Current-Density-Analyzer

Introduction

Short description

Current Density Analyzer or CSD.py is an application created for quick and easy analysis of the two dimensional cross section of current carrying conductor sets. It calculate the current density distribution in given conductors, generated ohmic power losses, equivalent single line circuit parameters and electrodynamic forces vectors for static case. There is as well simplified thermal solver available.

The whole idea is based on the assumption of quick-turnaround workflow: - sketch the cross section shape - define current parameters - analyze and get results

Application area and limitation

Tool is based on the AC circuit theory and therefore is capable to simulate only sinusoidal currents *(except the force solver - which assumes constant current)*. Basic underlying calculation methodology is a simple type of FEA method. It's approximating the given *(sketched)* geometry by set of squares ant treat each of such piese *(element)* as separate conductor in which the current density is uniform. Whit this assumption taken - the system wide ohm law based matrix equations are solved delivering the currents values in each element - and hence - the current density distribution.

With this principia the tool works well for a currents that are sinusoidal in shape of given frequency. The precision of the solution depends of the *elements* size and can be increased by subdividing the sketched geometry into smaller *mesh* approximation. But this comes with the cost of longer calculation times.

Used technology

This tool is written in python *(python)* language and by so shall be OS agnostic. It has been developed under linux and tested in Windows.

The following dependencies and packages are uses (and required to work): - python 3.xx - matplotlib - numpy - tkinter

It is required to have those packages available in the system to be able to use CSD.

Working with the application

Installation

1. Install python 3.xx language interpreter in your system (please refer to python.org). *I strongly suggest to install it with the pip python package manager for easier library installation.*
2. Install numpy library (please refer to numpy.org) *if the PIP is installed correctly it shall be possible to be done with the "pip install numpy" command.*
3. Install matplotlib plotting and visualization library (please refer to matplotlib.org) *again it can be done with PIP.*
4. Get the thInter working. It should be there wit your pyton installation but that may depend on your OS *(please check online if a separate installation is needed).*
5. Clone this repository to a folder on your machine, or download it as a ZIP file and extract to a folder on your machine.

Running the app

In general the app is a python script (or rather a set of those) and can be run by: - navigate to the folder where the repository was cloned or extracted - execute command "python csd.py" *(in case of unix like system you may have more than one python interpreter installed and the command might be "python3 csd.py").* If the installation of python was done by the installer the *.py* files might my linked to the interpreter and hence it would be enough to double click on the csd.py file.

Application window

After successful installation the app will display the main window:

The main window can be divided into three general areas: 1. Top Menu Bar 2. Main Working Canvas 3. Right Action Panel

The Top Menu Bar (1) serve the typical function. - **File** menu allows for: - geometry sketch load, - save, - reset canvas to empty. - **Analyze...** menu can run: - Power Losses Pro Solver - Electrodynamic Forces Solver - Impedance Model Solver - **Geometry** menu allows for: - Increase mesh density (subdivide) - Decrease mesh density (simplify) - **View** menu: - Display CAD like sketch representation with dimensions.

The Main Working Canvas (2) is the checkered area where a conductors geometry cross section can be sketched by setting particular squares set for electrical phase A,B or C of the system. It's done by **Left Mouse Click** - if proper **Active operation** is set in the Right Action Panel (3). The **LMC** always realize the selected **Active operation** at the current mouse cursor location which is marked by *gray shadow box* that's follow mouse move.

