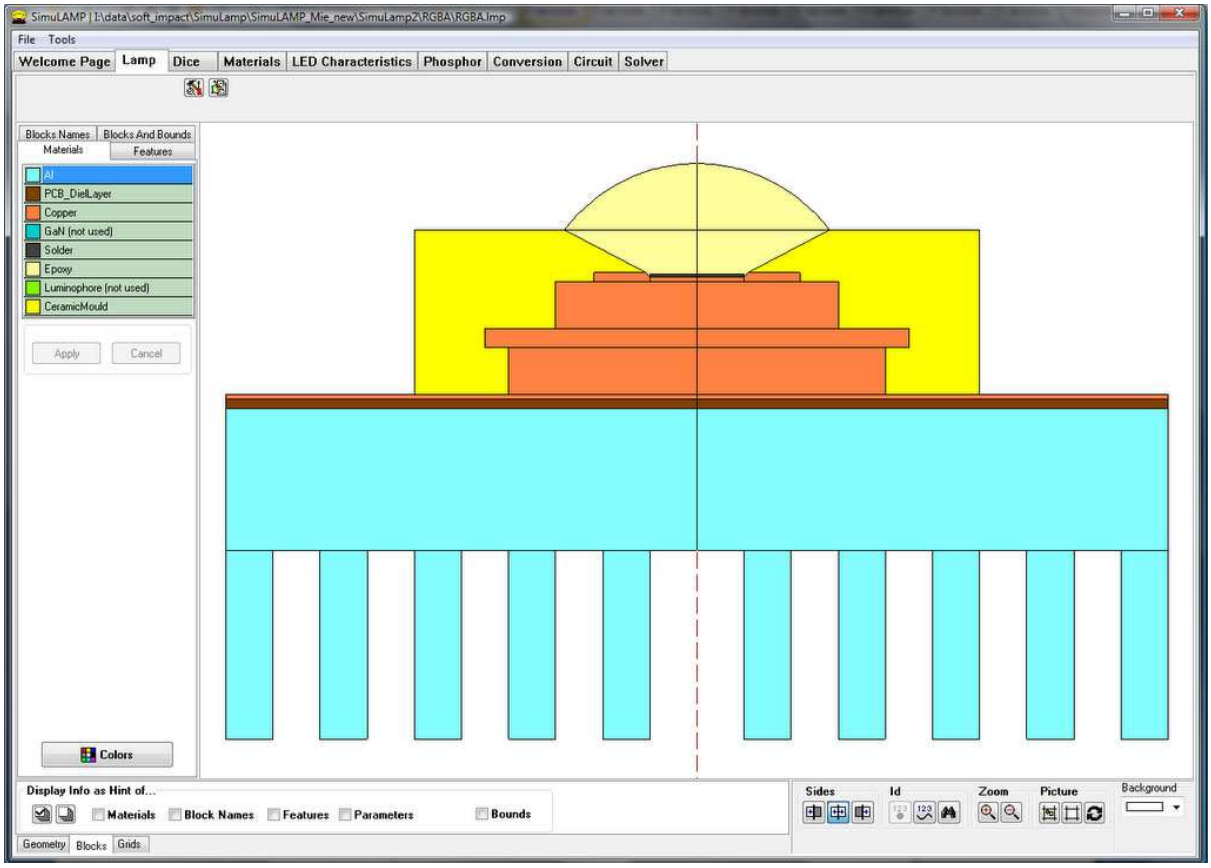


Fig.3

The **Blocks** tab window (see **Fig.3**) is designed for assigning the materials to the closed contours and assigning the boundary conditions to the external boundaries and block interfaces. It includes **Graphics Window** providing the visual lamp presentation and the control panel with the following tabs:

- Materials
- Features
- Blocks Names
- Blocks and Bounds

The **Materials** tab of the control panel includes the list of the materials to be applied to blocks. It includes all materials that are specified on the main **Materials** tab of **SimuLAMP**.

To assign the material, the user should do the following:

- Select the material by clicking on the material name in the list
- Click inside the block. The block will be filled with the color corresponding to the selected material.

- Press **Apply** button to accept the assignment.

If the same material should be assigned to the several blocks, the user should consecutively click inside all of them before pressing the **Apply** button.

To view which material is currently assigned to the blocks hints are used. The user should mark the corresponding checkbox on the panel below the graphics window and place the mouse cursor above the block without clicking the button.

Note that if the user clicks inside a block when the **Materials** tab is active, the material currently selected in the list is always assigned to the block. If it happens accidentally and no material specification is needed, the user should press **Cancel**.

The **Features** tab of the control panel includes the list of special types of blocks considered **SimuLAMP**, namely, LED and conversion medium. The user should assign the following type:

- **LED**. This type should be assigned to block representing the submount. The heterostructure itself is assumed to be thin compared to the other amp elements and is simulated by a surface layer of zero thickness. This surface is assumed to be the top surface of the block of the LED type.

Switching to the **Blocks & Bounds** tab of the control panel allows specification of various block parameters and boundary conditions.

In **Block Parameters** mode one can specify if the selected block is a conversion medium. Then the respective checkbox is checked, a drop-down list appears where the user is able to choose a conversion medium from ones available in **Conversion->Conversion Mediums** tab.

The **Boundary Conditions** are set on all internal and external boundaries for both heat transfer (thermal boundary conditions) and light extraction (optical boundary conditions) problems. Some boundary conditions are set by default, so the user should manually assign the boundary conditions which differ from the default ones.

The following boundary conditions should be specified:

- **Heterostructure** boundary condition – special boundary condition which identifies the heterostructure surface on top of the block to which **LED** type is assigned.

- **Optical** boundary conditions on external surfaces of the semitransparent blocks and on interfaces between semitransparent blocks. The default condition is a smooth surface.
- **Thermal** boundary conditions on external lamp boundaries. The common condition that is assigned to all boundaries by default is specified as a global parameter in the **Solver->Heat Transfer** tab.

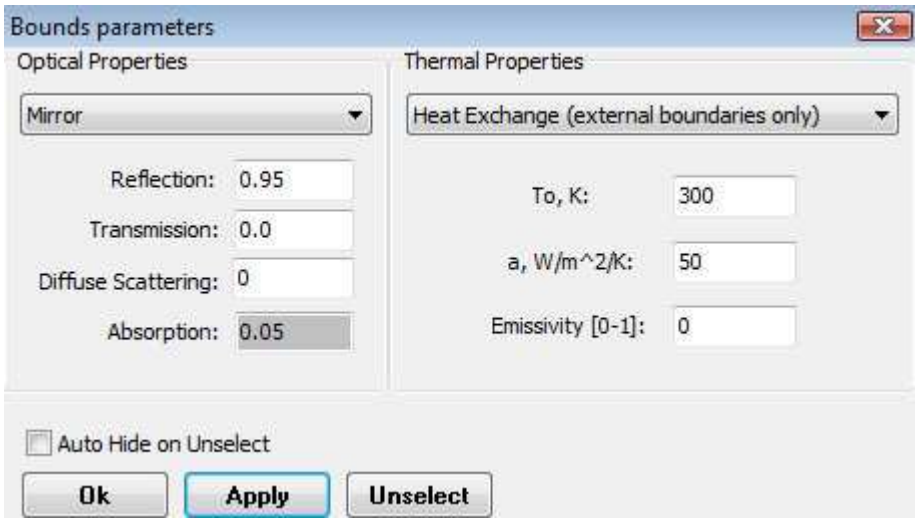


Fig. 4

To assign the boundary conditions, the user should click on the boundary with left mouse button. This will open Boundary Conditions dialog window. It contains the following sections:

- **Optical Properties** section contains a combobox with the following surface types:
  - ✓ **Smooth**. A default boundary condition. No parameters are required. The reflection and transmission angles obey Snell's law. The reflection and transmission coefficients of the reflected and absorbed rays are calculated for TE and TM polarized waves according to Fresnel's law.
  - ✓ **Mirror**. The reflection and transmission are defined by the user. The following parameters are available:
    - ❖ **Reflection** – the reflection coefficient (R), the reflection angle corresponds to a smooth surface
    - ❖ **Transmission** – the transmission coefficient (T) , the refraction angle corresponds to a smooth surface
    - ❖ **Diffuse Scattering** (S) – fraction of rays distributed uniformly in all directions
    - ❖ **Absorption** - the absorption coefficient (A), which is calculated as A = 1 – R – T - S. The assigned values are used unless the incidence angle exceeds the total

internal reflection angle. In case of total internal reflection T = 0, R = 1 – A, while A is assumed to be constant in the whole angle range.

- ✓ **Heterostructure**. Special surface type used to identify the heterostructure. For a heterostructure surface, the following options are available:
  - ❖ **Conversion Layer**. If this checkbox is enabled, heterostructure is assumed to be covered with a thin phosphor layer, which is modeled as a boundary condition rather than a bulk block. For a conversion layer the user should assign its thickness (in microns), refractive index and absorption index.

- **Thermal** Properties section contains a combobox with the following surface types:
  - ✓ **Default.** The condition that is specified as a global parameter in the **Solver->Heat Transfer** tab.
  - ✓ **Heat Exchange**. The ambient temperature T_0, heat exchange coefficient **a** and emissivity coefficient of radiative heat exchange are assigned.
  - ✓ **Thermal Resistance**. Condition to be assigned to block interfaces to account for thin layers which are modeled as boundary conditions rather than bulk blocks. The value of the thermal resistance can either be assigned explicitly or calculated from the layer thickness and thermal conductivity.

The **Block Names** tab includes the list of the block names assigned by the user. By default, no names are assigned. Specification of the block names is not required for the computations. Names are only needed for the user convenience to identify the blocks.

To assign the block name, the user should do the following:

- Click inside the block. The block will be marked with hatching.
- Type the name in the **Block Name** text field
- Press **Apply** button

3.3 Grid

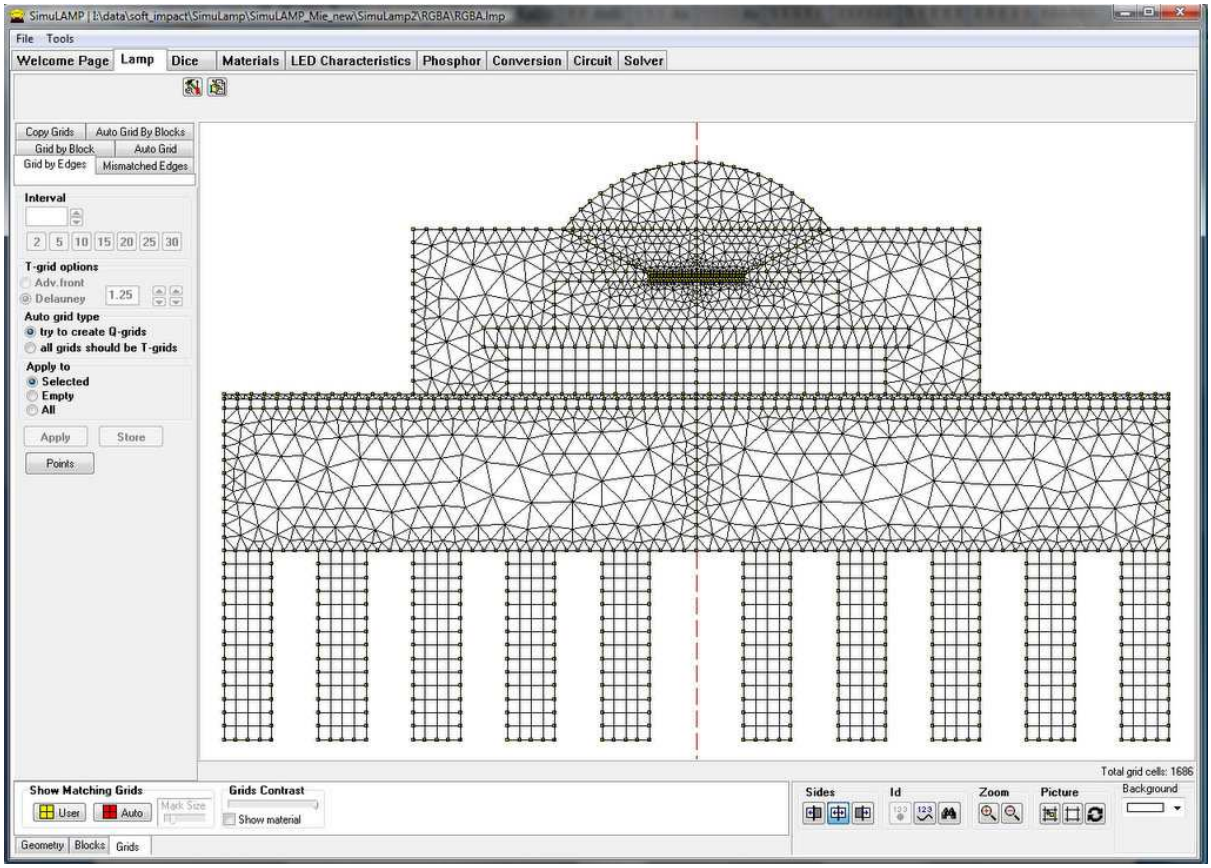


Fig. 5

The **Grid** tab window (see **Fig.5**) is designed to specify and visualize the computational mesh. 2D mesh is generated in all blocks according to the fragmentation of boundaries which is defined by the user. Boundaries to which no fragmentation is assigned are marked with grey circles. Mismatched grid in adjacent blocks is supported, so different fragmentation can be assigned to the same boundary within each of adjacent blocks.

There are several modes of the grid generation which are selected on the tab bar at the left hand side of the **Graphics Window**. The following tabs are available:

- **Auto Grid**. The fastest grid generation mode. The user should select a single boundary of the geometry and assign its fragmentation. This fragmentation will be assigned to all boundaries. Note that the fragmentation interval will be kept, while the number of intervals assigned to each boundary will depend on its length. The generated grid is matched.
- **Grid By Edges**. Fragmentation is assigned to the selected boundaries. The generated grid is matched. The user should select a single boundary one or a group of boundaries and assign

their fragmentation. In this mode, in addition to assigning the fragmentation to the selected boundaries the user can also select an option allowing assignment fragmentation to all boundaries at which no fragmentation is currently assigned or to all boundaries of the geometry.

- **Auto Grid by Blocks**. Fragmentation is assigned to the boundaries of selected blocks. The generated grid is mismatched. The user should select a single block or a group of blocks, then select one boundary of the selected blocks and assign its fragmentation (the number of edges). This fragmentation will be assigned to all boundaries of the selected blocks within these blocks.
- **Grid by Block**. Fragmentation is assigned to the boundaries of selected block. The generated grid is mismatched. The user should select a single block, then select one or several boundaries of this block and assign their fragmentation. This fragmentation will be assigned to the selected boundaries within the selected block.
- **Edges**. Using this tab one can prohibit generation of mismatched grid on some selected boundaries. The user should select the boundaries and select **Must Coincide** option for them. In this case, change of fragmentation of one side of the boundary in the **Grid by Block** or **Auto Grid by Block** mode will force its change in the adjacent block as well.
- **Copy Grid**. Using this tab one can translate meshing of one block to another block of similar geometry. If source and destination blocks have different dimensions, the user can specify whether fragmentation intervals (edge size) or the number of edges should be copied.

Most of the grid generation parameters to be specified are common for most of the above grid generation modes. The following elements are available:

- Section **Interval**. This section (see **Fig.6**) provides specification of the number of intervals. It contain the following elements:
  - ✓ **Interval** text field. If fragmentation is assigned to a group of boundaries which have different length this parameter is first converted to the grid density (edge size) using the length of the selected boundary (in case of multiple boundary selection the first selected boundary is used). Then the found grid density is assigned to all boundaries to which it should be assigned depending on the grid generation mode.
  - ✓ A group of buttons 

provides a fast assignment of some predefined Interval values.

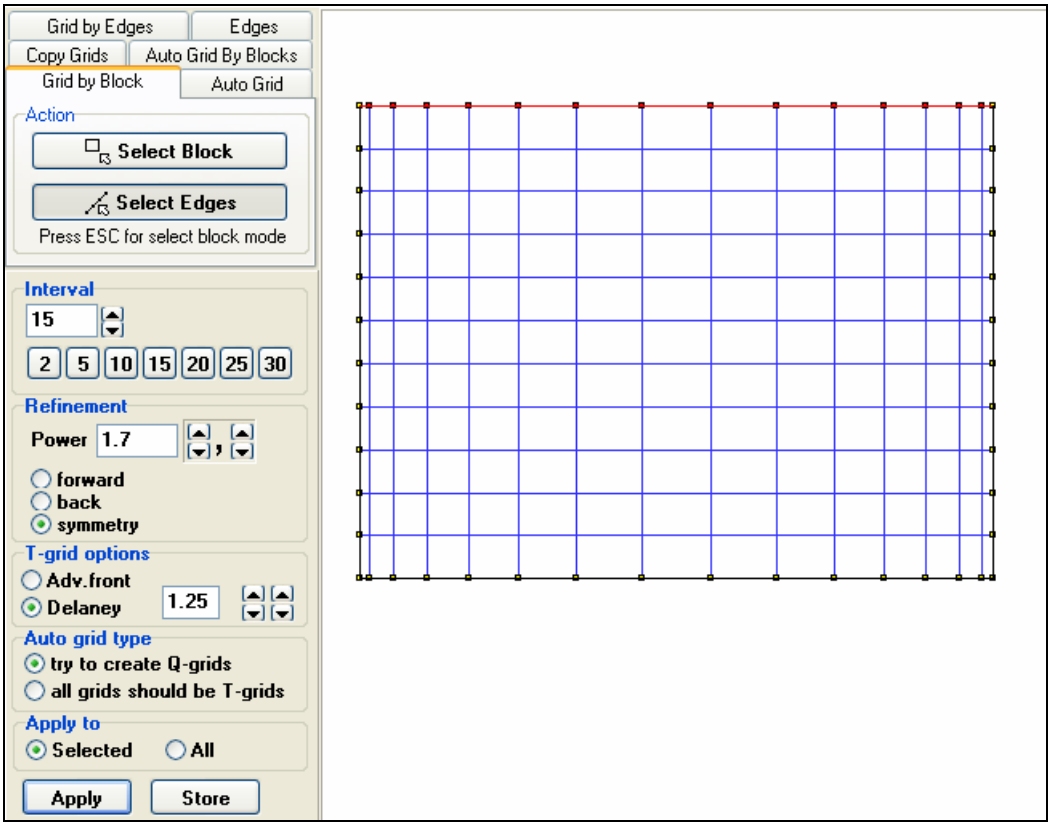


Fig. 6

- Section **Refinement** allows the user to specify a continuous increase of the internodal distance (i.e. distance between consecutive grid nodes) towards some of the edges of the selected boundary or to the center from both edges (see **Fig.6**). It is only available in the **Grid By Block** mode. It contains the following elements:
  - ✓ Text field **Power** specifying the exponent of the power function used to define the internodal distance variation. **Power = 1** corresponds to a uniform point distribution. The higher is power, the more considerable internodal distance increase is applied. **Fig.6** presents an example of grid refinement for a **Power** of **1.7**.
  - ✓ **Up** and **Down** arrows provide an easy way to increase or decrease the digits of the Power decimal representation. The left arrows govern the digit before the point, while left arrows govern the digit after the point.
  - ✓ Direction of increase of the internodal distance. The following variants are available: **Forward** (the internodal distance increases from the first ending point towards the second ending point), **Backward** (the internodal distance increases from the second ending point

towards the first ending point), and **Symmetrical** (grid refinement towards the ending points).

