

SpiceGUI Help

The SpiceGUI was developed at ISFH by Stefan Eidelloth. Please send bug reports and helpful suggestions to software@isfh.de

1. Installation.....	2
1.1 Description of the SpiceGUI starting process.....	2
1.2 Installation procedure.....	2
1.3 Alternative output terminals and formats.....	3
1.4 Alternative Interpreter and Spice tools.....	4
2. File handling with the SpiceGUI toolbar.....	4
3. Model specification.....	5
3.1 Indice range.....	5
3.2 Contact levels.....	5
3.3 Contact points.....	6
3.4 Element sizes.....	6
3.5 Scaling factors.....	7
3.6 Thermal voltage.....	7
3.7 Sub circuit models.....	7
3.7.1 Main panel.....	7
3.7.2 Resistor.....	8
3.7.3 Analytic_I.....	9
3.7.4 Analytic_II.....	9
3.7.5 JV-tables.....	10
3.7.6 Advanced Sub circuit.....	11
3.7.8 Build and edit matrices.....	13
3.8 Extra circuit elements on the Extra physics tab.....	14
4. Simulation.....	15
4.1 Voltage range.....	15
4.2 Batch parameters.....	15
4.3 Batch description.....	15
4.4 Solve and save simulation.....	16
5. Evaluation.....	16
5.1 Options.....	16
5.1.1 Ask_fem_range.....	16
5.1.2 Ask_fem.....	17
5.1.3 Eval @ MPP and Ask sol_num.....	17
5.1.4 Default output.....	18
5.1.5 2D.....	18
5.2 Batch evaluate.....	18
5.3 Evaluate.....	19
5.4 Plots.....	19
6. Figure handling.....	19
7. Script commands.....	19
8. Customization.....	20
8.1 Toolbar and File menu.....	21
8.2 Tabbed panels.....	22
8.3 Evaluation.....	22
8.4 Spice tool and model.....	22
8.5 User defined output formats.....	23

1. Installation

1.1 Description of the SpiceGUI starting process

- The SpiceGUI can be started with “**SpiceGUI.exe**”.
- This executable starts our **Octave** terminal and then runs the script “GUI.m”.
- “GUI.m” is the main script file of the SpiceGUI.
- The start command executed by “SpiceGUI.exe” can be found in the configuration file “setup.txt”:

```
C:\Program Files (x86)\Java\jre7\bin\javaw.exe  
-classpath .\javaclasses OctaveTerminal chdir('D:\SpiceGUIv2.0'); GUI
```

- This configuration file “setup.txt” is created during the installation procedure of SpiceGUIv2.0_Setup.exe, after asking for the Java path.
- The SpiceGUI.exe starts our Octave terminal and the Octave terminal starts the GUI.
- (The Octave terminal is required to enable the bidirectional communication between the java classes of the graphical user interface and Octave.)
- If you want to run the SpiceGUI with **Matlab** instead of Octave, please also see [section 1.4](#)

1.2 Installation procedure

- Install **Octave3.6.2** from <http://Octave.sourceforge.net> if you don't have Matlab available
 - The installation of the current Octave version requires several steps. If this bothers you, please feel free to create an Octave installer and share it with us.
 - Detailed Octave installation instructions:
http://wiki.octave.org/Octave_for_Windows
 - Download both Octave zip files from here:
<http://sourceforge.net/projects/octave/files/Octave%20Windows%20binaries/Octave%203.6.2%20for%20Windows%20MinGW%20installer>

The first zip file contains the Octave core, the second zip file contains the Octave Forge packages, which are required to run the SpicieGUI (e.g. the Octave forge java package).

- Unzip both zip files to a directory, e.g. to c:\dummyfolder
- Go to that folder and copy the "Octave3.6.2_gcc4.6.2" folder to where you want to have it on your hard drive, e.g. to c:\Octave\Octave3.6.2_gcc4.6.2
- (Delete the previously created dummy folder)
- Run octave, eg. with C:\Octave\Octave3.6.2_gcc4.6.2\bin\octave.exe
- Install packages: Enter each of the following commands and press Enter:
 - pkg rebuild -auto
 - pkg rebuild -noauto ad windows
 - pkg rebuild -noauto nan
 - pkg rebuild -noauto gsl

- pkg rebuild -auto java
 - Ignore following warnings at the startup of Octave (or try to install additional software; gmsh, dx, ... are not required for the SpiceGUI):
 - warning: gmsh does not...
 - warning: dx does not...
 - warning: function C ... fstat.m shadows ...
 - If the plotting software **Gnuplot** was already installed before the installation of Octave:
 - Check the Gnuplot path in your PATH system environment variable, e.g.: C:\Octave\Octave3.6.2_gcc4.6.2\gnuplot\bin
 - Delete old Gnuplot path from PATH system environment variable, e.g. C:\python27\gnuplot\binary
 - Create a new system environment variable GNUTERM (see next section) and set its value to “windows” (This tells gnuplot to use the “windows” **plot terminal**. If you want, have a look at the gnuplot documentation for other available terminals: <http://www.gnuplot.info/documentation.html>)
- Install **LTspice IV** from <http://www.linear.com/designtools/software/ltspice.jsp>
 - If you don't have a JRE, install the **Java Runtime Environment** from <http://www.java.com/de/download>
 - Install **SpiceGUI**: Run “SpiceGUIv2.0_Setup.exe”. The setup program will guide you through the installation of the SpiceGUI.
 - The installer will ask you to specify the directories of Octave, LTspice and Java.
 - After installation you can start the SpiceGUI with SpiceGUI.exe

1.3 Alternative output terminals and formats

- It is possible to change the gnuplot **output terminal** for standard plots: create a system environment variable “GNUTERM” and set its value to “windows”.
- You can create your own output formats (see [chapter 8.5](#)).
- You might want to use “.vsz” ([Veusz](#)) as output format
- Here is some help on environment variables: <http://vlaurie.com/computers2/Articles/environment.htm>



1.4 Alternative Interpreter and Spice tools

The Installer is written for the use of the SpiceGUI with Octave and LTspice. If you want to run the SpiceGUI with **Matlab** and **NGspice** or **SMARTspice** you might have to manually adapt some of the following configuration files for your platform after the installation of the SpiceGUI. (You can cancel the Octave path selection during installation.)