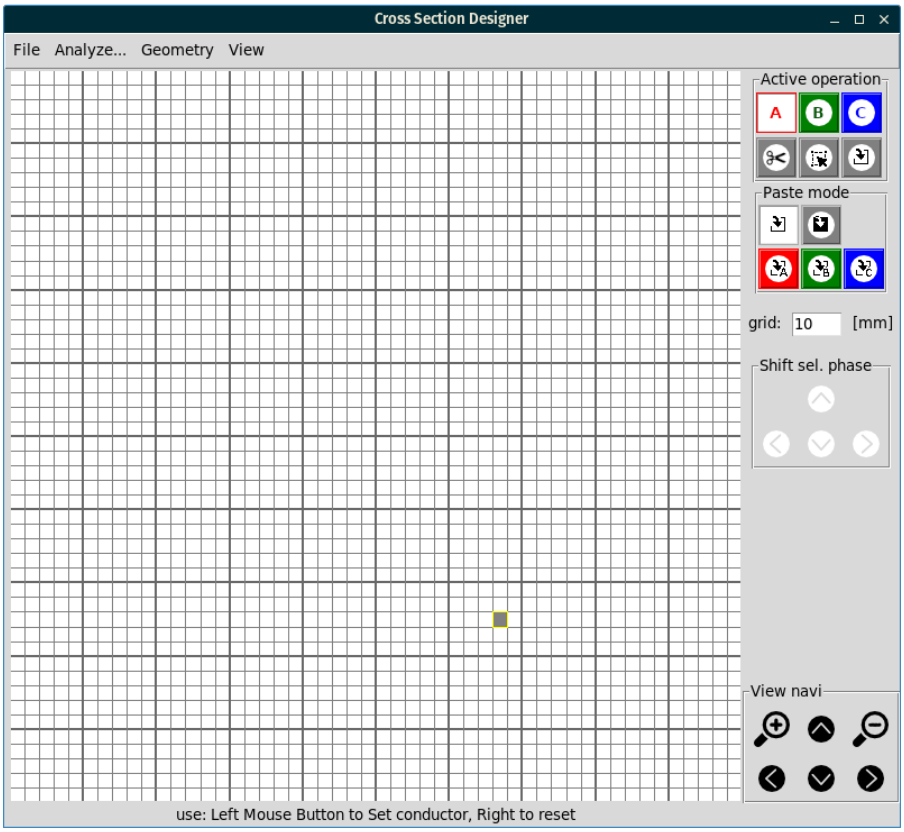
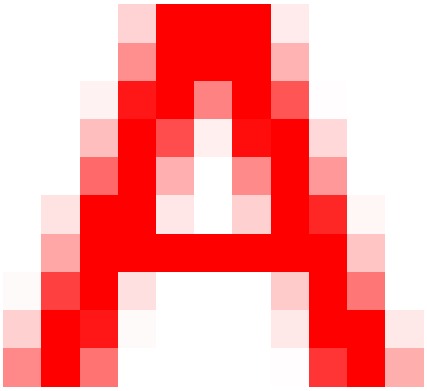


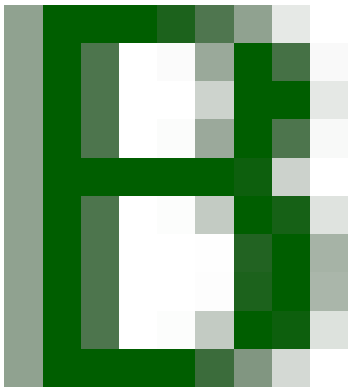
Figure 1: CSD main window v2020

In the Canvas area - default behavior of **Right Mouse Click** is set to clear the *element* under the cursor.

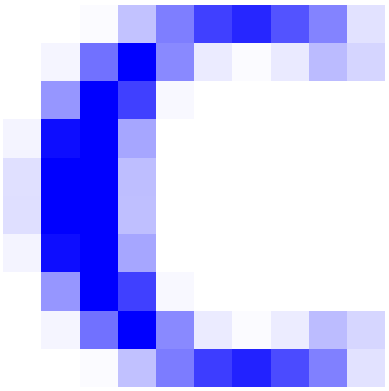
Right Action Panel (3) contains five main panels: - **Active operation** This panel define the current operation in the Canvas Area. It can be (in order): -



