

Coil Modeling Tutorials

Low Frequency Electromagnetics

Jonathan Velasco

September 23, 2021

Coil Modeling Tutorials

About this document

This document provides examples on the modeling of coils (eg., massive, stranded, foil) using the magnetodynamics $\mathbf{a} - v$ formulation. The tutorials use a preprocessing tool called the *Circuit Builder*. The Circuit Builder is a Python library that enables the creation of passive-element and ideal-source circuits, to perform circuit-field simulations. The Circuit Builder is not a simulation tool, but rather a preprocessing tool to build circuit network matrices used in alongside the CircuitsAndDynamics and CircuitsAndDynamicsHarmonic modules to model coils.

The present manual corresponds to Elmer software version 9.0.

Latest documentations and program versions of Elmer are available (or links are provided) at <http://www.csc.fi/elmer>.

Copyright information

This document is licensed under the Creative Commons Attribution-NonCommerical 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/>.

Initially this Template For Tutorials has been written by Rich Bayless. External contributions to the Template are welcome.

Circuit Builder

Description

Elmer circuit builder main:

Description: This is a tool to write circuits in Elmer format using pin-connection convention.

Instructions:

Contents

Table of Contents	3
1 Current source connected to a 2D massive conductor	4
2 Voltage source connected to a 2D massive conductor	10
3 Voltage divider connected to a 2D stranded conductor	16
4 Current divider connected to a 2D stranded conductor	22
5 Three-dimensional closed-coil connected to sinusoidal current TEAM Workshop Problem 7	28

Tutorial 1

Current source connected to a 2D massive conductor

Directory: current_source

Transient Solvers: CircuitsAndDynamics, MagnetoDynamics2D

Harmonic Solvers: CircuitsAndDynamicsHarmonic, MagnetoDynamics2DHarmonic

Tools: Circuit Builder, ElmerGUI

Dimensions: 2D

Simulation Type: Transient, and Harmonic

Author: Jonathan Velasco

Introduction

This tutorial addresses the implementation of a **circuit-field** coupled model, in which an **ideal current source** excites a solid conductor (i.e., massive conductor).

The circuit equations are setup using the **circuit builder** library and solved using Maxwell's magnetodynamics $\mathbf{a} - v$ formulation with **ElmerSolver**.

The model is setup as a transient simulation (CircuitsAndDynamics and MagnetoDynamics2D) and in frequency domain (CircuitsAndDynamicsHarmonic and CircuitsAndDynamicsHarmonic), pointing out the differences when pre-processing Elmer's input file (.sif).

Let the conducting region

$$\Omega_c = \bigcup_{i,ckt} \Omega_{c_i}^{ckt}$$

be the union of the i conductors associated to circuit ckt , and Ω_c^C be the complementary non-conduction regions. The computation domain is thus $\Omega = \Omega_c^C \cup \Omega_c$. The conductors Ω_{c_i} are supplied with a DC current source $I_{s_i} = 1A$.

In this tutorial the field problem is solved in current-driven solid (massive) round conductor surrounded by air as shown in Fig. 1.1.

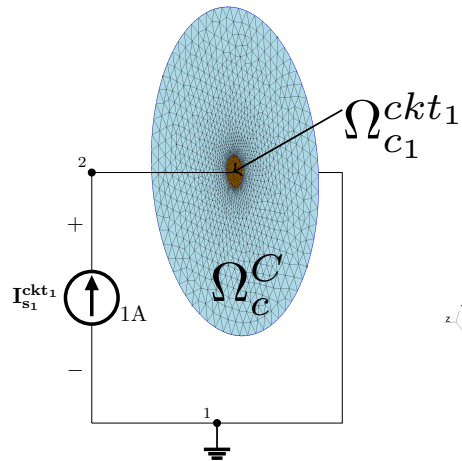


Figure 1.1: Single DC current source connected to FEM Round Wire

To find model in our Github repository click on [Current Source](#)

Tutorial Goals

After completing this tutorial the user will be able to:

- Setup an ideal current source using the Circuit Builder, and understand the methodology behind setting up circuit equations.
- Understand what a Massive coil is.
- Run 2D coil models in Transient and Frequency Domain.
- Post-process local and global quantities

Circuit Builder: Ideal current source connected to 2D massive conductor

The circuit schematics is shown in figure1.2.

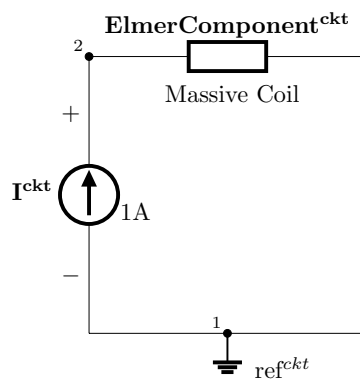


Figure 1.2: DC current source connected to FEM Component Schematics

To define the source circuit we use the **Circuit Builder**, which follows the nodal circuit analysis approach to build the circuit equations of the type

$$Bx = f \quad (1.1)$$

where, B is the stiffness matrix of the system containing the Kirchhoff laws and component voltage and current information, and f is the source term.