- {install_dir}\setup-matlab.txt (This is executed if you start SpiceGUI-Matlab.exe)
C:\Program Files\MATLAB\R2010a\bin\matlab.exe
-r "GUI;"
- {install_dir}\#1guiconfig\spice_tool.txt (LTspice, NGspice or SMARTspice)
NGspice
- {install_dir}\#1guiconfig\ngspice_path.txt
C:\NGspice\bin\ngspice.exe
- {install_dir}\#1guiconfig\smartspice_path.txt
C:\Silvaco\exe\smartspice.exe
- {install_dir}\SpiceGUI-Matlab (for Linux platforms)
export GNUTERM=wxt
export SPICE_ASCIRAWFILE=1
export SFLMMULTICOREONLY=1
/home/user/matlabR2010a/bin/matlab -r "run //home//user//SpiceGUI//GUI"
- If you want to run NGspice on windows, create a system environment variable SPICE_ASCIRAWFILE and set its value to 1.

2. File handling with the SpiceGUI toolbar

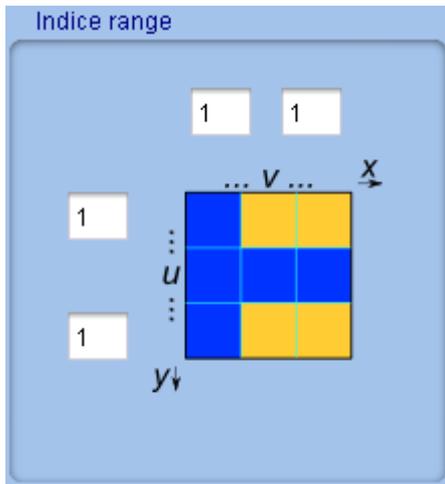
Following toolbar items represent file operations:

-  **Open:** Load simulation data from the default file (e.g. “Spice.data”), specified on the “Solve” tab (no file dialog box is shown).
-  **Save:** Save simulation options to current *.data file path (see “Solve” tab) and delete the current simulation results. (Simulation results can only be saved with the “Solve & Save !” button (found on the toolbar or on the “Solve” tab). This ensures that the saved simulation results correspond to the current simulation options.)
-  **Open from:** Open a file dialog box for selecting and opening a SpiceGUI *.data file.
-  **Save as:** Open a file dialog box for saving the current simulation options as *.data file. (The current simulation results are lost while saving the simulation options!)
-  **Explore:** Explore current file path. (The default path of the file selection dialogs is saved at any file selection task.)
-  **Help:** Open this help file. Please send bug reports and suggestions on how to improve this help file to software@isfh.de

3. Model specification

3.1 Indice range

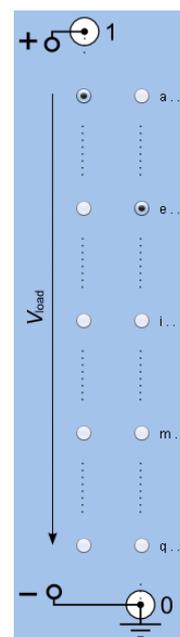
- The device area can be divided in **several elements** with the indices u and v .
- Please specify the range of the indices at the “**Indice range**” panel:



- Enter for example [1], [3], [1], [3] for a device with nine matrix elements.
- If you want to analyze a **limited region** of your device (without changing all parameter matrices), you might use starting indices that are not equal to 1, e.g.: [2], [3], [2], [3].
- The specifications on the “Indice range” panel, together with the element sizes on the “Element sizes” panel, define the **size of the simulated device**.

3.2 Contact levels

- Each element of the device is represented by an “element tree”.
- This “element tree” contains the **levels a, e, i, m and q**.
- The **load voltage** V_{load} has to be connected to two of these levels by activating the corresponding **radio buttons**.
- If you want to use for example the first two levels of the element tree, connect the main node 1 to the level **a** and the main node 0 to the level **e**.



3.3 Contact points

- At each contact level, you can activate **several contact points**.
- The load voltage will be directly connected to each contact point.
- Please specify the contact points at the “Contact points & thermal voltage” panel:

- In most cases the main node 1 will be connected to a “top” level and main node 0 to a “bottom” level.
- In the shown example both contact levels have a single contact point at $(u, v) = 1, 1$.
- The contact points can also be specified by a contact point matrix.
- If the value (u, v) of the matrix is 1, the load will be connected to the point (u, v) .
- If the value is 0, the point will not be connected.
- Here is an example for a 3x3 contact point matrix with two contact points:

1	0	0
0	0	0
1	0	0

- You can find help on how to edit matrices in [chapter 3.7.6](#)

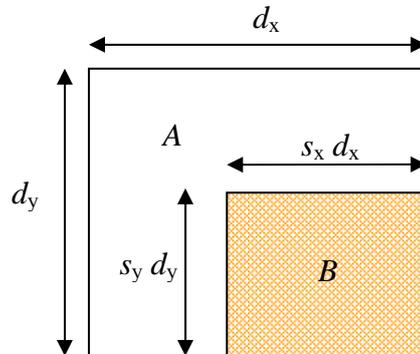
3.4 Element sizes

- The **element sizes** can be specified on the “Element sizes & scaling factors” panel:

- If you want that all device elements have the same x-width d_x , please activate “**constant**” at the d_x combo box and specify the single value in the text field.
- You can also specify a **vector of element sizes** $d_x(u)$, if you activate “vector”.
- Then the text field will be disabled and the vector “Edit” button will be enabled.
- If you want to run **several simulations** with different values for d_x , please activate the **batch** radio button.
- If “constant” is activated for batch simulations, you can enter the batch values $d_x(batch_number)$ as a vector in the text field, e.g. [1e-2 2e-2 3e-2], or [1 2 3]*1e-2.
- If “vector” is activated for a batch simulation, you can enter the batch numbers in the dialog that pops up if you click the “Edit” button.
- The y-widths d_y of the elements can be entered similarly.

