

# SEQUEL: using the GUI

Mahesh B. Patil

Department of Electrical Engineering  
Indian Institute of Technology, Bombay  
Mumbai-400076, India  
e-mail: [mbpatil@ee.iitb.ac.in](mailto:mbpatil@ee.iitb.ac.in)

# 1 Introduction

The purpose of the SEQUEL GUI is to make it easier for the user to enter the circuit schematic, specify analysis options, and view the simulation results. Using the GUI involves the following steps:

- (a) entering the circuit schematic
- (b) preparing the solve sections
- (c) generating the circuit file
- (d) running SEQUEL on the circuit file
- (e) viewing the results

When the GUI is started, one gets the display shown schematically in Fig. 1. Of the various

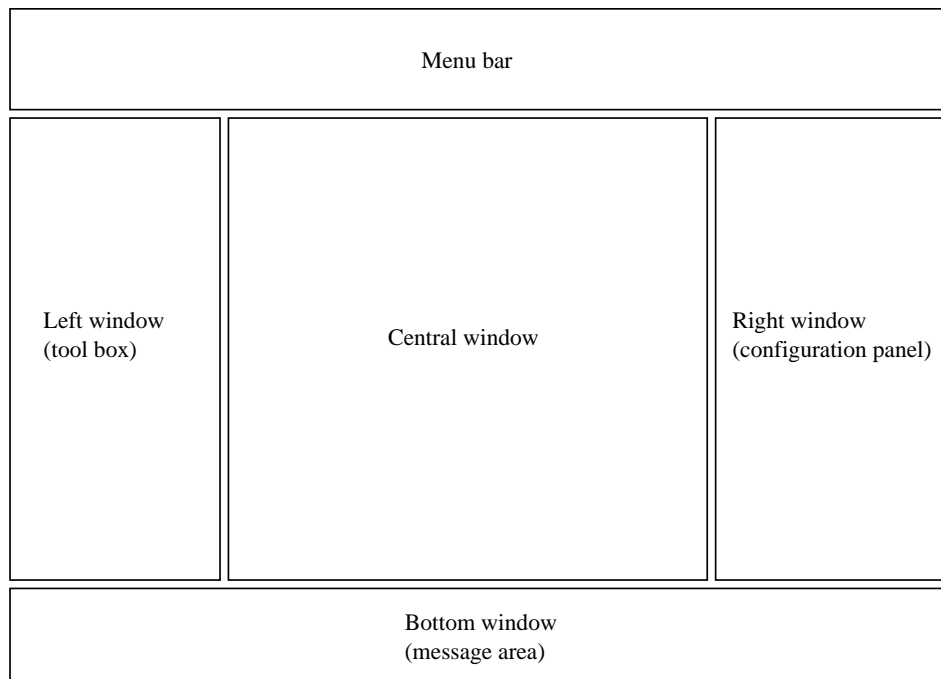


Figure 1: Schematic diagram of the GUI window.

windows, the menu bar remains fixed, but the contents of the other windows may change, depending on the context. For example, when the “circuit editor” menu is selected, a canvas for drawing the circuit schematic appears in the central window. When “graphs” is selected, plotting options appear in the central window, and so on.

The best way to get started with the GUI is to run an existing example and view the results, by following the steps below:

- (i) Click<sup>1</sup> on the “open project” menu. A new window will appear, showing the directory structure and files in the current directory, Select the directory<sup>2</sup> `electronics_gnr1`. The SEQUEL projects available in this directory (each with extension `.sqproj`) will appear in a separate window. Click on the project `rc1.sqproj`.
- (ii) Click on the “circuit editor” tab to view the circuit schematic. You may single-click on any of the components to view its parameters in the right window, by clicking on the tab “property editor” there. You could also view the list of output variables that have been selected for this project, by clicking on the “output variables” tab in the right window.
- (iii) Click on the “solve blocks” tab to view the details of the analyses which have been requested for this circuit. In the “output blocks” of the solve sections, you will find a list of output variables, a subset of those you would have seen in (ii) above. On execution, SEQUEL would make these variables available in the file specified in the output block. The output file is written to the directory `/SequelGUI2_Release/SGUI/Output`.
- (iv) Click on the “circuit file” tab and select “generate CF.” The circuit file corresponding to the circuit schematic and solve sections seen in (ii) and (iii) will appear in the central window. The circuit block in the circuit file (between `begin_circuit` and `end_circuit`) corresponds to the circuit schematic, and the solve blocks (between `begin_solve` and `end_solve`) to the solve sections. The circuit file may also be written to the hard disk by clicking on the “save CF” button; however, it is not required for running SEQUEL.
- (v) Run SEQUEL on the circuit file generated in (iv) by clicking on the “run solver” button of the menu bar<sup>3</sup>. The messages that the solver may generate while executing the circuit file are displayed in the bottom window and can be seen by clicking on the “solver output” tab of the bottom window. The messages include matrix size, iteration number (for transient analysis), etc., and error messages, if any, from the solver. When execution is completed, you will see the message, “SEQUEL: program completed.” In this particular example (`rc1.sqproj`), which is a small circuit, the execution is very quick, and you will see the program completion message instantaneously.

When the program is completed, the output data requested by the user are written to the output files, as described in (iii).

<sup>1</sup>By “click,” we would generally mean “left-click,” unless specified otherwise.

<sup>2</sup>The exact path for this directory will depend on where the SEQUEL GUI has been installed. This comment also applies to other directories mentioned in this document.

<sup>3</sup>The “run solver” button generates the circuit file and then invokes the simulation engine. In that sense, the “generate CF” step described above is not really required separately.

- (vi) Now click on the “graphs” button. The central window will turn into a three-column format. In the left column, you will see the names of the output files requested in the solve blocks (see (iii) above). When you select the desired file, the names of the variables contained in that file will appear in the second and third columns. One of the variables from the second column can be selected as the  $x$ -axis, and one or more from the third column as the  $y$ -axis variable(s). Clicking on the “graph it” button would plot the requested graph. If another plot is desired, click on the “back” button, and repeat the above procedure.

## 2 A Tutorial Example

Having seen the basic functioning of the GUI, we will now construct a SEQUEL project from scratch. In particular, we will simulate the circuit shown in Fig. 2, and plot  $i$ ,  $V_1$ ,  $V_2$  versus time.

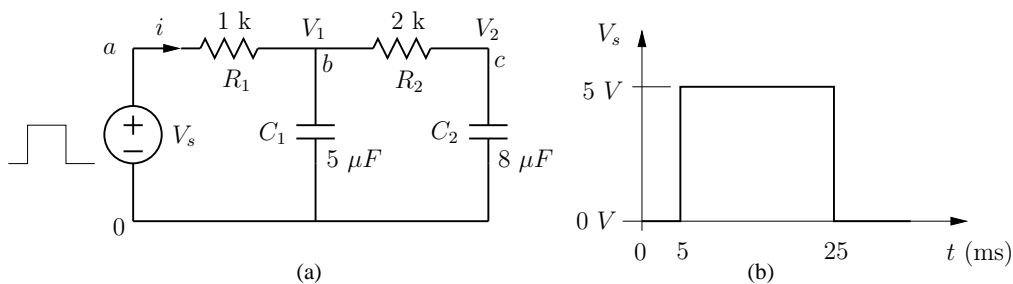


Figure 2:  $RC$  circuit for the tutorial example.

### 2.1 Creating the circuit block

SEQUEL allows the following types of elements:

- (i) electrical compound elements (**ece**)
- (ii) digital compound elements (**dce**)
- (iii) general compound elements (**gce**)
- (iv) thermal compound elements (**tce**)
- (v) general mixed elements (**gme**)

In the circuit of Fig. 2, only elements of type `ece` are involved, viz., resistors (`r.ece`), capacitors (`c.ece`), and a pulse voltage source (`vpulse.ece`). Click on the circuit editor tab. The central window will turn into a canvas for drawing the circuit schematic, and a list of the available types of elements will be displayed in the left window. Expand the electrical element list (`ece`) and click on `r.ece`. Take the cursor to some location on the canvas and double-click. A resistor will appear in the canvas.