- Section **T-Grid Options** provides specification of triangular grid generation. The following options are available:

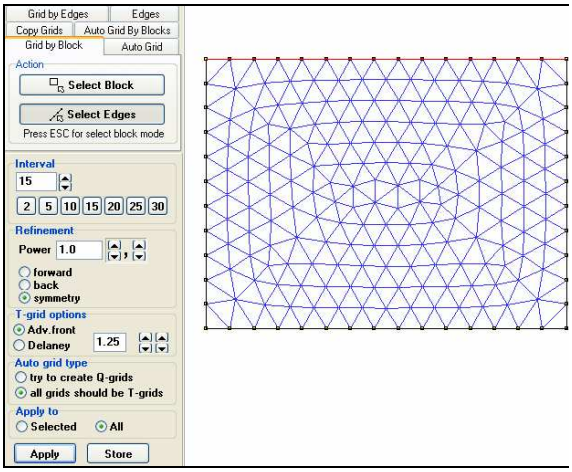


Fig. 7

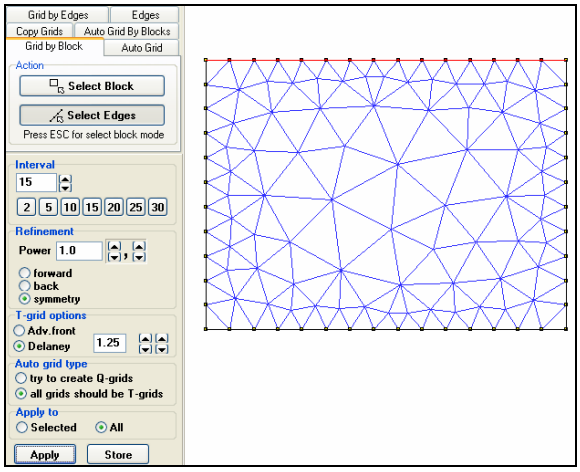


Fig. 8

  - ✓ **Adv.Front**. The advancing front method builds internal cells close to equilateral triangles whose side size is equal to the boundary edge size, using no additional parameters (see **Fig.7**).
  - ✓ **Delauney**. The Delauney method builds cells by scattering the internal points in the domain (see **Fig.8**) using the parameter **Cell Size Ratio** that is specified in the text field located at the right hand side of the of the **T-Grid Options** section. **Fig.8** presents an example of Delauney grid generation for a **Cell Size Ratio** of **1.25**.

- Section **Auto Grid Type** allows the user to specify the desired mesh type. The following options are available:
  - ✓ **Try To Create Q-Grid** (default). In this case the automatic generator tries to generate quadrangular grid in all domains. If the domain cannot be represented as a quadrangle, triangular grid is generated.
  - ✓ **All Grids Should Be T-Grids**. In this case the automatic generator generates triangular grid for all domains regardless to their shape.
- Section **Apply To** allows the user to specify the set of boundaries to which the action applies. It is available in Grid by Block (options are **Selected** and **All**) and **Grid by Edges** (options are

Selected, **Empty** and **All**). In **Auto Grid** or **Auto Grid by Blocks** action always applies to all boundaries.

- Section **Show Coincide** allows the user to mark the boundaries where mismatched grid is disabled.
- Section **Grid Contrast** allows the user to vary the brightness of the mesh lines with respect to the main geometry lines.

4 Dice Tab Window

The **Dice** tab window provides specification of positioning the led chip(s) on top of the circular axisymmetric submount. Like **Lamp**, **Dice** tab includes the same tabs of the second level: **Geometry, Blocks** and **Grid**. All dimensions within **Dice** tab are specified in microns.

4.1 Geometry

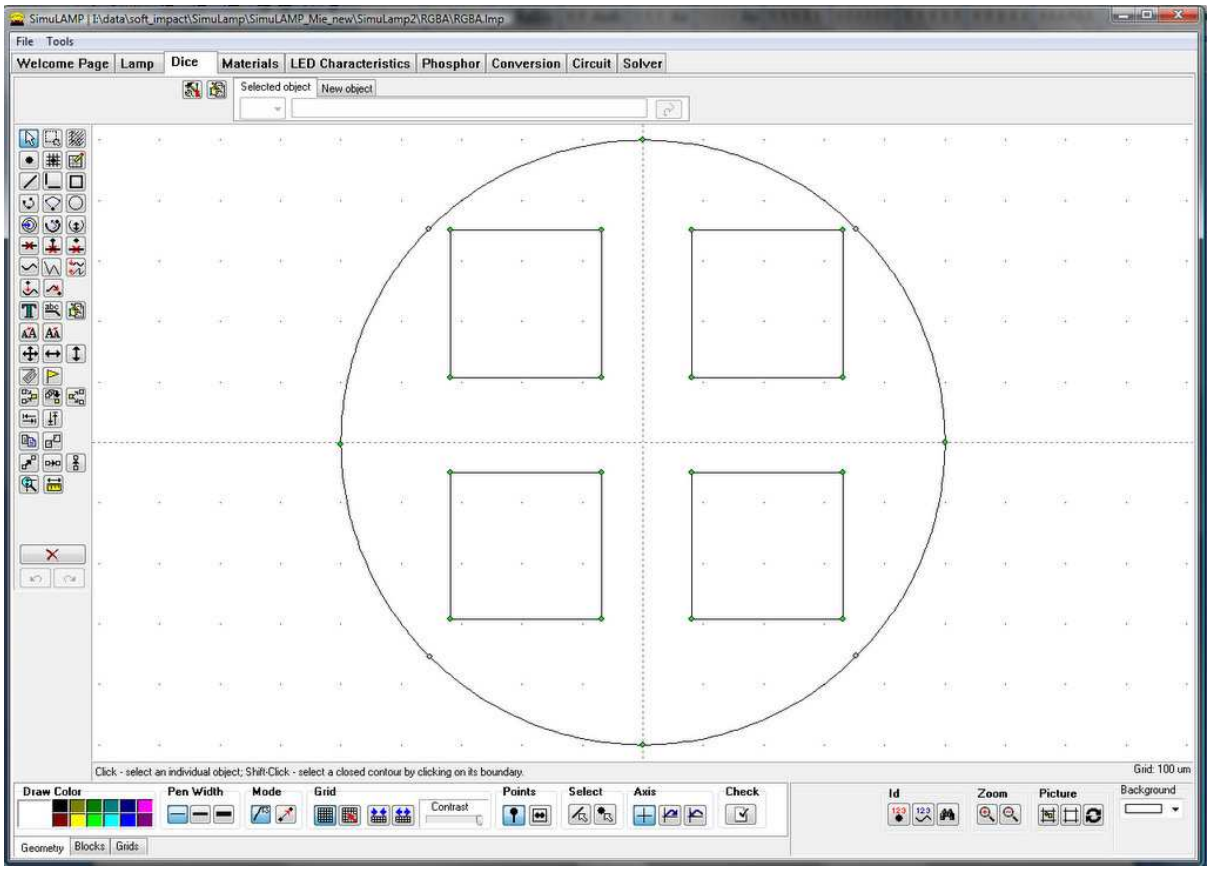


Fig. 9

Using **Geometry** tab (see **Fig.9**) one should specify LED contours inside the submount perimeter. In case of single LED lamp with a rectangular LED die matching the submount, the whole geometry to be specified within **Dice** should only include two objects: a rectangle inscribed in a circle. In actual 3D geometry, both submount and LED are quadrilateral, but due to axisymmetric specification of the lamp bulk objects and planar specification of the heterostructure, they have different shape in the top view. The submount is approximated by a cylinder whose top view is a circle, while LED is a rectangle. In case of a multi LED lamp, several LED contours should be placed inside the submount circle.

So the user should create the following elements:

- The submount contour. Though its shape and size are strictly determined by the lamp geometry specified in **Lamp** tab, it should be manually drawn by the user within the **Dice** tab as well. To provide a self-consistent specification of the whole lamp geometry, the radius of this circle should exactly match the radius of the boundary to which Heterostructure boundary condition is assigned.
- The LED contours. **Fig.9** presents three rectangular LED contours uniformly placed on the submount. The radial lines that split the rest of the submount area are only used in grid generation.

4.2 Blocks

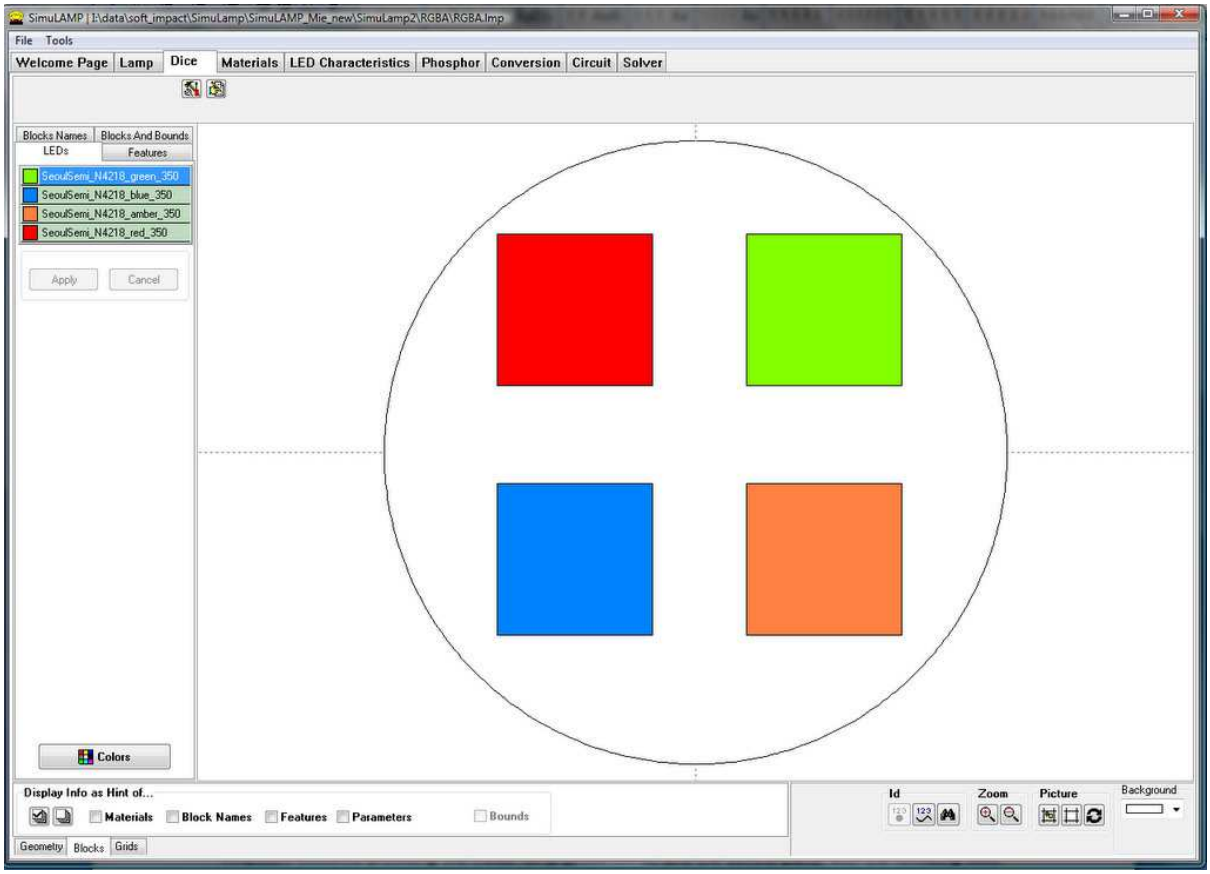


Fig.10

The **Blocks** tab window (see **Fig.10**) is designed to associate the LEDs with given I-V characteristics with geometrical blocks and assigning the current flowing through each LED block. It includes **Graphics Window** providing the visual lamp presentation and the control panel with the following tabs:

- **LEDs**
- **Block Names**

The **LEDs** tab of the control panel includes the list of the LEDs to be applied to contours, like **Materials** tab within **Lamp** tab is used to assign materials. It includes all LEDs that are specified on the **LED Characteristics** tab, which is used for assigning LED I-V and spectral characteristics.

To assign the LED, the user should do the following:

- Select an available LED by clicking on its name in the list
- Click inside the contour. The contour will be filled with the color corresponding to the selected LED.
- Press **Apply** button to accept the assignment.

Note that the same LED can be assigned to several blocks. This should be done if several diodes on the submount have identical characteristics, so that one can create a single LED in LED Characteristics tab and assign it to several blocks of the geometry.

The **Block Names** tab includes the list of the block names assigned by the user. By default, no names are assigned. Specification of the block names is used for assigning the current to dice in multichip lamp.

To assign the block name, the user should do the following:

- Click inside the block. The block will be marked with hatching.
- Type the name in the **Block Name** text field
- Press **Apply** button

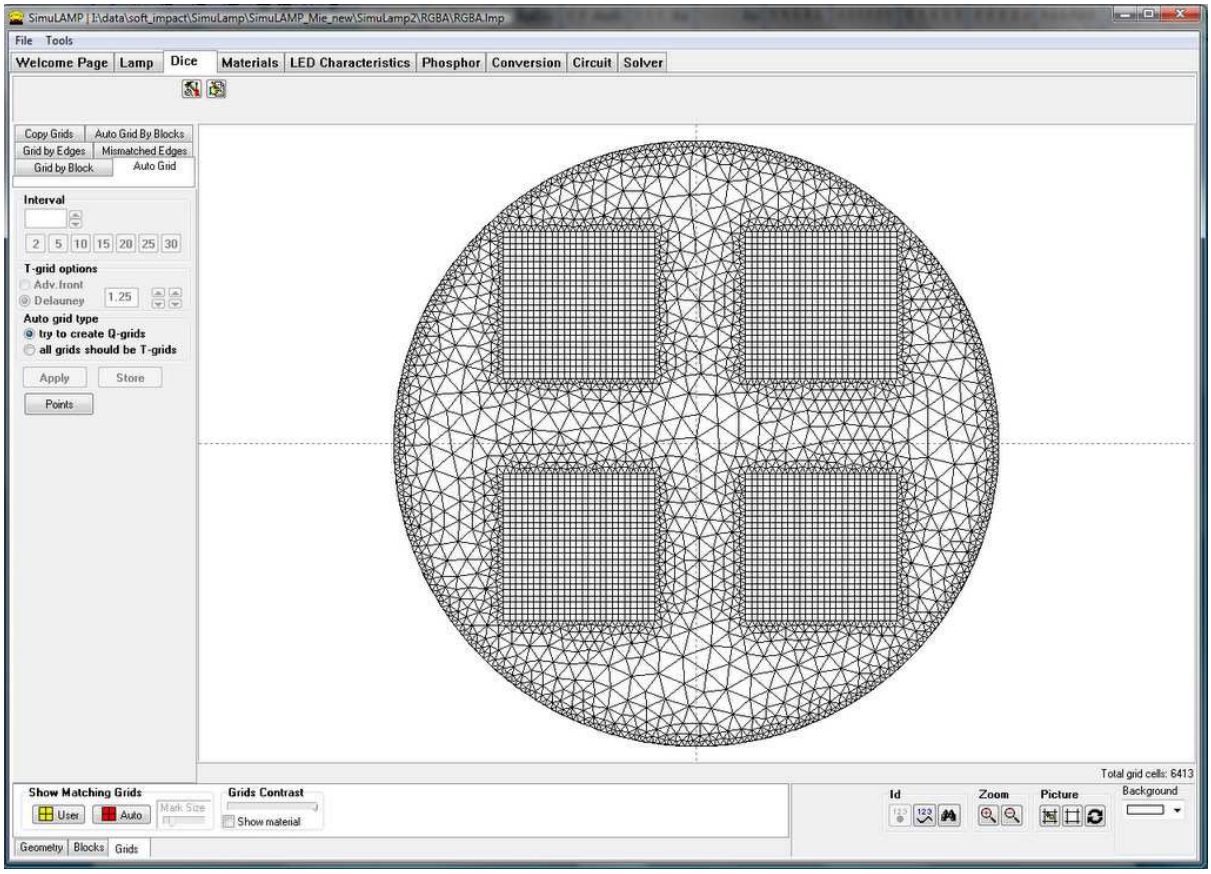


Fig. 11

The **Grid** tab window (see **Fig. 11**) is designed to specify and visualize the computational mesh. The **Grid** tab within main **Dice** tab window of **SimuLAMP** is identical to the **Grid** tab within **Lamp** tab window, which is described above.

5 Materials Tab Window

The **Materials** tab window allows the user to specify the thermal and optical properties of the materials to be assigned to the lamp elements. It contains the following elements (see **Fig.12**):

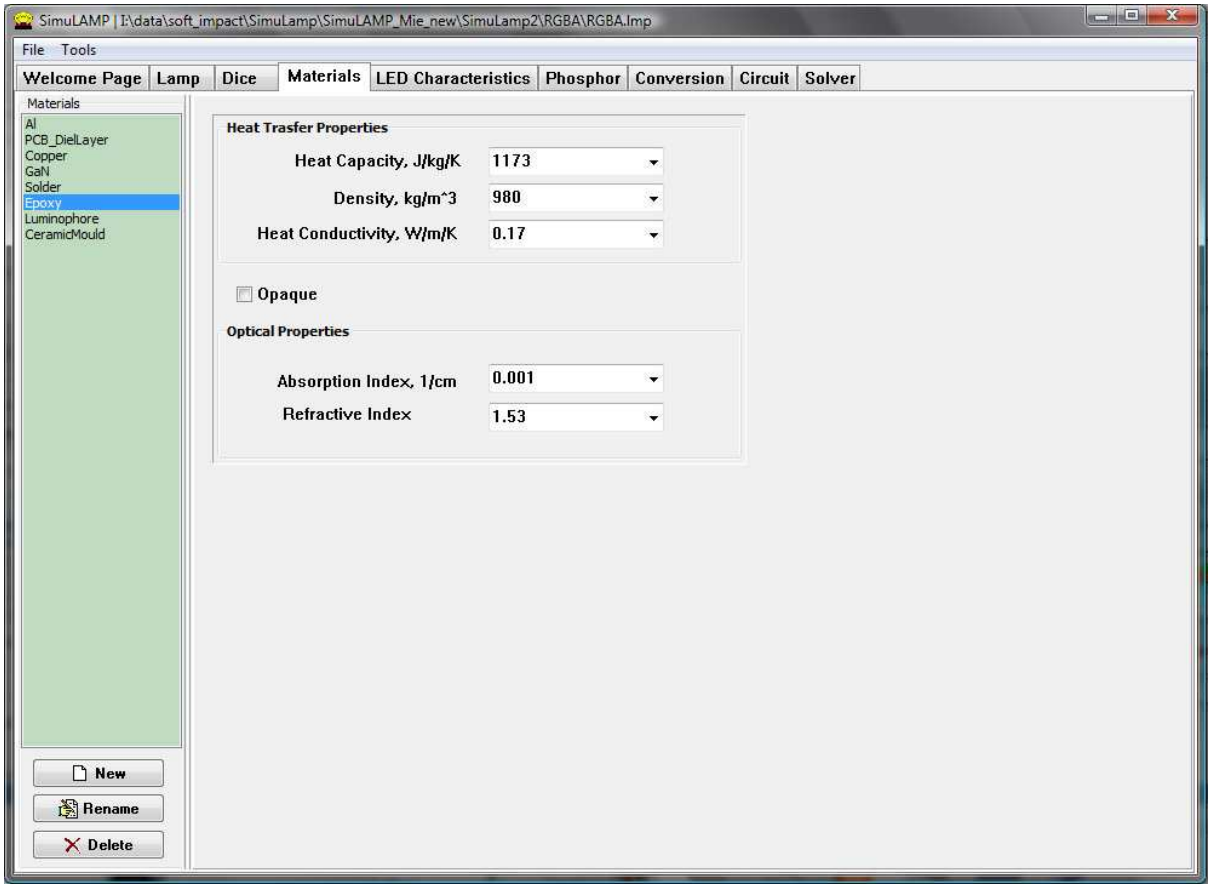


Fig. 12

- List of the materials.
- Button **New** adds a new material.
- Button **Rename** renames the selected material.
- Button **Delete** removes the selected material.
- Section **Heat Transfer Properties** contains specification of heat capacity , density and heat conductivity.
- Checkbox **Opaque** determines whether the material is semitransparent or opaque. The light propagation is traced inside semitransparent blocks only.
- Section **Optical Properties** contains specification of the absorption and refractive indices, which should be specified for all semi-transparent materials.

