Simulace a diagnostika plazmatu

Electrostatic field simulation in arbitrary geometries and media

Úkoly

1. Install the necessary software
2. Generate the 2D geometry specified by the teacher
3. Configure a simulation using the Elmer FEM simulation package on this geometry
4. View and process the data in ParaView software

Introduction

In this task, you will learn to carry out a 2D electrostatics simulation in a geometry of an arbitrary shape. Knowing the electric field is absolutely crucial in Plasma Physics, though it often requires additional equations to be solved to get the full picture. You will be simulating the electric field in Elmer Finite Element Method (FEM) software which contains a number of physics interfaces. Therefore, you can expand on the knowledge you have obtained in this task and learn to use different physics interfaces in Elmer as well (e.g. magnetics, heat transfer).

Tools installation and running

SALOME Platform

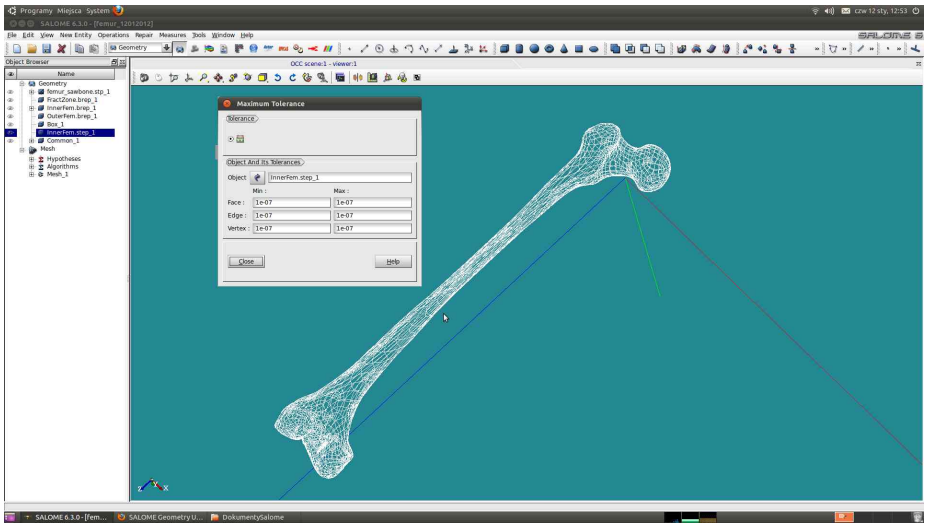
The SALOME Platform is a very powerful tool for generation of 1D, 2D and 3D geometries. It integrates an engine for geometry generation and, in addition, the Netgen meshing algorithm for generation of tetrahedral computational grids. SALOME is relatively straightforward to obtain. Please download a binary for your operating system on this webpage: `http://www.salome-platform.org/`.

Salome can be run in several ways - you can create your own scripts for it, it can run as a server with several clients or it can run as a standalone application. Here, we are only going to use the latter option. To run SALOME, simply use the desktop icon after you have installed it.

Elmer FEM library

Elmer software is a universal tool for numerical simulation using the Finite Element Method. It includes a number of physics interfaces, for example:

- Electrostatics
- Magnetodynamics
- Heat transfer
- Navier-Stokes equations
- etc...



Obrázek 1: *SALOME screenshot*

Obtaining Elmer is also quite straightforward - go to the downloads page at `https://www.csc.fi/web/elmer/` and either download the binaries (for Windows) or follow the instructions for adding the repository in Ubuntu and Debian-based system. If you are using a different operating system, ask your instructor for advice.

Elmer is typically run from the command line. On Windows, please use the "cmd"program (you can search for it in the Start menu). In Linux systems, use the terminal emulator that you prefer.

There are **two important executables that you will need to use**. The first one is called ElmerGrid and it is necessary for converting the computational mesh from SALOME format to Elmer-compatible format. To do so, run the ElmerGrid command with the following parameters:

```
ElmerGrid 8 2 name_of_mesh_file_from_SALOME.unv -autoclean
```

The other important executable is ElmerSolver. You run in by simply typing

```
ElmerSolver name_of_your_case_file.sif
```

Later on, we discuss how to compile the so-called "case file".

ParaView software

ParaView is a very powerful postprocesing tool capable of working with 2D and 3D data. It uses conventional .vtk or .vtu format for the files, which is the output of many open-source physics solvers. Binaries are, again, available at the homepage of the software `https://www.paraview.org/download/`.