3.5 Scaling factors

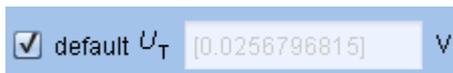
- The “**scaling factors**” are **disabled** per default.
- They get enabled if the **scaling option** is activated for a circuit element (see below.)
- The meaning of the scaling factors can be understood with the help of **two areas**:



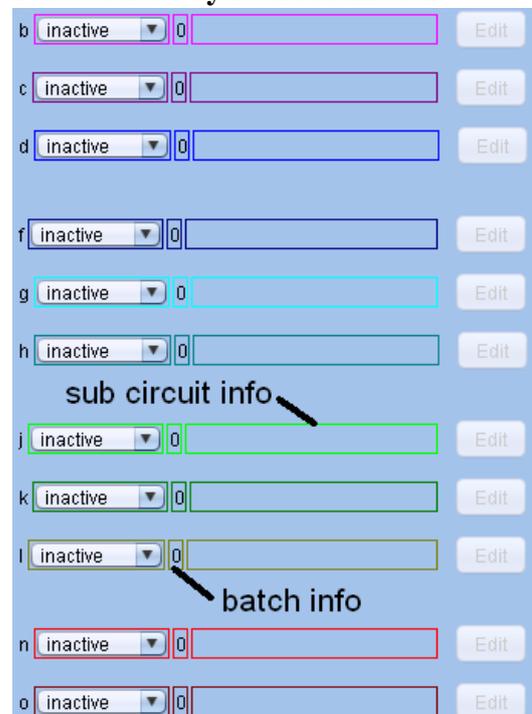
- The area **A** is the whole area of the element $\{u,v\}$, having the size $A = (d_x d_y)$.
- The area **B** is a part of the area **A**, having the size $B = [(s_x d_x)(s_y d_y)]$.
- If the scaling option is enabled for a current density j , the **element current** I will be calculated with $I = B j$ instead of $I = A j$.
- The sizes $(1 - s_x) d_x$ and $(1 - s_y) d_y$ can be used for scaling **sheet resistances**.

3.6 Thermal voltage

- The **thermal voltage** is $U_T = 0.0256796815$ V per default:



- This value applies for a temperature of 25 °C and can be changed for simulations at other temperatures.
- The thermal voltage U_T is used in circuit elements with **analytic diode models**.



3.7 Sub circuit models

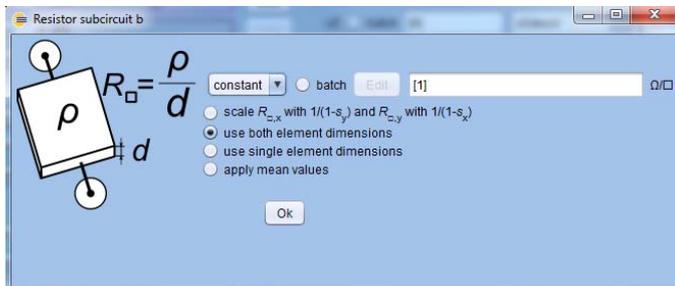
3.7.1 Main panel

- Each sub circuit (b c d f g h j k l n o) of the **element tree** can be edited:

- The sub circuits (b c f g j k n o) for lateral connections can be
 - inactive
 - **resistor** (see [chapter 3.7.2](#))
 - advanced (see [chapter 3.7.6](#))
- The sub circuits (d h l) for vertical connections can be
 - inactive
 - **analytic_I** (see [chapter 3.7.3](#))
 - **analytic_II** (see [chapter 3.7.4](#))
 - **jv-tables** (see [chapter 3.7.5](#))
 - **advanced** (see [chapter 3.7.6](#))
- If the batch mode is activated in a sub circuit, the **batch info** text field next to the sub circuit combo box shows the number of activated batch radio buttons (0, 1 or 2).
- The sub circuit info text field shows some **information about the current state** of the sub circuit.
- Please **close all sub circuit dialogs** before starting the simulation.

3.7.2 Resistor

- If you activate the “**resistor**” option for a sub circuit, you can edit the sub circuit with the “**Edit**” button. A resistor panel will pop up:

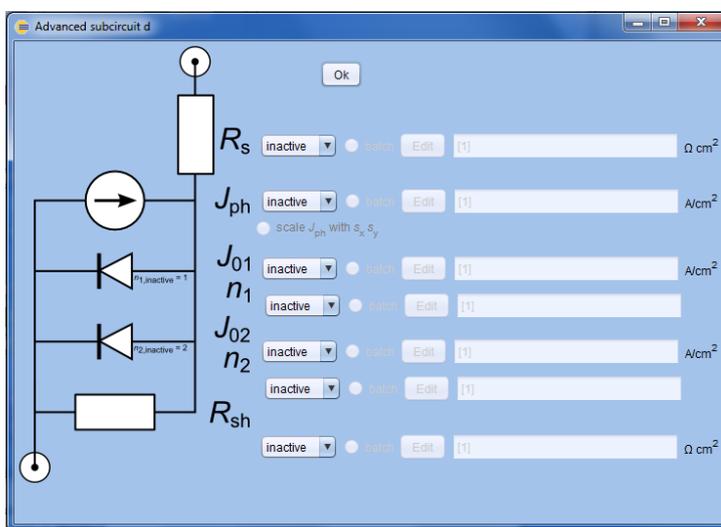


- You can specify a **constant** sheet resistance (“**constant**”) for all elements or a sheet resistance **distribution** (“**matrix**”).
- If you activate the **scale option**, the scaling factors on the “Physics” tab are enabled.
 - Use this advanced option if you want to be more flexible in the scaling of the sheet resistance with the element sizes.
 - See [section 3.5](#) for a description of the scaling factors.
- Independent from the scaling factors you have three possibilities to calculate the absolute value of the resistor that is used in the net list.
 - By default (“**use both element dimensions**”) the length for a resistor in x-direction is calculated with $l = (dx_1 + dx_2)/2$. The sheet resistance value is only taken from the current element. The absolute resistance is calculated with $R = R_{\square,1} l/b$.
Example: $dx = [1e-3 \ 10e-3] \Rightarrow l = 5.5e-3$
 - The option “**use single element dimensions**” only uses the size of the current element: $l_1 = dx_1$. The resistance is $R = R_{\square,1} l/b$. The sheet resistance value is only taken from the current element.
Example: $dx = [1e-3 \ 10e-3] \Rightarrow l = 1e-3$

