1. Introduction

This file is an instruction created to provide a basic introduction to the functions of our FEM program. Only information regarding the use of the program is presented here as the theory has no impact on its usability.

First the instruction will provide a basic installation guide, then a rundown of functions of all elements of the interface and finally an example case.

2. Installation and running the app

For the application to run the following is required:

- Python >= 3.7 (newest possible recommended)
- Kivy >= 2.3.0
- Numpy
- Scipy

The application was created using python 3.12 and Kivy 2.3, however the package webpage specifies that only python 3.7 is required. This should work but was not tested.

The download for python can be found at the official website at [Download Python | Python.org](). Meanwhile the packages can all be installed using pip by calling **pip install** with the necessary argument.

- pip install Kivy
- pip install numpy
- pip install scipy

If all the components are installed, then the application can be started by running the **main.py** file using python with **python main.py.** Outside of the **main.py** file there are two more files needed to run the application. The **main.kv** provides the UI layout and the **FEM_module.py** contains the actual algorithm.

If all is well then, the app should launch the following window.

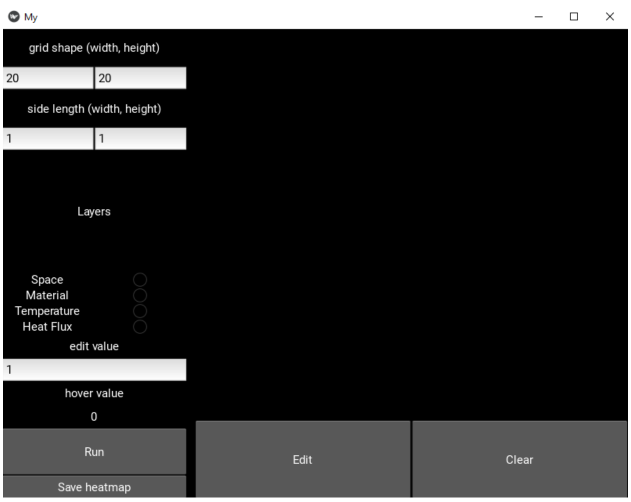

Figure 1 screenshot of the app window

3. Left panel

By the left panel we understand this part of the UI.

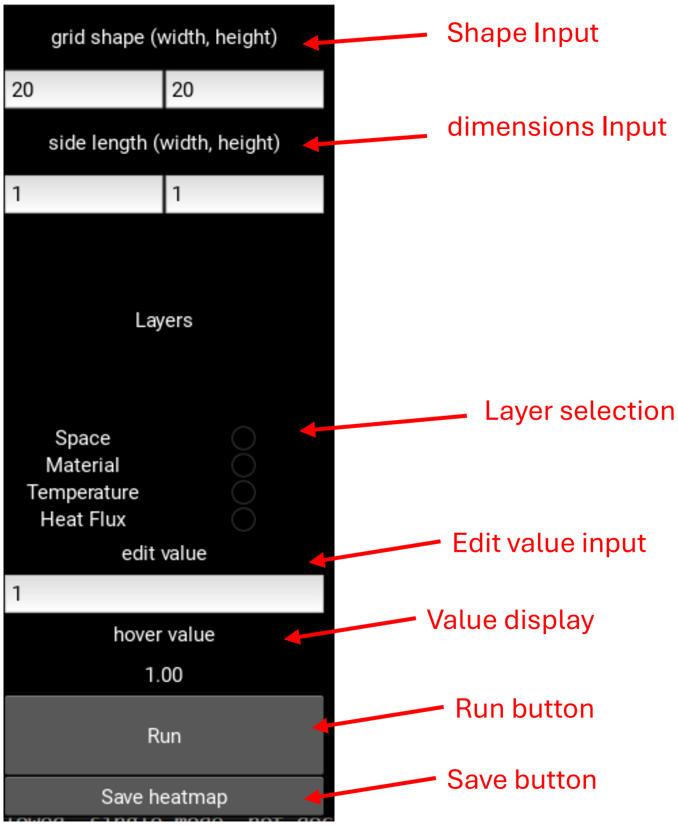

Figure 2 screenshot of the left panel

Shape input

Shape input controls the number of nodes in x and y direction. Specifically, the left input box sets the number in the x direction and the right box in the y direction. Both boxes only accpect integer inputs.

