JOINUS SIMULATION TOOL

Simulation tool for Josephson circuit engine

FEBRUARY 1, 2019

SAVOIE MONTE BLANC UNIVERSITY
Chambery, France

TABLE OF CONTENTS

LICENSE

The software is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office grant W911NF-17-1-0120. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation herein.

INSTALLATION

JOINUS is a program developed in Savoie University, France as a part of super tool project for simplifying the simulation process of the Josephson junction based analog and digital circuits. This program uses the JSIM and JoSIM simulation engines as its core. JOINUS has a built-in plotter to automatically plot the output result from the simulation or you can link it to other open source plotters.

COMPILING JOINUS

JOINUS is a program written in the Qt C++ integrated development environment in order to be multi-platform. The source code can be run in Qt IDE in any platform. In order to compile the code, just open the "*SIMGUI.pro*" file and run it either with the standalone version to create the executable file without any dependencies or you can compile it normally and add all the dependencies to installed folder.

To add the dependencies, there is a batch file *"dependencies.bat"* Included for windows OS. For other OS, you can use the dependency finder software to make a standalone program. For the windows version, to open the help files from program, you need to have Adobe Acrobat Reader ActiveX on the system, or you can easily get it from Adobe site.

The plotters for this program are a custom plotter that uses Qcustomplot library to plot the graphs and output data, XMGrace and GNUplot could also be called from the program if they are installed on the system.

THE RESOURCES

JOINUS does not have a simulator engine inside it and uses opensource engines like JSIM and JoSIM for the simulations. Therefore, at least one of these programs are needed for the correct function of JOINUS. The windows compile uses Adobe Acrobat ActiveX to view help PDF files. There are also three other programs that are launchable from the interfaces main page. These programs are also developed in Savoie University in the same scope and will eventually get published.

SIMULATOR ENGINES

As stated before, there are two simulator engines. For the JSIM [ref] you can get it form:
http://www0.sun.ac.za/ix/?q=tools_jsim

And for the JSIM and JoSIM you can download from:
https://github.com/JoeyDelp?tab=repositories

Please check the license and terms of use on their respected sites.

WSIM

WSIM is the program that is developed for investigation of the quasiparticle effect on the current and voltage output of the junction at high frequencies using Werthamer model [ref].

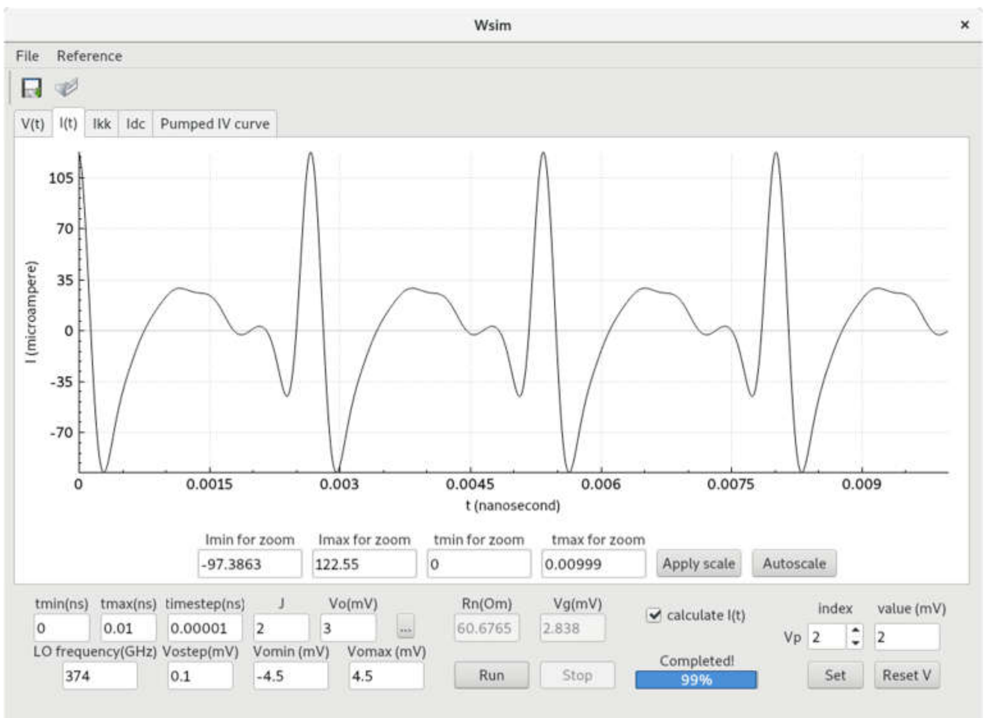

Figure 1 The quasiparticle current result from the microwave signal pump to the junction.

