Rozvoj lidských zdrojů TUL pro zvyšování relevance,
kvality a přístupu ke vzdělání v podmínkách Průmyslu 4.0

CZ.02.2.69/0.0/0.0/16_015/0002329

Electrotechnology

Studijního program: N0714A150003 - Mechatronics

Description of the FEMM program interface Finite Element Method Magnetics - FEMM

Miroslav Novák

TECHNICKÁ UNIVERZITA V LIBERCI
www.tul.cz

Obsah

Version:

2017/10 1.0 original document

1 Introduction of the FEMM SW package

FEMM is a software package for solving problems of low-frequency electromagnetic problems in 2D, namely planar problems or axially symmetric problems. Linear / non-linear magnetostatic problems, linear / non-linear magnetic problems with harmonic excitation, linear electrostatic problems, and steady state heat flow problems can be solved.

FEMM is divided into three parts:

• Interactive editor (femm.exe). This program is a pre-processor and post-processor for all types of problems solved by FEMM. It contains a simplified interface similar to CAD for plotting the geometry of the problem and for defining material properties and boundary conditions.

For convenience, Autocad DXF files can be imported to analyze existing geometries. The postprocessor allows the user to view the field at any point, and evaluate a number of different integrals and plot the quantities on the user-defined contours.

• triangle.exe is an automatic finite element mesh crosslinker, ie it divides the geometric areas of the model into a large number of triangles.

• Solver (fkern.exe for magnetics; belasolv for electrostatics; hsolv for thermal problems; and csolv electromagnetic problems with harmonic excitation). Each solver processes data files that describe the problem and solves the appropriate partial differential equations to obtain values for the required field.

The Lua scripting language is integrated into the interactive editor. Lua can compile and analyze geometry and evaluate the results of post-processing, thus simplifying the creation of various types of "batch" calculations.

In addition, all edit fields in the user interface are analyzed using Lua, which allows you to enter equations or mathematical expressions in any edit field instead of a numeric value. In any edit field in FEMM, the selected piece of text can be evaluated by Lua by selecting it in the menu on the right mouse button.

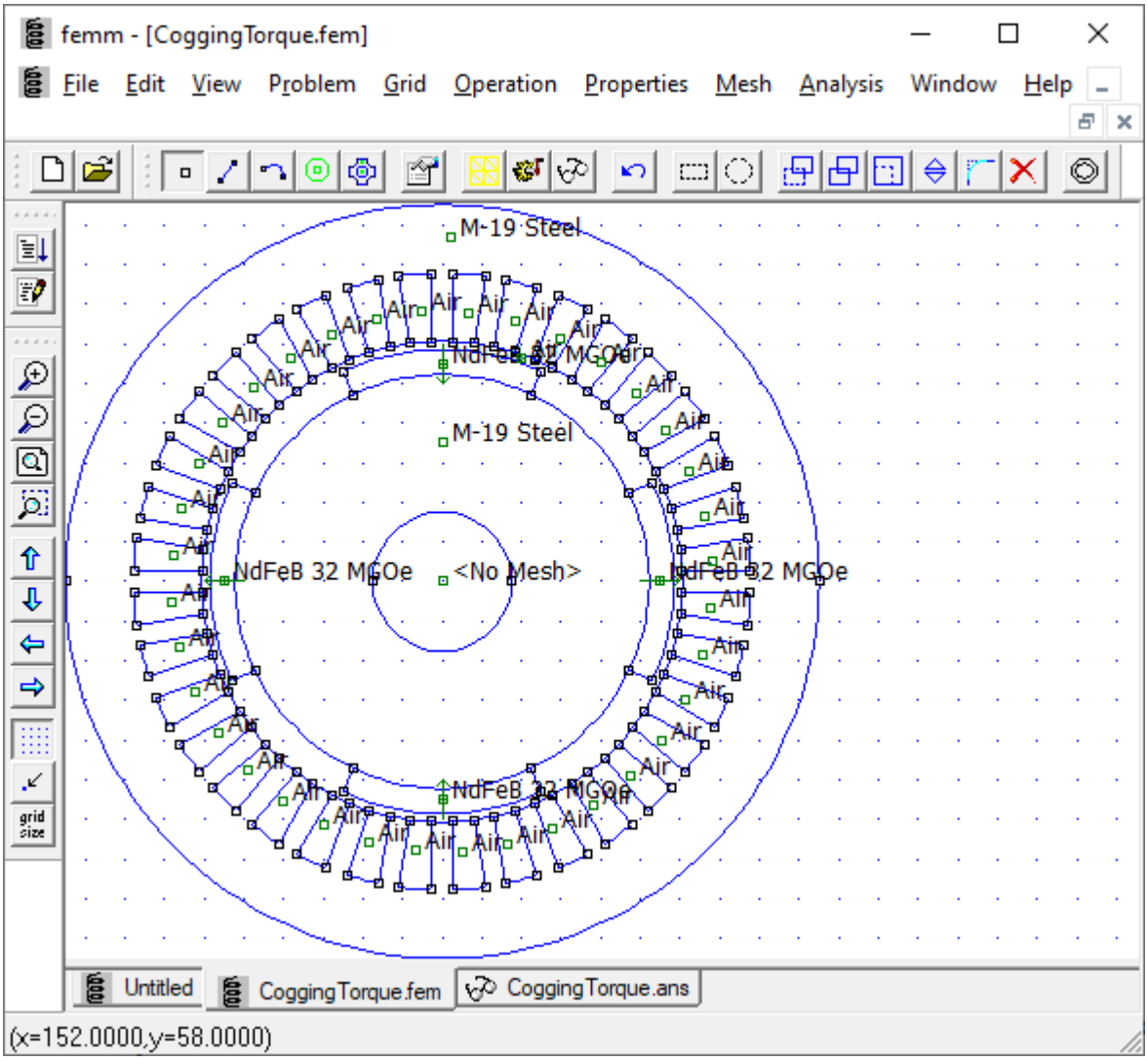

Fig. 1: Interactive editor

2 Installation

We can download the program from the website https://www.femm.info/ Download section. The installation will proceed smoothly. It is advisable to set the destination directory to the root of the disk. It is not advisable to use national characters to name the installation directory. The default setting is correct and we can use it.

The Dwonload section also describes the integration of the SW package into other products: Matlab, Octave, Scilab, Mathematica, Python.