Dimensions input

Dimensions input controls the physical side length of the entire scene. Simillarly to the shape input the left box controls the length in the x direction and the right box controls the length in the y direction. Both boxes accept floating point numbers, but negative values are ignored and set both lengths to 1.

Layer selection

Layer selection is a collection of four toggle buttons which allow us to select which layer do we want to view/edit. Information about editing layer will be further presented in the canvas section.

The layers are as follow:

- Not selecting any of the buttons will set the layer to the heatmap. The heatmap cannot be edited and is empty by default. After running the simulation it contains the Temperature distribution.
- The space layer contains information about where nodes should and should not be. By default, all nodes are set to 1 (node does exist). This layer can be edited and can take on only two values 1(node does exist) and 0(node does not exist). This is how we can introduce opening to the structure.

- The material layer contains information about the thermal conductivity of the material. The value of the thermal conductivity can be set separately for each node by editing the layer. All positive values are allowed.
- The Temperature layer sets the Dirchlet conditions on nodes. This is how we can set boundary conditions, however there is no problem with setting this on any other element. This layer can be edited and can take on any positive value.
- The Heat Flux layer sets the Neumann conditions on nodes. This is how we can set boundary conditions, however there is no problem with setting this on any other element. This layer can be edited and can take on any real value.

Edit value input

This field controls the value which is used when editing the layers with the canvas. All floating-point numbers can be inputted, but some layers have their own logical restrictions.

Value display

This fields displays the value of the layer under the cursor. When the cursor is outside of the canvas, the last value is stored.

Run button

Runs the simulation and updates all values. If the grid shape is changed running the simulation will clear all layers, since there is no consistent way of keeping the existing values.

Save button

Saves both the values of the temperature and coordinates of the nodes to .csv files. The files are named heatmap.csv, and coords.csv.

4. Canvas

Canvas is used to view and edit individual layers. At startup the canvas is completely black, but any input will update it.

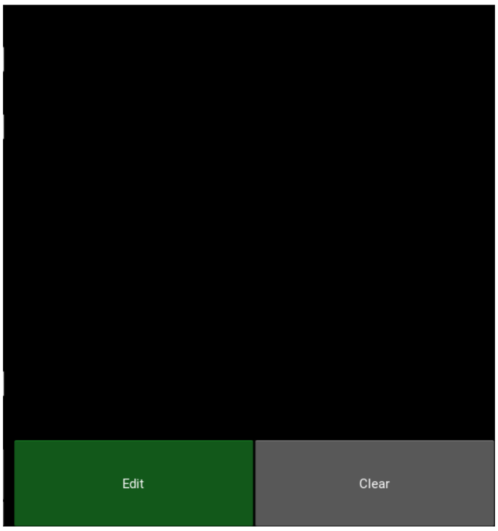

Figure 3 canvas at startup

The Edit and clear buttons at the bottom switch between editing and clear mode.

Editing any layer, which can be edited is done in the same way. Pressing an holding the left mouse button will create a draggable rectangle. After letting go of the left mouse button all values of the selected layer under the rectangle will be set to a new value. In edit mode this value is whatever is typed into the edit value box, and in clear it is 0 or 1 depending on the context.

When the cursor is within the canvas boundary the display value will show the value of the element under the cursor.

5. Example case

Let us consider a case which utilizes all the elements of the app.

After launching the app and running the simulation we are presented with the default scene.

Figure 4 the default scene

First let's change the geometry. We will use an asymmetric grid with width 14 and height 19, and an object with side lengths of 2.3, 4.1. The specific values do not matter. After inputting all the values and hitting Run we should now see a uniform blue scene.

Figure 5 Scene after geometry edits

This is because after changing the geometry the program cleared all layers as explained before.

Lets start by adding the Dirchlet conditions. To do this, first select the Temperature layer on the left panel. The canvas display should not change, since it is also filled with zeroes. In our example we will set the entire left edge to $T = 200K$ and the right edge to $T = 400K$. Starting with the left edge, type 200 into the edit value box and draw a rectangle over the left edge, while trying to only select the left most elements as shown below.

Figure 6 setting the boundary conditions on the left edge

Repeat the same process to set the right edge to $T = 400K$. The result should look like this.