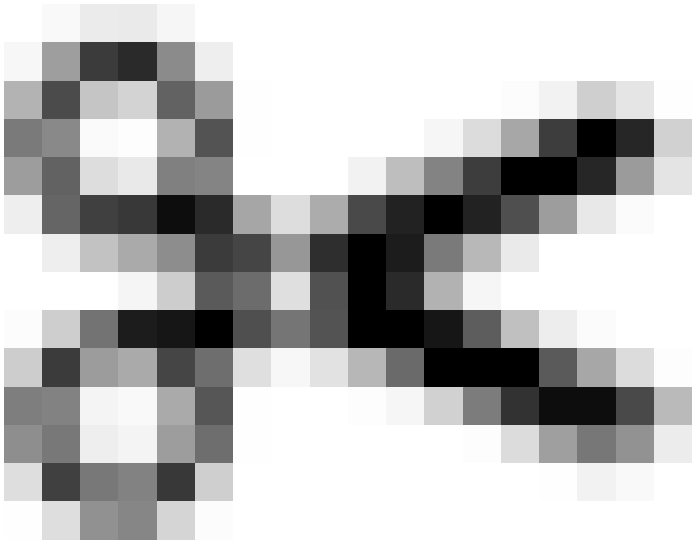
Painting the phase A elements



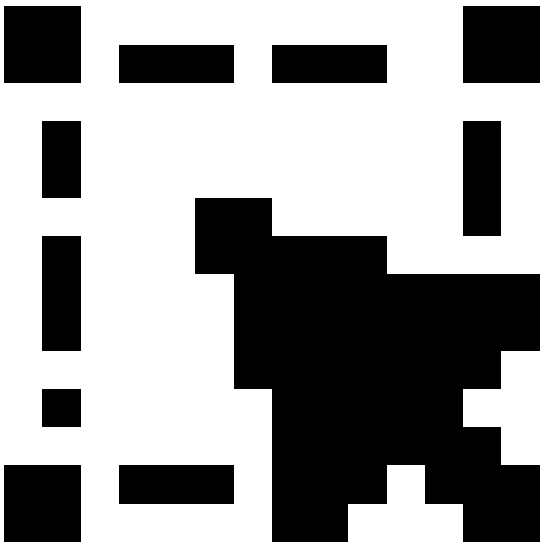
- Painting the phase B elements



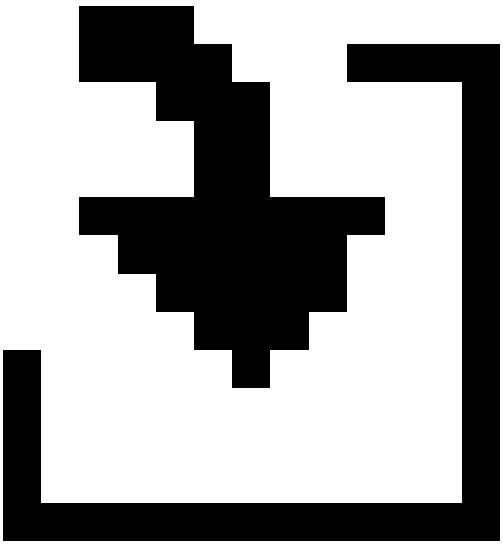
- Painting the phase B elements



- Clearing the selected element
- Selecting & copying to clipboard rectangular area of sketch (by click LMB +

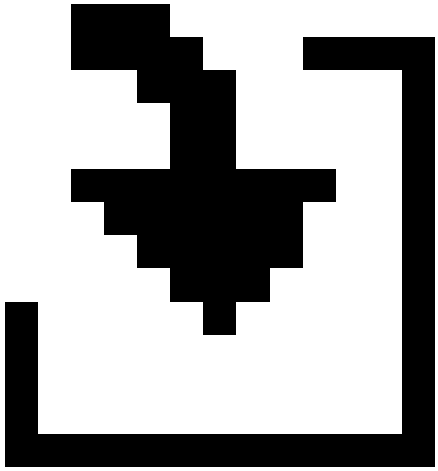


drag) .
The operation is automatically switched to Paste after copying to clipboard. - Paste the data from clipboard to current location (left top corner) by apply se-