FEMM is licensed under the terms of the Aladdin Free Public License. The network generator and the Lua scripting language that are part of the distribution each have their own license terms.

The network generator is free to use except for sale to other parties. The LUA language is completely free in the MIT license, including subsequent sales.

3 Creating a task

We define the task from the menu - File - New. In the following dialog, select the type of problem. Each job type creates a file with a different extension. When opening files, it is necessary to choose which extensions are filtered.

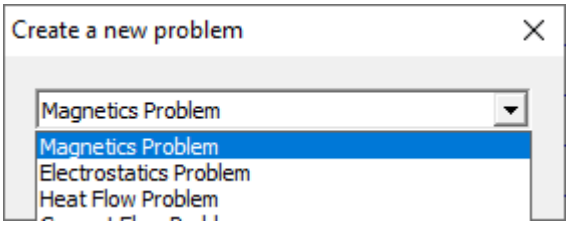

Fig. 2: Problem type selection dialog

Subsequently, it is necessary to define the parameters of the task in the menu - Problem. In the dialog we will discuss whether the task will be planar or rotationally axially symmetric. We will not forget to select units. We then define the geometry of the task in these units.,

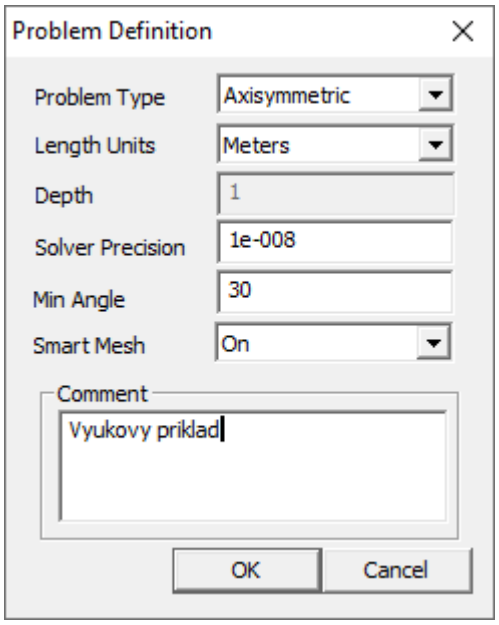

Fig. 3: Task definition

4 Model geometry creation

The geometry editor works with three different types of files corresponding to the problems solved: magnetics, electrostatics, and heat fluxes.

The finished 2D geometry can be imported from a * .dxf file (AutoCAD version 13). Import is performed from the menu - File - Import DXF.

Manual geometry entry is done with the mouse. The input philosophy is sequential: point - line / arc - area. Therefore, you must first specify the endpoints of the objects. This is done by a small square icon. In the status bar of the application (bottom left) we monitor the location of the point. The values correspond to the units specified in the problem definition. Each point can be assigned to a group (In Group). This is done by selecting the yoke with the right mouse button - the point turns red and then from the menu - Properties or the eighth hand icon with the form.

Fig. 4: Interactive editor icon bar

We can connect the distributed points with lines. Select the fourth icon (line) and click on the start and end point. Similarly, we can enter circular arcs. Fifth icon with an arch. Select the start and end points and enter the angle of the arc in the dialog. We influence the direction of the arc by which point we select first.

For lines and arcs, we can redefine the group assignment in the properties and also assign a boundary condition.

If we want to define an area, we have to create it using points, lines and arcs. The area must be closed. Then we add an area handle inside the area (the sixth green circle icon around the point). We then right-click the area handle and set its properties. The most important thing is to select the material type - popUp Block type. For electromagnetic problems, we can also assign the part of the circuit to which the area belongs (In Circuit) and determine the number of turns of the coil. For magnetic problems, we determine the polarization direction of the magnet in degrees.

For convenient editing, we will use the 12th back arrow icon. A The following two icons for grouping objects bounded by a rectangle or circle. We must first select the type of objects we want to select (points, lines, arcs, area definitions) and then make a selection with a rectangle or circle. Another set of icons is used to move objects, copy, transform, mirror, corner, and delete. Entering points in exact mouse positions is not very reliable. Advantageously, therefore, we use the icons copy, move, mirror in their dialogs, you can enter the exact rates of shifts and transformations. The geometry created in this way is accurate. Advantageously, we can copy and mirror entire complex parts of already defined geometry.

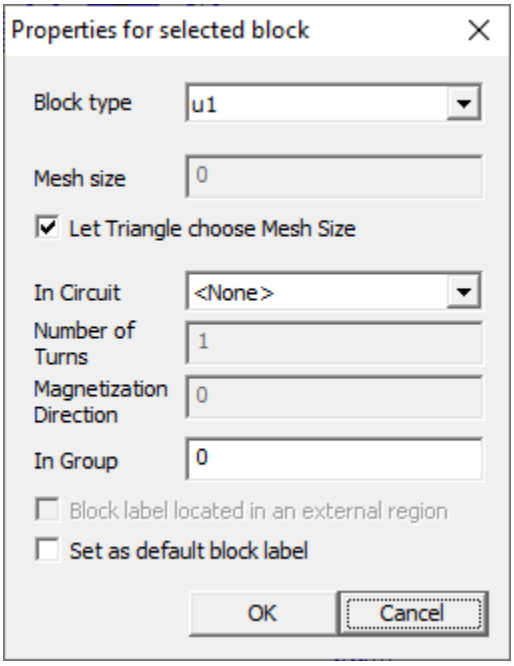

Fig. 5: Pop up window of area parameters

The last icon in the horizontal speed dial bar is used to generate a boundary condition simulating open space / infinity. It generates a set of concentric circles and assigns them the correct values of the Dirichlet boundary condition.

We will also use the vertical bar of icons to manipulate objects. The first of the top is to **run the LUA script**. The LUA console with the editing and status window opens. The following are 4 icons for zooming the image. The next 4 icons are used to move the drawing area. The next icon turns the auxiliary raster on and off.

Fig. 6: Vertical icon bar

5 Setting material properties

We define the properties of areas for individual areas. A database of the most common materials is integrated in the Editor. It can be accessed via the menu - Properties - Materials Library. In the subsequent dialog, we select the material from the left tree and add it to the model by dragging it with the mouse to the right half. Only materials dragged into the right window are then accessible in the model.

By double-clicking on the material we can change its properties. I do not recommend this procedure for material from the library. It is better to create new material. Both in the library and in the right window of the model, materials can be added in the popup menu of the right mouse button.

