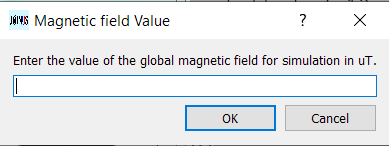
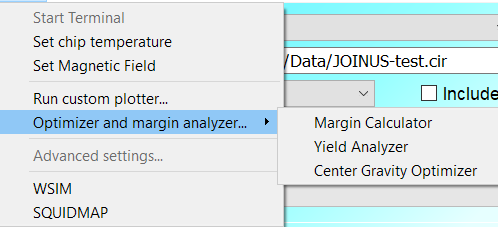
# JOINUS New features:

Added magnetic effect to the menu and now magnetic field will change critical current. Critical magnetic field is affected by the temperature and can be set from “myglobalvars.cpp” under “BC2” parameter. Now the default is set to 1mT.

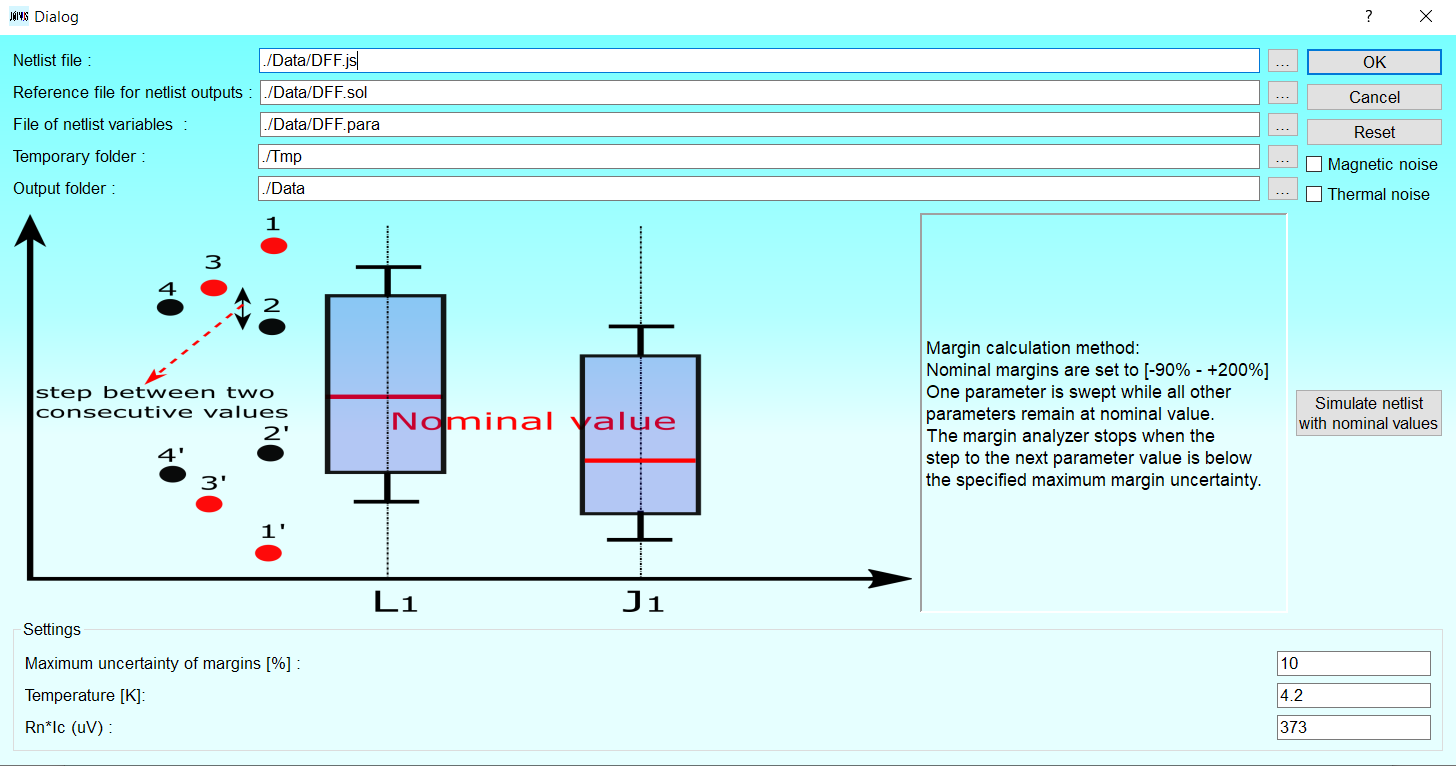


## Margin and optimizer

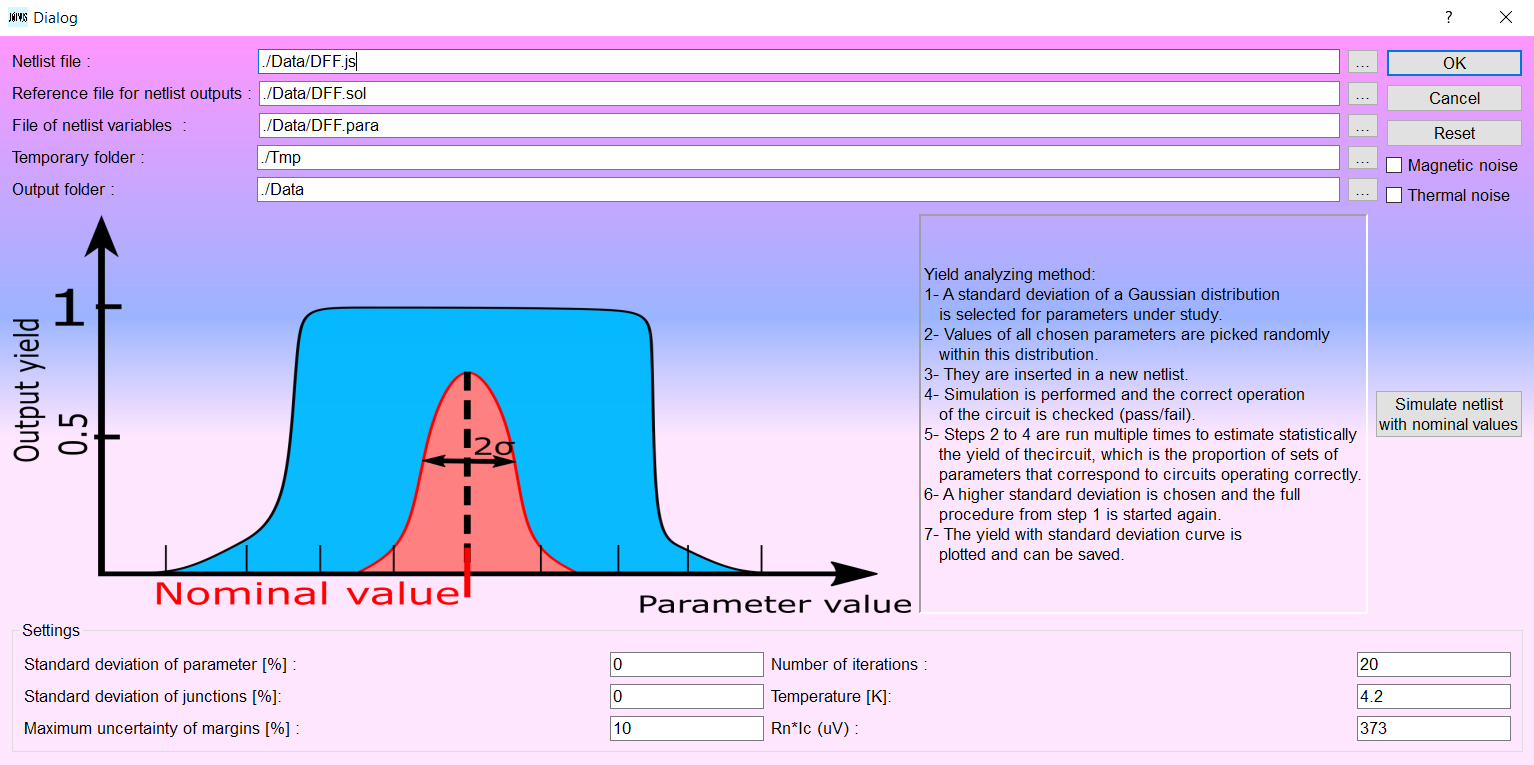
The optimizer is now in three different parts:



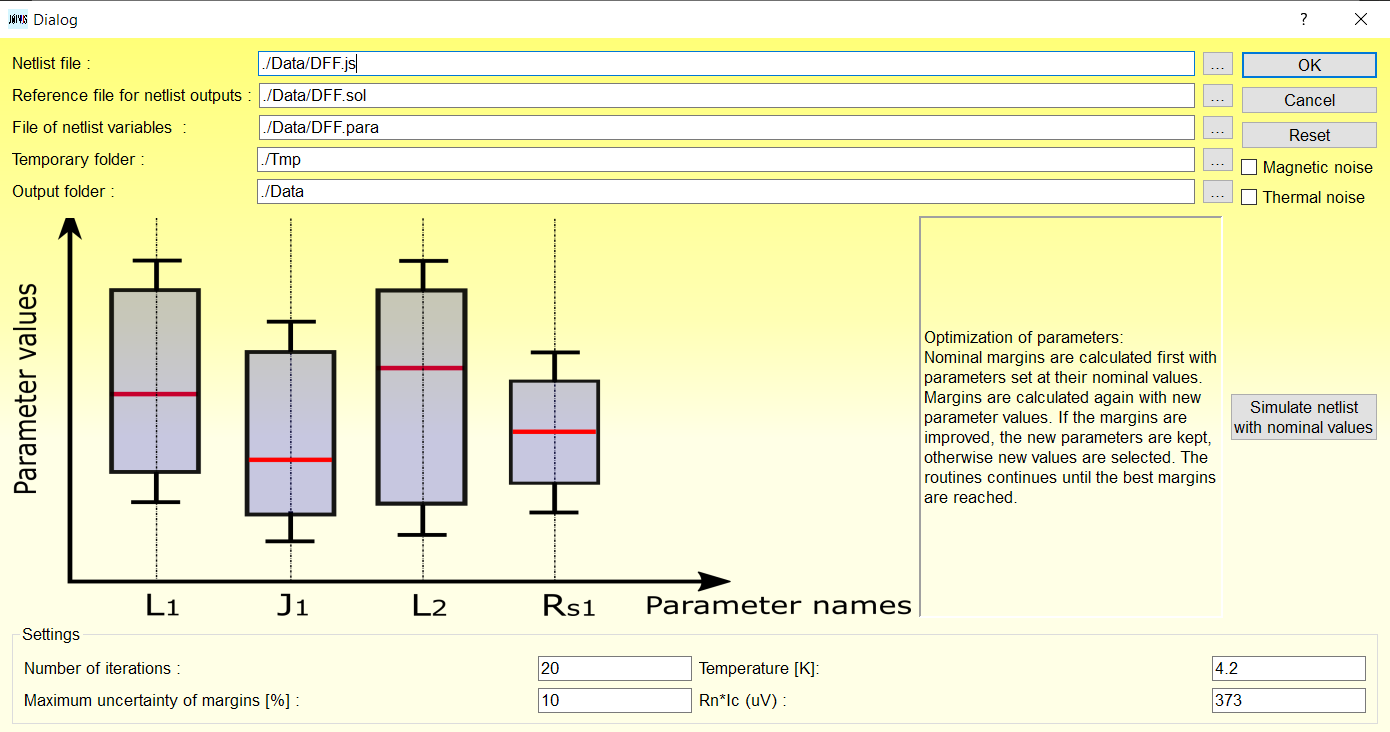
Margin calculator includes files {“margincalc.cpp”, ” margincalc.h” and ” margincalc.ui”}.