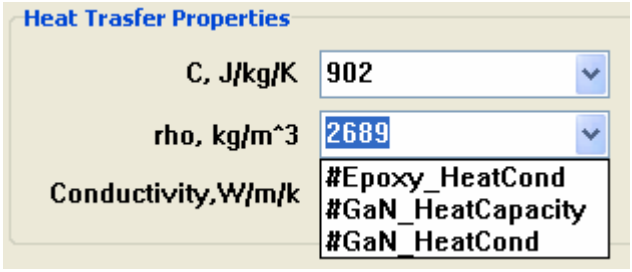


Fig. 13

All properties can be either defined as constant values or as a temperature and coordinate dependent function. Selection is made using the text fields with the property values which can either contain a numerical value or a name of the function (see **Fig.13**). The comboboxes associated with these text fields contain a list of all functions that specified in **Materials (X, Y, T)** group on **Functions** window. So the user should first define a function representing the material property in **Functions** window (see **Section 11**) and then apply it to the corresponding property in **Materials** tab.

6 LED Characteristics Tab Window

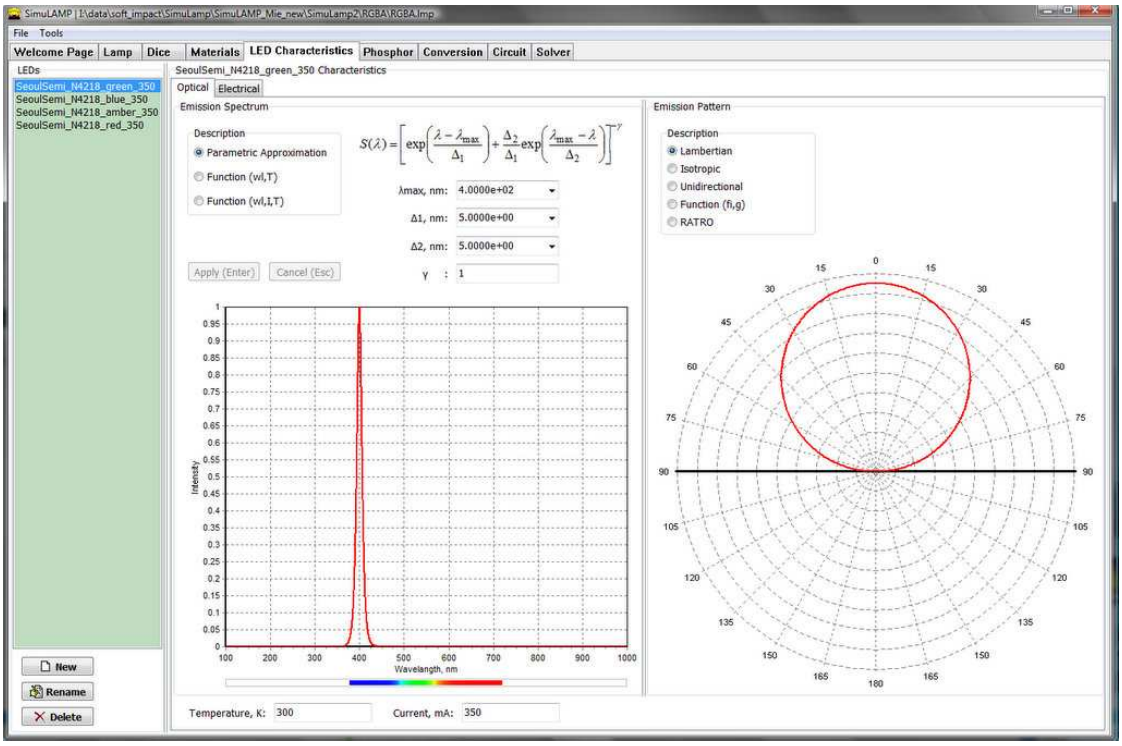


Fig. 14

The **LED Characteristics** tab window allows the user to specify the I-V and I-L characteristics, emission spectra and emission pattern of the LED chips used in the lamp. It contains the following elements (see **Fig.14**):

- List of the LEDs.
- Button **New** adds a new LED with no parameters specified.
- Button **Rename** renames the selected material.
- Button **Delete** removes the selected material.
- Tab **Optical** contains specification of the LED optical characteristics (emission spectrum and pattern).
- Tab **Electrical** contains specification of the LED I-V and I-L characteristics.

Note that there is another possibility to edit list of LEDs, through the context menu which can be accessed by mouse right click on the list of the LEDs (**Fig. 15**).

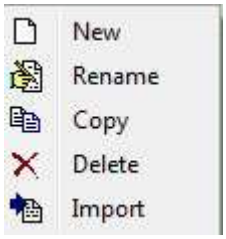


Fig. 15

- Button **Copy** adds a new LED with parameters copied from the selected one.
- Button **Import** imports the list of LEDs from a SimuLAMP project file (*.lmp) created before.

In **Optical->Emission Spectrum** section (**Fig.14**) the user should assign the LED emission spectrum. Two options are available:

- **Parametric Approximation**. The user should assign parameters Peak Wavelength ($\lambda_{\max}$), Spectrum Width Parameters (Δ_1, Δ_2) and γ parameter of the following approximation:

$$f(\lambda, I, T) = \frac{1}{\left[\exp\left(\frac{\lambda - \lambda_{max}}{\Delta_1}\right) + \frac{\Delta_2}{\Delta_1}\exp\left(\frac{\lambda_{max} - \lambda}{\Delta_2}\right)\right]^{\gamma}} \quad ,$$

  which works well for approximation of the emission spectrum of the manufactured LEDs.
  Note that all the parameters except γ can be set temperature and current dependent using the group of functions **Characteristic (I; T)** .
- **Function**. The user should select the name of the function specified in **Functions** window. There are two possibilities to specify spectra: wavelength and temperature dependence or wavelength, temperature and current dependence. In this case the user should first define a function representing the LED spectrum in **Functions** window and then apply it to the LED in **LED Characteristic->Optical** tab.

The selected spectrum is plotted on the chart. The values of the temperature and current for which the spectrum is plotted should be specified in the fields below the chart.

In **Optical->Emission Pattern** section (**Fig.14**) the user should assign the LED emission pattern. Five options are available:

- **Lambertian**. The far field emission pattern is described by the Lambert's law.
- **Isotropic**. The far field emission pattern is isotropic.
- **Unidirectional**. Rays propagate in one direction normal to the chip surface.
- **Function**. The user should select the name of the function specified in **Functions** window in the group **Diagram(fi; g)**. In this case the user should first define a function representing the LED emission pattern in **Functions** window and then apply it to the LED in **LED Characteristic->Optical->Emission Pattern** section.
- **RATRO**. The user should import the near field emission pattern from RATRO ray file (*.bin).

The selected pattern is plotted on the polar chart below.

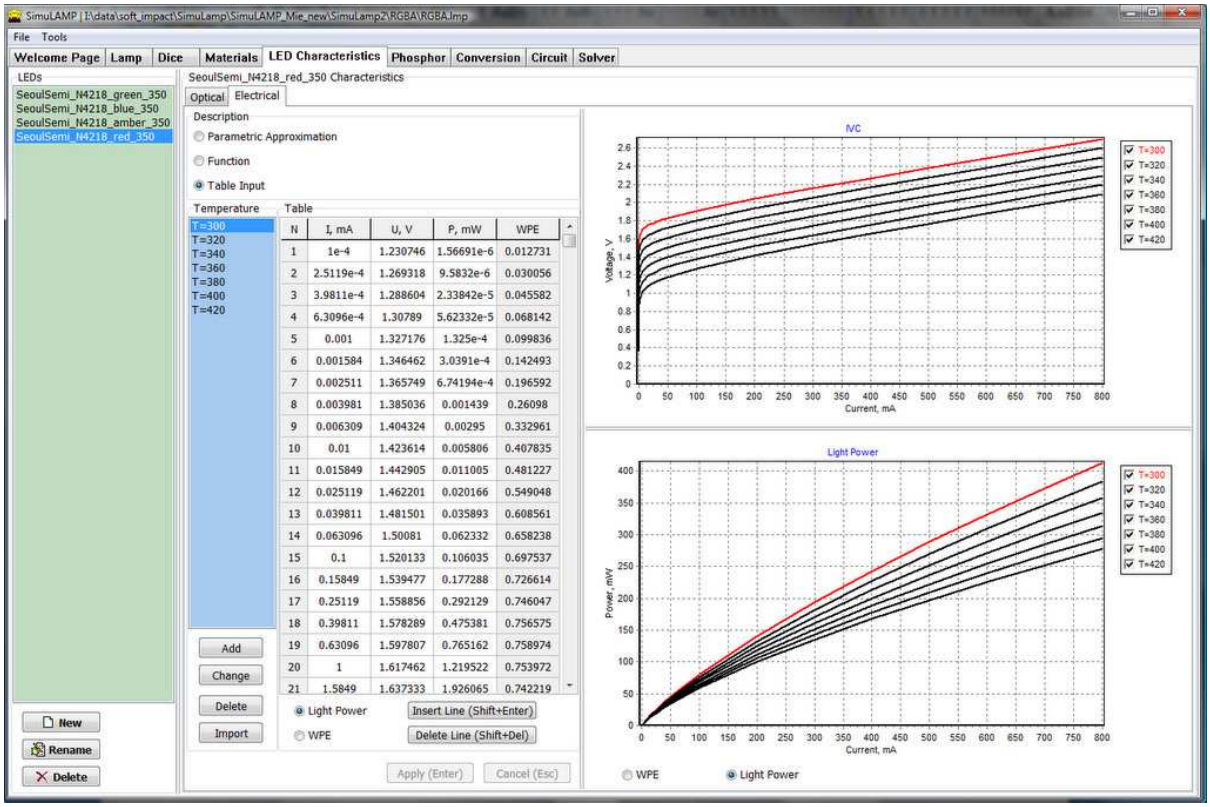


Fig. 16

On **Electrical** tab window (**Fig.16**) the user should assign the LED I-V and L-I characteristics. There are three options: by the table input, by the function of temperature and current or by the parametric approximation with temperature dependent parameters.

Table input section (see **Fig.16**) contains the following elements:

- List of the temperatures.
- Button **Add** adds a new temperature value for which characteristics will be assigned.
- Button **Delete** removes the characteristics for the temperature that is currently selected in the list.
- Button **Change** changes the value of temperature that is currently selected in the list.
- Button **Import** loads an ASCII file with I-V characteristics for a set of temperatures, which is generated by **SpeCLED 2008** software in **Temperature Series** mode.
- A table containing the **Current** (mA), **Voltage** (V), **Light Power** (mW) and **WPE** at the given temperature.
- Section with radio buttons **Light Power** and **WPE** specifying which of these parameters is calculated from the three parameters and cannot be changed by the user.
- Button **Insert Line** adds a new line in the table.
- Button **Delete Line** removes the selected line from the table.

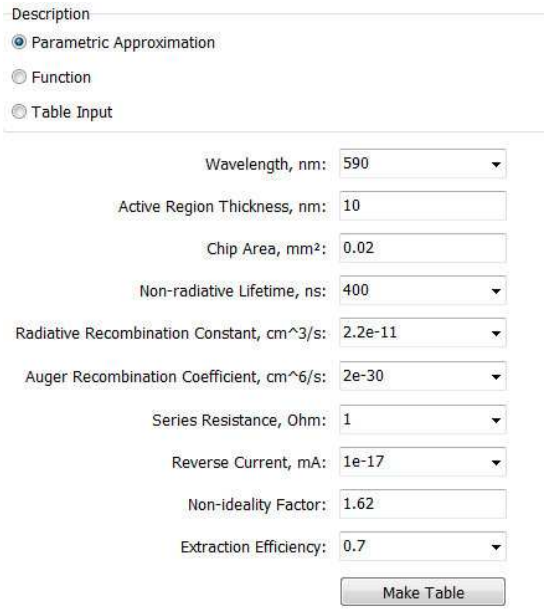


Fig. 17

Parametric Approximation section (see **Fig.17**) contains the following elements:

- List of fields with chip parameters. Some of them can be set temperature dependent.
- Button **Make Table** creates the table with user defined list of temperatures and currents. The values in the table are calculated using the list of chip parameters within simple approach (see the **Physical Summary** document for details) and can be later edited in the section **Table Input.** Note that **Apply** button creates the same table but the temperature and current range is program defined (which is [0, 2000] mA for the current and [300, 750] K with the interval of 50 K for the temperature).

In **Function** section the user can specify I-V and I-L characteristics using pre-defined functions of current and temperature. The user should select the name of the function specified in **Functions** window. In this case the user should first define a function representing the I-V and I-L characteristics in **Functions** window and then apply it to the LED in **LED Characteristic->Electrical** tab. Instead of using I-L characteristics the user can assign I-WPE characteristics. The choice is controlled by radio buttons **Light Power** and **WPE**.

The specified characteristics are plotted on the two charts at the right side for the specified list of temperatures (the user can control the visibility of characteristic by clicking on the checkbox with the respective temperature). On the bottom chart the user can switch between I-L and I-WPE characteristics.

7 Phosphor Tab Window

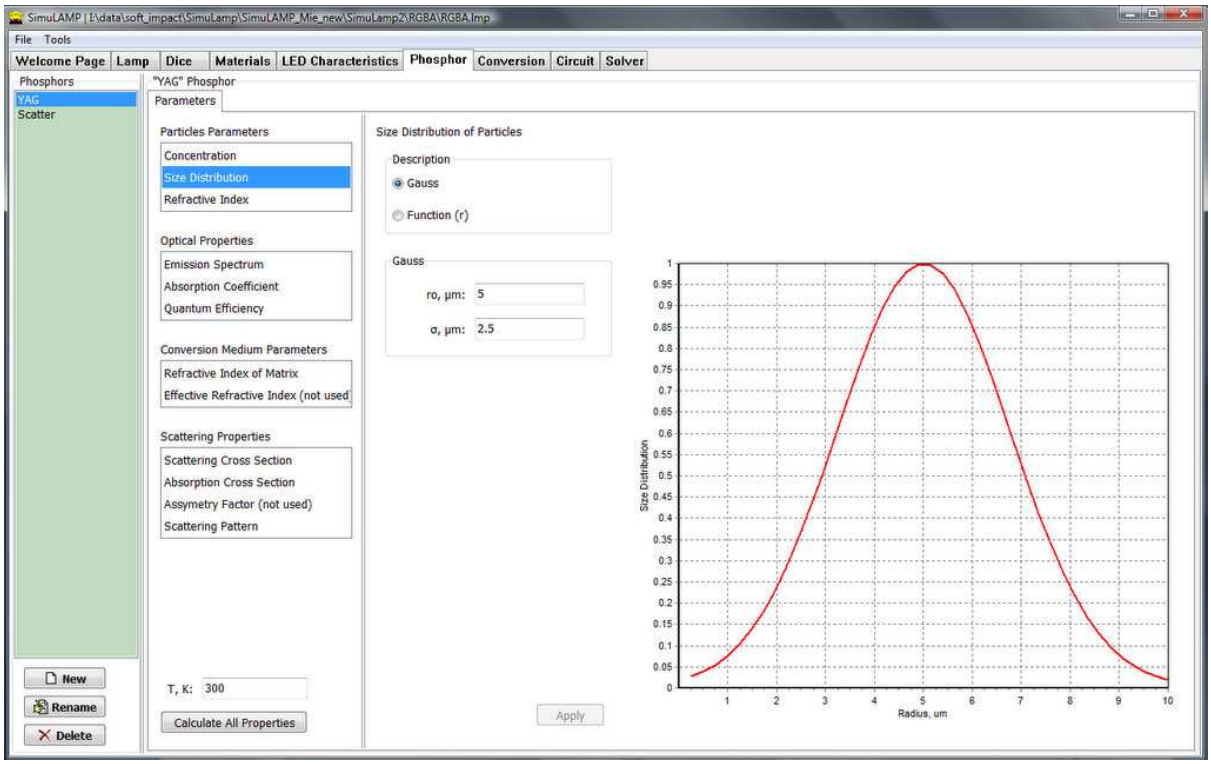


Fig. 18

The **Phosphor** tab window is designed to specify the parameters of the phosphors used in the lamp.

The **Phosphor** tab window contains the following elements (see **Fig.18**):

- List of the phosphors specified in the project. A project can include any number of phosphors with specified characteristics. The conversion medium used in the simulated lamp can include one phosphor or a mixture of different phosphors which should be checked in the list by the user.
- Button **New** adds a new phosphor
- Button **Rename** changes the phosphor name
- Button **Delete** removes the selected phosphor
- Section **Parameters** contains the following sub-tabs:
  - ✓ **Concentration** is a parameter of the phosphor particles concentration in the conversion medium. There is an additional read-only field with the calculated value of the phosphor particles volume fraction in medium.

✓ **Size Distribution** is a function of distribution of the phosphor particles by radius. The user should choose between Gauss distribution (with two user defined parameters: the center of distribution r_0 and the dispersion σ) and function from the group **Distribution(r)**.

✓ **Refractive Index** is the refractive index of particles spectrum. The user should choose between constant value and function of wavelength and temperature.

✓ **Emission Spectrum** is the emission spectrum of phosphor. It can be specified either by parametric approximation or function of wavelength and temperature. On the selection of the parametric approximation the user should assign parameters Effective Wavelength λ^*, Spectrum Width Parameters (Δ_1, Δ_2) and α Parameter of the following:

$$f(\lambda,T)=\frac{1}{\exp\left(\frac{\lambda-\lambda^*(T)}{\Delta_1(T)}\right)^{\alpha}+\exp\left(\frac{\lambda^*(T)-\lambda}{\Delta_2(T)}\right)}.$$

α Parameter is usually 2 for the emission spectrums of the manufactured phosphors. Note that all the parameters except α can be set either constant or temperature dependent. The specified spectrum is normalized by unity.

✓ **Absorption Coefficient** is the absorption coefficient spectrum of phosphor (**Fig.19**). It can be specified either by parametric approximation or function of wavelength and temperature. On the selection of the parametric approximation the user should assign parameters Effective Wavelength λ^*, Spectrum Width Parameters (Δ_1, Δ_2) and α Parameter of the following approximation of the excitation spectrum:

$$f(\lambda,T)=\frac{1}{\exp\left(\frac{\lambda-\lambda^*(T)}{\Delta_1(T)}\right)^{\alpha}+\exp\left(\frac{\lambda^*(T)-\lambda}{\Delta_2(T)}\right)}.$$

Parameter α is usually about 4 for the excitation spectrums of the manufactured phosphors. Note that all the parameters except α can be set either constant or temperature dependent.

The specified excitation spectrum is normalized by unity and to transform it into the absorption coefficient spectrum the user should specify the value of the absorption coefficient at the specific wavelength.