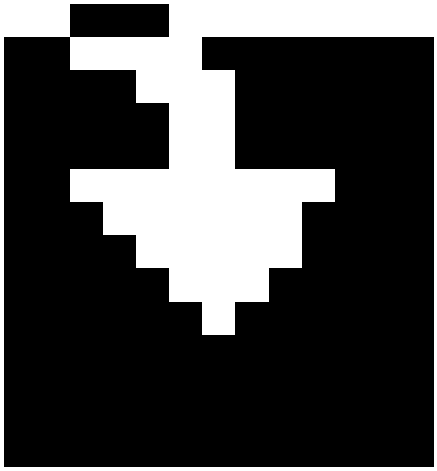


lected **Paste mode**

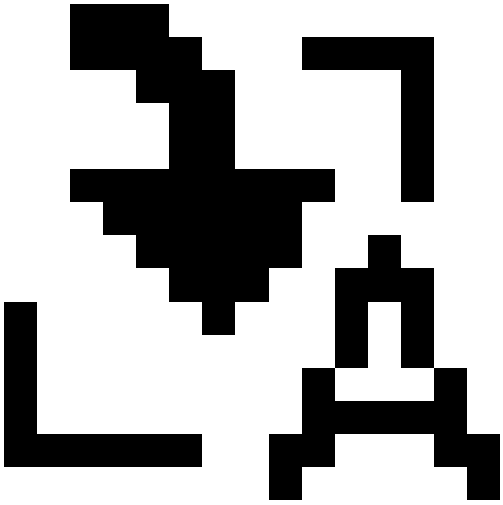
- **Paste mode** define the behavior during paste operation:



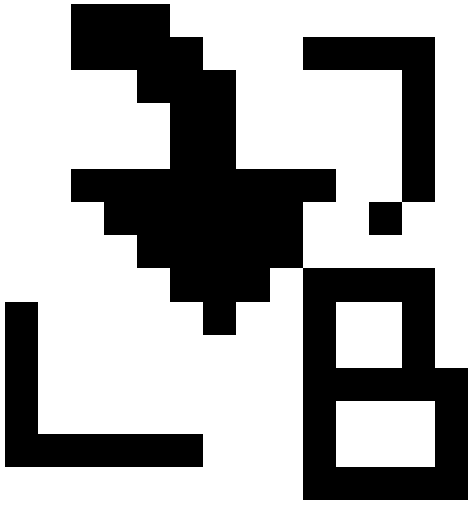
– Clean paste in place
  put only non empty elements in the current location (clipboard empty cells don't erase destination area).



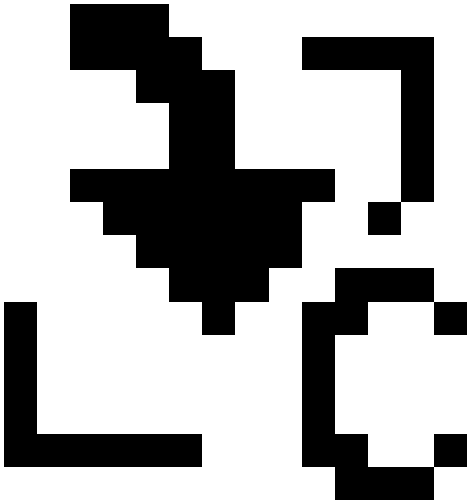
– Full paste in place
  put all clipboard elements in the current location (clipboard empty cells erase destination area).