Yield analyzer includes {“yieldcalc.cpp”, ” yieldcalc.h” and ” yieldcalc.ui”}.



Center of gravity optimizer includes {“cgoptimizer.cpp”, ” cgoptimizer.h” and ” cgoptimizer.ui”}.



I will write examples and more explanation for them. The only issue is that Center of gravity optimizer needs a working input netlist to start the process and cannot find an operation point on its own. I will include that in future codes. The functions that are in common by all three of these classes are in “autooptim.cpp”.

Three file types are used in optimizer and margin analyzer:

## Circuit file (.cir)

This is the netlist file similar to jsim netlist. The difference is that instead of numeric values, you should put your parameters. For example:

B13 16 34 jjmod area=2.16

RS13 16 34 1.73ohm \*SHUNT=3.73

B14 92 7 jjmod area=@AJ1@

RS14 92 7 @A2@ohm \*SHUNT=3.73

Vbias 43 0 PWL(0ps 0mv 5ps @M7@mv)

Here @AJ1@ , @A2@ and @M7@ are the parameters we want to affect.

## Parametric file (.para)

This file determines the behavior of each parameter. For the former example, we can write:

@AJ1@ 1.15 B14 1

@A2@ @AJ1@ RS14 4

@M7@ 10.5 Vbias 1

First column is the parameter with (@), second column is the nominal value, third is the main param name for plotting and forth is the type of the parameter (0=fix, 1=No gauss dist., 2=element value, 3= Junction area, 4= dependency).

0 means the parameter will stay at Nominal value Ex. {@AJ1@ 1.15 B14 0} @AJ1@ will be replaced with 1.15 in the netlist.

1 means the parameter changes but has no distribution {@AJ1@ 1.15 B14 1} @AJ1@ will be replaced with the range [1.15\*0.1 ... 1.15\*3] in the netlist.

2 means it has the distribution determined by the Parameter variation {@AJ1@ 1.15 B14 1} @AJ1@ will be replaced with the range [1.15 (%PVP)\*0.1 ... 1.15 (%PVP)\*3] in the netlist.

3 means it has the dist determined by the Junction spread {@AJ1@ 1.15 B14 1} @AJ1@ will be replaced with the range [1.15 (%JCCS)\*0.1 ... 1.15 (%JCCS)\*3].

4 means value of parameter is dependent on another parameter {@A2@ @AJ1@ RS14 4} Value for RS14 is determined from the parameter in @AJ1@ in relation with RnIc value.

## Solution file (.sol):

This file shows the correct outputs and the right timing. The file should have columns the same size as output file from simulations. First column is time, second is the first output and so on.