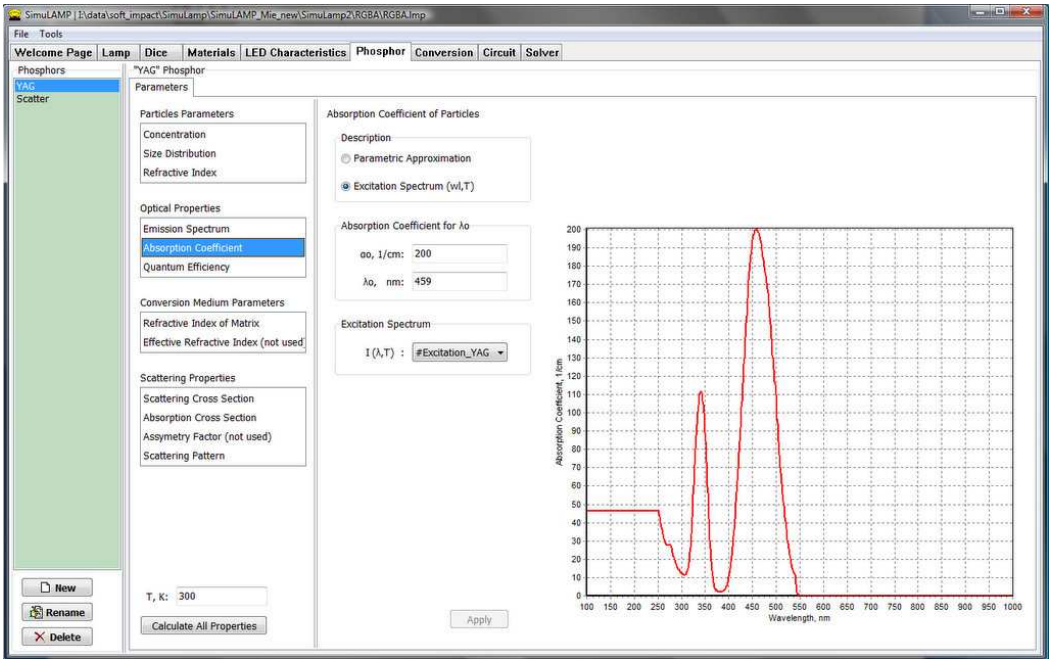


Fig.19

✓ **Quantum Efficiency** is the temperature dependence of the internal quantum efficiency of phosphor. There are three ways to specify it: constant value at the whole temperature range, parametric approximation and user-defined function of temperature. On the selection of the parametric approximation the user should assign parameters Effective Quantum Efficiency η_0 and Effective Temperature Parameters (T_1, T_2) of the following:

$$f(T) = \frac{\eta_0}{1 + \exp\left(\frac{T - T_1}{T_2}\right)} .$$

On the selection of function the user should first define a function representing the quantum efficiency in **Functions** window (**Temperature (T)** group) and then apply it to the phosphor in **Phosphor** tab.

✓ **Refractive Index of Matrix** is the refractive index of conversion medium matrix. The user should choose between constant value and function of wavelength and temperature.

✓ **Effective Refractive Index** is the effective refractive index of conversion medium in the Rayleigh model. The user should choose between Bruggeman Model and Monecke Model (for more details see the **Physical Summary** document).

✓ **Scattering Cross Section and Absorption Cross Section** are the scattering and absorption cross section spectrums of the phosphor particles. The user can either specify it manually by choosing **Function** radio button and assigning the function of wavelength and temperature, or let the program calculate it within two available models of scattering: Mie model and Rayleigh model.

✓ **Asymmetry factor** is the parameter used for specifying the scattering pattern (see below). It can be either set manually by function of wavelength and temperature or calculated within Mie theory.

✓ **Scattering Pattern** is the scattering pattern of the phosphor particles for different values of wavelength (**Fig. 20**). There are five options to specify it: function of polar angle and asymmetry factor (specified above), function of polar angle and wavelength, Henney-Greenstein approximation using specified above asymmetry factor, and calculations within Mie or Rayleigh models.

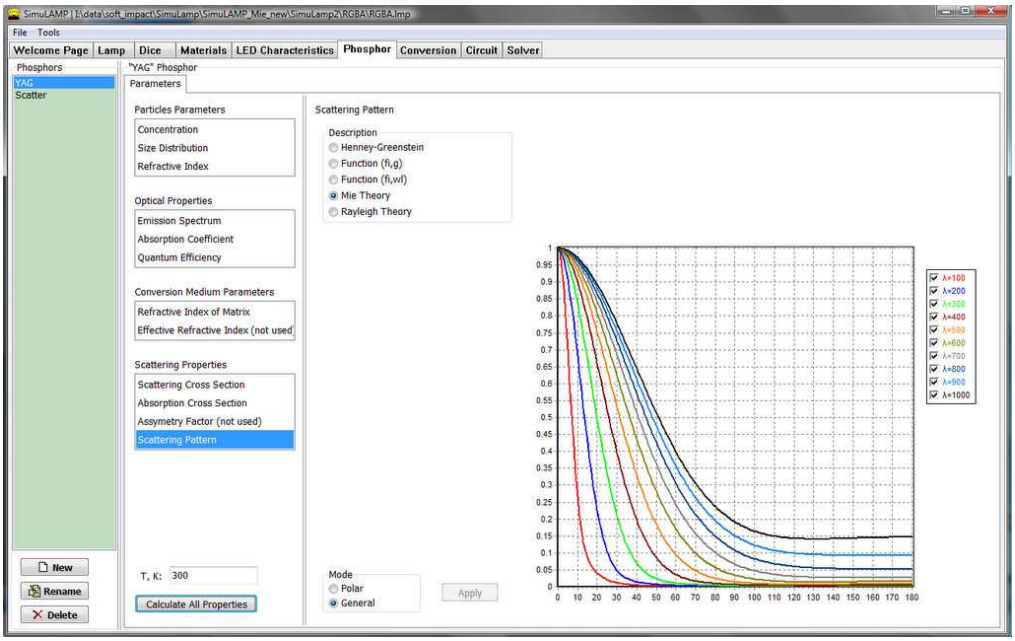


Fig. 20

➢ **Calculate All Properties** button runs the calculation of all parameters within **Scattering Properties** group (which are not set manually) at the above specified temperature value. All the calculated parameters are shown on the charts at the right section. While right clicking on the chart the user can change some visual settings and save the chart if necessary. Note that **Apply** button at the right section is used only for visualizing manually set characteristics, other characteristics should be calculated first.

8 Conversion Tab Window

The **Conversion Tab Window** allows the user to specify what phosphors are included into the conversion medium, as well as estimate the light conversion by given absorption medium by tracing the light beam penetrating test semitransparent medium between two parallel surfaces.

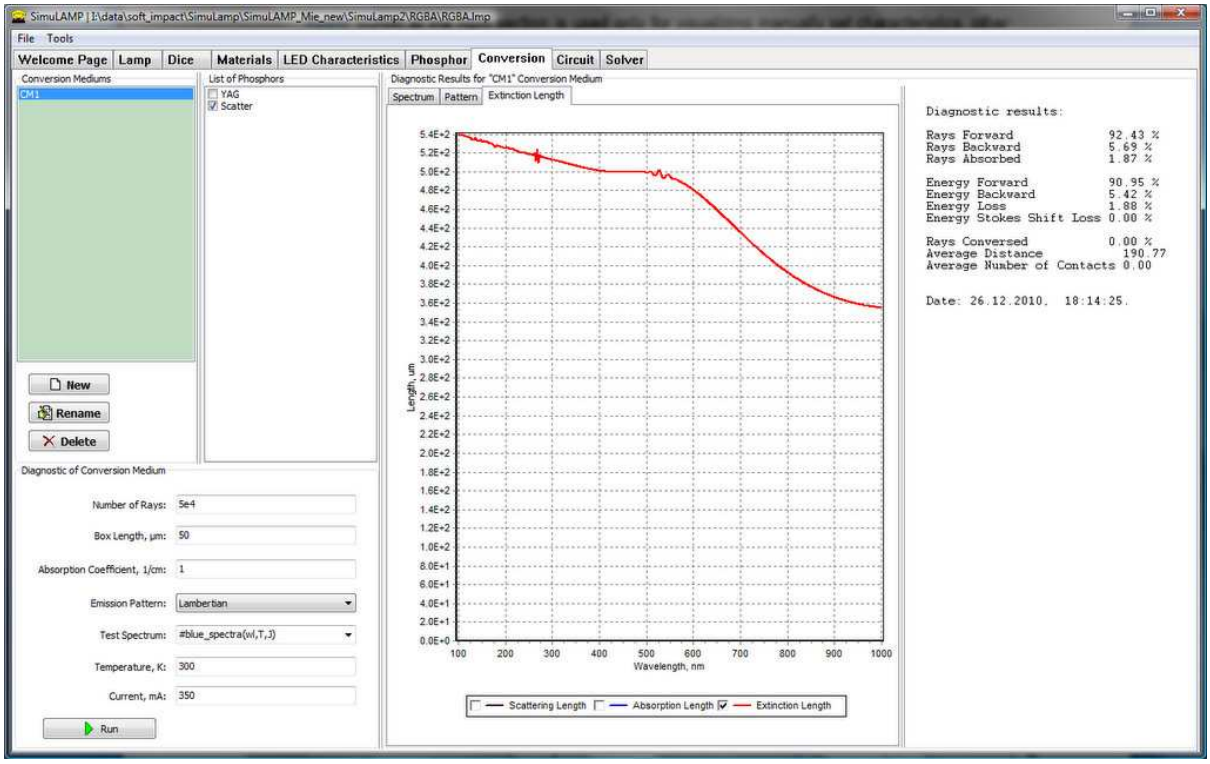


Fig. 21

It contains the following elements (see **Fig.21**):

- List of the conversion mediums specified in the project. The conversion medium used in the simulated lamp can include one phosphor or a mixture of different phosphors which should be checked in the list by the user. Created conversion medium is then available on the **Lamp->Blocks->Blocks and Bounds** tab and can be applied to a selected block.
- Button **New** adds a new conversion medium
- Button **Rename** changes the conversion medium name
- Button **Delete** removes the selected conversion medium
- List of phosphors specified on the **Phosphor** tab window

- **Diagnostic of Conversion Medium** section with user defined parameters which allows to estimate the light conversion by given absorption medium by tracing the light beam penetrating test semitransparent medium between two parallel surfaces.

The user should specify the following parameters:

- **Number of rays** – the number of traced rays, which governs the calculation accuracy.
- **Box Length** – thickness of the test medium filled with phosphor.
- **Absorption Coefficient** - absorption coefficient of the test medium filled with phosphor.
- **Temperature** – operating temperature.
- **Current** – operating chip current.
- **Test Spectrum** – spectrum of the incident light beam.
- **Emission Distribution** – angular distribution of the incident rays.

The results of the computation include:

- Fractions of energy transferred forward through the phosphor, reflected backward and lost, respectively. Energy is proportional to the number of rays multiplied by the photon energy at the frequency of the given ray.
- Energy loss due to the Stokes shift in the phosphor.
- Fractions of rays transferred forward through the phosphor, reflected backward and lost, respectively (they differ from the fractions of energy due to the Stokes shift)
- Fraction of conversed rays, i.e. rays adsorbed and reemitted by the phosphor.
- Fraction of rays propagated through the phosphor without a single contact with the phosphor particles.
- Average distance covered by an individual ray before adsorption or leaving the phosphor.
- Average number or contacts of an individual ray with the phosphor particles before adsorption or leaving the phosphor.
- The charts with visual information:
  - ✓ **Spectrum** tab contains the spectrums of forward and backward radiation.
  - ✓ **Pattern** tab contains angular emission distribution of the test medium.
  - ✓ **Extinction Length** tab contains the spectrums of the average scattering, absorption and extinction length of the test conversion medium particles.

9 Circuit Tab Window

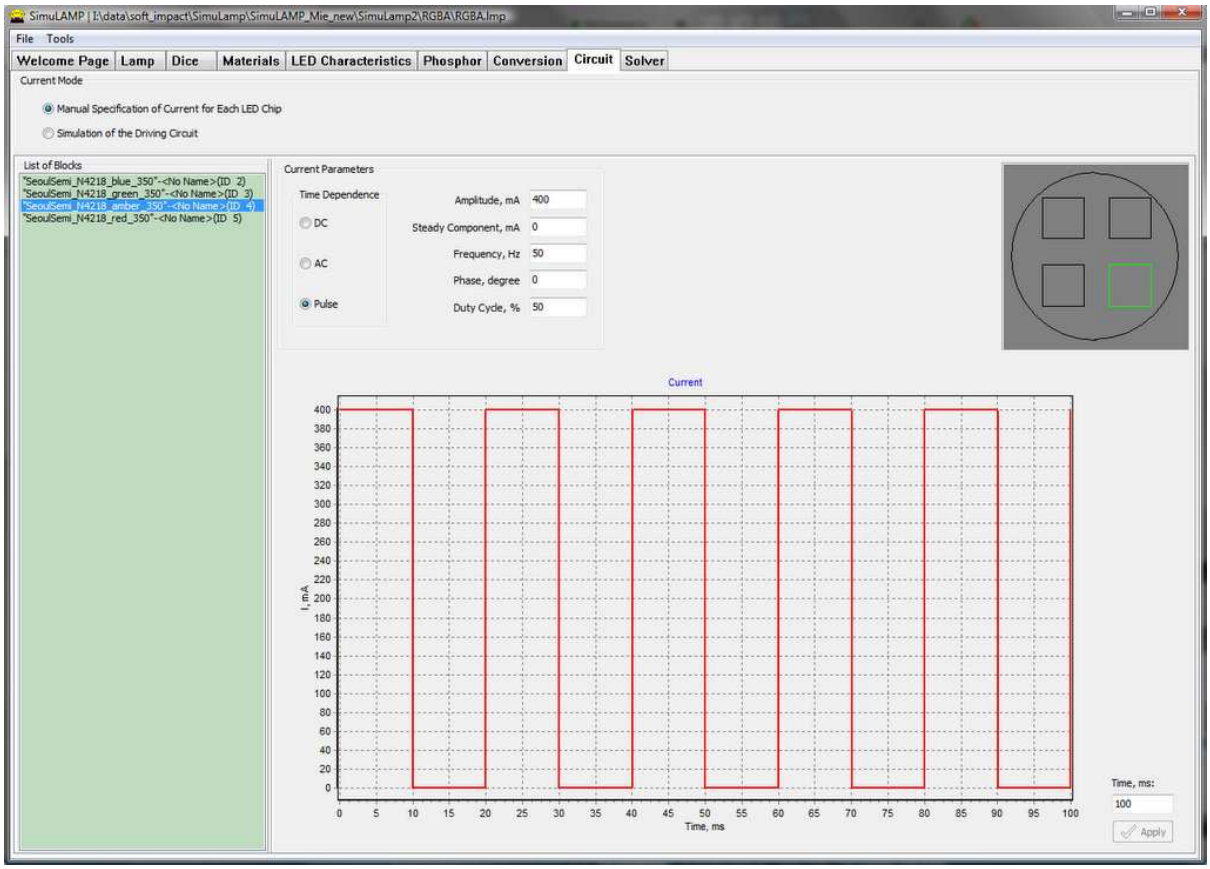


Fig. 22

The **Circuit** tab window provides specification of the current flowing through the LED dice. Two modes of the current specification are available:

- **Manual Specification of Current for Each LED Chip** – values of current in each LED are explicitly assigned by the user
- **Simulation of the Driving Circuit** – the user specifies the circuit design and assigns the total voltage supplied to the LED array.

9.1 Manual Specification of Current

In this mode the Circuit tab window contains the following elements (see **Fig.22**):

- List of LED dice. The number of dice in the list corresponds to the number of blocks assigned in **Dice** tab. Each die is identified by the name of LED chip assigned to this block and the block name.
- Image of the submount layout highlighting the selected die.
- Group of the LED operation modes. Three options are available:

✓ **DC**. In this mode, the user should assign the current magnitude only.

✓ **AC**. In this mode, the user should assign the **Amplitude**, **Frequency** and **Phase**. In addition, non-zero **Steady Component** of the current can be assigned. **Rectifier** flag activates the mode of LED operation on both half-periods of the oscillations.

✓ **Pulse**. In this mode, the user should assign the **Amplitude**, **Steady Component**, **Frequency**, **Phase** and the **Duty Cycle** defined as $\gamma = \frac{\tau_1}{T}$ (see **Fig.23**).

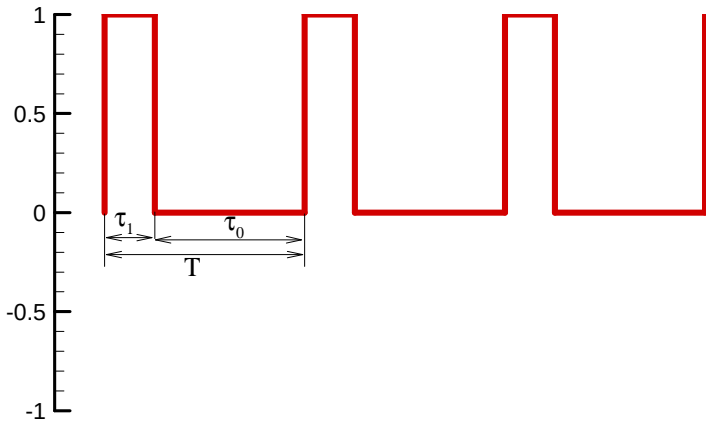


Fig. 23

9.2 Simulation of the Driving Circuit

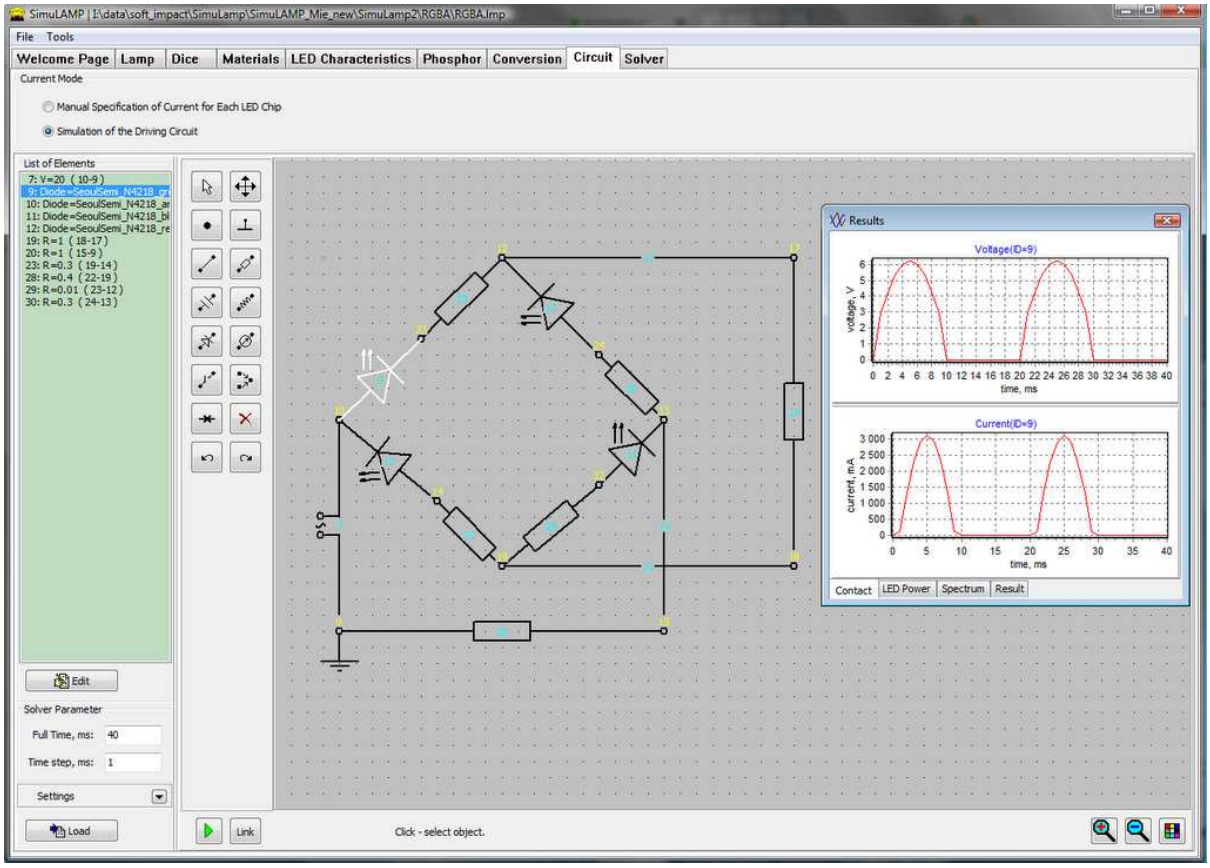


Fig.24

In this mode the user specifies the circuit design and assigns the total voltage supplied to the LED array (**Fig.24**).