Figure 1 and Figure 2 will show the output results for the current vs. time and current vs. voltage of the quasiparticles for the applied microwave signal to the junction. The quasiparticle current gets very important at high frequencies and near the critical temperatures. More information is available in the WSIM help file.

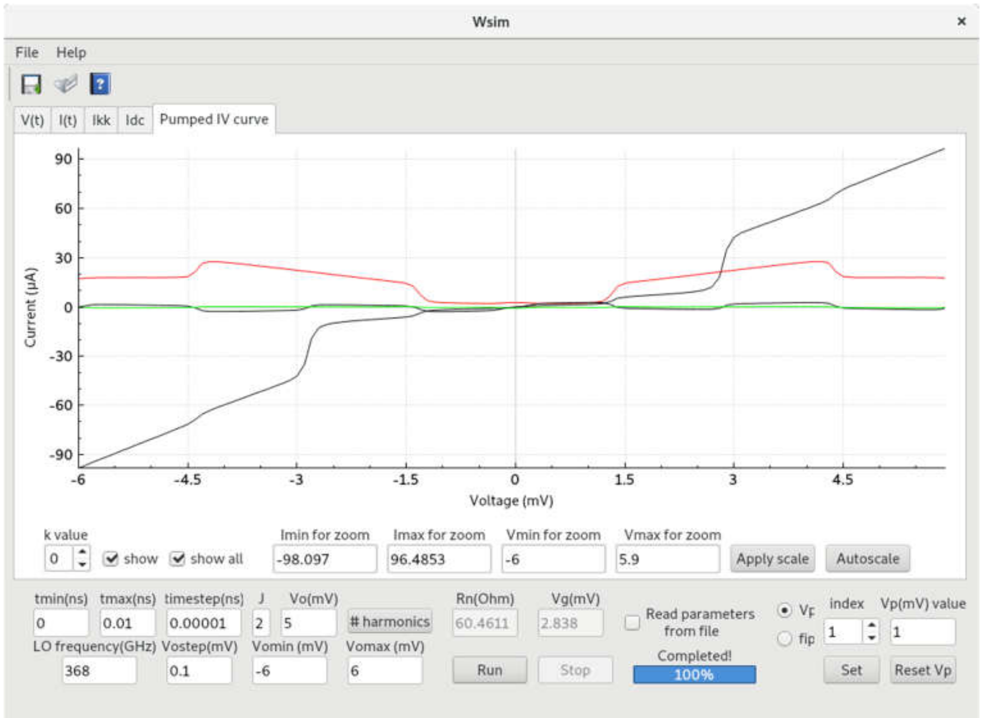

Figure 2 The quasiparticle I-V characteristic for different harmonics result from the microwave signal pump to the junction.

SQUID MAGNETIC PATTERN

SQUID magnetic pattern software as its names suggest was developed to calculate the magnetic pattern for the SQUID based on the junctions critical current and the inductance of the junction on both side of the SQUID loop.