- The option “**apply mean values**” takes the average, considering both element length and both element values.
 $l_1 = dx_1/2, R_1 = R_{\square,1} l_1/b,$
 $l_2 = dx_2/2, R_2 = R_{\square,2} l_2/b,$
 $R = R_1 + R_2$
- If the scaling factors are active, adapted element sizes are used.
- These options are only important if you use non constant element sizes and/or non constant sheet resistances.
- The calculations for the y-direction work similarly.

3.7.3 Analytic_I

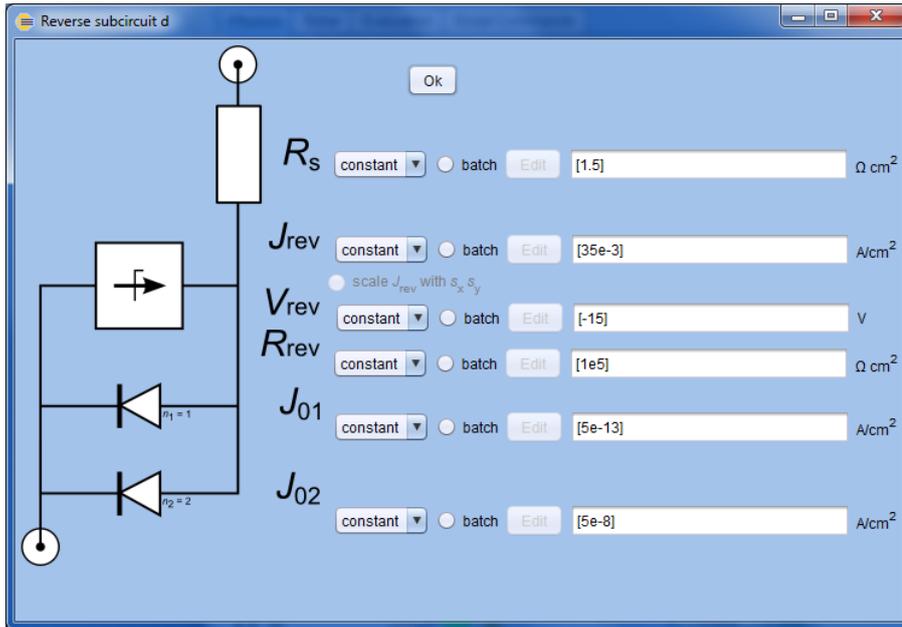
- The “**analytic_I**” sub circuit model supplies **several sub circuit elements**:



- Each of the sub circuit elements can have the combo box states “inactive”, “constant” or “matrix”
- If all elements are inactive, the input and output sub circuit nodes are not connected. The sub circuit is **open**.
- If the **series resistance** R_s is inactive, the other elements are directly connected to the input node.
- If the **shunt resistance** R_{sh} is inactive, the “shunt resistance connection” between the left and the right side of the sub circuit does not exist.
- If the **first ideality** factor n_1 is inactive, the first diode has an ideality factor of 1.
- If the **second ideality** factor n_2 is inactive, the second diode has an ideality factor of 2.
- The **generation current density** J_{ph} can be scaled with the scaling factors $s_x s_y$.

3.7.4 Analytic_II

- The “**analytic_II**” sub circuit model includes some **reverse bias behavior**.
- The **breakdown model** is from M.C. Alonso-Garcia, J.M. Ruiz, Solar Energy Materials & Solar Cells 90 (2006) 1105-1120
- Following parameters have to be entered:



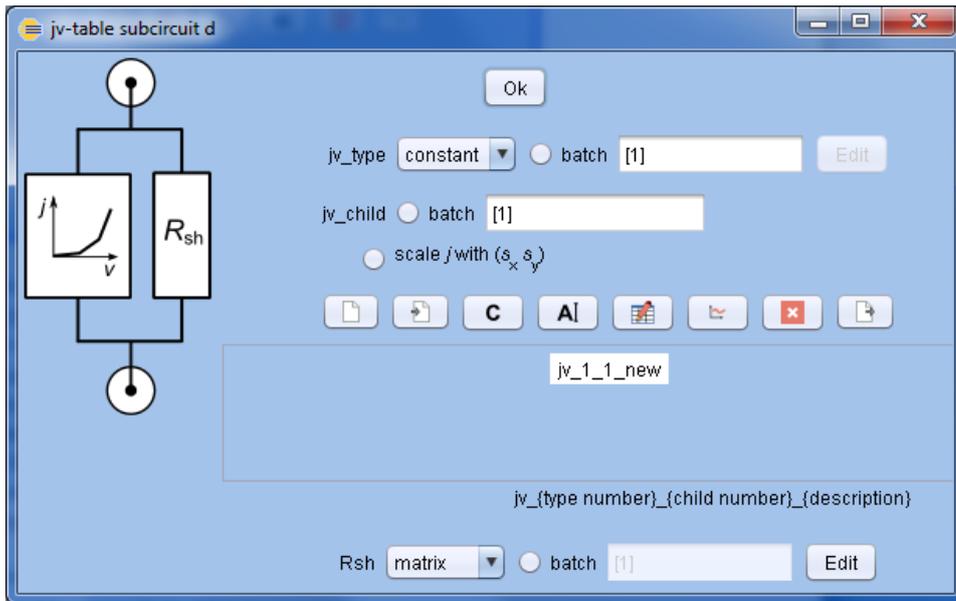
- The parameters J_{rev} , V_{rev} and R_{rev} describe a **voltage dependent current source** with following behavior:

$$I_{\text{rev}}(V_D) = \frac{\left(J_{\text{rev}} A - \frac{V_D}{R_{\text{rev}}} \right)}{1 - \exp \left\{ 3 \left(1 - \sqrt{\frac{0.85 \text{ V} - V_{\text{rev}}}{0.85 \text{ V} - V_D}} \right) \right\}}$$

- V_D is the voltage applied to the diodes in the above schematic in [V] and A is the element size in [cm^2].
- If the scaling option is enabled for J_{rev} , the term $J_{\text{rev}} A$ will be replaced by $J_{\text{rev}} A s_x s_y$, where $s_x s_y$ are the scaling factors from the “Scaling factors” panel.