A circuit can include several types of elements which are characterized by the following parameters:

- **Input Voltage**
  - ✓ Selection of DC, AC or Pulse mode. In DC mode the user assigns the voltage magnitude. In AC mode, the user assigns the voltage amplitude and frequency. In Pulse mode, the user assigns the voltage amplitude, frequency and the pulse ratio $R = \frac{T}{\tau_1}$ (see **Fig.23**).
- **Resistance**
  - ✓ Resistance (Ω)
- **Capacity**
  - ✓ Capacity (F)

- **Inductance**
  - ✓ Inductance (H)
- **Diode**
  - ✓ **Diode** - selection of the LED name from the list containing all LEDs specified in **LED Characteristics** tab
  - ✓ **Repeat** – the number of identical LED dice in series connection.
  - ✓ **Light Emitting Diode** – flag governing the type of the diode.
  - ✓ **Direction** – diode orientation in the circuit.
- **Additional Current Source**
  - ✓ Voltage
  - ✓ Resistance
  - ✓ Current direction
- **Lock**
  - ✓ Initial State (locked/unlocked)
  - ✓ Time instant of the state changing
- **Ground**
  - ✓ None

The **Circuit** tab contains the following elements:

- List of Elements. It always contains the input voltage. The other elements are added by the user.
- Button **Edit** opens **Contact** dialog window that specifies the parameters of the selected element.
- Section **Solver Parameters**.
  - ✓ **Full Time** – duration of the time interval to be modeled
  - ✓ **Time Step** – time step of the numerical solution of the unsteady problem, which affects the accuracy of the computation.
- Section **Settings** containing parameters of the circuit visualization.
- Tool bar allowing the user to select and existing element or to add a new one.
- The graphics window providing the circuit visualization
- Button **Run** providing the computation of the circuit design at a given temperature.

- Button **Link** allowing the user to set association between the diodes in the circuit and the LED blocks specified in **Dice** tab.

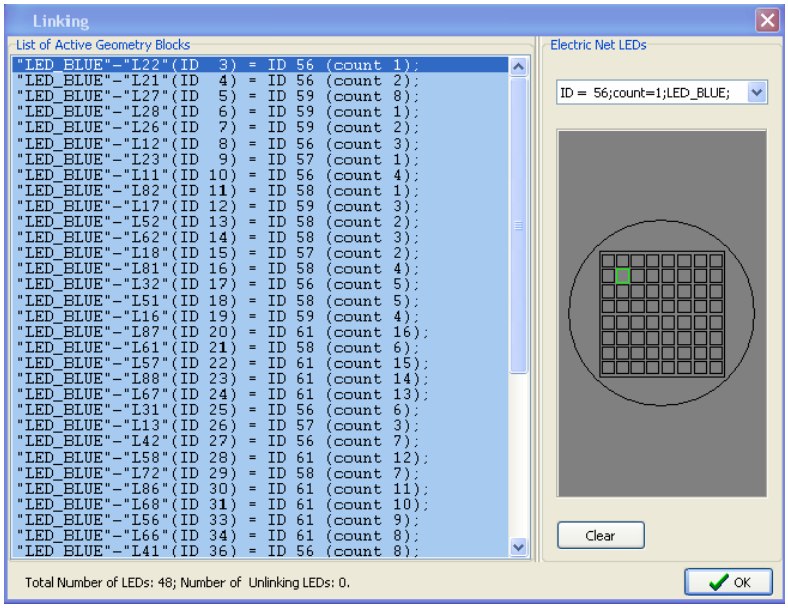


Fig. 25

It opens the **Linking** window which contains the following items (see **Fig.25**):

- List of LED dice specified on **Dice** tab. Each die is identified by the name of LED chip assigned to this block and the block name.
- Image of the submount layout highlighting the selected die.
- A combobox providing selection of the diode on the circuit. At each moment the list of circuit diodes contains all available relevant diodes, i.e. diodes which have the same LED type as the selected block and are not yet assigned to any other block.
- Button **Clear** resets the linking association.

Visualization of the isothermal circuit computations started by Run button is available on finishing the computation in **Results** window that is opened by clicking on a circuit element on the graphics window.

- **Contact**. Voltage and current in the given element as a function of time (see **Fig.26**).

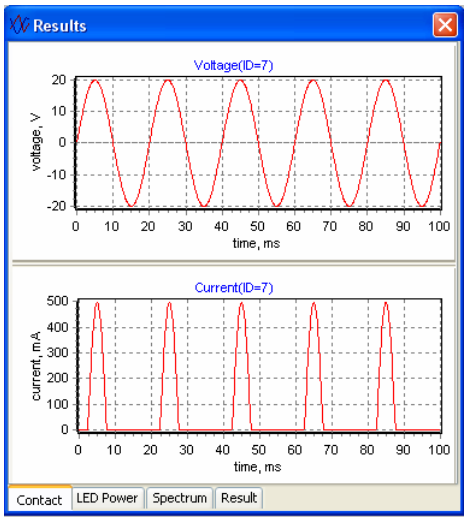


Fig. 26

- **LED Power**. The power generated in all LEDs (black line) and in the selected LED (red line shown if the selected element is an LED) as a function of time (see **Fig.27**).

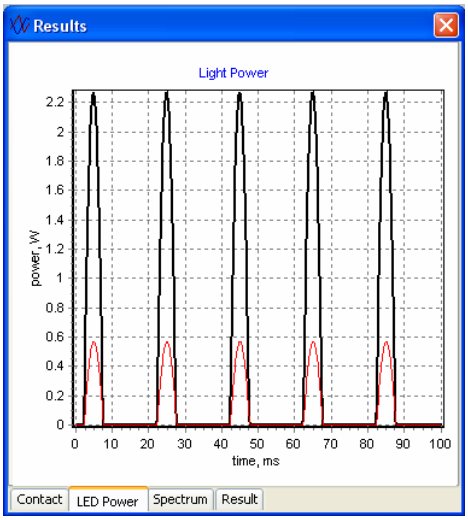


Fig. 27

- **Spectrum**. The total lamp emission spectrum (see **Fig.28**).

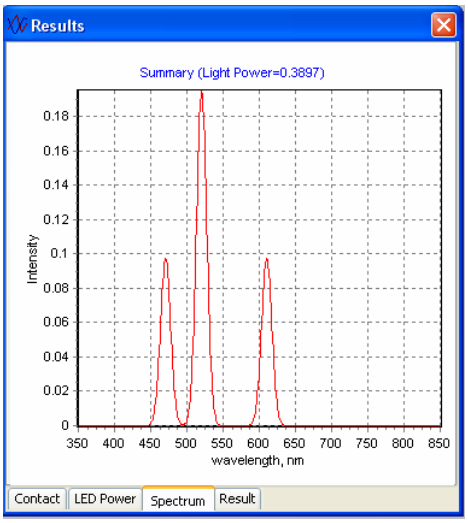


Fig. 28

- **Result**. Summary on the average power released in each element.

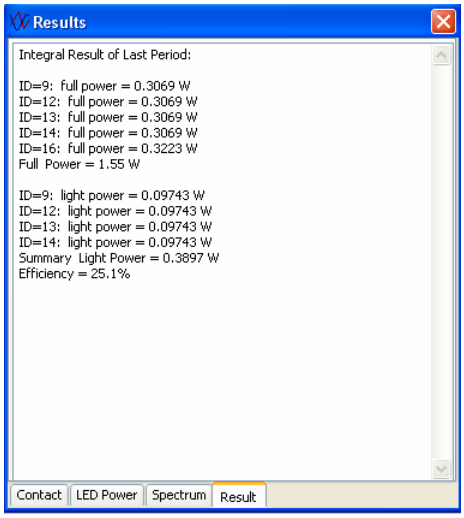


Fig. 29

10 Solver Tab Window

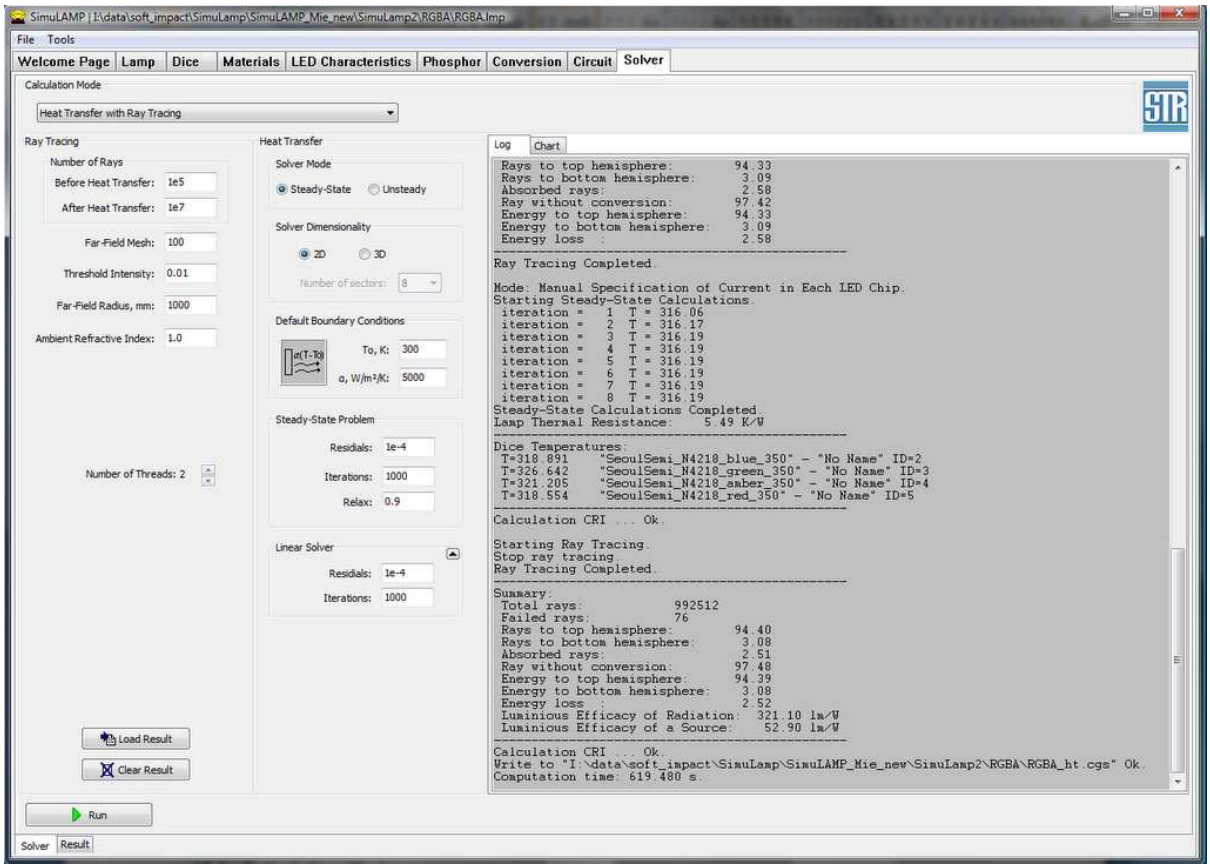


Fig. 30

10.1 Calculation Modes

Solver tab window provides specification of the computational settings, running the solver, and visualization of the results.

The **Calculation Mode** section on top of the window contains a combobox providing selection of the three computation modes:

- **Ray Tracing** – an isothermal light emission problem.
- **Heat Transfer** – a heat transfer problem ignoring the heat release in the light absorption.
- **Heat Transfer with Ray Tracing** – a coupled heat transfer and light extraction problem which is made in three stages: First preliminary isothermal ray tracing problem is run, then heat transfer accounting for the light absorption is simulated, and finally the ray tracing at the calculated temperature distribution is run.

Visualization of ray tracing computation is available in a built-in visualization tool on **Result** tab of Solver tab window (see section 10.4 Visualization below). Besides, results of the computations are

stored in a ProjectName.cgs file to be viewed in external visualization tool **SimuLEDView** (see **Appendix 1: Visualization of SimuLAMP Results in SimuLEDView** for details). The far-field light distribution can be viewed by both the built-in and the external visualization tool, while the near-field light emission distributions and all heat transfer computations are only available in **SimuLEDView**.

Online monitoring of the computation is available using **Log** text field and **Chart** tab which provides visualization of variation of the residuals and the maximum temperature during the heat transfer computation.

10.2 Ray Tracing Options

The **Ray Tracing** section contains the following options of the light extraction problem:

- **Number of Rays** (N). For a single LED lamp without light conversion N = 10^5 – 10^6 is usually sufficient. For a multi-LED or multi-phosphor lamps reasonable accuracy of the color distribution may require N = 10^7 – 10^8. In **Heat Transfer with Ray Tracing** mode the user can assign different values of the number of rays for the preliminary and final ray tracing computations.
- **Temperature** – operating temperature (in **Ray Tracing** mode only).
- **Far-Field Mesh**. Linear meshing of the far-field sphere (the number of cells is proportional to the square of the assigned value).
- **Threshold Intensity**. Minimal intensity of the traced ray measured as a fraction of its initial intensity. When its intensity after absorption and a series of reflections becomes lower than this limit, the ray is considered completely absorbed and its tracing stops.
- **Far-Field Radius** – dimension of the far-field which is used for the light angular distribution visualization.
- **Ambient Refractive Index** – the refractive index of the ambient outside of the lamp.
- **Emission Distribution** – the distribution of the light emission from the chip surface.
- **Number of Threads**. Number of computational processes employed by the solver, which governs parallelization of the computation. This parameter can be used on multi-CPU computers.

The button **Load Results** loads earlier computed distributions from another project file.

The button **Clear Results** removes the computed results from the project file. This can be used to decrease the file size.

10.3 Heat Transfer Options

The **Heat Solver** section contains the following elements:

- **Solver Mode** section. Two modes are available:
  - ✓ **Steady-State** – calculation of the steady-state temperature distribution. In AC or Pulse current operation modes the heat temperature is found using the power values averaged over the period.
  - ✓ **Unsteady** – calculation of temperature distribution as a function of time. It accounts for both unsteady heating from the initial temperature (in all current operation modes) and for unsteady character of the current (in AC or Pulse modes).
- **Solver Dimensionality**. Both 2D and 3D solvers are available. In case of 2D solver, the planar heat source distribution in LED dice is averaged of the angle of 360° over the submount surface. In case of 3D solver the user should specify the mesh accuracy parameter **Number of Sectors**, which governs the mesh fragmentation along the angle.
- **Steady-State Problem Settings** section specifies the following computational options of the non-linear heat transfer solver in steady-state mode:
  - ✓ **Iterations**. The maximum number of global iterations. If the required residuals are not obtained after the prescribed number of iterations, the iterative process automatically stops.
  - ✓ **Residuals**. A parameter that defines the threshold residual of the heat transfer equations. When the actual residual at certain iteration becomes smaller than this parameter, the convergence is considered to be obtained. A default residual is 10^{-4}. This value can be lowered to obtain a better convergence or raised to reduce the computation time.
  - ✓ **Relax**. The relaxation factor which affects the convergence rate. Relax can lay within the interval 0 < Relax <= 1. Higher value provides faster convergence rate, while the lower values provide more stable convergence. In case of solution divergence at the default relaxation value, this parameter should be lowered.

- **Linear Solver** section specifies the computational parameters (**Residuals** and **Iterations**) of the linear solver which is called at each iteration of the non-linear problem solution.
- **Unsteady Problem** section specifies the parameters of the simulated unsteady process:
  - ✓ **Time Max** – duration of the modeled process
  - ✓ **Save Time Step** – time interval of storing the temperature distributions in the *.cgs file.
- **Initial Temperature** – specification of the initial temperature distribution. Two options are available:
  - ✓ **Ambient Temperature** – computation of the unsteady lamp heating starting from the room temperature.
  - ✓ **Load from File** – computation of the unsteady operation of lamp which is already heated. This mode should be used to simulate the effect of oscillations of current in AC or Quasi-CW (Pulse) modes on the temperature distribution eliminating the process of its heating, whose simulation at a time step that allows resolution of the current oscillations is very time consuming. In this case it is beneficial first to find steady-state temperature distribution and then use it as the initial temperature in the unsteady simulation.
- **Auto Time Step** section includes the parameters of time step evaluation. It includes the following options:
- **Quasi-Unsteady Calculation** flag governs whether the heat transfer problem and the circuit operation problems are solved with the same or with different time steps. If it is activated, the time step of the heat transfer is independent on the current frequency and is only limited by the accuracy of the temperature computation, which allows efficient simulation of lamp heating by high frequency current. Otherwise, both problems are solved at the same time step which is found from the fastest of the processes. In this mode, temperature distribution follows all temporal variations of the current.
- **Delta min**, **Delta max** and **Increase Step** parameters govern the evaluation of the time step used for heat transfer problem.
  - ✓ The time step is reduced (divided by 2) if the average temperature distribution accuracy exceeds **Delta max**. This means that the accuracy is insufficient and the time step can be decreased to improve the accuracy.

- ✓ The time step is increased (multiplied by the value of **Increase Step** parameter) if the average temperature distribution accuracy is lower than **Delta min**. This means that the accuracy is excessive and the time step can be raised to reduce the computation time.

➢ **Limited Step by Save Time** flag governs whether the time step used for the calculation is limited by the value of saving time step. In this case the computations are run with the time step equal or lower than the assigned step of saving the data to the file, which is necessary to store the results for the subsequent visualization at the assigned time intervals.

10.4 Visualization

Tab **Result** is used to visualize the far-field distributions of the light emission found in the **Ray Tracing** or **Heat Transfer with Ray Tracing** modes. It includes the following elements:

➢ **Distribution** tab presents the 2D distributions across the far-field sphere. The following quantities are visualized:

- ✓ **Intensity** (W/sr) – distribution of the light intensity.
- ✓ **Color**. Visual presentation of the lamp from various points of the far-field.
- ✓ **Temperature** (K) – the correlated color temperature distribution.

➢ **Radiation Pattern** presents the angular distributions of the light color coordinates and intensity.

➢ **Spectra** tab includes the lamp emission spectrum. The top chart presents the average spectrum of the lamp. The bottom chart presents the spectrum of the emission in the far-field point under the cursor.

➢ **Color** tab presents the following information (see **Fig.31**):

- ✓ Section containing the color rendering index, color coordinates and the color temperature at the point under the cursor
- ✓ Section **Summary** containing the **Color** average values of the color rendering index, color coordinates and the color temperature
- ✓ The color coordinates of the point under the cursor on the color locus.