– Paste elements as phase A ,

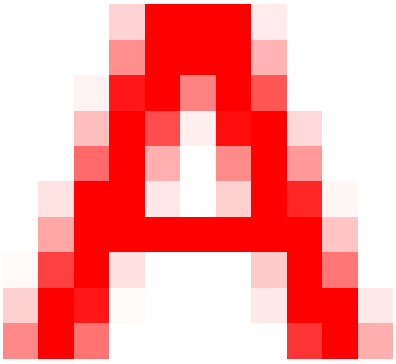


B

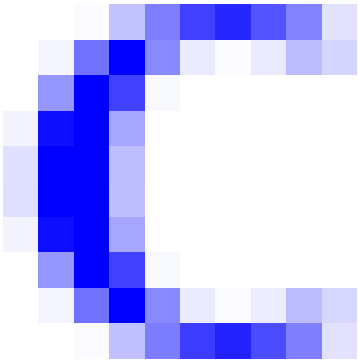


or C
put all non empty elements in the current location turning them into elements of phase A,B or C respectively.

- **grid size** allows to define the real world size of single grid *(mesh)* element. This can be modified at any time just by entering any number and "de click" the text field area.
- **Shift selected phase** those directional controls allows to shift all elements



of active phase (active by means of



buttons) in canvas by a 1 *element* in given direction. Tis functionality is reminiscence of previous version and might be removed with future updates.

- **View navigation** allows to zoom in and out in the canvas view and move view around if in zoommed mode.

Old release notes:

So – long story short – after putting together a round of 600 lines of code (in python off course) and learning some basic usage of the tkinter GUI building framework I can now for the first time release to you beta version of the CSD app. This app is a python based mirror of the calculator of the Current Density Distribution that was available on my webpage (actually still is http://tomasztomanek.pl/pub/webapp/IcwThermal).

Please give it a try!

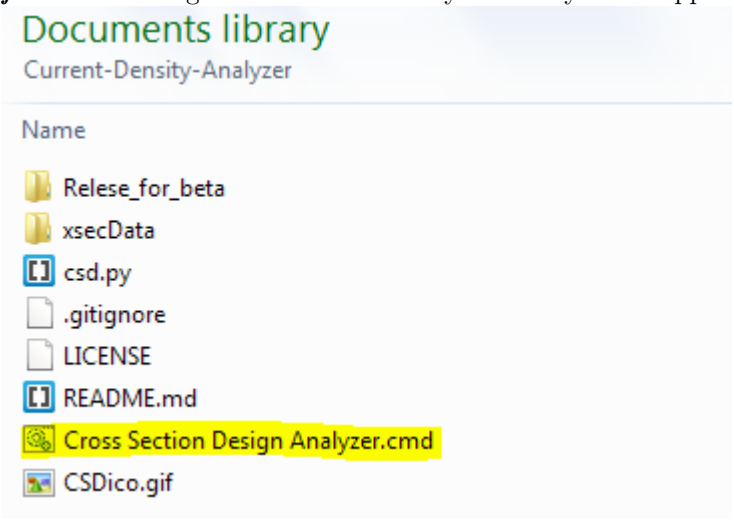
Installation: 1. If you already have python installed you can jump

to point 2 a. Install on your system the anaconda package from: https://www.continuum.io/downloads b. This should be all you need to do. 2. Download the CSD app script from GitHub (use the Clone or Download button and take a ZIP or clone the repo depending on your preferences): a. https://github.com/tymancjo/Current-Density-Analyzer/tree/master b. If you get ZIP then Unzip the content to a directory of your choice. If you colone the repo - you will know how to handle this ;)

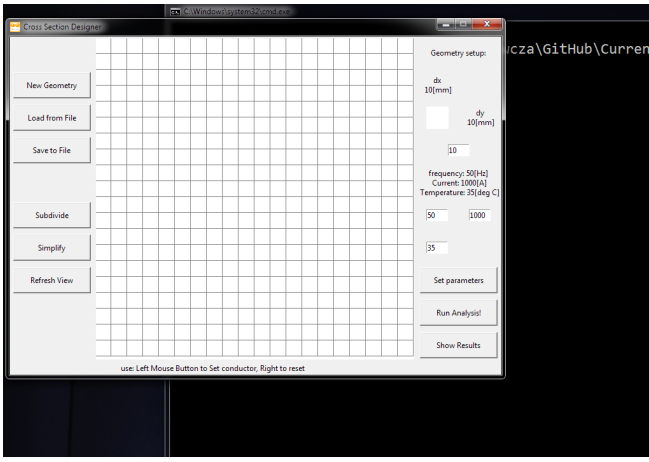
You should be good to go! If something is not working – let me know!