Select the “property editor” option in the right window. Now, go back to the canvas and single-click on the resistor. The resistor will get highlighted, and a list of properties of `r.ece` will appear in the right window. Change its name<sup>4</sup> to `r1` and its resistance (which appears as a real parameter in the list) to `1k`.

Following the above procedure, create the other required elements, and edit their properties as desired. For the voltage pulse, we can let the pulse start at  $t_1 = 5$  ms and end at  $t_1 = 25$  ms. The relevant parameters are therefore `t_1=5m`, `t_2=25m`, `v_1=0`, and `v_2=5`. The parameters `delt_1` and `delt_2` represent the rise/fall time of the pulse and can be set to a suitable (small) number, say, `10u`.

Rotate the capacitors clockwise through  $90^\circ$ . This can be done by selecting the element and typing `r` or by setting “rotation” to  $90^\circ$  in the property editor window.

Place the elements suitably by either (a) clicking and dragging or (b) clicking, releasing the mouse button, and moving the selected element with the arrow keys. Note that, if an element is selected (i.e., highlighted), it can be de-selected by simply left-clicking on an empty area of the canvas.

We are now ready to make the required connections. To connect `node 1` to `node 2`, click on `node 1`, release the mouse button, take the cursor to `node 2` or, if necessary, to a grid point where a bend in the wire is desirable. Continue this process until you arrive at node 2. The GUI will automatically exit the wiring mode when you click on `node 2`. If you are in the middle of making a connection and would like to abandon the present wire (without completing the connection), press F4, and the GUI will exit the wiring mode.

After completing the connections, we need to inform SEQUEL about the node that needs to be designated as the “reference node.” For this purpose, bring the element `ground.ece` on the canvas and connect it to the desired wire or element node.

As with element names, the GUI generates a default name for each node. The program would run with the default names. However, for convenience, the user may want to assign node names which are more meaningful. This can be done in two ways:

<sup>4</sup>The GUI generates a default name for each element, and that is good enough for running SEQUEL. However, for easy identification, it may be desirable to assign familiar names to some of the elements.

- (i) Click on the wire (node) and edit its name in the right window (make sure that the “property editor” tab is selected in the right window).
- (ii) Click on an element to which the node is connected and edit the node name for that element in the right window.

If a node name is changed in the above manner, the node assignments for *all* elements sharing that node are updated simultaneously.

For the circuit in Fig. 2, we can assign the node names **a**, **b**, **c**, **0** as shown in the figure. A convenient way to figure out the name (default or assigned) of a node or an element is to keep the cursor (*without* clicking a mouse button) on the concerned node (or element) for two seconds or so. The name will then appear in a box near the cursor<sup>5</sup>. You can check this out for your circuit in the canvas.

The final step in completing the circuit description is definition of the output variables. In our example, we want to define the current through  $R_1$ , and node voltages at **b** and **c** as output variables. This can be done as follows<sup>6</sup>.

- (i) Click on the “output variables” tab in the right window and select “add variable.” The GUI is now in the “add output variable” mode. Click on the resistor **r1**. A menu will appear, showing **i1**, **v1**, etc. Select **i1**. The GUI will automatically exit the “add output variable” mode.
- (ii) Enter the “add output variable” mode and click on node (wire) **b**.
- (iii) Enter the “add output variable” mode and click on node (wire) **c**.

As you execute the above steps, you will notice that the GUI has generated the following lines in the right window:

```
var1=i1_of_r1
var2=nodev_of_b
var3=nodev_of_c
```

The names **var1**, **var2**, **var3** have been generated by the GUI as default names. You could change these to more meaningful names by double-clicking on the concerned line and typing the desired name. For example, we could use **i1**, **v\_b**, **v\_c**, respectively, instead of **var1**, **var2**, **var3**.

<sup>5</sup>If it is an element, the element type will also appear in the box.

<sup>6</sup>There is also an “expert” add variable mode which one can enter by clicking on the “Output variables” tab while keeping the **cntrl** key pressed. In that case, the user can keep adding output variables one after another and finally, click on the “Add variable” button once again to exit the expert add variable mode. Exiting the mode is rather important!