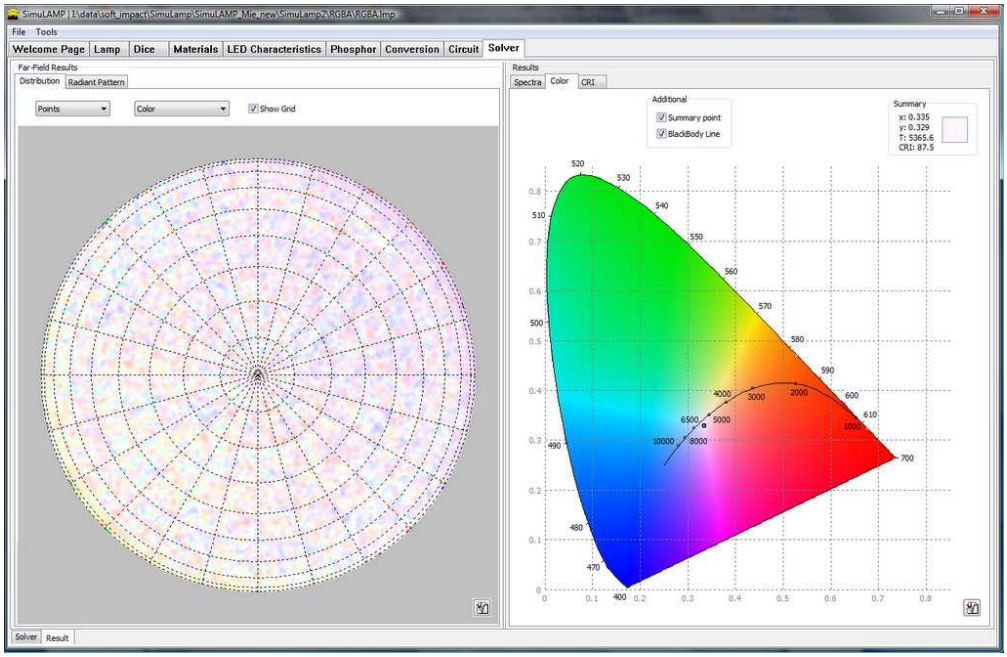


Fig. 31

- **CRI** tab (see **Fig.32**) presents color coordinates and CRI for 14 standard test color samples for the standard illuminant and for the modeled lamp. The values for the standard illuminant present the actual sample colors, while the values for the modeled lamp demonstrate the resulting color of the given sample illuminated by the modeled lamp.

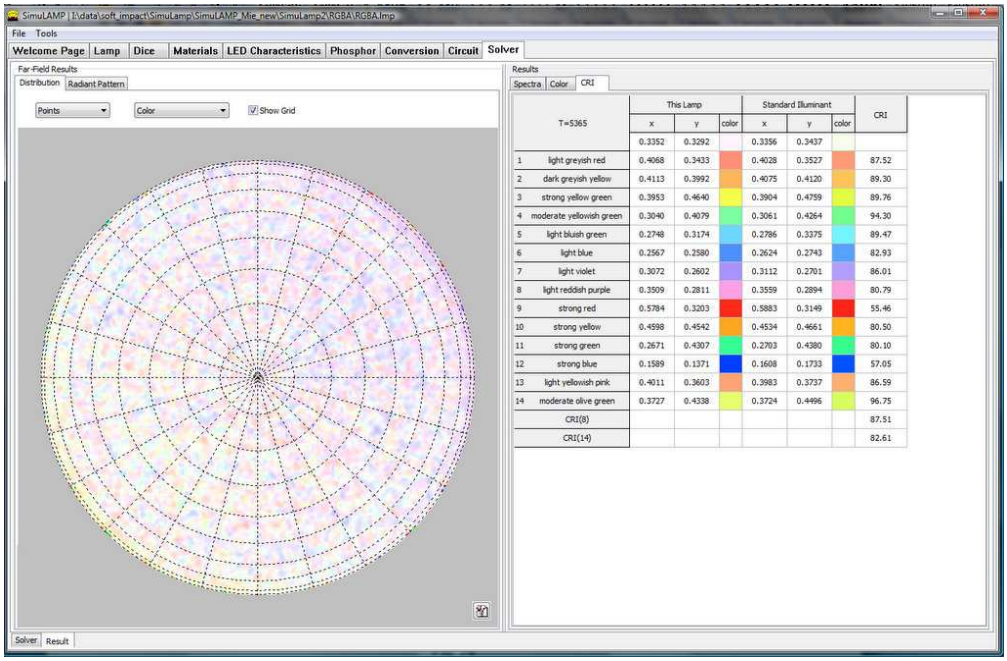


Fig. 32

11 Functions Window

The **Functions** window[3] (see **Fig.33**) allows the user to specify a set of functional and piecewise linear scripts describing complex dependencies of properties used in **SimuLAMP** on various parameters. Five groups of functions are available:

- **Wavelength Dependencies**. Functions of wavelength can be used to specify complex spectra of the LED emission, phosphor emission and excitation, phosphor scattering and absorption cross sections spectra, and of any wavelength-dependent optical parameters of the lamp materials. The user can also assign functions with additional current variable, which can be used to specify emission spectra of the LED as a function of temperature and current. This group consists of two function types named **Wavelength(wl; T)** and **Spectrum(wl; T; I)**.
- **Temperature Dependencies**. Functions of temperature can be used to specify temperature and current dependent parameters in the analytical approximations of the LED emission spectrums, LED I-V and I-L curves, phosphor emission and excitation, internal quantum efficiency, etc. This group consists of two function types named **Temperature(T)** and **Characteristic(I; T)**.
- **Coordinate Dependencies**. Functions of coordinates can be used to specify approximations of the parameters of the lamp materials as a function of local coordinates. Additional temperature variable can also be used in this type of functions. This group consists of functions named **Materials(x; y; T)**.
- **Angular Dependencies**. Functions of polar angle and wavelength or polar angle and asymmetry factor can be used to specify scattering pattern of the phosphor particles (asymmetry factor can also be wavelength and temperature dependent) and emission pattern of the LED chip (asymmetry factor is replaced by g-parameter here). This group consists of two function types named **Diagram(fi; g)** and **Diagram(fi; wl)**.
- **Radial Dependencies.** Functions of radial coordinate can be used to specify size distribution of the phosphor particles. This group consists of functions named **Distribution(r)**.

[3] In version 1.0, Functions were listed in a separate tab of the main window. In version 2.0, Functions window can be opened by Tools->Functions menu item.

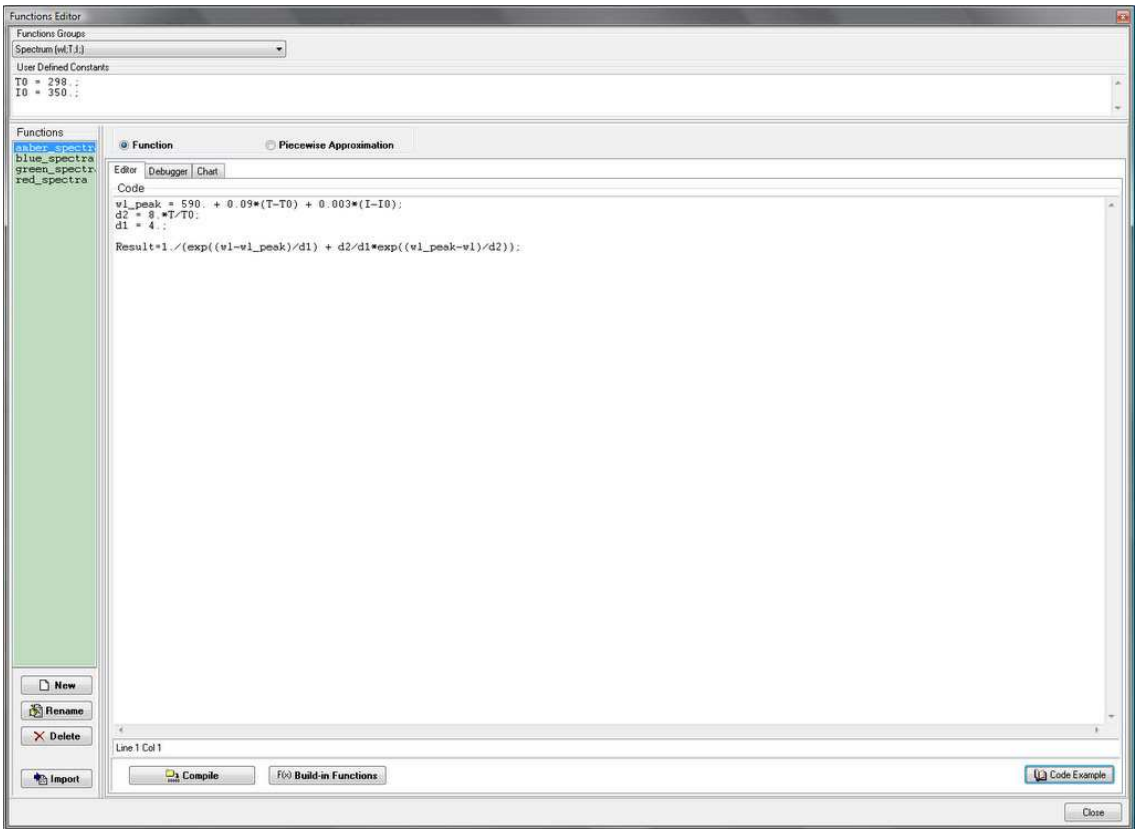


Fig. 33

The tab **Functions** contains the following elements:

- Section **User-Defined Constants** on top of the window allows the user to define a set of global parameters that can be used in different functions.
- Section **Functions** containing the list of all defined functions and the following buttons:
  - ✓ Button **New** adds a new function.
  - ✓ Button **Delete** removes the selected function.
  - ✓ Button **Rename** allows the user to rename the selected function.
  - ✓ Button **Import** allows the user to import one or several functions from another project file (*.lmp).
- Selection of the function specification mode. A functional approximation can be specified by a script (**Function** mode) or buy a table (**Piecewise Approximation** mode).

In the **Function** mode, the following tabs are available:

- Tab **Editor** provides specification of the function. It contains the following items:
  - ✓ Section **Code** contains the text definition of the function. The function must end with assigning a value to the variable named **Result**.
  - ✓ Button **Compile** provides compilation of the function text and checks for the errors.
  - ✓ Button **Built-In Functions** opens a dialog window providing a list of built-in functions (see **Fig.34**). The user can either type the function name in the code or just select the needed function in the list and press button **Copy to Code**.

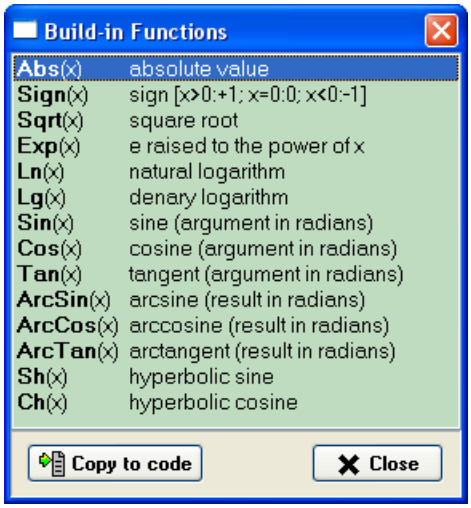


Fig. 34

  - ✓ Button **Code Example** provides a set of code fragments to be used as examples of function programming.
- Tab **Chart** provides graphical visualization of the specified function. It contains the following items (see **Fig.35**):
  - ✓ Selection of the X- and Y- variables of the plot.
  - ✓ **From** and **To** text fields specify the plot X-axis range
  - ✓ **Create Chart** button is used to generate the plot and to updates when the axis ranges are changed.

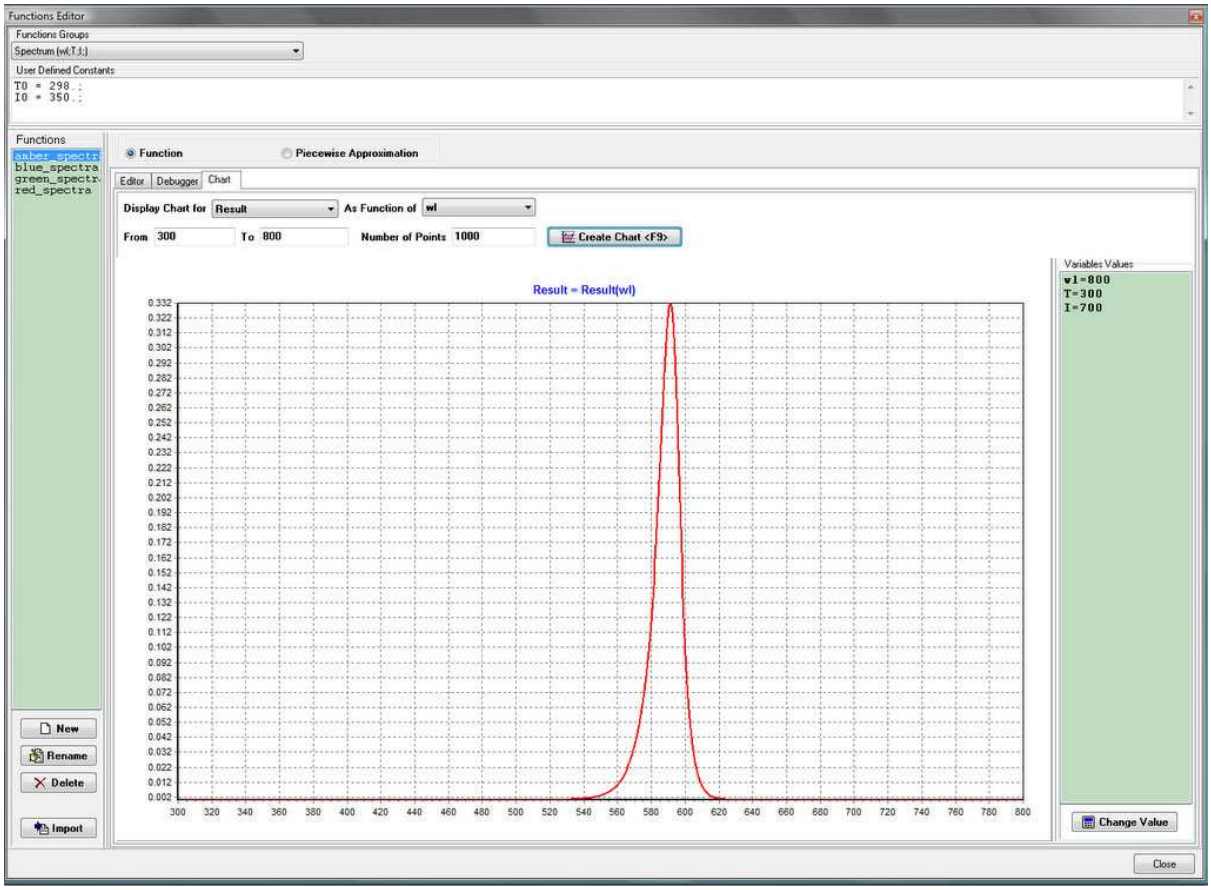


Fig. 35

- Tab **Debugger** allows the user to trace the code execution and check the intermediate results. It contains the following items (see **Fig.36**):
  - ✓ Section containing the lines of the function code, which are executed consecutively.
  - ✓ Button **Execute Line** providing execution of the code of the current line.
  - ✓ Button **Execute All** providing execution of the whole code of the function.
  - ✓ Button **Reset Calculations** providing restart of the code debugging.
  - ✓ Section **Variables** at the right hand side visualizing the current values of all variables defined within the function.
  - ✓ Button **Change Value** allowing the user to assign manually the value of a selected variable during the code execution.

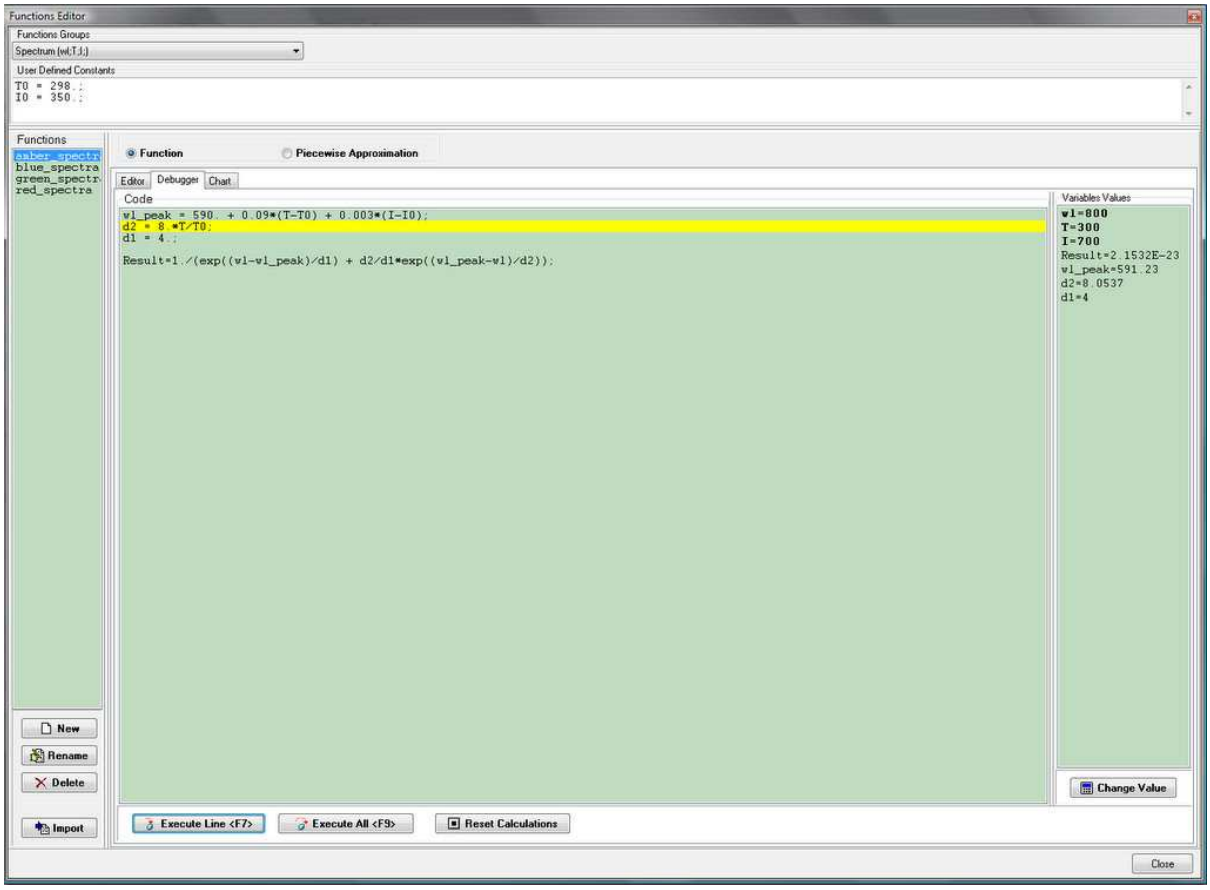


Fig. 36

In the **Piecewise Approximation** mode, the following items are available:

- Section **Argument** contains list of variables to be used as argument of the approximation. The following arguments are available:
  - ✓ Wavelength functions: **w** (wavelength) and **T** (temperature)
  - ✓ Materials functions: **X**, **Y** (radial and vertical coordinates) and **T** (temperature)
- Section **Points** containing the list of points (table with two columns corresponding to the argument and value of each point) in the current approximation. A new line is added to the table by pressing Shift-Enter key. A selected line is removed from the table by pressing Shift-Delete key.