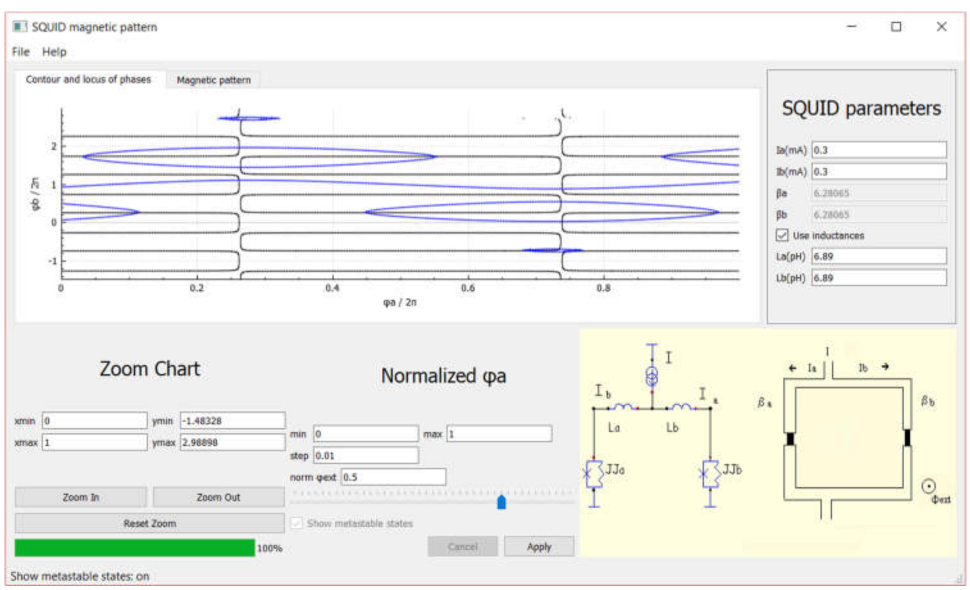

Figure 3 SQUID Magnetic pattern software front pannel.

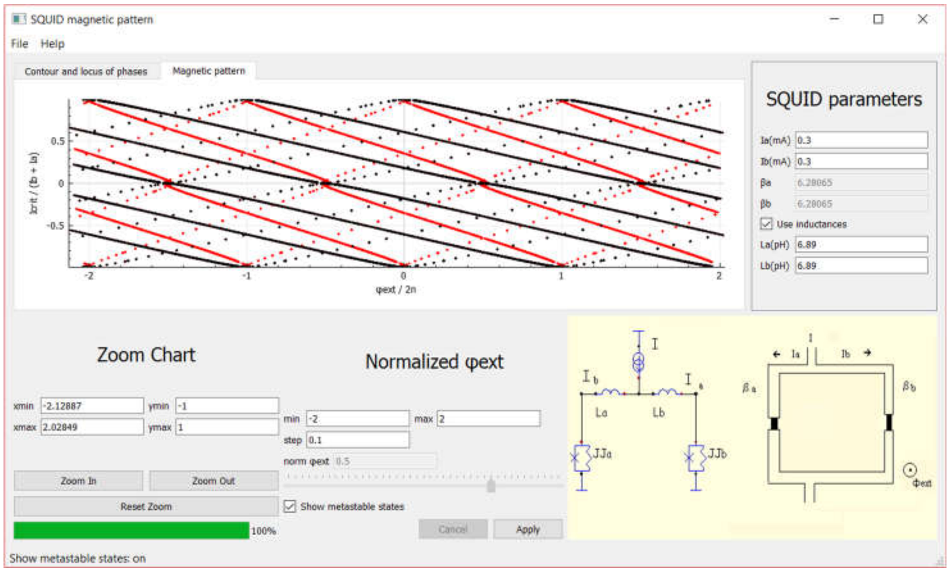

Figure 4 The magnetic patterns of the software. The red lines are the patterns with no physical meaning.

FOLDER ARRANGEMENT

In this version of the program, the arrangement and the name of the files should be as stated here for windows. Joinus.exe is the main file that launches the program. You should also have two other folders (Data & Help) and an optional WSIM folder for extra programs.

```
C:.
|   dependencies.bat
|   Joinus.exe
|   STATE.TMP
|
+---Data
|       dcsfq_jtl_sink.js
|       JoSIM_n.exe
|       jsim_n.exe
|       si.inp
|       si.js
|       volmul2.inp
|
+---Help
|       Manual.pdf
|       Manual2.pdf
|
\---WSIM
    |   Idc.dat
    |   Ikk.dat
    |   parameters.dat
    |   SMP.exe
    |   Wsim.exe
    |
    +---Help
```

Figure 5 The files and folders of the program after installation.

JSIM and JoSIM should be inside the Data folder and they are named jsim_n.exe and JoSIM_n.exe for Windows OS. The other files in the Data folder are the examples that are provided. The Help folder has two PDF manuals for the program. One is this manual and the other related to the syntax of the JSIM and the netlist. They are both accessible via JOINUS.

RUNNING PROGRAM

In order to run the program just click on the executable file. The interface should look like Figure 6.