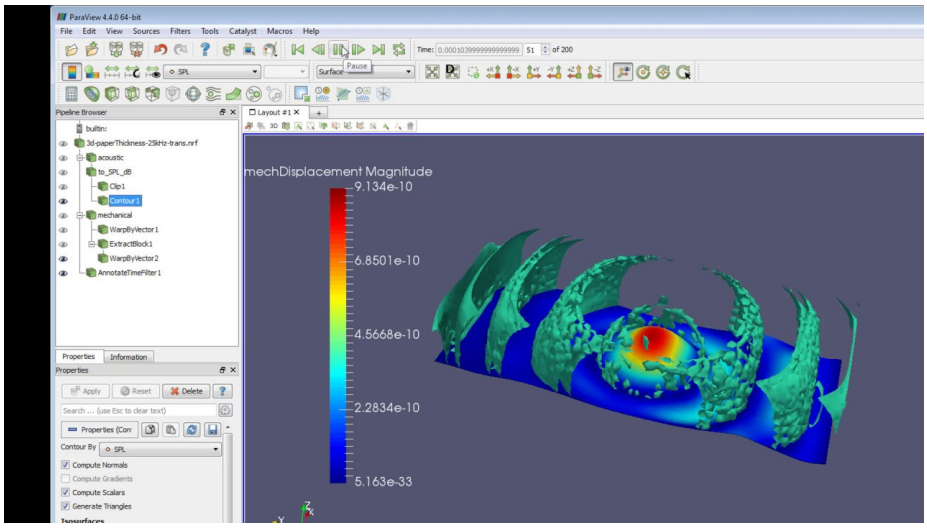
Electrostatic Simulation 1: Hyperbolic electrodes

In the first simulation, we will investigate the case of two opposing hyperbolic electrodes. This example has an analytical solution which we will compare with the numerical solution.

The configuration of this case is very simple because there are only two boundary conditions and one volume, in which we are solving the partial differential equation.

Geometry preparation

The geometry of the first problem is very simple - we will only simulate the electric field in vacuum in between two hyperboloid electrodes in 2D axially symmetrical geometry. The investigated geometrical configuration can be found in the attached publication [Celestin, S. et al., J. Phys. D: Appl. Phys. **42** (2009) 065203]



Obrázek 2: *ParaView screenshot*

There is a specified sequence, that you will need to follow. Generally, it can be summarized into the following steps:

1. Create the geometrical shapes corresponding to the domains in your simulation
2. Assemble all the geometrical shapes (domains) into a so-called **Partition**
3. In the Partition, you need to create **face groups** and **edge groups**. On the domain groups, you will later specify the material and the equations to be solved, on the edge groups, you will specify the boundary conditions.

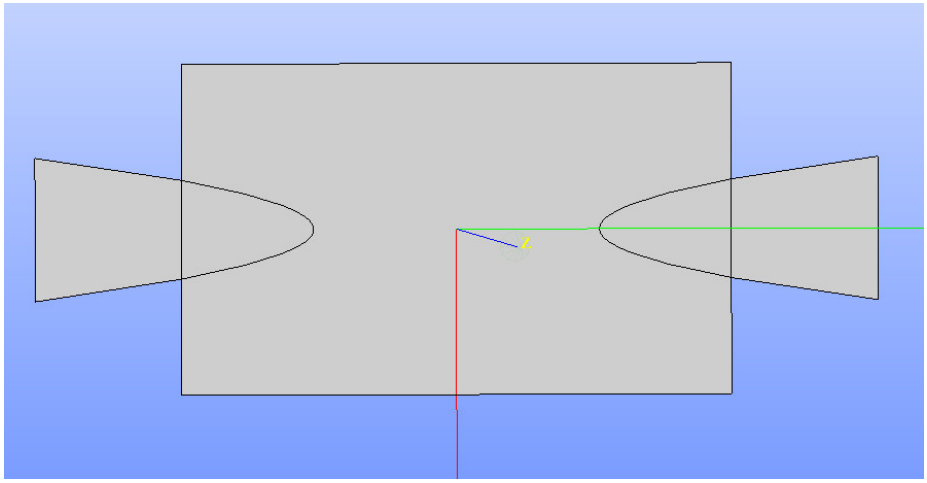
To create the geometry with the hyperbolic electrodes, you first need to create the volume enclosing these electrodes. We will use a Rectangle and create it by clicking **New Entity** > **Primitives** > **Rectangle**. A dialog will pop up, asking you to specify the basic properties of the rectangle.

Then, we need to create our hyperbolas. We will create the hyperbolic faces from hyperbola curves. First, to create the hyperbolic curve, select **New Entity** > **Basic** > **Curve**. Switch the **Creation Mode** to **Analytical** and use the fields provided to write parametric equations $x = x(t)$, $y = y(t)$ for the hyperbola, where t is your parameter. Once we draw the hyperbola, we need to convert it to a face, so that we can subtract it from our background rectangle. To do so, create a line connecting the opposing ends of the hyperbola by clicking **New Entity** > **Basic** > **Line**. Once you have the hyperbola and the line connecting its ends, you need to create a closed contour of these shapes. Select the line you created and the section of the hyperbola and do **Operations** > **Boolean** > **Fuse**. Finally, once you have the closed contour, you can select it and by clicking **New Entity** > **Build** > **Face**, you can convert it to a planar face.