Figure 7 both edges set

Now we run the simulation again and if we unselect the Temperature layer we can see the temperature distribution.

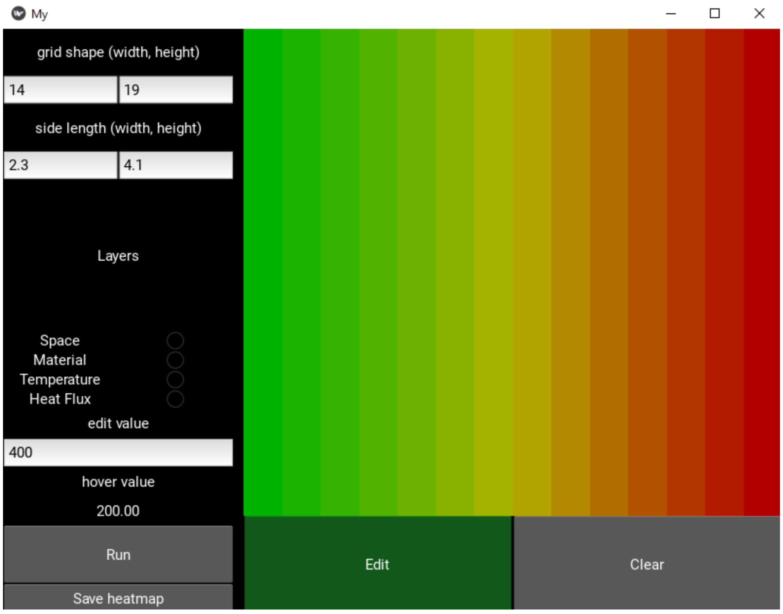

Figure 8 First results

This is decent, but we can make it more interesting.

Now select the material layer and let us change the default material to something more complex. Right now the entire layer is set to 1.

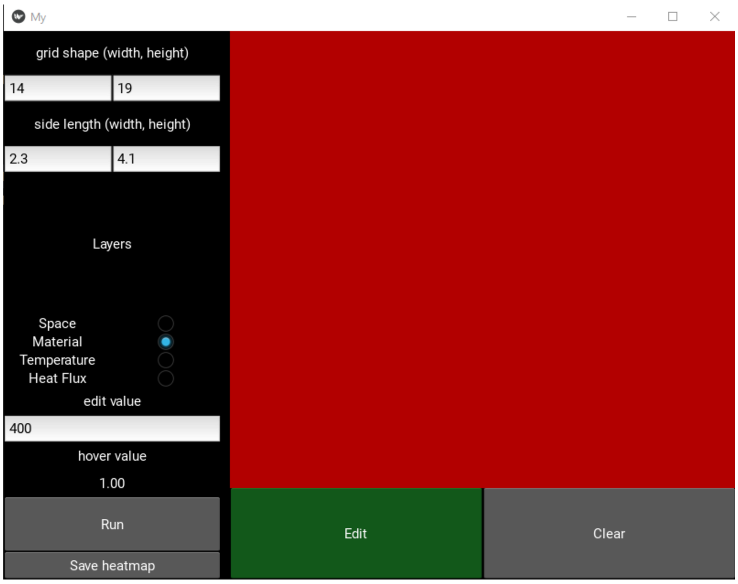

Figure 9 the default material layer

Let's make it more uneven. To do this set to edit value to 20 and drag the rectangle over the entire canvas. It should look like the same but the hover value should show 20 instead of 1.

Now set the edit value to 100 and set a part that looks roughly like this to it.