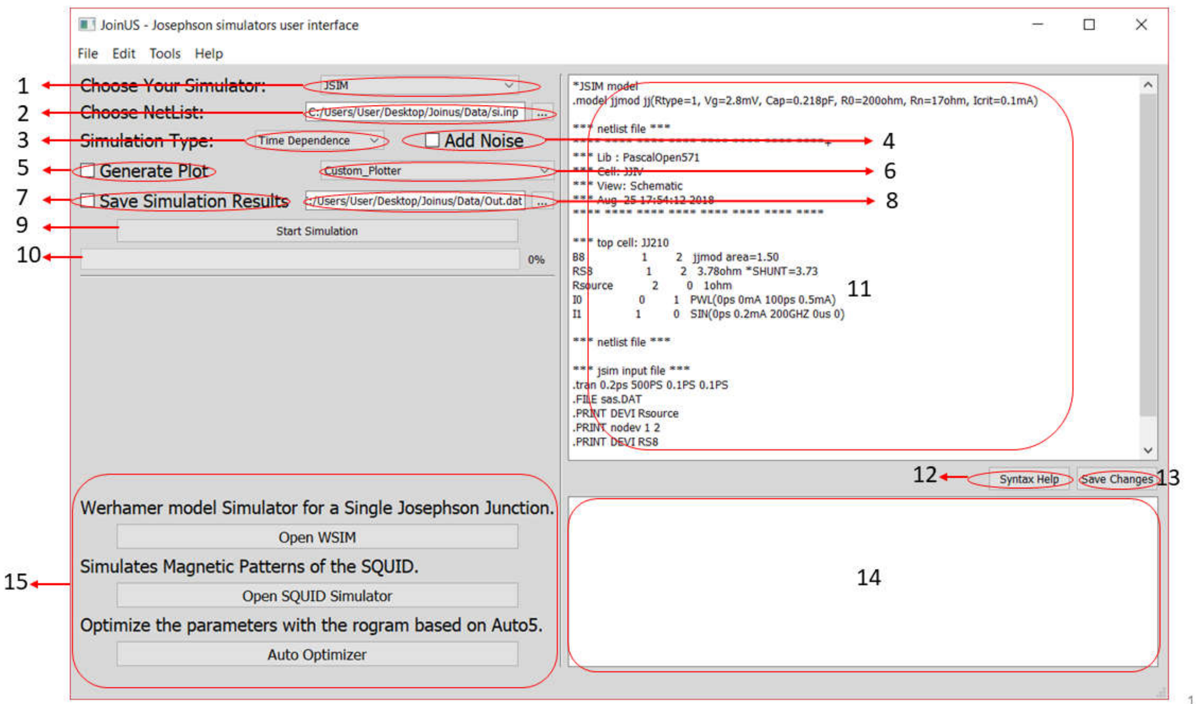

Figure 6 The Interface for JOINUS software.

The interface have different parts that are explained here:

1) The simulator engine, JSIM or JoSIM.
2) Load the netlist file. JoSIM works with *.js files.
3) The function that you want to perform. All the choices are explained in the next section.
4) Adds noise to the netlist depending on the temperature and resistor values.
5) If checked would generate plot from output file.
6) Choose the plotter type. Default is the built-in plotter.

7) If checked saves the output file in the chosen destination. The file can be in ASCII or CSV format.
8) Destination of the output file.
9) Starts the simulation process.
10) Progress bar for simulation.
11) Loads the netlist in that window and you can edit the netlist. If not saved, changes would not apply.
12) Shows the JSIM help file for syntax help.
13) Saves the changes that are done to netlist.
14) The window that shows the terminal or console output and errors. This also acts as a built-in terminal.
15) Launching pad for other connected programs.

The Interface for the plotter interface is shown in Figure 7.