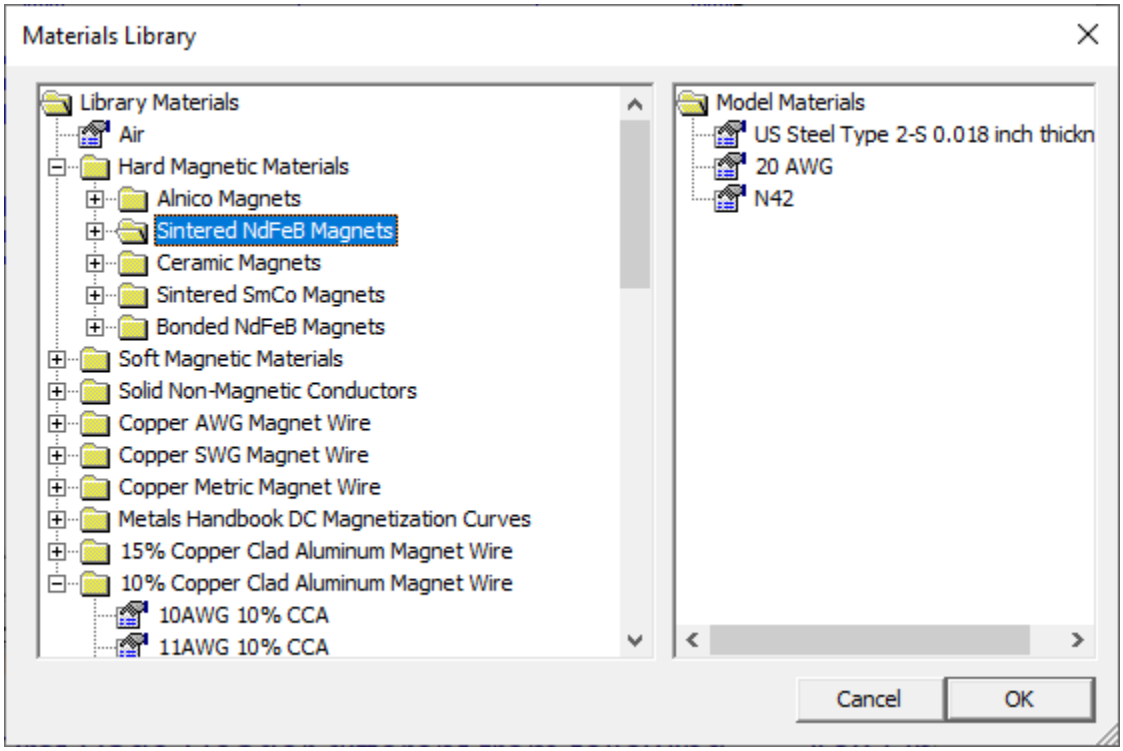

Fig. 7: Browsing the library of materials

Each material must have a unique name. The main parameter is the permeability or BH curve. The program therefore allows you to enter linear materials and into isotropic and anisotropic in the x and y axes. Nonlinear magnetics are entered using points on the BH curve. The minimum number is 3 points. All points are entered in the 1st quadrant. The system automatically considers the curve to be symmetric. The material hysteresis is entered by the magnitude of the coercive force. The conductivity of the material is also taken into account for magnetodynamic problems.

At the bottom of the dialog, we can reduce eddy currents by defining the lamination of the material and its filling factor. For lamination, we choose the direction relative to the geometry of the model. Here, we set the cross-sections of the coil conductors and the filling factor for the windings.

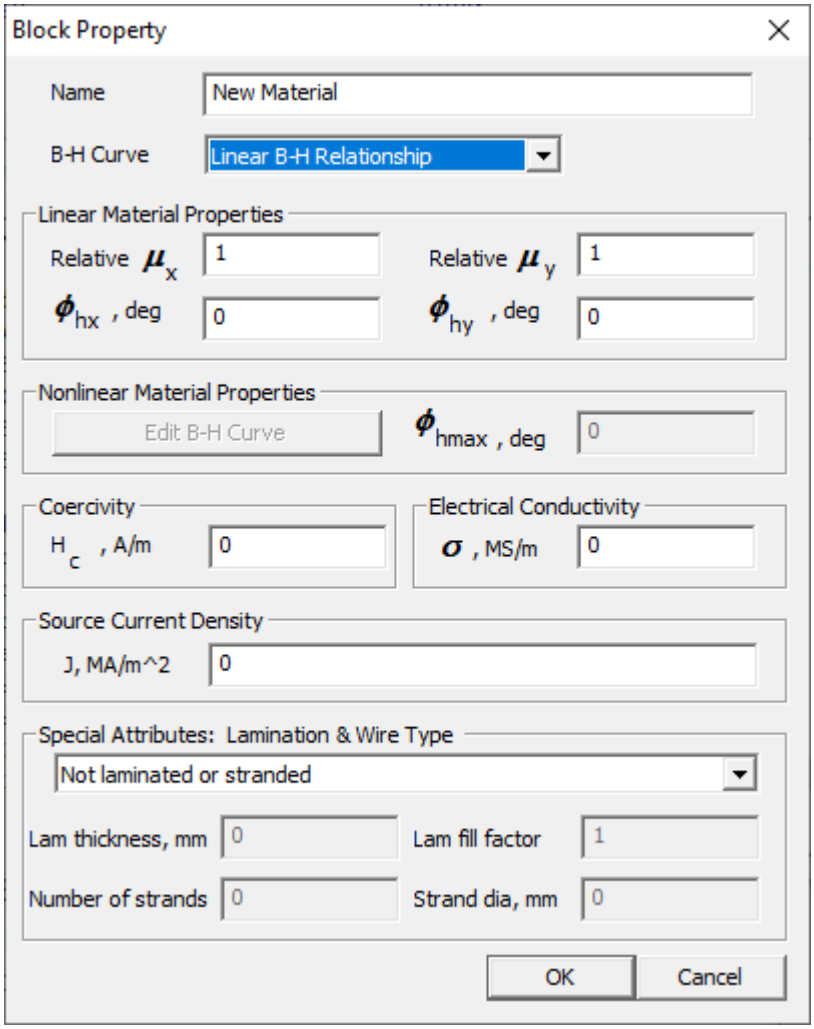

Fig. 8: Material definition dialog in electromagnetism tasks

6 Setting circuit parameters

Defining a circuit allows you to apply constraints to the current flowing in one or more blocks. Circuits can be defined as "parallel" or "series" connected. If "parallel" is selected, the current is divided between all areas marked by this circuit property based on impedance (the current is divided so that the voltage drop is the same on all sections). Only fixed conductors can be connected in parallel. If “series” is selected, the specified current is applied to each block marked by this circuit. Blocks that are marked with a series circuit property can also be assigned a number of turns, so that the area is treated as a stranded wire in which the total current of the series circuit times the number of turns in the area. The number of threads assigned to a block can be a positive or negative number. The sign on the number of turns indicates the direction of current flow associated with a positive value of the circuit current. For magnetostatic problems, it is possible to alternatively use the source current density and achieve the same result.