Congratulations, you have now created a hyperbolic face. Using **Opertations** > **Transformation** > **Translation**, you can move the hyperbolic face wherever you want to move it and using **Opertations** > **Transformation** > **Mirror**, you can create its opposing copy.

Now we have all the three ingredients for our simulation and need to create a Partition out of them. Partition is an assembly of objects, on which we will then compute the mesh. It is not a simple Boolean Union of objects because it maintains the information about the objects, from which it was created. Your partition should look something like the figure below:

Then, we need to create face groups on the partition. Right-click the partition and select **Create Group**. To create a face group in the partition, check the little square on the top and select the parts of the volume that you want to include in the partition. Basically, you should end up with three face groups: **Electrode1, Electrode2, Air**.



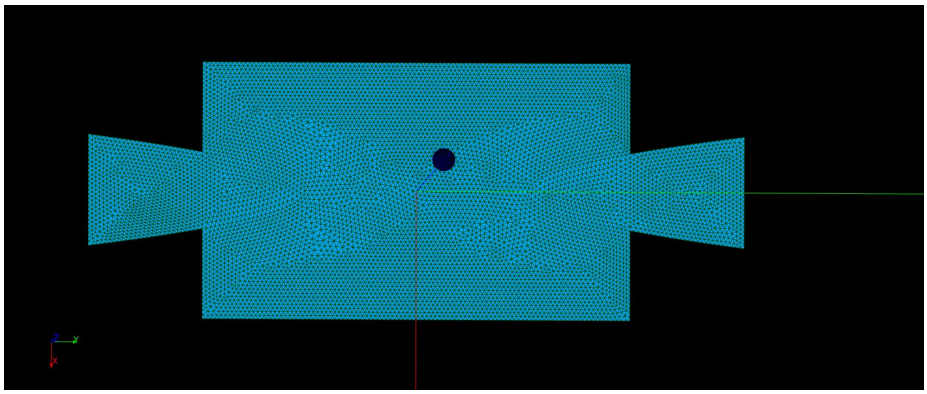
Obrázek 3: *What your partition should look like*

Having created the face groups, we need to create edge groups for boundary conditions. Use the same right-click+ **Create Group** trick but this time, select the little line icon at the top. You should have two boundary conditions in the end: Electrode1terminal and Electrode2terminal.

Mesh computation and export

Once you have created the partition with the correct volume and boundary conditions, we need to create the mesh. To do so, use the drop-down list at the top to switch from **Geometry** to **Mesh**. Select the partition in the tree on the right and click **Mesh** > **Create Mesh**. In the window that has appeared, select the **NETGEN 1D-2D** algorithm and select **Netgen 2D parameters** under the gear icon on the right. You can select the minimum and maximum size. After you are done, click **Mesh** > **Compute** to compute the mesh.

Before exporting the mesh, there is one last thing we need to do - copy the groups from geometry to the mesh. To do so, right-click the mesh and select **Create Groups from Geometry**.



Obrázek 4: *What your mesh should look like*

With this, your mesh is complete. To export it, right-click it and select **Export** > **UNV file**.

Preparing the case-file

The equation that we are solving is the Poisson equation in the usual form:

$$\nabla \cdot \varepsilon \nabla \phi = -\rho \tag{1}$$

where ϕ is the potential of the electric field, ε the permittivity and ρ the charge density.

To set up the solution of this equation, we need to create the so-called Elmer case file. The case file is a usual text file which contains several important control segments:

Simulation The simulation section gives you the fundamental information about the simulation - especially whether the simulation is time-dependent or steady-state, where to store the results, etc...

Solver i For each physics solver you define the Solver section, where you specify the settings of each of the physics solvers. Most importantly, you need to specify the solver for the linear system of equations and the numbers of linear and non-linear iterations.

Body i for each of the domains in your geometry, in which you want to solve the differential equation, you have to define the body segment. The body segment clearly identifies, which equations need to be solved and links the body to its material properties. Additionally, it can contain various "generalized body forces"(e.g. charge density in the case of electrostatics)

Equation i Each body links to an equation segment, in which we specify which equations need to be solved within this body

Material i In this section, we specify the material properties keywords.

BC i BC = Boundary Condition. Here, we specify the values of the field at the boundary (Dirichlet boundary condition) or values of the variables' derivatives at the boundary (Neumann boundary conditions)

We would like you to create the case file by analogy. Attached to this study is a simple electrostatics case of **potential distribution in a cartesian geometry**. By studying the case file, try to impose $V = 1000\,\mathrm{V}$ on Electrode1terminal and $V = 0\,\mathrm{V}$ on Electrode2terminal. Then run the case using the `ElmerSolver` command.

Postprocessing in ParaView

Elmer has created .vtu files in the folder which has the same name as your mesh. Try opening these files in ParaView and visualizing the potential using these files.