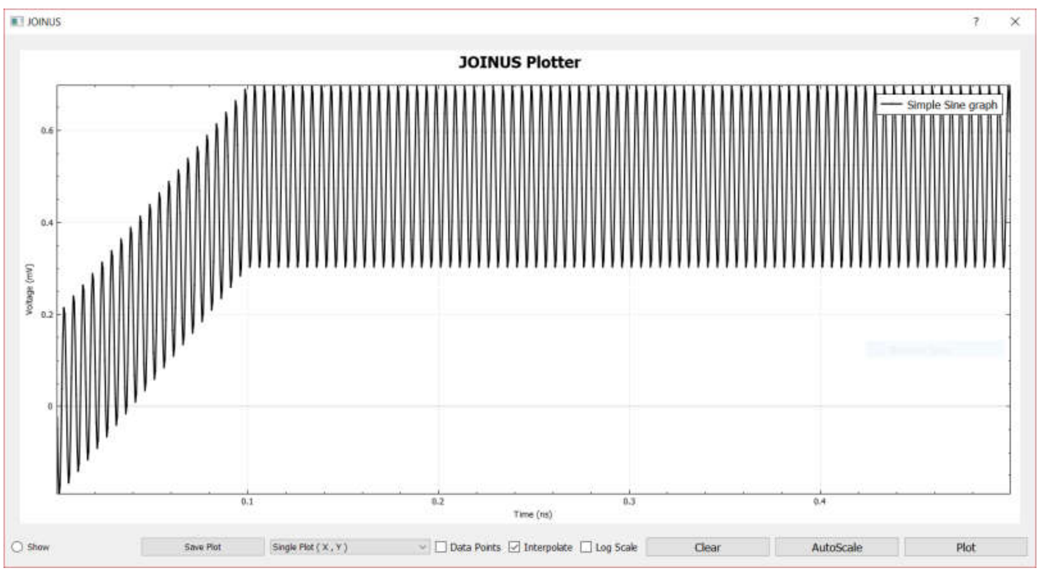

Figure 7 The Interface for JOINUS plotter.

In the plotter interface, you can change the scale of each axis by mouse wheel when you are on that axis or both when you are on the graph. The “autoscale” button set the scale to fit screen. The “clear” button will clear all the graphs, you can clear each individual graph by right-click on them. All the title, legends and axis texts could be changes by double click on them, the legend place could be set by right-click on it.

The scale can be set to log or linear, you can linear interpolate between data and show the data points on the graph. There are different graphing modes that could be selected:

Single plot (X , Y): plots the first two columns of the data from data file.

Single plot (Y , X): Plotting the first two column of the data in a single plot and inverting X and Y.

Multiple Plot (X , Y1 , Y2 , ...) : Plotting the all the columns, first column is X and the rest are Y.

Multiple Plot (X1 , Y1 , X2 , Y2 , ...) : Plotting the all the columns, odd columns are X and the rest are Y.

Also, you can plot data files that are already saved on the system to compare your graphs with old data. This is possible by clicking the show radio button and selecting the datafile and plotting it. You can choose the color of the plots in this part.

FUNCTIONS

Apart from being a user-friendly interface for simulation engines, JOINUS can perform various tasks and functions. Some functions like Monte Carlo optimization and Bit Error Rate calculation and frequency response will be activated in future versions.

TIME DOMAIN SYNTHESIS

The standard form of synthesis that is done by the simulation engines is time domain. In this synthesis, JOINUS would just apply the noise effect and temperature effect to the netlist and continues the simulation.

I-V CURVES

In the I-V curve mode, JOINUS would take the first output as Current (Y) and second output as Voltage (X) and plot them versus each other. In this mode you should determine Current Source name, boundary values for current, number of points per period and number of periods or loops as Figure 8. After that the program will run by start button and draw the I-V curve.