Usage - this shows the old screenshots - will be updated!

Run the app: ***Please note that the screenshots are from previous version of the app. Current one have options for 3 phase analysis. But for information purposes should be just fine*** 1. Navigate to the directory where you unzipped the app files:



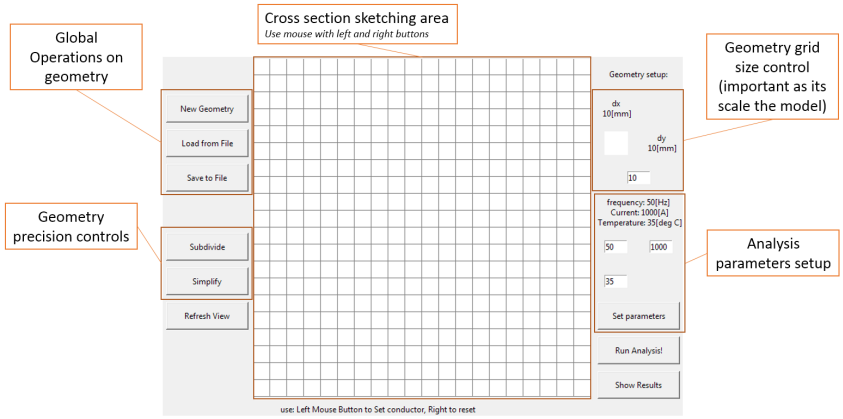
2. Double click on the "Cross Section Design Analyzer.cmd" if you are in windows world
3. Or run the csd.py script from your python interperer



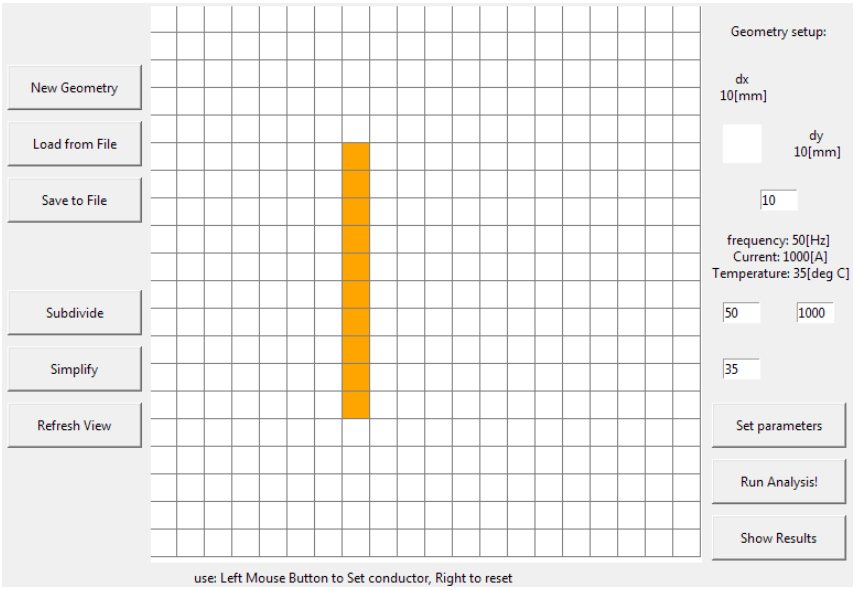
4. You should have app and terminal window pop-up
5. Now you can start using the app. All your actions are done in the main app window. Terminal will display some data that reflect particular calculations steps. But this is nothing you need to worry about.