For eddy current tasks, defining a circuit is much more useful - they allow the user to directly define the current and allows you to assign specific connections to different areas of geometry. This information is used to obtain impedance, mutual inductance, etc., in a relatively simple way in the postprocessor. By using circuit properties, it is also possible to force a connection in case of eddy current problems. By default, all objects in eddy current tasks are "shorted to infinity".

We define the circuit in the menu - Properties - Circuits - Add Property. We can define more circuits in the task.

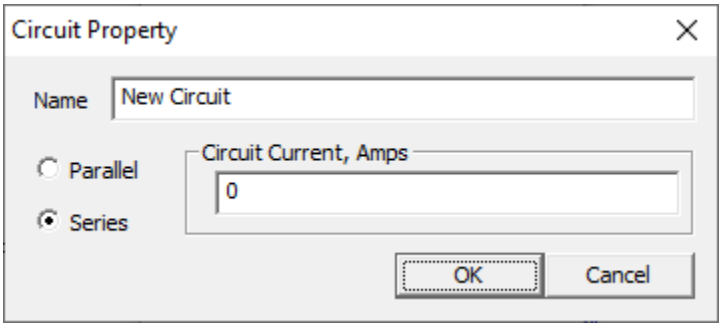

Fig. 9: Circuit parameter setting dialog

7 Setting boundary conditions

Okrajové podmínky jsou nezbytné pro zaručení korektního výpočtu úlohy.

7.1 Magnetické a elektrostatické okr. podmínky

Boundary conditions for magnetic and electrostatic problems exist in five variants:

- **Dirichlet**

In this type of boundary condition, the value of potential A or V is explicit defined at a given boundary, eg A = 0. The most common use of a Dirichlet-type boundary condition in magnetic problems is to define A = 0 along the boundary in order to maintain a magnetic flux exceeding this boundary. For electrostatic problems, Dirichlet conditions are used to define the stress on the surface of the part.

- **Neumannova**

This boundary condition specifies the derivative of the potential at the normal to the object boundary. In magnetic problems, the homogeneous Neumann boundary condition, ∂A / ∂n = 0 is defined along the boundary so that the flux crosses the boundary exactly at an angle of 90 ° to the surface. We use this type of boundary condition for materials with high permeability.

- **Robin**

Robin's boundary condition is a kind of mix between Dirichlet and Neumann, prescribing the relationship between the value of A and its normal derivative at the boundary. An example of this boundary condition is: A / ∂n + cA = 0.
This boundary condition is used to define "impedances" that allow a restricted domain to mimic the

behavior of an unbounded region. In the context of heat flow problems, this boundary condition can be interpreted as a convection boundary.

- **Periodic**

Periodic boundary conditions connect the two boundaries together. The limit values at the corresponding points of the two borders have the same value and sign.

- **Anti-periodic**

The anti-periodic boundary condition also connects the two boundaries. However, the thresholds are designed to be the same size but with the opposite sign.

If no boundary conditions are explicitly defined, each boundary is by default a homogeneous Neumann boundary condition. However, at least one non-derivative boundary condition must be defined in the solved geometry (or the potential at one reference point must be defined) for the problem to be solvable.
For axially symmetric magnetic problems, A = 0 on the line r = 0. In this case, the solution can be calculated without explicitly defining any boundary conditions if it is part of the geometry of the problem boundary lying along r = 0. However, this does not apply to electrostatic problems. For electrostatic problems, we have a solution with a nonzero potential along r = 0.

7.2 Boundary conditions for heat flow problems

There are six types of boundary conditions for heat flux problems:

- **Constant temperature**

The temperature along the limit is set to the prescribed value.

- **Heat flow**

The heat flux across the boundary is prescribed. This boundary condition can be mathematically represented as: $k*\partial T/\partial n + f = 0$,
where n represents the direction of the normal to the boundary.

- **Convection**

Convection occurs when the boundary is cooled by a fluid stream. This boundary state can be represented as: $k*\partial T/\partial n + h(T - T_o) = 0$
where h is the "heat transfer coefficient" and T_o is the ambient coolant temperature.

- **Radiation**

The heat flux through the radiation can be mathematically described as: $k*\partial T/\partial n + \beta k_{sb}(T^4 - T^4_o) = 0$
where β is the surface emissivity (dimensionless value between 0 and 1) and k_{sb} is the Stefan-Boltzmann constant.

- **Periodic**

Periodic boundary conditions connect the two boundaries together. In this type, the boundary condition defines the boundary values at the corresponding points of the two boundaries in the same way.

- **Anti-periodic**

The anti-periodic boundary condition also connects the two boundaries. However, the thresholds are designed to be the same size but with the opposite sign.

If no boundary conditions are explicitly defined, each boundary is the default isolated condition (ie no heat flux across the boundary). However, the non-derivative boundary condition must be defined somewhere in the problem to be solved (or the potential must be defined at one reference point).

7.3 Use of boundary conditions

Boundary conditions must first be defined - the top menu of the application - Properties - Boundary - Add property - Selection of the condition type (BC Type) + addition of parameters. We will also not forget to meaningfully name the condition (Name). Under this name, we then assign the condition to the boundaries of the geometry.