Following the schematics in 1.2 we apply the following steps in the Circuit Builder

- Enter file output name with .definition extension. This file includes all the circuit information needed by Elmer's input file (.sif)

```
# name output file
output_file = "transient_current_source.definition"
```

- Enter number of circuits in definition

```
# initialize circuits: number of circuits - do not remove
c = number_of_circuits(1)
```

- Enter the reference node number belonging to the given circuit

```
# reference/ground node needed - do not remove.
c[1].ref_node = 1
```

- Setup circuit components and populate given circuit with them

```
# Components
Is = I("Is", 1, 2, 1)
FEM_Wire_1 = ElmerComponent("FEM_Wire_1", 2, 1, 0, 1, [1], "Massive")

# store components in array components = [comp1, comp2,...] - do not remove
c[1].components.append([Is, FEM_Wire_1])
```

```
# generate elmer circuit.definitions - do not remove / do not edit
generate_elmer_circuits(c, output_file)
```

Pre-processing with ElmerGUI

```
File
Definitions
Append -> zzz.xml
Close
```

The zzz.xml definition file is, by default, located in Linux in:

\$ELMER_HOME/share/ElmerGUI/edf-extra

and in Windows, located in:

C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf-extra

We will also be using the zzz Solver, which is one of the default, pre-loaded GUI definitions, so it does not need to be manually activated. For reference, the zzz.xml definition file is, by default, located in Linux in:

\$ELMER_HOME/share/ElmerGUI/edf
and in Windows, located in:

C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf

Solution procedure

The mesh is given in ElmerGrid format in file zzz, load this file.

File

Open -> zzz

After we have the mesh we start to go through the Model menu from the top to bottom. In the Setup we choose things related to the whole simulation such as file names, time stepping, constants etc.

The simulation is carried out in 2-dimensional Cartesian coordinates.

zzz 2nd order bdf time stepping method is selected with zzz steps and with step size of zzz seconds.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Setup

Simulation Type = Transient

Steady state max. iter = 20

Time Step Intervals = 200

Gravity = 0 -1 0 9.82

Apply

In the equation section we choose the relevant equations and parameters related to their solution.

zzz In this case we'll have one set of equations (named "Natural Convection") which consists of the heat equation and of the Navier-Stokes equation.

When defining Equations and Materials it is possible to assign to the bodies immediately, or to use mouse selection to assign them later. In this case we have just one body and therefore its easier to assign the Equation and Material to it directly. It is important to select the convection to be computed since that couples the velocity field to the heat equation.

zzz The system may include non-linear iterations of each equation and steady state iterations to obtain convergence of the coupled system. It is often a good idea to keep the number of non-linear iterations in a coupled case low. Here we select just one non-linear iteration for both equations.

For the linear system solvers we are happy to use the defaults. One may however, try out different preconditioners (ILU1, ...) or direct Umfpack solver, for example.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Equation

Name = Natural Convection

Apply to Bodies = 1

Heat Equation

Active = on

Add

OK

The Material section includes all the material parameters. They are divided into generic parameters which are direct properties of the material without making any assumptions on the physical model, such as the mass. Other properties assume a physical law, such as conductivities and viscosity.

zzz Discuss which materials are used, why they are used, and any interesting details.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Material
  Apply to Bodies = 1
  Material library
    Water (room temperature)
  General
    Reference Temperature = 293
  Add
  OK
```

A Body Force represents the right-hand-side of a equation. It is generally not a required field for a body.
 zzz In this case, however, we apply the buoyancy resulting from heat expansion as a body force to the Navier-Stokes equation.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Body Force
  Name = Buoyancy
  Apply to Bodies = 1
  Navier-Stokes
    Boussinesq = on
  Add
  OK
```

Initial conditions should be given to transient cases, and probably are not needed for steady state solutions.

zzz In this case we choose a constant Temperature field and an small initial velocity that initializes the symmetry break.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Initial Condition
  Name = Initial Guess
  Heat Equation
    Temperature = 293
  Navier-Stokes
    Velocity 1 = 1.0e-9
    Velocity 2 = 0.0
```

Only one boundary condition may be applied to each boundary and therefore all the different physical BCs for a boundary should be grouped together.

zzz In this case the Temperature and Velocity. The side walls are assumed to be adiabatic.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
BoundaryCondition
  Name = Bottom
  Heat Equation
    Temperature = 293.5
  Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
  Add
```

```

New

Name = Top
Heat Equation
    Temperature = 293
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
OK

```

The conditions may also be assigned to boundaries in the Boundary condition menu, or by clicking on each boundary with the mouse. Here we use the latter approach as that spares us of the need to know the indexes of each boundary.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

```

Model
Set boundary properties
    Choose Bottom -> set boundary condition Bottom
    Choose Top -> set boundary condition Top
    Choose Sides -> set boundary condition Sides
OK

```

For the execution ElmerSolver needs the mesh files and the command file. We have now basically defined all the information for ElmerGUI to write the command file. After writing it we may also visually inspect the command file.

```

Sif
Generate
Edit -> look how your command file came out

```

Before we can execute the solver we should save the files in a directory. The ElmerGUI project includes all the files needed to restart the case.

```

File
Save Project

```

After we have successfully saved the files we may start the solver.

```

Run
Start solver

```

A convergence view automatically pops up showing relative changes of each iteration.