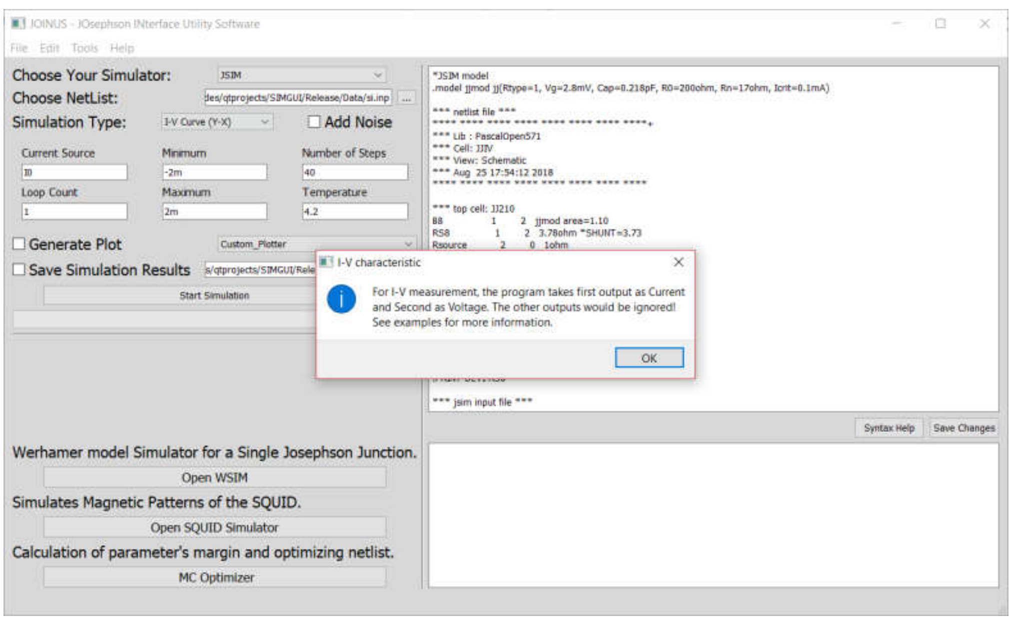

Figure 8 The I-V curve calculation front panel.

PARAMETER SWEEP

In parametric sweeps, JOINUS will ask for parameter name, parameter value in the netlist, boundary values and number of steps. After that, the program will simulate the netlist for each of those parameters and save the outputs in the DATA folder at the program root. Figure 9 shows the front panel for parametric sweep.

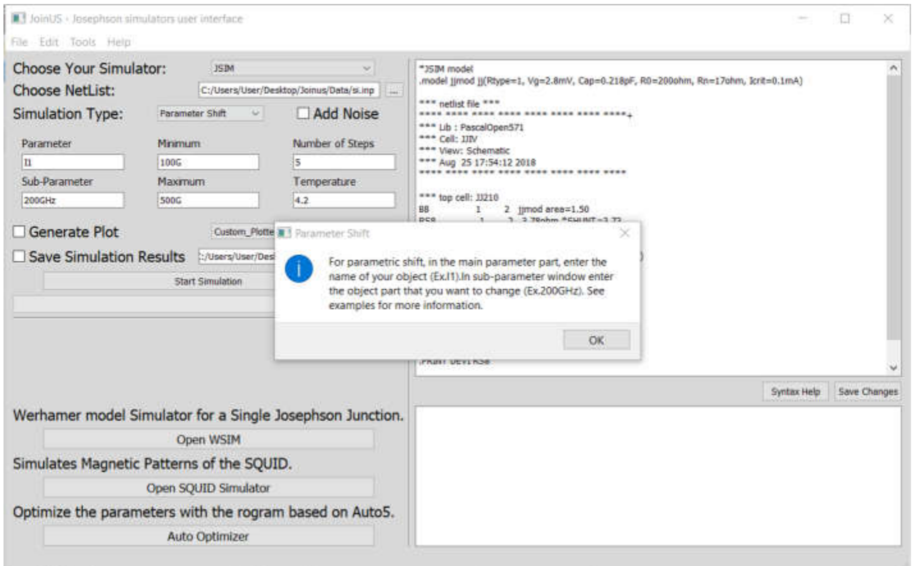

Figure 9 The parametric sweep front panel.

TEMPERATURE SWEEP

Like parametric sweep, temperature sweep will also simulate the netlist in temperatures determined by user. The front panel for temperature change is shown in Figure 10. The temperature effect is discussed in the appendix part formulation that is used for this program. The parameters for the temperature change are set for the Niobium. If you want to simulate for another material, you need to change the "Tc" value in the "myglobalvars.cpp" source file and then build the source code again.