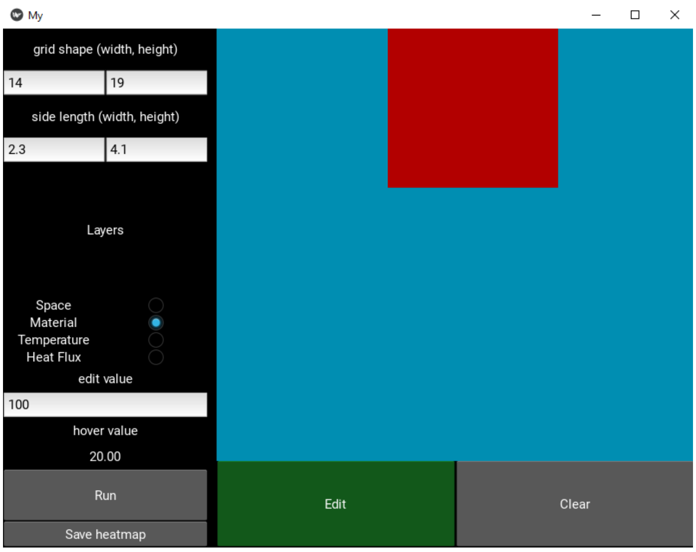

Figure 10 material layer after the edit

Now if we run the simulation we get something like this

Figure 11 result after the material layer edit

Now let's add an opening to the structure. Select the space layer, change the edit mode to clear and add a hole at the bottom. It should look like this.

Figure 12 Space layer after the edit

Now the simulation result looks like this.

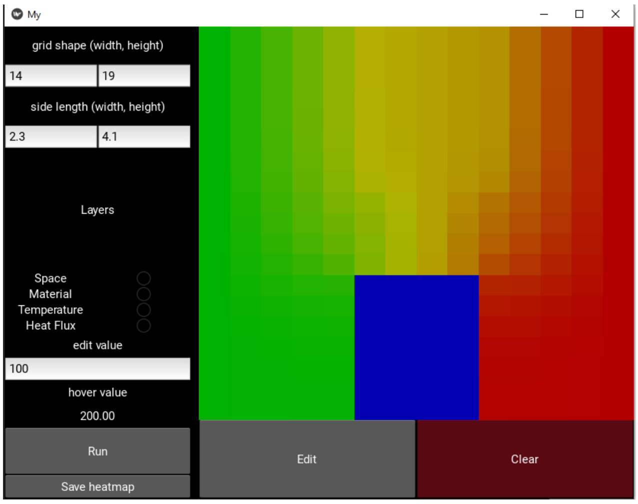

Figure 13 the result after the space edit.

This look a bit silly since the opening is fully isolated. So let’s add a negative heat flux to every node that touches the opening. This can be tricky since it’s not easy to know exactly which nodes should be included. I did just by trial and error while switching between the space layer and the heat flux layer trying to make the selection included one node more in every direction. For the value of heat flux -400 seems to work quite nicely.

Figure 14 the heat flux layer after the edit

Finally, after running the simulation, we get this.

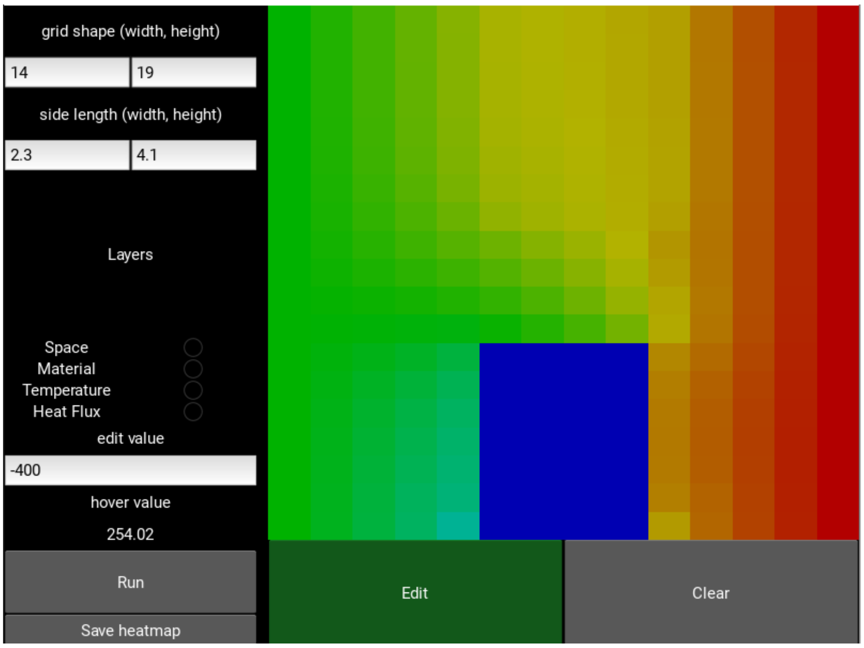

Figure 15 The final result.