3.7.5 JV-tables

- The jv-characteristics of the **vertical sub circuits** can be specified with **jv-tables**.



- The jv-characteristics are mapped to the elements with **jv_type** numbers.
- Each jv-table has a **type number** and a **child number**.
- The **syntax** for the jv-table names is `jv_{type number}_{child number}_{description}`
- If you want to use a single jv-table for all device elements (u, v), but **batch** this jv-table in several simulations, you can build some jv-tables with different child number and use a child number **batch**.
- If you want to **batch** the jv-table mappings, build several type number matrices and batch their matrix number.
- You can use following **buttons** for building and editing jv-tables:

	Build new jv-table
	Import jv-tables from *.xls files
	Copy jv-table
	Rename jv-table
	Edit jv-table with Calc or Excel
	Plot jv-curve
	Delete jv-table
	Export jv-table as *.xls file

- A resistor Rsh can be used parallel to the jv-curve.

3.7.6 Advanced Sub circuit

- The “advanced” sub circuit model can be used to develop new models.

- The SPICE commands can be directly entered in a script window.
- Sometimes this might be less effort than building user defined jv-tables.
- The “advanced” sub circuit model has **six parameters** that can be included in the **script**.
- These parameters (p1...p6) can be batched like the other GUI variables.
- Additionally to these six parameters you are able to access the element size with “dA”. (You are not able to use “dA” in the extra elements on the “Extra physics” tab. They do not have an associated element size.)
- You are also able to create additional parameters, e.g. with `.param UT={0.0256796815}`
- You are not able to use the SpiceGUI variables in the script: `r1 100 200 {v1}` does not work
- You are able to use the SpiceGUI variables in the parameter values: `r1 100 200 {p1}`, `p1 = v1`, `v1=1` does work
- Please see the **LTspice documentation** (e.g. page 98, “Circuit Elements”) <http://ltspice.linear.com/software/scad3.pdf> for more information on LTspice commands.
- If you don’t use LTspice but another spice tool, the script language will be slightly different and the examples in the default text might not work.

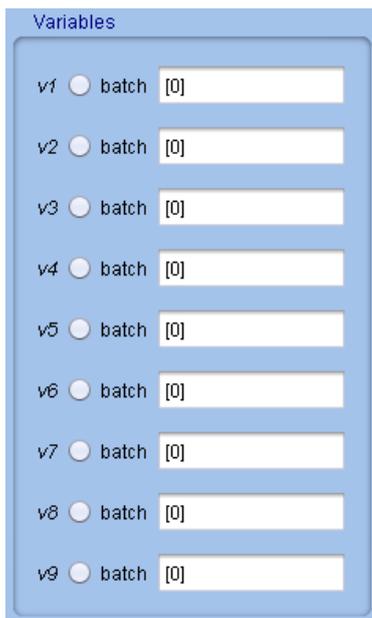
3.7.7 Variables

- It is possible to enter values **as function of variables**, e.g.

$$R_{\square} = v1 ./ v2 .* v3 + v4$$

(Please be aware of the points in front of “*” and in front of “/” for the **element-wise** operators that operate on the variable vectors.)

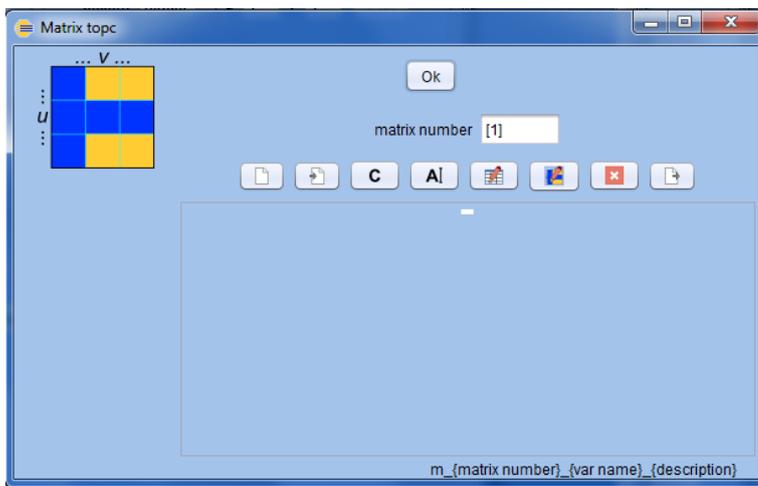
- The values of these variables can be entered at the “Variables” panel:



- The values of the variables can also be batched.

3.7.8 Build and edit matrices

- If you activate the “vector” or “matrix” state of a sub circuit element and hit the “Edit” button, a window like this will pop up:



- You can use following **buttons** to create and **edit matrices**:

	Build new matrix
	Import matrix from *.xls, *.tiff or *.m files
	Copy matrix
	Rename matrix
	Edit matrix with Calc or Excel
	Edit matrix with a special matrix editor
	Delete matrix
	Export matrix

- The **syntax** for the matrix name is `m_{matrix number}_{var name}_{description}`
- You can build several matrices with different matrix numbers and use a “matrix number batch” to simulate several value distributions in a **batch** simulation.
- Here is an example for a *.m file that can be imported:

```
%save this text as Matrix.m and import it  
matrix = zeros(3,3);
```

```

matrix(1,1)=1;
matrix(3,1)=1;
result__matrices{1}=matrix;
result__names{1}='m_1_imported';

result__matrices{2}=zeros(3,3);
result__names{2}='m_2_imported';

%msgbox('Hello world');

```

After evaluating the text in the *.m file, the value of “**result_matrix**” will be **imported** as matrix.