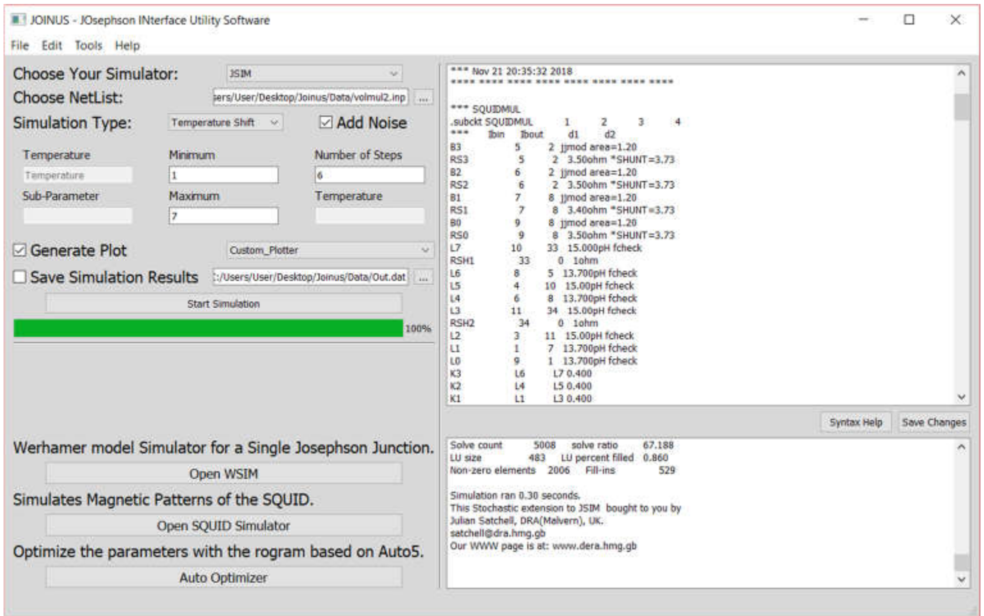

Figure 10 Main window of the software in parametric mode for temperature.

FILE MANAGEMENT

In JOINUS you can export the output of your simulation if you tick the "Save Simulation Results" box. There are two kind of files, ASCII and CSV. The difference is the delimiter in them. You can also save the state of the simulation from the file menu tab. You can load the state later from the same menu. The software would generate some temporary files mostly in the DATA folder at the root and delete them after exiting the program.

From the custom plotter, you can export graphs in different image formats. For now the font size and type is fixed, but it will be changed in future versions.

APPENDIX

In this part, we will discuss a little about the formulation that is used in the software. For more information on the engine formulation, you can see JoSIM help file.

TEMPERATURE DEPENDENCIES

The temperature is important for several aspects. The temperature modifies the Cooper pair density and therefore most of the characteristics of superconductors and Josephson junctions, such as the critical current, the critical magnetic field, the penetration depth and the energy gap. A change of temperature has an impact on active and passive elements in any circuit. Some effects were already shown in the September 2018 report. The other consequence of the temperature change is the thermal fluctuations and thermal noise due to the resistive / quasiparticles currents.

For the gap voltage from BCS theory we have:

$$\Delta_j = \sum_k \sum_i v_{ji} \Delta_j \frac{\tanh(\beta E_K^{(i)}/2)}{2E_K^{(i)}}$$

By solving this equation at the bulk limit and for elemental superconductors (like Nb) with symmetry:

$$1 = V_0 N(0) \int_0^{\hbar\omega_D} \frac{dE}{\sqrt{E^2+\Delta^2}} \tanh(\frac{\sqrt{E^2+\Delta^2}}{2k_B T})$$

We can estimate Δ value as:

$$\Delta = \Delta_0 \sqrt{\cos\left(\frac{\pi}{2}(T/T_C)^2\right)}$$

where Δ_0 is calculated from the weak coupling limit:

$$2\Delta_0 = 3.53 k_B T_C$$

The value of the normal resistance of Josephson junction can be found as:

$$R_N = \frac{\pi\Delta}{2eI_0} \times \tanh(\frac{\Delta}{2Tk_B})$$

The critical current of the superconductor can be estimated as:

$$I_C = I_0(1-(T/T_C)^2)\sqrt{1-(T/T_C)^4}$$

The resistance dependency to the temperature can be calculated from the more generic form of Ohm's law:

$$J = \sigma_0 E exp\left(-\alpha T^{-1/n}\right)$$

$$R = R_0 \exp\left(\alpha T^{-\frac{1}{n}}\right) \xRightarrow{linear\ estimation} R = R_0(1 + \propto (T - T_0))$$

The Johnson Noise value for resistance is calculated as:

$$I_N = \sqrt{\frac{4KT}{R}}$$