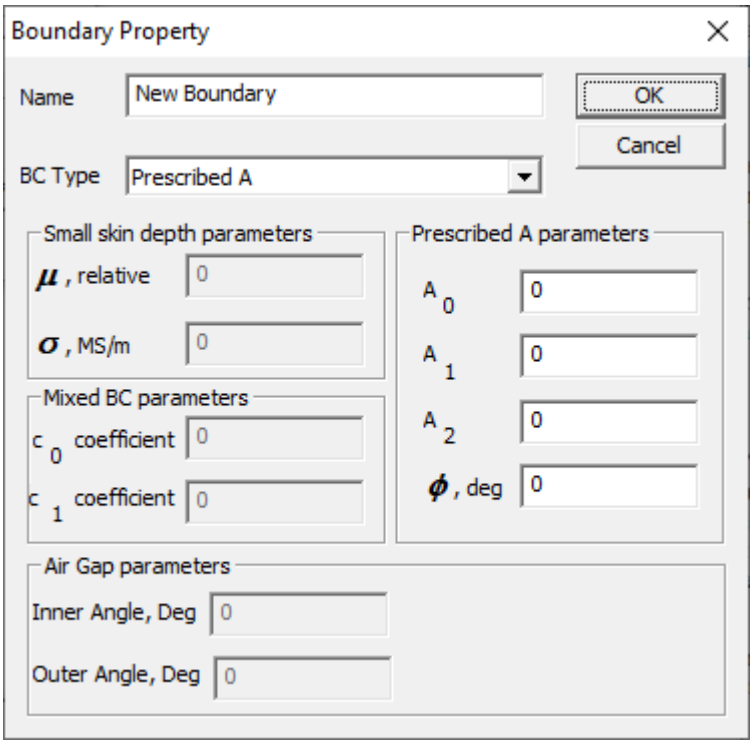

Fig. 10: Boundary condition settings dialog

We define boundary conditions in the finished geometry by selecting lines or arcs (right mouse button) - after selecting the lines turn red and you can select more than one at a time. Then select - the top menu - Operation - Open Selected or the hand icon above the dialog from the speed dial bars. In the dialog, select a predefined condition from the rollup.

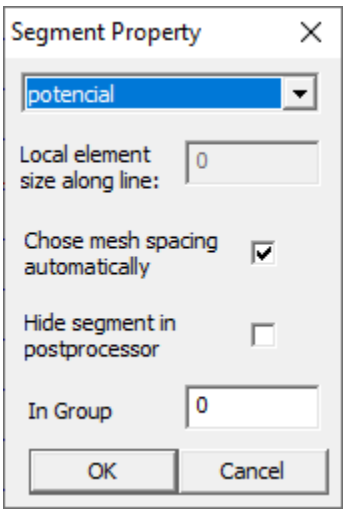

Fig. 11: Assign a boundary condition to an edge or arc

8 Meshing

Before the actual solution, it is necessary to mesh the defined geometry. To do this, use the icon with a yellow triangular mesh or program menu - Mesh - Create mesh. The geometry plotter draws the designed mesh. If the mesh density does not suit us, we can influence it:

- Adding boundaries in sensitive areas
- By lowering the limit angle in the problem definition.

If the task is not networked, networking will start automatically after clicking on the solver icon.

9 Solver

The solver is started from the menu - Analysis - Analysis or the wheel icon with a handle. The calculation configuration is specified to define the problem.

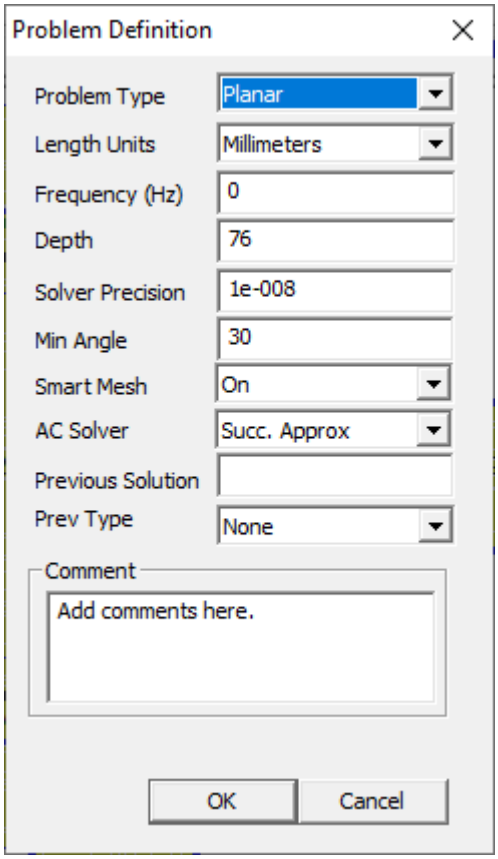

Fig. 12: Task definition

10 Postprocesor

The postprocessor displays the solved fields. We can choose to display equipotential. We can display the magnitude in the form of a colored field. You can also display a vector field in the form of a network of arrows. These views are arbitrarily combined. For each view, we can set limits and the number of lines or color smoothing, or the density and size of the arrows.