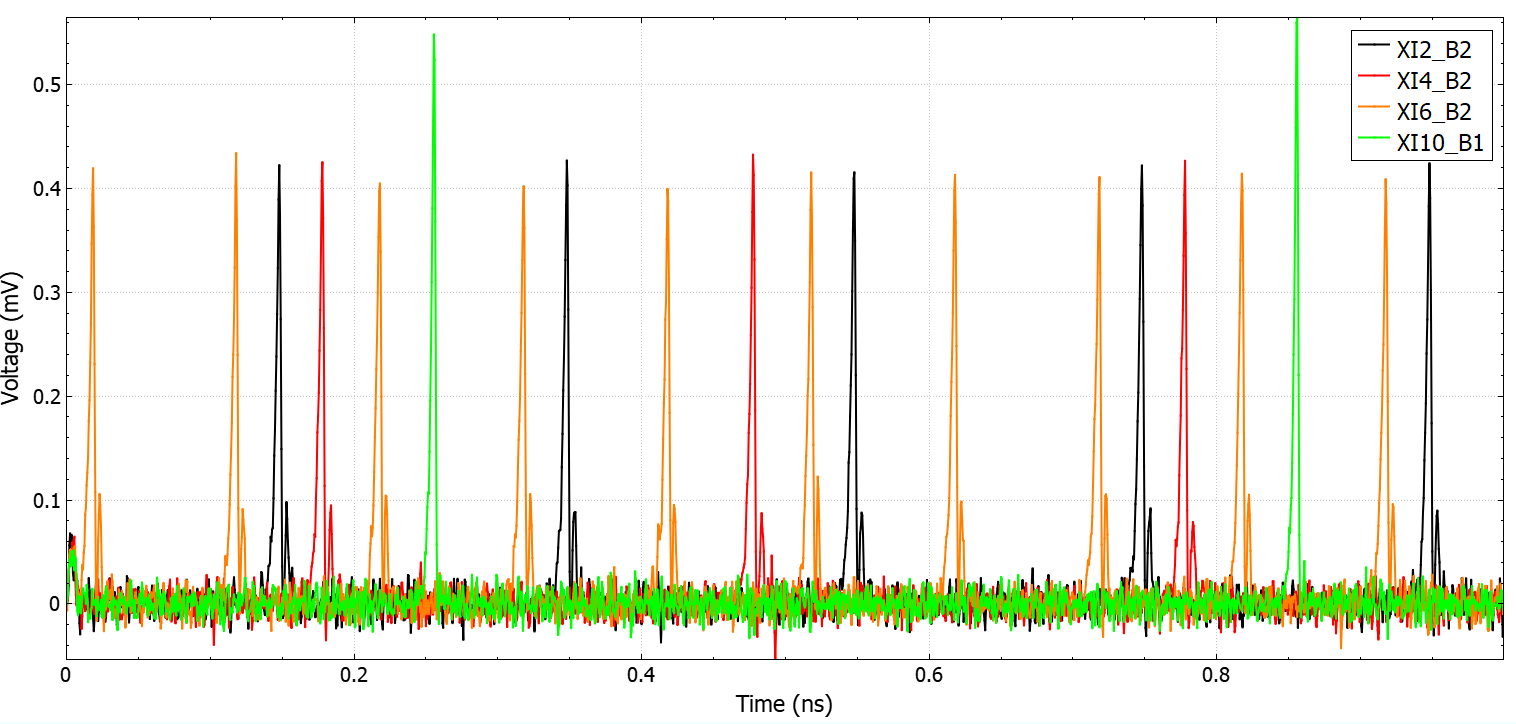
The format for writing solution file is in first column you write the time in picoseconds. In the other columns you will write the number of pulses that you expect to come until that time. For example, for a circuit with following output:

.print devv XI2\_B2 \*A\_In

.print devv XI4\_B2 \*B\_In

.print devv XI6\_B2 \*Clk

.print devv XI10\_B1 \*OutPut



We can write solution file as:

200 1 1 2 0

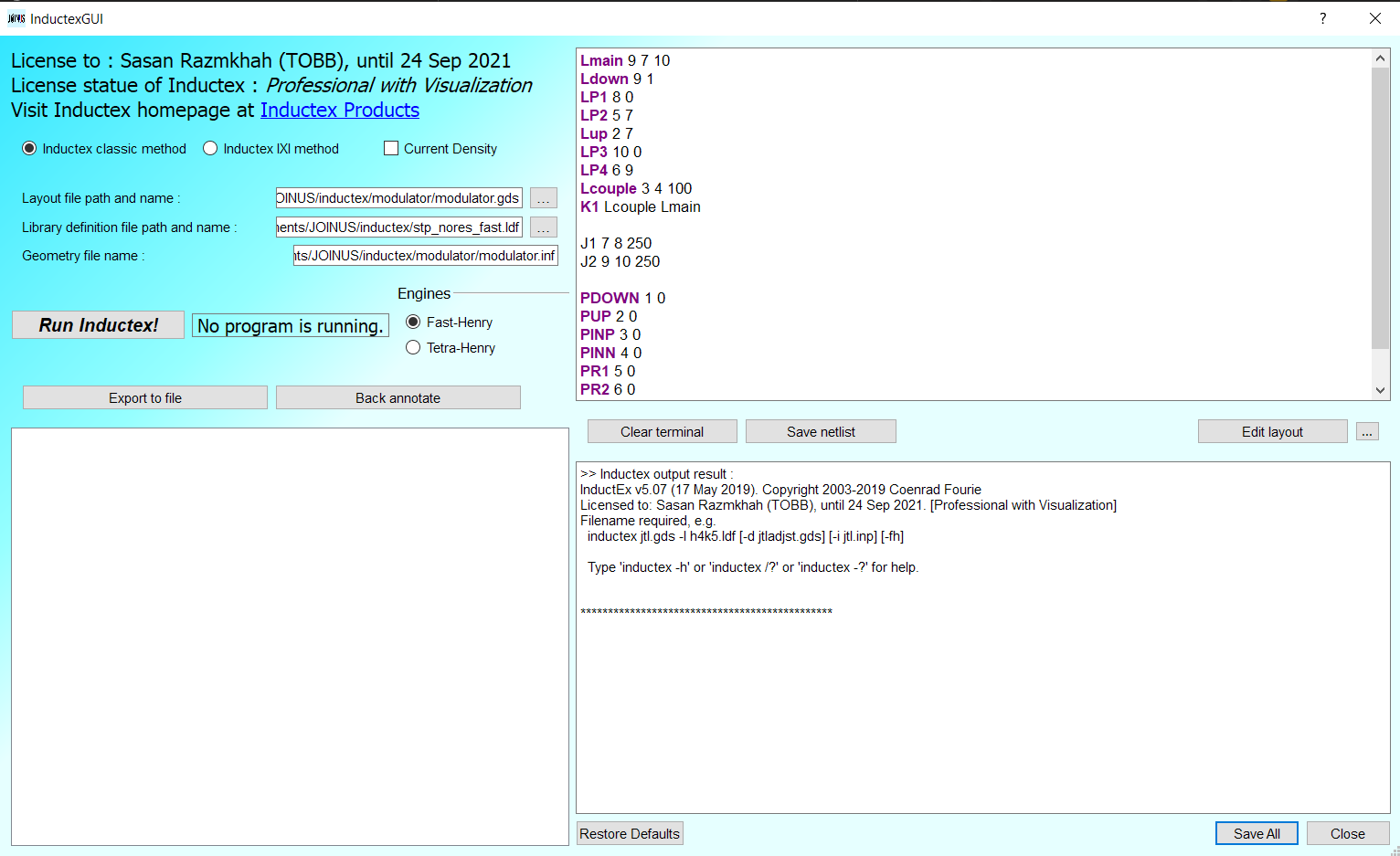
400 2 1 4 1

600 3 2 6 1

800 4 3 8 1

1000 5 3 10 2

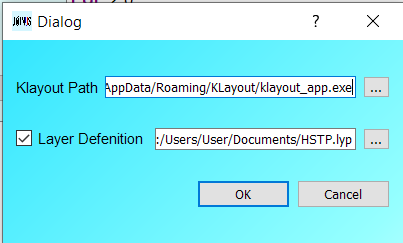
## Inductex:



Inductex interface will run your current version of Inductex and will solve the netlist based on old multi-file or single IXI inductex input format.



It shows three different fields including inductances, mutual inductances, and junctions. The layout edit is now set with Klayout and program path and library can be included to show the correct layers in Klayout editor.



The Inductex is not yet function on MAC OS X.

## Threading issue for MAC OS:

There was a thread issue with MAC OS X that the program created different threads for each run, and it could consume a lot of memory. The threading is now solved for MAC and optimization and simulation should run with no issue.

## Linux install:

For Ubuntu, QT cannot work with *clang 10+* and needs *clang 8*. If you are on new version of Ubuntu and get errors related to not recognizing QT core libraries, run the following commands:

sudo apt install clang-8

sudo update-alternatives --install /usr/bin/clang++ clang++ /usr/bin/clang++-8 100

sudo update-alternatives --install /usr/bin/clang clang /usr/bin/clang-8 100