- It is possible to use the variables to specify a matrix value.
- A matrix value "inf" has a special meaning:
 - If a matrix value of the "resistor subcircuit" is set to "inf", the sub circuit call for the infinite resistance Rsheet(u,v) is not included in the LTspice main net list.
 - If you use the analytic_I subcircuit, only activate the series resistance and set a matrix value to "inf", the sub circuit call for the infinite resistance is not included in the LTspice main net list.
- Example for a 1x10 resistance matrix:

v1	v1	1e-3	1e-3	inf	inf	inf	inf	inf	inf
----	----	------	------	-----	-----	-----	-----	-----	-----

3.8 Extra circuit elements on the Extra physics tab

- On the “**Extra physics**” tab you can add additional circuit elements to the network.
- These elements can **connect any existing nodes** of the already created network. They are not limited to the rectangular crystal-like structure that is spanned by the reproduction of the unit tree. Therefore, you are able to make “**diagonal connections**” and to “**jump over several nodes**”.
- The extra elements do not scale with element sizes. All parameters are given as **absolute values**. For example, there are no “jv-tables”, but “iv-tables”.
- The start- and end-nodes are specified with a **coordinate table**, e.g:

start_level	start_u	start_v	end_level	end_u	end_v
a	1	2	e	1	1
a	1	1	e	1	2

- Each row in the table corresponds to a single instance of the circuit element.
- The power dissipation of the extra elements is included in the loss analysis (FELA).
- In order to compare power densities in the loss analysis, the **power dissipation** of the extra elements is **scaled with the total area** of the device.
- The power dissipation of the extra elements is **not** included in (x,y)- or (u,v)- plots, since it is not possible to generally associate the extra elements with a specific location (level, u, v).
- The power dissipation of the extra elements can only be calculated, if the voltages at the start nodes and the end nodes are known. The **voltages are only measured for the nodes that are created by the rectangular network of the “Physics” tab**. => If you

specify non-existing start- or end-nodes in the coordinate table, the spice simulation might run, but the evaluation of the power dissipation fails.

- If you use the “**advanced**” mode for an element on the “Extra physics” tab, you are not able to use an element size parameter “dA” in your script, since no element area is associated with the extra elements.

4. Simulation

4.1 Voltage range

- The **voltage range** specifies an Octave vector containing the voltage values of the simulated jv-curve(s).
- Advanced users have the possibility to disable the **default** range and enter custom voltage values.
- The vector starts with the highest voltage value.
- **Octave vectors** can be entered by a list of single values in square brackets (separated by “,” or “”), e.g. [0.7 0.6 0.3]. Octave vectors can also be specified with commands like “logspace(...)”. You can enter “help logspace” in the Octave prompt for further help on how to use this command.
- If the default range **checkbox** is enabled again, the custom values are lost and the default vector is entered automatically.
- The default range can be changed in the text file “**defaultrange.txt**” in the subfolder “#1guiconfig”.

4.2 Batch parameters

- The names, values and units of the **current batch parameters** are shown on the “Batch paramters” panel on the **Solve tab**.
- Use the **Physics tab** to change the batch parameters.
- Selecting batch parameters results in a “**batch matrix**”. If you enter two values for the first batch parameter (e.g. dx = “[1e-2 2e-2]”) and three values for the second batch parameter (e.g. dy = “[1 2 3]*1e-3”) you will get a 2 (rows) x 3 (columns) “batch matrix”:

{1e-2, 1e-3}	{1e-2, 2e-3}	{1e-2, 3e-3}
{2e-2, 1e-3}	{2e-2, 2e-3}	{2e-2, 3e-3}

- Each element {...} of this matrix (“batch element” or “fem”) is going to represent a set of simulations, a related JV curve and corresponding JV characteristics.

4.3 Batch description

The batch **description** is a custom text, saved with your simulation data. You can enter the name of the simulation author, the purpose of the simulation, and so on.

4.4 Solve and save simulation

- Click the “**Solve & Save!**” button on the **Solve tab** or the  button in the SpiceGUI **toolbar** to start the simulation.
- All simulation results are included in an Octave structure called “dat” in the Octave workspace. This data structure can be saved as “*.data” file and it can be accessed by all post processing functions (Enter “dat” in the Octave prompt to see the dat structure or see [chapter 5](#) for more information.)
- While solving, the Octave prompt shows some information about the **solution progress**, e.g. the start time and an estimated time of completion.
- A message will pop up when the batch simulation is **finished**.
- Following icon **buttons** are available on the “Solve” tab:



Open a file dialog for selecting a *.data file. The path of the file is written to the text field and used for saving the simulation results.



Open the *.data file specified in the text field.



Save the simulation data without solving (using the file path specified in the text field).

- Please activate the check box near the  symbol if the computer should **shut down** after completion.
- If you activate the check box near the **LTspice** symbol , LTspice will be opened and show the status of each simulation. You will have to close LTspice for each simulation after the simulation is finished and the LTspice status bar shows “**ready**”.