Using the app:

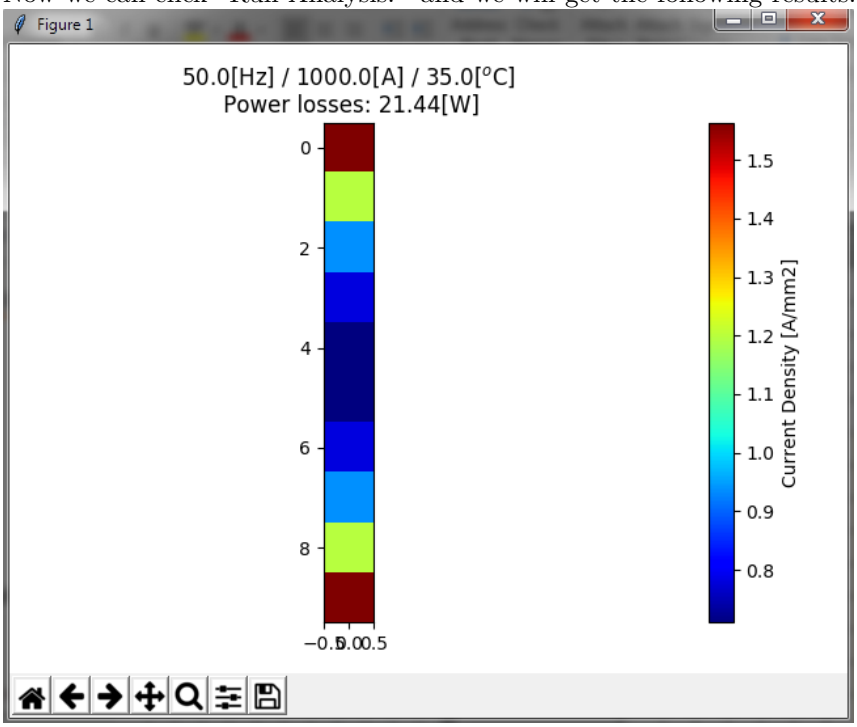
1. Let's use the app for simple case of one copper bar



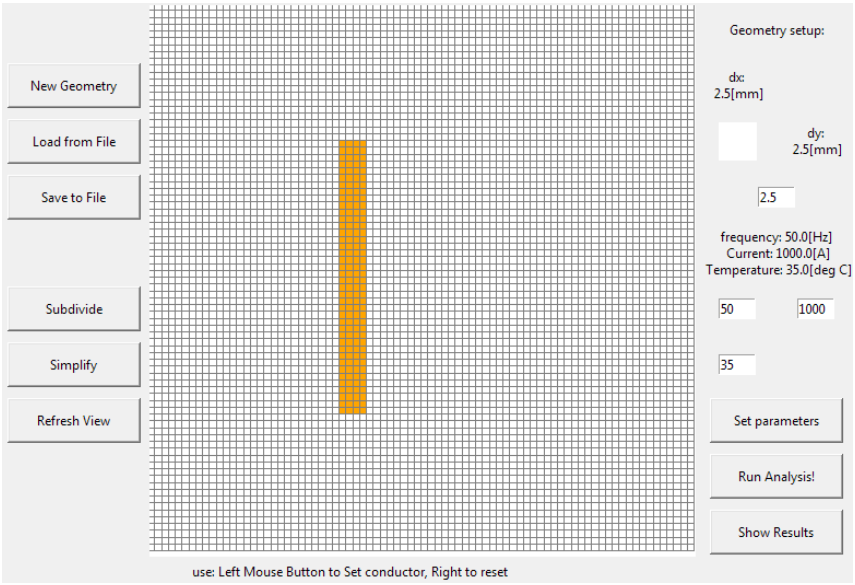
2. Main window description:
3. Defined geometry of copperbar 100x10 on initial grid 10x10mm size:

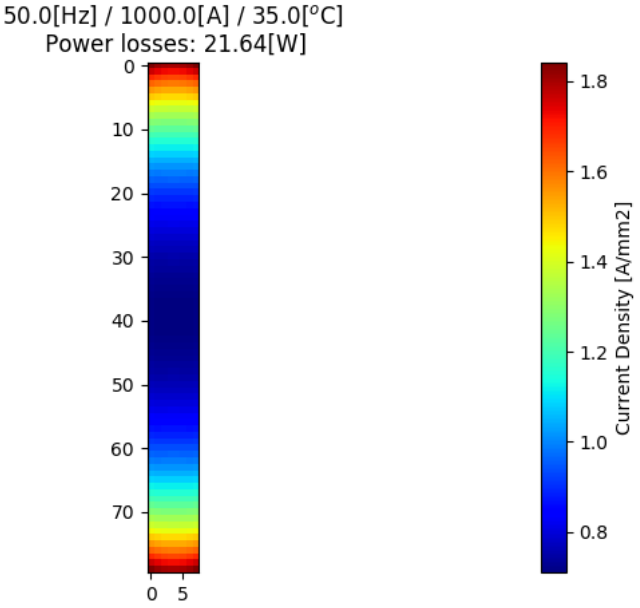


4. Now we can click "Run Analysis!" and we will get the following results:



5. The above results are not very spectacular due to low number of samples per bar. Let's fix it by clicking "Subdivide" 2 or 3 times getting:





6. And Run it again with results:

7. From here we can go and try other cross sections and shapes.

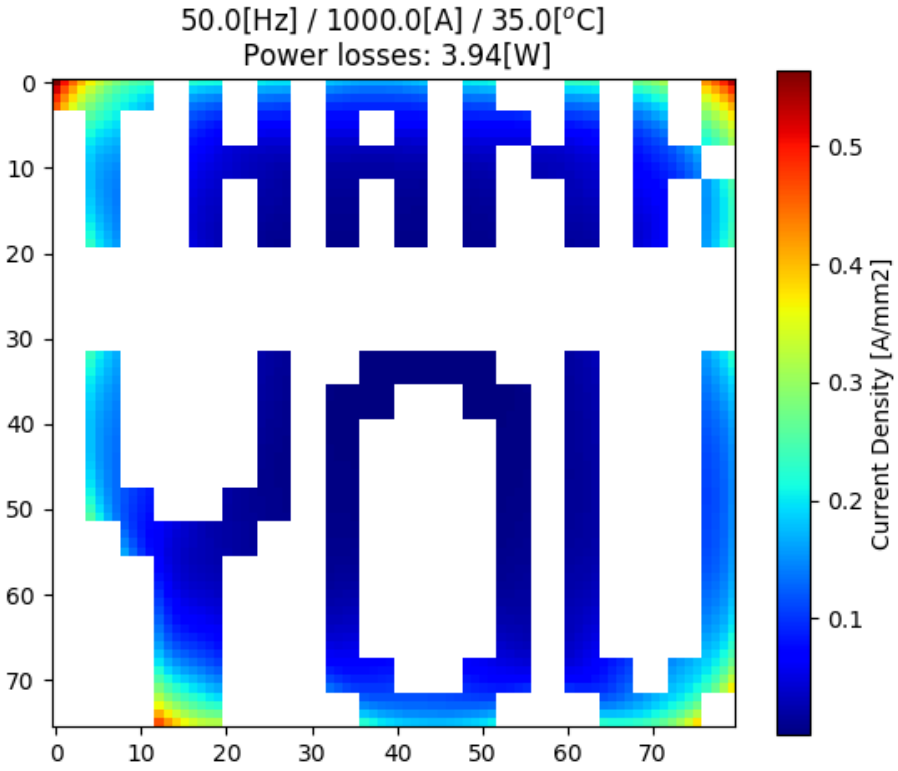


Figure 2: thank you!