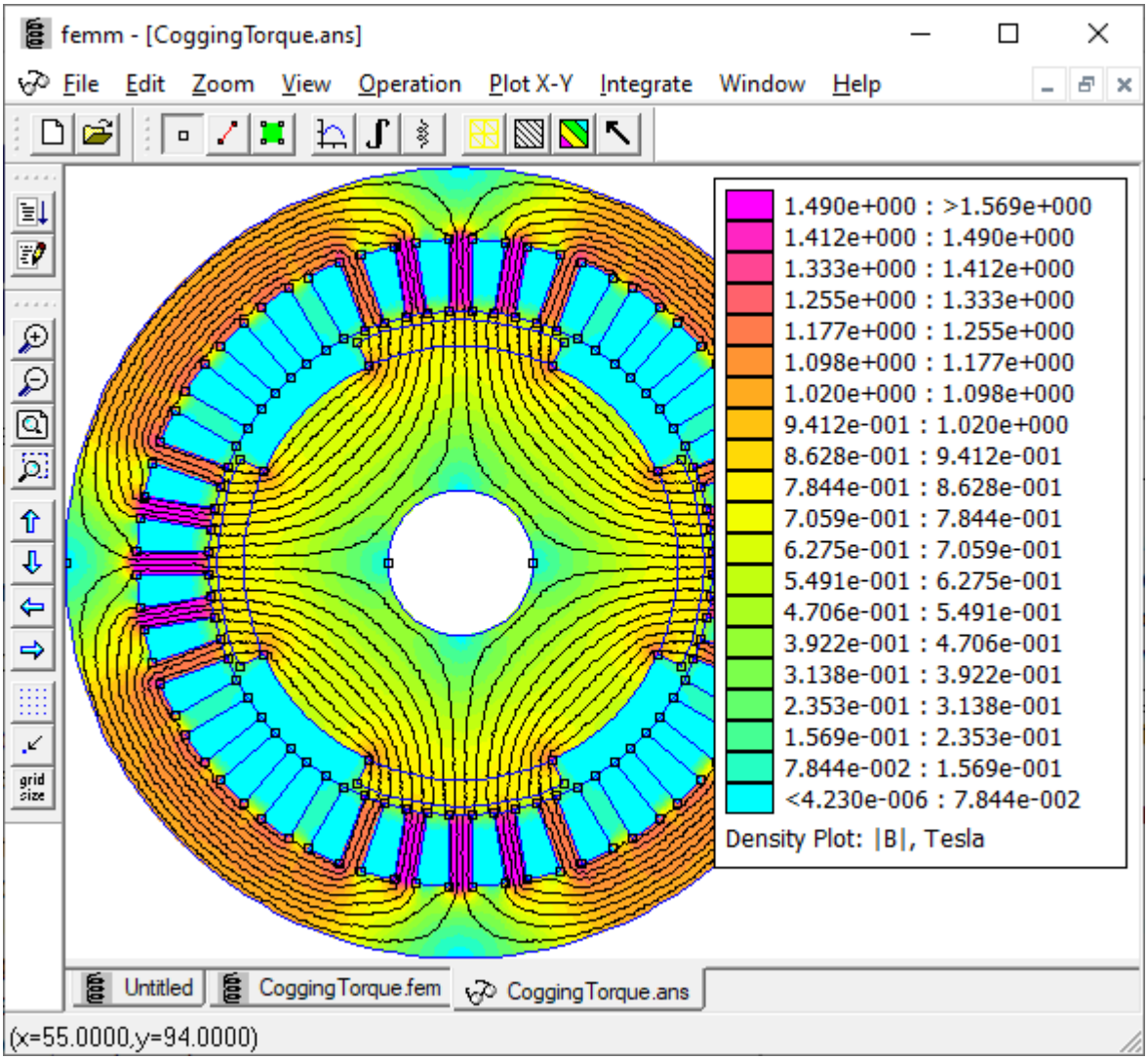

Fig. 13: Interactive editor in postprocessor mode

An important function is to display values at a point, boundary, or area. It's a menu - Operation. When selecting Point properties or selecting a point icon, we must click on one of the geometry points. The application displays a pop-up window with the values corresponding to the point.

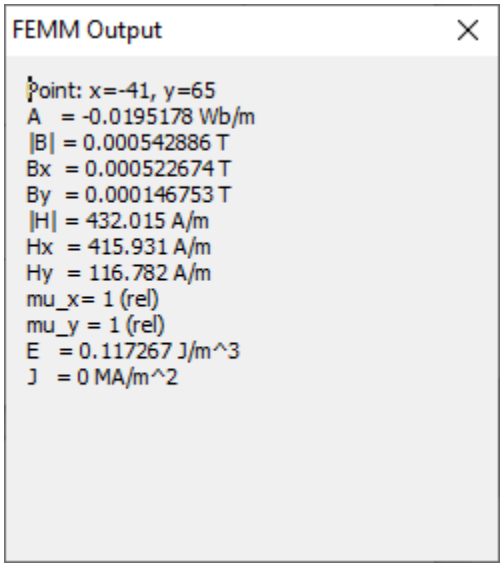

Fig. 14: Point property

To display the course in the section / boundary of the geometry, we must first define the boundary - red line icon or menu - Operation - Contour. Then, by clicking into the geometry, we create a polyline on which by activating the graph icon or from the menu - Plot X-Y opens a dialog box and then displays the plot/graph. In this dialog we can save the values to disk for processing in another SW product.

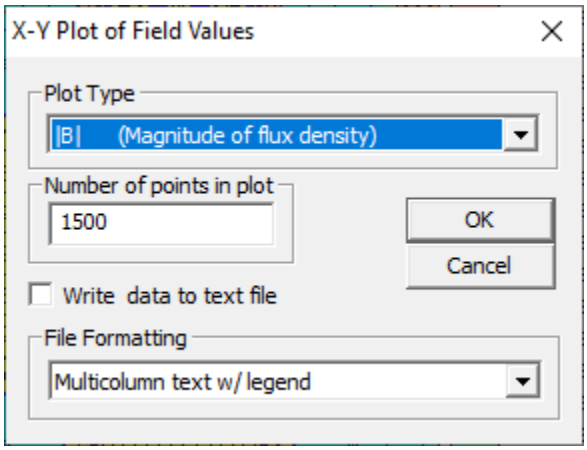

Fig. 15: Plot definition

At this limit we can also calculate the value of the curve integral from the selected quantity. This is done by pressing the integral icon or from the menu - Integrate.

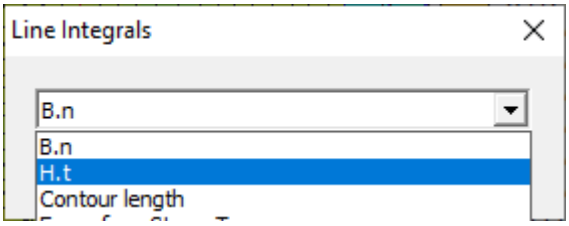

Fig. 16: Pop-up window for selecting the integration variable

11 Examples and documentation

To make it easier for beginners, the installation contains a number of interesting examples. I recommend trying at least some. Examples are called by default when you open a file in interactive editors. The default directory is named examples (in the program installation directory). Please note in the dialog it is necessary to select the extension = type of task we want to upload at the bottom right.

Further examples can be found at https://www.femm.info/wiki/Examples. There is also a description and explanation of the task.

Documentation in the form of pdf manuals and tutorials is available at https://www.femm.info/wiki/Documentation/.

12 Scripting in LUA

LUA scripting is a very powerful tool. We have to write scripts in a flat text editor outside the FEMM program's own interface. Run the script with the first icon from the top in the vertical icon bar. This option is not available from the menu.

In addition to standard commands and operators, we can use the problem-oriented functions described in the FEMM manual. In this way we can create a file with the task, define its parameters. We create geometry by using commands to create a point, line or arc connection. With other commands we insert materials from the library or we do not define our own. We can also define circuit properties and boundary conditions. We then assign the material to the areas and boundary conditions to the edges in the geometry.