5. Evaluation

The “Evaluation” tab contains several panels:

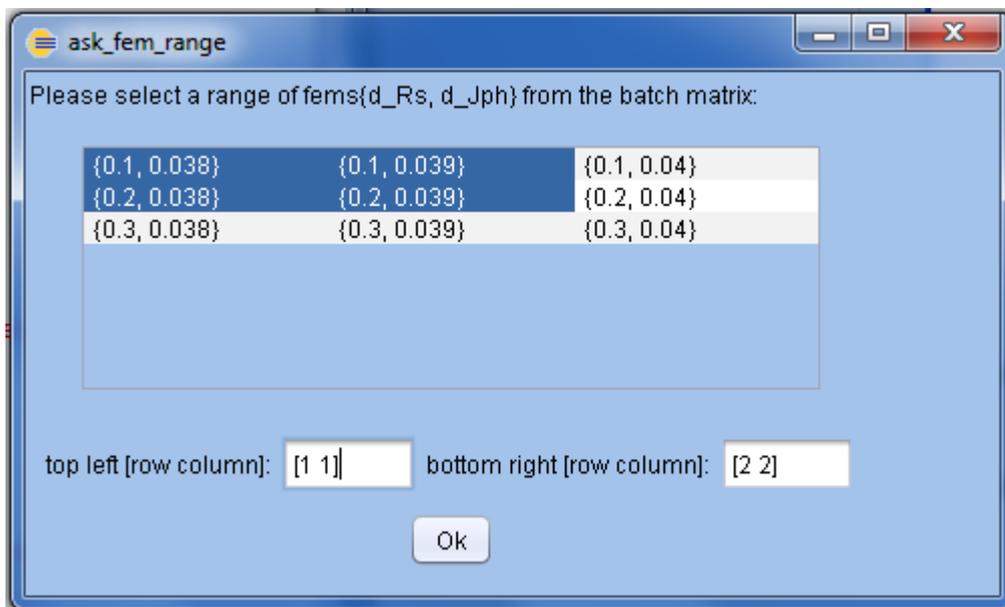
5.1 Options

- The “Options” panel can be used to modify the post processing functions.
- The **post processing functions** require some **input arguments**.
- The user can be asked for these arguments each time he runs a post processing function.
- The function arguments can also be specified directly on the “Options” panel. This **disables the user input popup windows**.

5.1.1 Ask_fem_range

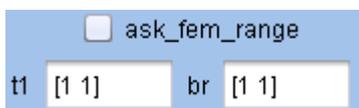
- The functions for evaluating **several batch elements** require a batch element **range** as input parameter.
- This range is specified by the coordinates of the top left (tl) element and the coordinates of the bottom right (bl) element of the **batch matrix**.

- If the “ask_fem_range” **check box** is enabled, the user can select the range from a pop up window at each start of the post processing function. This pop up window shows all batch elements with corresponding patch parameter values:



After selecting a range, the coordinates are transferred to the “Option panel” and the post processing function starts.

- If the “ask_fem_range” is disabled, the range selection window is not shown and the range has to be entered in the **text fields** of the “Options” panel:



- If there is only a **single batch element** or if the post processing function does not require a batch element range, the range selection window does not pop up.

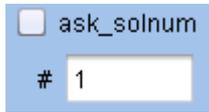
5.1.2 Ask_fem

- Some post processing functions require a single batch element as an input argument and not a batch element range as described above.
- The functionality of the “**ask_fem**” **check box** is similar to the “ask_fem_range” check box (see [section 4.1.1](#)), but for **single batch elements**.

5.1.3 Eval @ MPP and Ask sol_num

- If the “eval @ MPP” check box is selected, the evaluation will be done at the voltage of the Maximum Power Point. If it is not selected you will be asked for the voltage:

- If the “ask_solnum” check box is selected, a **voltage selection** window pops up at the start of some post processing functions.
- Each voltage of the evaluation range corresponds to a single solution of the LTspiceIV simulation.
- If the “ask_solnum” check box is not selected, the solution number has to be entered in the **text field** of the “Options” panel:



5.1.4 Default output

- If the “**default_output**” check box is selected, the results of the post processing functions are shown in Octave.
- If the “default_output” box is not checked, some **alternative output formats** can be selected:



- The corresponding Octave functions (e.g. output_data_ods.m) can be found in the subfolder [guifunctions](#). Please see [chapter 8.5](#) if you want to define your own output format.
- For some formats, e.g. “*.vsz” (Veusz), “*.jnb” (SigmaPlot) some additional software might be required.

5.1.5 2D

- This option is not used yet.
- Future post processing functions may access the option “**2D**” to show a two dimensional plot with a color bar instead of a three dimensional “height data plot”.
- This option can only be selected, if the “default output” box is checked.

5.2 Batch evaluate

The panel “Batch evaluate” of the “Evaluation” tab shows some post processing functions for evaluating **several batch elements**:

- “Eff_FF_Voc_Jsc” shows the **jv-characteristics** as function of the batch parameters.
- “JV_curves” can show several **jv-curves** in logarithmic and linear plots.

5.3 Evaluate

The panel “Evaluate” of the “Evaluation” tab shows some post processing functions for evaluating **single batch elements**:

- “JV_curve” shows the current density values as function of the voltage of a single batch element.
- “LTspice” lets the user choose a folder, generates LTspice files in that folder and opens the files in LTspiceIV. A plot settings file “LTspice_plot_settings.plt” is also generated and can be opened from within LTspiceIV.

5.4 Plots

Further evaluation options are shown on the “Plots” panel.

6. Figure handling

There might be some more features in future.

7. Script commands

If you need a functionality that is not implemented yet, you can write your own **Octave scripts**. The “**Script Commands**” tab has two editor areas where you can edit and run Octave scripts. You can also work in the Octave terminal or use the Octave editor (enter “**edit** <function_name>” in the Octave terminal, e.g. “edit output_data_usr”).

- Enter “dat” in the Octave terminal to show the simulation **data structure** (may take some time).
- The command “dat.JVcharact{1,1}” will show you the JV characteristics of the first batch element:

```
ans =
  Voc: 0.6697
  Jsc: -5.3347
  Vmpp: 0.5310
  Jmpp: -5.1699
  FF: 0.7684
  Eff: 0.0027
```

- You may use similar commands to access the **fem structures** (“i = 1; j = 1; dat.fems{i, j}”), the **JVcurves** (e.g. “dat.JVcurves{1,1}”), ...