12 Appendix 1: Visualization of SimuLAMP Results in SimuLEDView

SimuLAMP computations are stored in ProjectName_rtr.cgs (running computations on **Ray Tracing** tab) and ProjectName_rtr.cgs (running computations on **Heat Solver** tab), which are visualized by

SimuLEDView tool. The variables stored in these files are listed below. Results of coupled heat transfer and ray tracing computations (ProjectName_rtr.cgs file) include all the listed variables. The results of ray tracing computations (ProjectName_rtr.cgs) and results of heat transfer in the mode ignoring the light emission do not include the temperature and light emission variables, respectively.

The data are organized in the following groups which can be selected by the user in a combobox on top of the graphics window:

- **View2D** (distribution in the axial vertical section):
  - ✓ **T** (K) – temperature
  - ✓ **QVolume** (W/mm^3) – volumetric heat source due to light adsorption in semitransparent medium
  - ✓ **HeatCond** (W/(m·K)) – thermal conductivity
  - ✓ **Refractive Index** – the refractive index. In opaque materials, zero values are output.
  - ✓ **Absorption Index** – the absorption index. In opaque materials, zero values are output.
  - ✓ **dT/dx** (K/m) – radial component of the temperature gradient
  - ✓ **dT/dy** (K/m) – axial component of the temperature gradient
  - ✓ **gradT** (K/m) – magnitude of the temperature gradient
  - ✓ **QSurf** (W/mm^2) – surface heat source due to i) heat dissipation in the chip which is considered in SimuLAMP as a zero-thickness layer and ii) light adsorption on the boundaries of the semitransparent blocks where non-zero adsorption coefficients are assigned. The surface heat source is visualized in 2D mesh by distributing the heat in the adjacent block cells.
- **Active Zone**:
  - ✓ **T** (K) – temperature
  - ✓ **QSurf** (W/mm^2) – surface heat source due to heat dissipation in the chip
  - ✓ **Light Power** (W/mm^2) – emission power density
  - ✓ **Radius** (mm) – local radial coordinate
- **Far-Field Region**:
  - ✓ **Radiant_Intensity** (W/sr) – radiant light emission intensity
  - ✓ **Luminous_Intensity** (cd) – luminous light emission intensity
  - ✓ **x** – x color coordinate

- ✓ **y** – y color coordinate
- ✓ **CRI** [%] – color rendering index
- ✓ **Color_Temperature** (K) – correlated color temperature
- ✓ **Phi** (degree) – local **φ** coordinate in the spherical coordinate system
- ✓ **Theta** (degree) – local **θ** coordinate in the spherical coordinate system

➢ **View3D** (distributions over all lamp surfaces):

- ✓ **Intensity** (W/mm^2) – radiant light emission intensity
- ✓ **HeatFlux** (W/mm^2) – heat flux through the surface
- ✓ **QSurf** (W/mm^2) – surface heat source due to heat dissipation in the chip (for the Heterostructure surface) or light adsorption (for the boundaries of the semitransparent blocks where non-zero adsorption coefficients are assigned).
- ✓ **T** (K) – temperature
- ✓ **x** (mm) – local x coordinate (in horizontal plane)
- ✓ **y** (mm) – local y coordinate (in horizontal plane)
- ✓ **z** (mm) – local vertical coordinate

13 Appendix 2: Tools for Geometry Specification

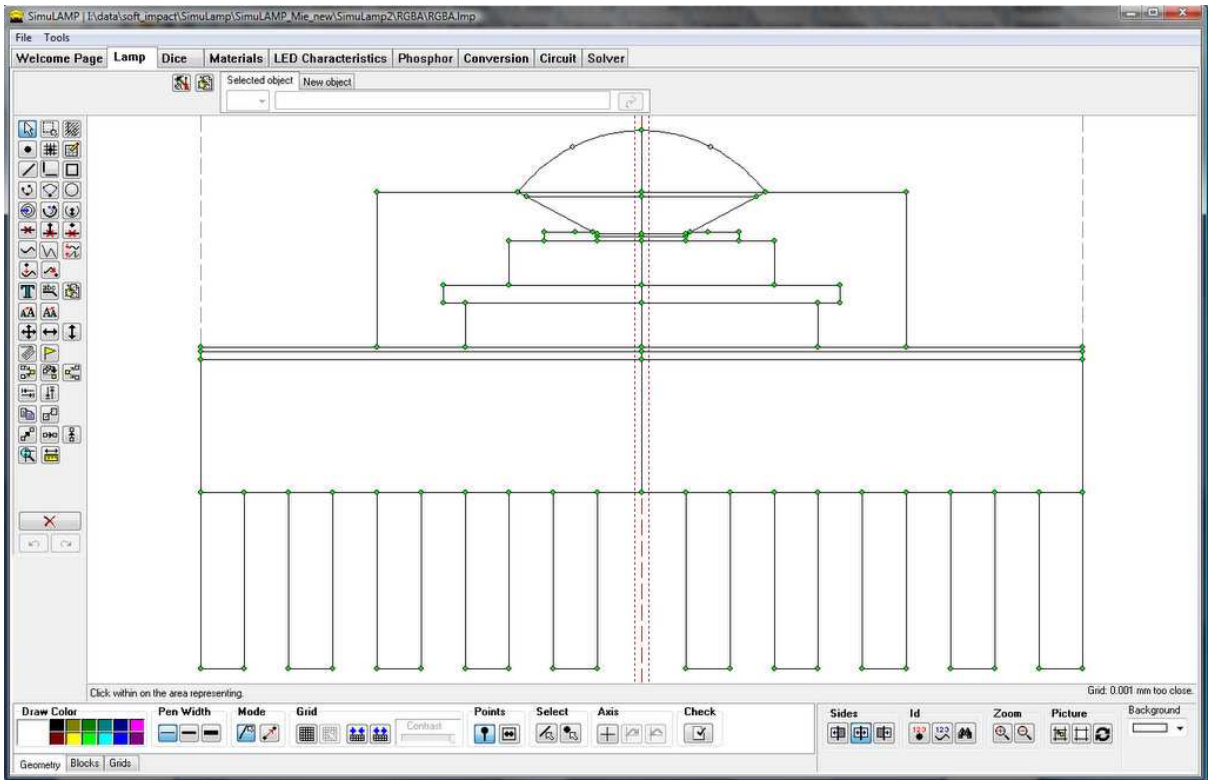


Fig. 37

Tools for the geometry specifications are located on the **Drawing Toolbar** at the left hand side of the window.

 The **Select Object** tool allows the user to select points, lines, arcs and splines with the left mouse button.

A point is selected when the mouse cursor is positioned in the vicinity of the point. In this case, the point starts flickering in white to identify that this specific point will be selected on clicking the mouse button. A selected point is marked with a red contour.

A line, polyline, arc or spline is selected when the cursor is positioned on the boundary at a distance from any point (otherwise, the point will be selected instead of the boundary). The selected boundary is marked in red.

Repeated clicking on a selected object deselects it. Consecutive selection a series of objects provides their multiple selections.

Clicking on a line that is a fragment of a closed contour with **Shift** key pressed selects the whole contour.

Pressing **Spacebar** key clears all selection.

 The **Select Object in the Rectangle** tool allows the user to select a group of points, lines, and arcs by a lasso, providing selection of all objects placed in a rectangular area. Moving the mouse with left button pressed assigns the rectangle dimensions specifying the bottom right corner keeping the top left corner. On the other hand, moving the mouse without pressing the button displaces the rectangle keeping its size, specifying the top left corner.

Note that in the lasso selection the boundaries are identified by the segment end points. So if no end point of a boundary lie within the area (i.e. if a single boundary segment crosses the lasso rectangle), the boundary is not selected.

 The **Clear Selection** tool clears the selection of all objects. This button is duplicated by the **Spacebar** key.

 The **Add Point** tool allows the user to add a new individual point. The point is created when the left mouse button is pressed.

If the mouse cursor is positioned in the vicinity of an existing point or existing line, the new point is aligned to the existing object.

If the **Align To Grid** mode is activated and the mouse cursor is positioned in the vicinity of a node of the reference grid, the new point is aligned to this node.

 The **Align Selected Point to Grid** tool allows the user to align an existing point to reference grid. Note that this tool does not assume prior selection of the points. The user should first press the **Align Point to Grid** button and then click consecutively on the points to be aligned.

 The **Explicit Coordinate Specification** tool allows the user to assign explicit coordinates of a point, line or polyline in a table. This option is an alternative to drawing the lines by clicking the mouse

on the **Graphics Window**. When The **Explicit Coordinate Specification** button is pressed, **Points List** window is shown. The user should select the object to draw (point, line or polyline), assign the coordinates in the table and press **Apply** button.

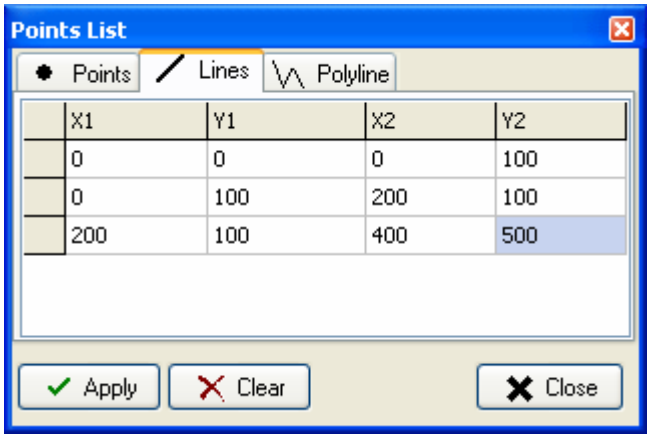


Fig. 38

 The **Add Line** tool allows the user to draw a line. To draw a line the user should consecutively click with the left mouse button on the first and last ending points.

There are two modes of line drawing.

- **Elastic Lines** mode (set by default). While the mouse cursor is moved after the first point is specified, the current line position is previewed for each cursor position. In this mode drawing cannot be interrupted by displacement of the view area by the right mouse button, so locations of both points of the line should be visible on the screen at the start of drawing. Thus, this mode is optimal for drawing short lines.
- **Point-To-Point Line** mode. The line is not shown until it is created, while the ending points are marked with crosshairs  symbol. The line is drawn when both points are prescribed. In this mode the user can shift the view area by the right mouse button, so this mode can be preferable in drawing long lines where zoom/move operations are needed to specify the exact point positions.

Switching between the drawing modes is made using the **Mode** section on the **Viewing** toolbar.

 The **Add Vertical or Horizontal Line** tool allows the user to draw a line (see the above section **Add Line** for details) aligning it in vertical or horizontal direction.

When the line is created using this tool, the position of the second ending point is displaced from the exact point under the cursor so that the line is oriented in the strictly vertical or horizontal direction.

 The **Add Rectangle** tool allows the user to draw a closed rectangular contour. The rectangle is specified by its diagonal line. So to create a rectangle the user should specify two points: its top left corner and the bottom right corner (or the other pair of opposite corners).

 The **Add Arc by 3 Points** tool allows the user to specify an arc assigning three points lying on it. First the user should specify two arc ending points by clicking the left mouse button. They are shown as marking points with crosshairs  . Then the middle point should be assigned. In **Elastic Line** mode the user can move the middle point keeping the left button pressed, the current arc position is previewed. When the left button is released, the arc is created. In **Point-To-Point** mode the arc is shown after all three points are specified.

 The **Add Arc by Center and 2 Points** tool allows the user to specify an arc assigning its center position, radius and angles of the ending points. First the user should specify the arc center (the center of the circle comprising the arc) and first arc ending point by clicking and releasing the left button. These points are shown as marking points with crosshairs  . Then the second ending point should be assigned. When the user releases the left button after assigning the first ending point and moves the mouse, the arc is previewed for each cursor position. Clicking the left button the user specifies the final position of the second ending point.

 The **Add Circle** tool allows the user to draw a closed circle. First the user should specify circle center by clicking the left mouse button. The center is shown as marking point with crosshairs  . Then the circle radius should be assigned by specifying a point that lies on the circle. The user should click the left button and can move the point keeping the left button pressed, the current arc position is previewed. When the left button is released, the circle is created.

 The **Make Two Arcs Concentric** tool allows the user to adjust an arc center to center of another arc. The user should first select the arc (circle) to be moved, and then the reference arc (circle).

 The **Move Arc Point** tool allows the user to change arc length by moving an arc ending point along the arc. To move an arc point the user should select the point, then press the left button and keeping the button move the mouse towards the new point position. Then the point is displaced strictly along the arc.

 The **Change Arc Radius** tool allows the user to increase or reduce the radius of an existing arc or circle. To change circle radius the user should select the circle (arc), then press the left button and keeping the button move the mouse toward or outward the circle center to increase or decrease the radius, respectively.

 The **Split Line or Arc by Point** tool allows the user to split a line at a given point. To split the line the user should just specify the splitting point position by clicking the left mouse button. Prior selection of the line (arc) to be split is not required. The point under cursor is automatically aligned to the nearest object which is split. If the splitting point is an intersection point of two or more lines, all of them are split.

 The **Add Normal Splitting an Existing Line, Arc…** tool allows the user to create line normal to an existing line or arc which is split at the intersection point. The user should select the line (or arc) to be split and then to specify the reference point, i.e. a point from which perpendicular is dropped on the selected line. The reference point can either be an existing point (an ending point of some other lines) or a new point.

Note that creation of a normal to a line is successful if only the reference point is positioned so that its projection is within the selected line, which is actually a finite-length line segment with two ending points. Otherwise, if this segment and the normal line do not intersect (i.e. the intersection point of the normal and the infinite line that comprises the segment lies beyond the segment itself), the intersection point is shifted to the nearest ending point, so that the created line is not normal to the selected line.

A normal to an arc (circle) is a line connecting the reference point and the arc (circle) center.

 The **Split Line, Arc… by a Normal Projection** tool allows the user to split a line at a point created as a normal projection of some reference point. The user should select the line (or arc) to be split and then to specify the reference point, i.e. a point from which perpendicular is dropped on the selected line to find the projection position. The reference point can either be an existing point (an ending point of some other lines) or a new point. Unlike the **Add Normal Splitting an Existing Line or Arc** tool, this tool does not add the line connecting the reference point and the splitting point.

Note that splitting by normal projection is only available when the reference point is positioned so that its projection is within the selected line, which is actually a finite-length line segment with two ending points.

A normal to an arc (circle) is a line connecting the reference point and the arc (circle) center. So the arc is split at the point of intersection of the arc with circle radius oriented toward the reference point.

 The **Add Spline** tool allows the user to draw a smooth curve. The user should consecutively specify the spline points by clicking the left mouse button.

 The **Add Polyline** tool allows the user to draw a polyline. The user should consecutively specify the intermediate polyline points by clicking the left mouse button.

 The **Spline-To-Polyline Conversion** tool allows the user to convert a polyline to a spline and vice versa. Note that this tool does not assume prior selection of the object. The user should first press the **Spline-To-Polyline Conversion** button, then click on the object and select the target object type (spline or polyline) in the popup window.

 The **Add Point to Spline or Polyline** tool allows the user to modify a spline or polyline by adding an intermediate point to it. The user should press the **Add Point to Spline or Polyline** button and then specify the point to be added by clicking the left mouse button. If the assigned point is found to be in the vicinity of an existing spline or polyline, it will be inserted into it.


The **Continue a Spline** tool allows the user to modify a spline or polyline by adding a new point to it after the ending points. The user should press the **Continue a Spline** button, select the spline, and then specify the point to be added by clicking the left mouse button. The point will be added to the spline at a side which is closer to the new point.


The **Add Text** tool allows the user to add a text to the drawing. The user should specify the text position with the left mouse button and enter the text in a **Text** dialog window.


The **Callout** tool allows the user to add a callout to some element of the drawing. The user should specify the text position with the left mouse button and enter the text in a **Text** dialog window.


The **Edit Text** tool allows the user to edit an existing text. To edit a text, the user should first press the **Edit Text** button and then click on the text to be edited. Then a **Text** dialog window will appear.

Increase Font Size


The **Increase Font Size** tool allows the user to increase size of the font used in text and callouts added to the drawing. To enlarge a text, the user should first select the text and then to press the **Increase Font Size** button. A single click application of this tool increases the font size by 10 per cent of its current value.


The **Decrease Font Size** tool allows the user to decrease size of the font used in text and callouts added to the drawing. To reduce a text, the user should first select the text and then to press the **Decrease Font Size** button. A single click application of this tool decreases the font size by 10 per cent of its current value.

 The **Move Selected Objects by Mouse** tool allows the user to move objects (lines, arcs, or points) in an arbitrary direction using the mouse. The user should first select the objects to be moved, then press the left mouse button, move the mouse cursor in the desired direction, and then release the mouse button specifying the new object position. If new positions of some points of the moved object are located in the vicinity of the existing objects, the cursor position specifies the intermediate positions only. To finalize the object displacement the user should clear the object selection (by pressing the **Spacebar** key). Then such points are aligned to the existing points.

Note that releasing the mouse button does not finalize the moving operation (it does not clear the object selection) so that the object can be consecutively moved several times without distortion at the intermediate positions.

 The **Move Horizontally by Mouse** tool allows the user to move objects in horizontal direction using the mouse. Compared to the **Move Selected Objects by Mouse** tool, the vertical coordinate is fixed during cursor displacement.

 The **Move Vertically by Mouse** tool allows the user to move objects in vertical direction using the mouse. Compared to the **Move Selected Objects by Mouse** tool, the horizontal coordinate is fixed during cursor displacement.

 The **Move Selected Objects by Explicit Shift** tool allows the user to move selected objects by explicit specification of the displacement value. First, the user should select the objects to shift, then to press the **Move Selected Objects by Explicit Shift** tool button. It opens the Translation dialog window that has two text fields **Translation along X** and **Translation along Y**, respectively. The shift values should be specified in microns.

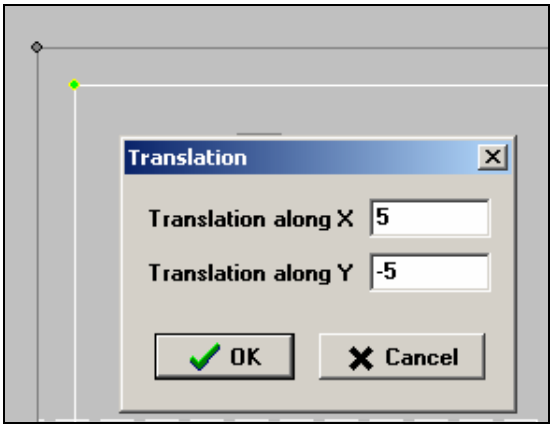


Fig. 39

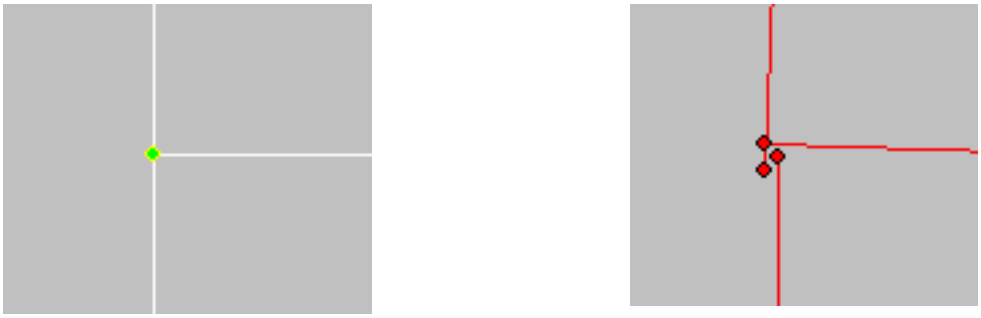
 The **Put Flag** tool allows the user to add a mark to some drawing position.