The commands run the solver and control the postrocessor.

and	break	do	else	elseif	end
false	for	function	goto	if	in
local	nil	not	or	repeat	return
then	true	until	while		

Fig. 17: List of LUA keywords

The advantage of scripting is that we enter the geometry with exactly the given coordinates. We can modify the finished model and run the calculations in a loop repeatedly. This makes it easy to obtain dependencies on the change of various parameters.

Example LUA script:

```
--open("pukus.FEM") --opens the desired file
showconsole() --shows konsole
clearconsole() --clears konsole
newdocument("pokus.FEM")
```

```
--problem setting
mi_probdef(0,'millimeters','planar',1e-8,44,30)
mi_zoom(-150,-150,100,150)

--material definiton
mi_getmaterial("Air") --load material def. From library
mi_getmaterial("Soft magnetic ferrite (Fe-Ni-Zn-V)")  --core
mi_getmaterial("Cold rolled low carbon strip steel")  --sample
mi_getmaterial("0.5mm") --Cu wire

--circuit definition
current = 1
mi_addcircprop('civka',current,1)  --serial connection, current 1A
mi_addcircprop('civka1',0,1)       --serial connection, current 0A

----------------------------------------------------geometrie definice
gr_oblastR = 150 –size of surrounding area for simulation
gr_magMaterial = 'Soft magnetic ferrite (Fe-Ni-Zn-V)';
gr_vzorekMaterial = 'Cold rolled low carbon strip steel';
gr_civkaMaterial = '0.5mm'

--core size U100/57/25
gr_a = 101.6     --whole height
gr_b = 57.1      --whole width
gr_c = 25.4      --side width
gr_d = 31.7      --side height

gr_vzorek_a = 3.5    --sample height
gr_vzorek_b = 250     --sample width

--coil dimensions
gr_civkaVyska = 0.5
gr_civkaSirka = 40

gr_AirGap = 16 –air gap between yoke and sample
gr_civkaH = 18 –thickness of coil
gr_bobinh = 2.2+0.9/2  --thicknesst of coil former
gr_bobinL = gr_a - 2;

gr_sp = 0.2   --error of yoke assemply

coilTurns = 500
----------------------------------------------------sample
gr_name="vzorek";
gr_num=100;
gr_BodyX={-gr_vzorek_b/2, gr_vzorek_b/2, gr_vzorek_b/2, -gr_vzorek_b/2}
gr_BodyY={-gr_vzorek_a/2, -gr_vzorek_a/2, gr_vzorek_a/2, gr_vzorek_a/2}

--points sample
for n=1,4,1 do
  print(gr_BodyX[n])
```

```
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--lines sample
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end
--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1]-1)/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1]-1)/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_vzorekMaterial,0,0,'<None>','90',gr_num,0);

mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1]+1)/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1]+1)/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_vzorekMaterial,0,0,'<None>','90',gr_num,0);

-----------------------------------------------------bottom core
gr_name="mag_dole";
gr_num=101;
--points, groupe definition
gr_BodyX={-gr_a/2, gr_a/2, gr_a/2, gr_a/2-gr_c, gr_a/2-gr_c, -gr_a/2+gr_c, -gr_a/2+gr_c, -gr_a/2}
gr_BodyY={-gr_b-gr_vzorek_a/2, -gr_b-gr_vzorek_a/2, -gr_vzorek_a/2, -gr_vzorek_a/2, -gr_d-gr_vzorek_a/2, -gr_d-gr_vzorek_a/
2, -gr_vzorek_a/2, -gr_vzorek_a/2}

--points bottom core
for n=1,8,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--line drawing
for n=1,8,1 do
 print(gr_BodyX[n])
 m=n
 if m==8 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[2]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[1]-gr_BodyY[6])/2+gr_BodyY[6])
```

```
mi_selectlabel((gr_BodyX[2]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[1]-gr_BodyY[6])/2+gr_BodyY[6])
mi_setblockprop(gr_magMaterial,0,0,'<None>','90',gr_num,0);

-----------------------------------------------------upper core
gr_name="mag_dole";
gr_num=102;
--points drawing, group def
gr_BodyX={-gr_a/2, gr_a/2, gr_a/2, gr_a/2-gr_c, gr_a/2-gr_c, -gr_a/2+gr_c, -gr_a/2+gr_c, -gr_a/2}
gr_BodyY={gr_b+gr_vzorek_a/2, gr_b+gr_vzorek_a/2, gr_vzorek_a/2, gr_vzorek_a/2, gr_d+gr_vzorek_a/2, gr_d+gr_vzorek_a/2,
gr_vzorek_a/2, gr_vzorek_a/2}

--points upper core
for n=1,8,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--lines drawing
for n=1,8,1 do
 print(gr_BodyX[n])
 m=n
 if m==8 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[2]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[1]-gr_BodyY[6])/2+gr_BodyY[6])
mi_selectlabel((gr_BodyX[2]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[1]-gr_BodyY[6])/2+gr_BodyY[6])
mi_setblockprop(gr_magMaterial,0,0,'<None>','90',gr_num,0);

-------------------------------------------------------------coil 1a
gr_name="civka 1a";
gr_num=201;

--points drawing, groups
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={gr_b+gr_vzorek_a/2, gr_b+gr_vzorek_a/2, gr_b+gr_vzorek_a/2+gr_civkaVyska, gr_b+gr_vzorek_a/2+gr_civkaVyska}

for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--lines drawing
for n=1,4,1 do
```

```
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
   mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
   mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
   mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka',0,1,coilTurns);

------------------------------------------------------------------coil 1b
gr_name="civka 1b";
gr_num=202;

--points drawing, groups
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={gr_d+gr_vzorek_a/2, gr_d+gr_vzorek_a/2, gr_d+gr_vzorek_a/2-gr_civkaVyska, gr_d+gr_vzorek_a/2-gr_civkaVyska}

for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--lines drawing
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
   mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
   mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
   mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka',0,1,-coilTurns);

------------------------------------------------------------------coil 2a
gr_name="civka 2a";
gr_num=203;

--kresleni body, prirazeni skupiny
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={gr_vzorek_a/2, gr_vzorek_a/2, gr_vzorek_a/2+gr_civkaVyska, gr_vzorek_a/2+gr_civkaVyska}
```

```
for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--kresleni usecek
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka1',0,2,coilTurns);

------------------------------------------------------------coil 2b
gr_name="civka 2b";
gr_num=204;

--kresleni body, prirazeni skupiny
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={-gr_vzorek_a/2, -gr_vzorek_a/2, -gr_vzorek_a/2-gr_civkaVyska, -gr_vzorek_a/2-gr_civkaVyska}

for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--kresleni usecek
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka1',0,2,-coilTurns);
```

```
------------------------------------------------------------coil 3a
gr_name="civka 3a";
gr_num=205;

--kresleni body, prirazeni skupiny
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={-gr_d-gr_vzorek_a/2, -gr_d-gr_vzorek_a/2, -gr_d-gr_vzorek_a/2+gr_civkaVyska, -gr_d-gr_vzorek_a/2+gr_civkaVyska}

for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--kresleni usecek
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
  mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
  mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
  mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka',0,3,coilTurns);

------------------------------------------------------------coil 1b
gr_name="civka 3b";
gr_num=206;

--kresleni body, prirazeni skupiny
gr_BodyX={-gr_civkaSirka/2, gr_civkaSirka/2, gr_civkaSirka/2, -gr_civkaSirka/2}
gr_BodyY={-gr_b-gr_vzorek_a/2, -gr_b-gr_vzorek_a/2, -gr_b-gr_vzorek_a/2-gr_civkaVyska, -gr_b-gr_vzorek_a/2-gr_civkaVyska}

for n=1,4,1 do
 print(gr_BodyX[n])
 mi_addnode(gr_BodyX[n],gr_BodyY[n])
 mi_selectnode(gr_BodyX[n],gr_BodyY[n])
 mi_setnodeprop(gr_name,gr_num)
end

--kresleni usecek
for n=1,4,1 do
 print(gr_BodyX[n])
 m=n
 if m==4 then m=0 end
```

```
   mi_addsegment(gr_BodyX[n],gr_BodyY[n],gr_BodyX[m+1],gr_BodyY[m+1])
   mi_selectsegment((gr_BodyX[n]+gr_BodyX[m+1])/2,(gr_BodyY[n]+gr_BodyY[m+1])/2)
   mi_setsegmentprop(gr_name,1,0,0,gr_num)
end

--material
mi_addblocklabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_selectlabel((gr_BodyX[3]-gr_BodyX[1])/2+gr_BodyX[1],(gr_BodyY[3]-gr_BodyY[1])/2+gr_BodyY[1])
mi_setblockprop(gr_civkaMaterial,0,0,'civka',0,3,-coilTurns);

----------------------------------------------------edge drawing
gr_name="okraj";
gr_num=300;
gr_BodyX = {0,0}
gr_BodyY= {-gr_oblastR, gr_oblastR} –area of simulation

mi_addnode(gr_BodyX[1],gr_BodyY[1])
mi_selectnode(gr_BodyX[1],gr_BodyY[1])
mi_setnodeprop(gr_name,gr_num)

mi_addnode(gr_BodyX[2],gr_BodyY[2])
mi_selectnode(gr_BodyX[2],gr_BodyY[2])
mi_setnodeprop(gr_name,gr_num)

mi_addarc(gr_BodyX[1],gr_BodyY[1],gr_BodyX[2],gr_BodyY[2],180,1)
mi_addarc(gr_BodyX[2],gr_BodyY[2],gr_BodyX[1],gr_BodyY[1],180,1)

mi_selectarcsegment(gr_BodyY[1],gr_BodyX[1])
mi_setarcsegmentprop(1,gr_name,0,gr_num)
mi_selectarcsegment(gr_BodyY[2],gr_BodyX[2])
mi_setarcsegmentprop(1,gr_name,0,gr_num)

--material
mi_addblocklabel(10,gr_oblastR*0.95)
mi_selectlabel(10,gr_oblastR*0.95)
mi_setblockprop('Air',0,0,'<None>',0,0,0);

--boundary condition
bo_name = "okraj";
bo_c0 = 0;
bo_c1 = 0;
bo_BdryFormat = 1; --0=fixed voltage, 1=mixed, 2=surface charge density, 3=priodic, 4=anti-periodic
mi_addboundprop(bo_name, 0, 0, 0, 0, 0, 0, bo_c0, bo_c1, BdryFormat)

mi_selectarcsegment(gr_BodyX[1]+0.1,gr_BodyY[1])
mi_setarcsegmentprop(1,bo_name,0,gr_num)
mi_selectarcsegment(gr_BodyX[1]-0.1,gr_BodyY[1])
mi_setarcsegmentprop(1,bo_name,0,gr_num)

-----------------------------------------------space between sample and ferite core
--material up
```

```
mi_addblocklabel(0,20)
mi_selectlabel(0,20)
mi_setblockprop('Air',0,0,'<None>',0,0,0);

--material bottom
mi_addblocklabel(0,-20)
mi_selectlabel(0,-20)
mi_setblockprop('Air',0,0,'<None>',0,0,0);

----------------------------------------------------line for mesurig in postprocessor
gr_BodyX = {0,0}
gr_BodyY= {gr_vzorek_a/2, -gr_vzorek_a/2} --velikost oblasti simulovani

mi_addnode(gr_BodyX[1],gr_BodyY[1])
mi_selectnode(gr_BodyX[1],gr_BodyY[1])
mi_setnodeprop(gr_name,gr_num)

mi_addnode(gr_BodyX[2],gr_BodyY[2])
mi_selectnode(gr_BodyX[2],gr_BodyY[2])
mi_setnodeprop(gr_name,gr_num)

mi_addsegment(gr_BodyX[1],gr_BodyY[1],gr_BodyX[2],gr_BodyY[2])
mi_selectsegment(gr_BodyX[1],(gr_BodyY[1]+gr_BodyY[2])/2)
mi_setsegmentprop('mereni',1,0,0,500)

------------------------------program
showconsole() --show console
--clearconsole() --clear console

print("nastaveni prvotni vzdalenosti")
print(gr_magMaterial)
--print into the console

mi_saveas("temp.FEM") –it will operate over sopy of original file
a = {} --create a tabole with value
k = "x" –put value x
a[k] = 0 –set the value into 0

        mi_analyze()
        mi_loadsolution() --show results
        mo_showdensityplot(1,0,2,0,"bmag") –show colored mag. field


        mo_seteditmode("contour") –switch to editation of groups
  mo_clearcontour()
  mo_selectpoint(0, gr_vzorek_a/2)
  mo_selectpoint(0, -gr_vzorek_a/2)
  mo_makeplot(1,20,"model.txt",1)
```