- Each substructure in the block “---Simulation data -----” has the format of the **batch matrix**:
Batch matrix format: 3x3 elements => Format of “JVcurves”: 3x3 elements.
- Here is an **example script** for plotting a JV curve:

```

if exist('dat')
jv=dat.JVcurves{1,1};
%get JVcurve
x=jv(:,1);      %get voltage
y=jv(:,2);      %get current
plot(x,y,'o-'); %plot JVcurve
else
msgbox(['JVcurve is not available. ' sprintf('\n') ...
'Please use the Solve tab or load a *.data file.'], 'Error', 'warning');
end

```

- There is also a structure “**gui**” in the Octave workspace. It contains text fields, buttons and some other elements and data of the SpiceGUI.

8. Customization

- The layout and functionality of the SpiceGUI is based on several configuration files. If you want to add extra buttons to the **toolbar** you can do so by editing the corresponding *.csv file toolbar_objects.csv.
- All subfolders of the SpiceGUI installation folder starting with “#” include files that can be adapted for GUI customization.
- The other folders include more general Octave functions that should not be changed in a first attempt.
- All subfolders of the SpiceGUI installation folder are included in the Octave path (see GUI.m) and all functions in those folders can be called from within Octave.
- Here are some **configuration files**:

Purpose	Files and folders
Toolbar	toolbar_panels.csv, toolbar_objects.csv, guifiles
Menu	menu_panels.csv, menu_objects.csv, guifiles
Tabbed panels	tabbed_panels.csv, guifiles
Default range	defaultrange.txt
Default description	description.txt
Evaluation	eval_panels.csv, eval_objects.csv
Script Commands	#6scriptfiles
Name of GUI	GUIname.txt
Application Info	info_text.txt
Images and icons	ltspice.svg, icons.svg
Simulation	LTSPICE IVsimulation.m, #2simulationfiles
Batch	run_batch.m
Time estimation	single_sim_time.txt
Installer script	inno_setup_script.iss
SpiceGUI start	setup.txt

8.1 Toolbar and File menu

If you want to customize the SpiceGUI **toolbar**, edit `toolbar_objects.csv`. This `*.csv` file might look like this:

toolbar_item	label	tooltip	icon	action m-file	toolbar
Open		Open...	open.png	sol_action	standard_toolbar
Save		Save	save.png	sol_action	standard_toolbar
Explorer		Open Explorer	explorer.png	explore	standard_toolbar

Each row represents a toolbar item:

toolbar_item: Identifier of the item. It is also passed to the action m-file as an argument “info_string”. If you enter “Seperator” as menu_item and leave the other fields empty, a separator will be drawn in the toolbar.

label: Label of the toolbar button.

tooltip: Tooltip that is shown if the mouse pointer hovers the toolbar button.

icon: Name of a `*.png` file in the “#1guiconfig” folder that is shown as image icon.

action m-file: An Octave function that is called when the user clicks the toolbar button. Here is an action m-file example:

```
function sol_action(handle, event, info_string)
if isequal(info_string, 'Open')
    msgbox('Open');
else if isequal(info_string, ' Save')
    msgbox('Save');
end
end
```

toolbar: The toolbar in which the toolbar item is shown.

If you need another toolbar, add a row to `toolbar_panels.csv`:

toolbar
standard_toolbar
custom_toolbar

You can specify the new “custom_toolbar” for some toolbar items in `toolbar_objects.csv` instead of “standard_toolbar” in the column **toolbar**.

The File menu can be customized with `menu_panels.csv`:

(sub)menu	label	mainflag	submenu
File	File	true	
Help	?	true	

and menu_objects.csv:

menu_item	label	(sub)menu	icon	action m-file
Open	Open	File	open.png	sol_action
Save	Save	File	save.png	sol_action
Separator		File		
Info	Info	Help	help.png	help_action

It is also possible to specify submenus in menu_panels.csv: Change the **mainflag** of the submenu to “false” and add an extra row to the table, including the parent **(sub)menu** and the **submenu**:

(sub)menu	label	mainflag	submenu
File	File	true	
Help	?	false	
File			Help

8.2 Tabbed panels

The **main layout** of the SpiceGUI can be customized with tabbed_panels.csv:

panel	label	layout	m-file
physics	Physics	PAGE_AXIS	phy_tab
solve	Solve	PAGE_AXIS	sol_tab
evaluation	Evaluation	LINE_AXIS	eval_tab
scripts	Script Commands	PAGE_AXIS	scr_tab

If you want to remove a tab, remove the corresponding row in that table. If you want to add a custom tab, add a new row and write a corresponding Octave file that includes the source code for that tab. The Octave files for the tabs can be found in the folder “guifiles”.

8.3 Evaluation

- The objects of the “**Evaluation**” **tab** are also specified in *.csv files.
- Each tab can have several panels and subpanels
- The objects are specified in post_objects.csv.

8.4 Spice tool and model

- The definition of the Spice networks can be found in the Octave file **LTSPICEIVsimulation.m** in the subfolder **#2simulationfiles**.
- This function is called from **run_batch.m**.
- If you want to use other SPICE tools, change the file **spice_tool.txt** in the folder **#1guiconfig**. There is support for LTspice, NGspice and SMARTspice.
- Please make sure that the output *.raw files of your SPICE tool are in ASCII format. For SMARTspice you have to create an environment variable `SPICE_ASCIRAWFILE=1`
- If you want to use another SPICE tool, you have to write your own net list generation function. You can start with **LTSPICEIVsimulation.m** and adapt according to your needs.

8.5 User defined output formats

- If the “default output” check box on the “Options” panel is disabled, **alternative outputs** can be selected, e.g. “*.ods”.
- Please have a look at “output_data_usr.m” and “output_data_pdf.m” if you want to create your own **output formats**.
- The output functions take two input arguments:
- The input argument “**file_name**” is a default name for the file selection dialog box.
- The input argument “**data_matrix**” is a cell array including string data that is going to be saved, e.g:

```
data_matrix =
{
  'Ua [V]'  'J [mA/cm^2]'
  '0'      '-7.8134'
  '0.1'    '-7.8134'
  '0.2'    '-7.8134'
  '0.38478' '-7.8127'
  '0.63019' '0.72243'
  '0.70449' '78.1293'
}
```