 The **Merge Selected Points** tool allows the user to merge several close points. To merge the points the user should first select them and then press the **Merge Selected Points** tool button. The points will be merged into a single point whose position is determined by the order of the point selection, so that the coordinates of the first selected points are assigned to the new common point.

 The **Merge All Points in the Rectangle** tool allows the user to merge all points inside the selected area. To merge a group of points the user should first press the **Merge Selected Points** tool button. Then the cursor shape changes to a small rectangle. The user should position this rectangle so that it would comprise the points to be merged. The rectangle has a fixed size in pixels so that if it is too small to surround all needed points the user should enlarge the relative size of the rectangle by zooming out the image. Clicking with the left mouse button will merge the points into a single point whose position is determined by one of the merged points that is closest to the rectangle center. This point starts flickering in white to identify this before the points are merged.

 The **Unlink Lines with a Common Point** tool allows the user to unlink lines that have a common point. To apply this tool the user should first select the lines (see **Fig.40 a**) and then press the **Unlink**

Lines with a Common Point tool button. Then the lines will be separated (see **Fig.40 b**) and could be edited (moved, etc.) independently.



(a) (b)

Fig. 40

 The **Align Selected Points Horizontally** tool allows the user to align the **R**-coordinates of a group of points. The user should first select two or more points and then press the **Align Selected Points Horizontally** tool button. The **R**-coordinates of all selected points will be changed to the **R**-coordinate of the first selected point.

 The **Align Selected Points Vertically** tool allows the user to align **Z**-coordinates of a group of points. The user should first select two or more points and then press the **Align Selected Points Vertically** tool button. The **Z**-coordinates of all selected points will be changed to the **Z**-coordinate of the first selected point.

 The **Multiply Selected Objects** tool allows the user to multiply several objects with moving them in some direction, rotating with reference to a selected point, or flipping vertically or horizontally. This tool was designed to provide an easy specification of repeated contour fragments in some complex, for example, periodical or symmetrical, die geometries.

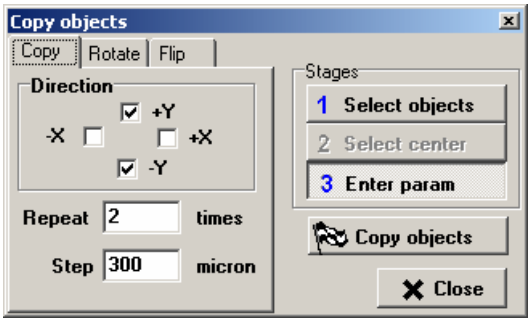


Fig. 41

Pressing **Multiply Selected Objects** tool button opens the **Copy Objects** dialog window that includes three tabs corresponding to different types of the tool operation and a set of buttons governing the following stages of the user actions:

- **Select Objects**. The initial stage at which the user should select the original objects by the mouse.
- **Select Center**. For **Rotate** type the user should specify a point that will be used as the rotation center. For **Flip** type, the user should specify a point that determines a horizontal or vertical line that will be used as the reference axis.
- **Enter Param**. At this stage the user should specify the parameters described below.
- **Execute**. A button applying the operation.

The following types of the operation are available:

- **Copy**. This operation represents translation of the selected objects in a vertical or horizontal direction. Simultaneous copying in several directions is available. The **Copy** tab includes the following elements (see **Fig.41**):
  - ✓ Section **Direction** contains checkboxes **–X**, **+X**, **-Y** and **+Y** providing translation to the left, to the right, downward and upward, respectively.
  - ✓ **Step**. The distance of the translation.
  - ✓ **Repeat**. The number of objects to be created in each specified direction. Multiple translation (**Repeat** is above 1) creates equidistant objects consecutively, so that **i**-s new object is translated by **Step** value with reference to the **(i-1)**-s object, i.e. by **Step*i** with reference to the original one.

- **Rotate**. This operation creates new object by rotation of the selected objects. Simultaneous creation of objects by rotating at consecutively increasing angles is available. The **Rotate** tab includes the following elements (see **Fig. 42**):
  - ✓ **Angle**. The rotation angle of the translation (in degrees).
  - ✓ **Repeat**. The number of objects to be created. Multiple rotation (**Repeat** is above 1) consecutively creates objects so that **i**-s new object is rotated by **Angle** value with reference to the **(i-1)**-s object, i.e. by **Step*i** with reference to the original one.

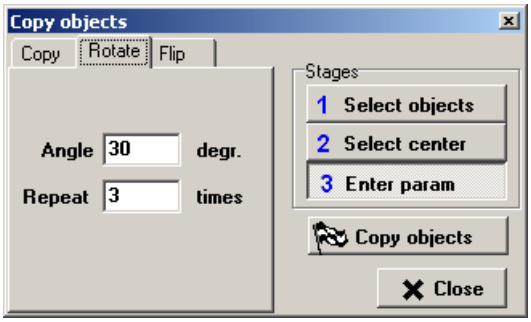


Fig. 42

- **Flip**. This operation represents flip of the selected object with reference to some vertical or horizontal axis. The **Flip** tab includes the following elements (see **Fig.43**):
  - ✓ **Flip Vertical**/**Flip Horizontal** selection allows the user to specify the flip direction.

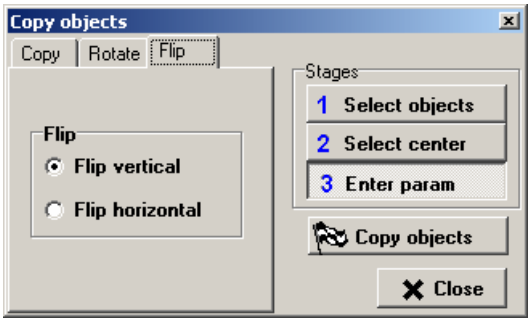


Fig. 43

 The **Duplicate Selected Objects at a Specified Shift** tool allows the user to duplicate an object by creating a copy at a specified position. The user should first select an object, then specify the starting point of the translation vector (as a rule, a point on the original object), and then specify the ending point of the translation vector (at the new position of this point).

 The **Duplicate Selected Objects at a Preset Shift** tool provides a fast way to duplicate an object creating the copy at some predefined position shifted both to the right and upwards with respect to the original object. The user should only select an object and press the **Duplicate Selected Objects at a Preset Shift** tool button.

 The **Duplicate at Preset Shift Horizontally** tool provides a fast way to duplicate an object creating the copy at some predefined position shifted to the right with respect to the original object. The user should only select an object and press the **Duplicate at Preset Shift Horizontally** tool button.

 The **Duplicate at Preset Shift Vertically** tool provides a fast way to duplicate an object creating the copy at some predefined position shifted upwards with respect to the original object. The user should only select an object and press **the Duplicate at Preset Shift Vertically** tool button.

 The **Analyze for Dangling Lines** tool performs analysis of self-consistency of geometry specification, providing the user information on the presence of dangling lines (in other terms, points with only one contacting line or arc) and overlapping lines.

 The **Measure Distances and Orientation** tool provides the user the information on the length and orientation of lines in the drawing. The user should consecutively specify the first and the last ending points of a vector. A resulting message window contains the following parameters:

- Coordinates of each points
- Length of the **R** and **Z** projections of the vector.
- Vector length
- Angle of the vector orientation in the counter-clockwise direction.

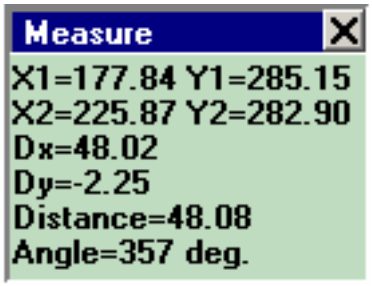


Fig. 44

The **Scale Geometry** tool provides a way to scale the whole geometry or some of its fragment. This is required, in particular, when the geometry is imported from a file where it was specified in coordinates different from those used in **SimuLAMP**.

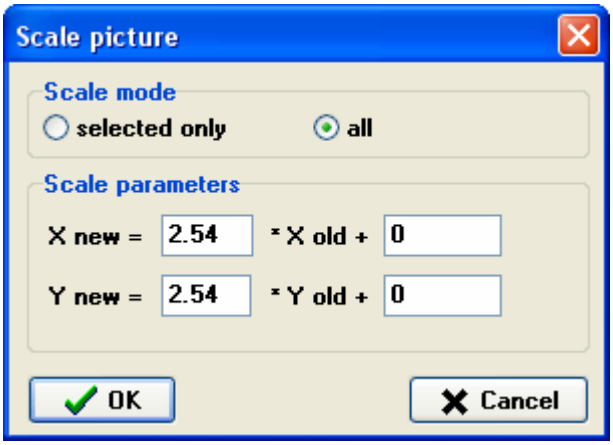


Fig. 45

Pressing **Scale Geometry** button opens the **Scale Geometry** dialog window that includes the following element:

- **Scale Mode.** The user can specify whether the whole geometry or only the selected objects should be scaled
- **Scale Parameters.** The user should specify the scaling factor and displacement for both radial and vertical coordinates.

 The **Delete Selected Objects** tool allows the user to remove any objects from the current layer of the drawing.

 The **Undo** tool allows the user to undo the last action.

 The **Redo** tool allows the user to redo the last action.

14 Appendix 3: Options of Geometry Visualization

The **Viewing** toolbar is located at the bottom of the **Geometry** tab window and contains a set of buttons governing the options of geometry specification and visualization.

The **Draw Color** section allows the user to specify the color of the geometry objects. The selected color is applied to the new lines to be drawn and to the existing lines which are selected at the moment of the color assignment. It does not have an effect on the earlier drawn unselected contours.

The **Pen Width** section  allows the user to specify the thickness of the drawn lines. The selected thickness is applied to the new lines to be drawn and to the existing lines which are selected at the moment of the thickness assignment. It does not have an effect on the earlier drawn unselected contours.

The **Mode** section contains  **Elastic Lines** and  **Point-to-Point Lines** buttons that allow the user to switch between two modes of line drawing, which are described above in **Add Line** section.

The **Grid** section allows the user to specify visualization of reference coordinate grid and alignment of the entered coordinates to the grid nodes. The **Grid** section contains the following buttons:

-  **Show Grid** governs visualization of the reference grid.
-  **Draw Aligning To Grid** governs the creation of new objects. If this button is pressed, the ending points of new objects are aligned to the nodes of the reference grid closest to the cursor position.
-  **Decrease Step** button decreases the grid step in two times.
-  **Increase Step** button increases the grid step in two times.

The **Points** section allows the user to specify the modes of point visualization. It contains the following buttons:

-  **Show Points** governs visualization of ending points.
-  **Show Frame Around Close Points** enables identification of close points which might correspond to a single physical point but were separated unintentionally during geometry specification.

The **Select** section allows the user to specify the modes of point and line selection. It contains the following buttons:

-  **Select Lines Only**. If this button is pressed, points cannot be selected, which makes easier to pick a line in a complex geometry with close points lines.
-  **Select Points Only**. If this button is pressed, lines and arcs cannot be selected, which makes easier to pick a point in a complex geometry with short lines.

The **Axis** section allows the user to shift temporarily the coordinate origin to make easier specification of the coordinates known with respect to some intermediate element of the geometry. For example, the user may need to specify a rectangular element while he knows the values of its length and height and coordinate of the left bottom corner. Normally, the user should first calculate the coordinates of all its corners and then type the calculated values in the text fields. As an alternative, the user may shift the origin to the known left bottom corner of the rectangle using the Axis tool and then specify explicitly the length and height values. This section contains the following buttons:

-  **Show Axes**. When this button is pressed, the **R** and **Z** axes are shown and the origin can be shifted. Repeated pressing this button hides the axes and resets the origin to its basic position.
-  **Move Origin**. To specify the temporal origin poison? The user should click **Move Origin** button and then pick the new origin on the drawing.

The **Analyze** section contains the **Analyze Geometry**  button providing analysis of the specified geometry. The program checks if there are crossing and dangling lines and identifies them by marking with yellow triangles if they are found.

The **Sides** section governs the visualization of the left-hand and right-hand sides of the axisymmetric geometry. It contains the following buttons:

-  **Show Left-Hand Side**.
-  **Show Both Sides**.
-  **Show Right-Hand Side**.

The **ID** section governs the visualization of the ID of the geometry objects which may be needed to identify them in case they are mentioned in some error messages. It contains the following buttons:

-  **Show Point ID**.
-  **Show Line ID**.
-  **Find Object by ID.**

The **Zoom** section allows the user to zoom the image. It contains the following buttons:

-  **Zoom In** button allows the user to enlarge the image.
-  **Zoom Out** button allows the user to reduce the image.

The **Picture** section allows the user to refresh and zoom the image. It contains the following buttons:

-  **Show All** button automatically calculates the drawing scale to fit all objects in the drawing.
-  **Show Page** button assigns the drawing scale to fit the page. The page is visualized by white dashed lines, its dimensions and position can be specified in the **Options** dialog window available via **Options** button on the top tool bar.
-  **Refresh** button forces image refreshing.

15 Appendix 4: Options Dialog Window

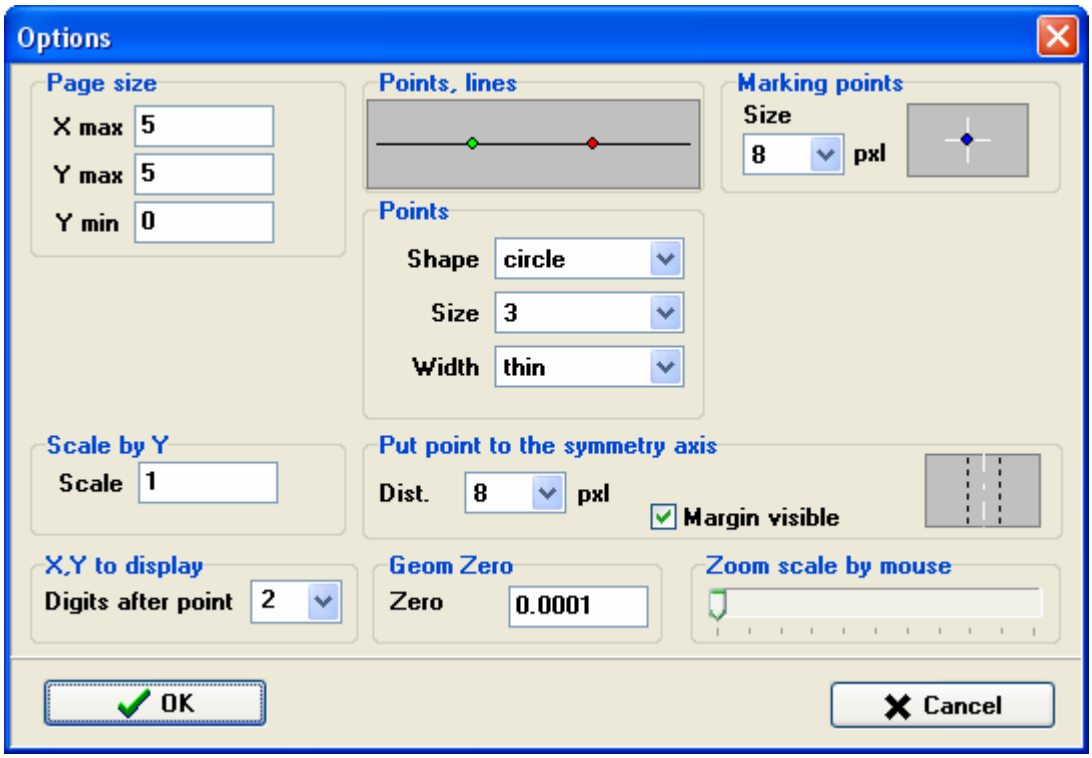


Fig. 46

The **Options** dialog window (see **Fig.46**) is activated by **Options** button that is located on the top tool bar of the **Geometry** tab window. It includes the following items:

- Section **Page Size.** This section governs the size and position of the page where the whole drawing is placed. The page perimeter is shown with dashed white lines. The user can specify the minimum and maximum values of both X and Y coordinates. By default, the page is 1000×1000 microns.
- Section **Scale Y.** This section can be used to scale the vertical coordinates of the geometry. By default, it is equal to unity, so that the geometry is shown keeping actual proportions. Otherwise, the visible coordinates are multiplied by YScale.
- Section **X,Y to Display.** This section contains parameter **Digits After Decimal Point** that governs the accuracy of specification of point coordinates and other values. It corresponds to the number of digits after decimal point.
- Section **Points.** This section governs representation of the points. It contains the following parameters:

- ✓ **Shape**. The shape of the symbol used to mark a point. By default, it is circle. Rectangular points are also available.
- ✓ **Size**. The point size (circle diameter or rectangle side) in pixels.
- ✓ **Width**. The width of black contour surrounding the point. Thin (1 pixel) and thick (3 pixels) contours are available.

- Section **Marking Points.** This section governs representation of the points used as markers in Point-To-Point mode of line creation. The size of the point itself is the same as the size of the other points and is assigned by the **Size** parameter in **Point** section. The **Marking Points** section contains parameter **Size** that governs the length of white lines of the cross-hairs.
- Section **Put Point to Symmetry Axis** specifies the distance from the symmetry axis at which cursor position is aligned to the axis. If **Margin Visible** checkbox is marked, the symmetry axis zone is visualized by the dashed lines.
- Section **Zoom Scale By Mouse.** This section allows the user to change degree of image scaling by moving the mouse with middle button pressed. A higher scale value requires a smaller mouse displacement for a given zooming.

16 Appendix 5: Graphics Window

All tab windows contain the **Graphics Window** that is designed to specify and visualize the layout of all die layers, providing mouse specification of the objects to be created or modified. Multiple boundary selection is also available. The cursor view depends on the type of the active operation.

The following basic operations are available within the **Graphics Window**:

- Image translation (displacement of the whole image including the origin of coordinate system, i.e. the coordinates of all objects are unchanged).
  - ✓ Right mouse button is pressed and the mouse is moved along the translation direction.
- Zoom In (enlarging the image).
  - ✓ The mouse wheel is rolled towards the user.
- Zoom Out (reducing the image).
  - ✓ The mouse wheel is rolled away from the user.
- Selecting a point.
  - ✓ In the selection mode, the left mouse button is pressed on the point. When the mouse cursor is positioned in the vicinity of a point so that clicking the mouse button will lead to its selection, this point starts flickering in white. A selected point is marked in white.
- Selecting a boundary.
  - ✓ In the selection mode, the left mouse button is pressed on the boundary at a distance from any point (otherwise, the point will be selected instead of the boundary). The selected boundary is marked in white.
- Deselecting an object (point or a boundary).
  - ✓ Second selection of a selected object clears its selection.
- Selecting a group of objects (points or boundaries).
  - ✓ Consecutive selection a series of objects provides their multiple selections. To suppress multiple selection (i.e. to select a different object only when some other objects are already selected) the user should manually deselect them in advance.
- Filling a contour (assigning a closed contour to the current die layer).
  - ✓ In the Fill mode, the left mouse button is pressed inside of the contour.
- Specifying an ending point of a new boundary.

- ✓ New point: The left mouse button is pressed at a distance from any existing point. If a new point should be added near an existing point, the user should zoom in the image so that the point positions on the screen become far enough from each other.
- ✓ Selecting an existing point: The left mouse button is pressed in the vicinity of an existing point which starts flickering to identify this.

The color of an unselected point in the **Graphics Window** depends on the number of boundaries adjacent to it.

- If it is equal to 0 (an individual point) or 1 (an ending point of a dangling boundary), it is filled in red.
- Otherwise, it is filled in green.

The color of unselected boundaries (lines or arcs) forming a closed contour is brown. Individual unselected boundaries that do not form closed contour are marked in yellow.