## 2.2 Creating the solve blocks

We have now completed the circuit to be simulated. Next, we must inform the simulator about the analyses that should be performed on the circuit. That is the purpose served by the solve sections.

For the circuit of Fig. 2, there are two analyses that we would like to perform:

- (a) Start-up analysis to obtain the solution at  $t = 0^-$ .
- (b) Transient analysis to obtain the solution for  $t > 0$ .

Click on the “solve blocks” button. The central window can now be used for creating and editing solve blocks, as follows:

- (a) Click on the “add solve block” button. This will display a solve block. Make sure that the “property editor” tab is selected in the right window. Click on the title bar of the solve block. In the right window, you can now select the type of the solve block (**Start-Up**) in the drop-down menu that appears after clicking on the bar showing “DC.” Next, we need to select an option for the initial solution to be used for this block. Do this by dragging `initial_sol` from the left window into the solve block. The default value for `initial_sol` is `initialize`, and it is appropriate for this solve block.
- (b) Add another solve block and change its type to **transient**. Drag `initial_sol` from the left to the current solve block in the central window, click on it, and select **previous** in the right window<sup>7</sup>. This will add the assignment `initial_sol=previous` in the solve block. Drag `back_euler` and set it to **yes**, by clicking on it in the right window and then toggling the square box there. Drag `t_start` and set it to 0 in the right window. Using the same procedure, make the following assignments:

```
t_end=100m
delt_const=10u
```

The above choices will instruct SEQUEL to perform transient analysis from  $t_{\text{start}} = 0$  to  $t_{\text{end}} = 100$  ms, with a constant  $\Delta t = 10 \mu\text{s}$ , using the backward Euler method. We now need to specify the names of the output files and the output variables to be written to each of the output files. Say, we decide to write the node voltages `v_b` and `v_c` to a file called `rc_1.dat` and the current `i1` to another file called `rc_2.dat`. This can be specified as follows.

<sup>7</sup>Many of the options for solve blocks are specified in this manner, viz., by dragging a keyword from the left window to the central window, clicking on it, and then selecting an appropriate option in the right window.

Drag “output block” from the left window to the central window, and click on it. The right window will show the various options that may be set for the output block. Set `filename` to `rc_1.dat`. Click on the “output variables” field. This will display a list of the output variables (which we defined while creating the circuit block). Select `v_b` and `v_c`. Similarly, create another output block with `filename=rc_2.dat` and select the output variable `i1` for that block.

## 2.3 Executing the program and viewing plots

The circuit file can now be created by clicking on the “circuit file” tab and then on the “generate CF” button in the central window. The circuit file corresponding to the circuit schematic and solve blocks you have created will appear in the central window. While it is instructive to look at the circuit file, it is not necessary for the *user* to generate it since the GUI will automatically generate it when the “run solver” option is used.

Run SEQUEL on the circuit file thus generated, by clicking on the “run solver” button. Click on the “solver output” tab in the bottom window and make sure that the message, “SEQUEL: program completed” appears there. Click on the “graphs” tab, and view the plots of `v_b` and `v_c` with respect to time, following the procedure explained in Sec. 1.

The Graph interface allows quantities to be plotted with respect to different  $y$ -axes. This feature is useful when two quantities of different orders of magnitude are to be plotted together, e.g., a voltage varying from  $0\text{ V}$  to  $5\text{ V}$ , and a current varying from  $-1 \times 10^{-3}\text{ A}$  to  $2 \times 10^{-3}\text{ A}$ . The user can choose the left  $y$ -axis for the voltage and the right  $y$ -axis for the current in that case by simply selecting the appropriate box for the  $y$ -axis.

Exercise: Add the node voltage `v_a` to the output file `rc_1.dat` (i.e., the first output block), generate the circuit file, run SEQUEL, and plot `v_a`, `v_b`, `v_c` with respect to time on the same plot.

Once the reader has completed the above tutorial example successfully, the on-line manual (available by clicking on the “help” button in the SEQUEL GUI) can be viewed for a more detailed description of the circuit and solve blocks. Also, the “FAQ” page of the SEQUEL site would be useful in addressing some of the common concerns.