When there are some results to view we may start the postprocessor also.

```

Run
Start ParaView

```

Results

Due to the number of the time steps the simulation may take around zzz minutes.

You may inspect the results with Paraview or with ElmerVTK.

zzz Extra task:

zzz If you have time you may try to solve the case with different parameters.

Tutorial 2

Voltage source connected to a 2D massive conductor

Directory: voltage_source

Transient Solvers: CircuitsAndDynamics, MagnetoDynamics2D

Harmonic Solvers: CircuitsAndDynamicsHarmonic, MagnetoDynamics2DHarmonic

Tools: Circuit Builder, ElmerGUI

Dimensions: 2D, Transient, and Harmonic

Author: Jonathan Velasco

Introduction

zzz This Template For Tutorials is designed to help make it easy to create a new tutorial using ElmerGUI. Having an outline of a tutorial should allow for simpler development of new tutorials, and will help keep the look and feel of new tutorials in unison with existing tutorials.

zzz Anywhere you see ‘zzz’, it means that section needs to be replaced. Any text that begins with ‘zzz’, should be deleted and replaced by the appropriate text for your tutorial.

zzz Think of ‘zzz’ as a spot where you should ‘fill in the blanks’.

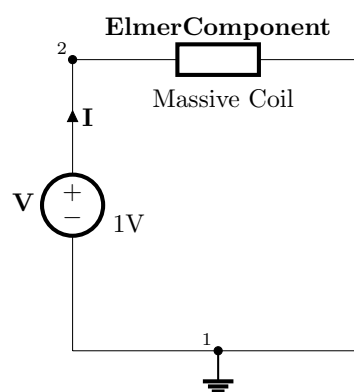


Figure 2.1: DC voltage source connected to FEM Component Schematics

Case definition

zzz Enter your case definition here.

The geometry for this tutorial looks like as shown in figure 5.2.

zzz Enter your geometry figure here.

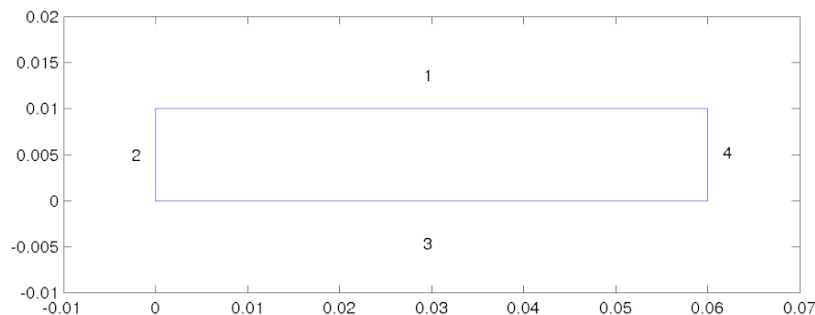


Figure 2.2: Geometry

ElmerGUI Equation Menu

zzz Note that we include both edf and edf-extra examples. If you don't need the edf-extra example, the first section can be deleted. It is suggested to keep the edf example, to show where the edf file comes from.

zzz The GUI definitions of zzz solver utilized in this tutorial are located in the `edf-extra` folder and thus need to be manually activated in the ElmerGUI.

zzz Note that the extra definition needs to be loaded first, before any other steps in building any tutorial. Once the extra definition has been loaded, and the ElmerGUI project has been saved, then the reference to the extra definition will be stored in the project, and you won't have to load the extra definition again.

```
File
  Definitions
    Append -> zzz.xml
  Close
```

The `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf-extra
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf-extra
```

We will also be using the zzz Solver, which is one of the default, pre-loaded GUI definitions, so it does not need to be manually activated. For reference, the `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf
```

Solution procedure

The mesh is given in ElmerGrid format in file `zzz`, load this file.

```
File
  Open -> zzz
```

You should obtain your mesh and may check `Model Summary...` that it consists of `zzz` elements. Your mesh should look like as shown in figure 5.3

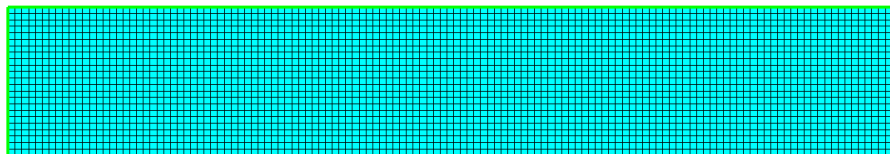


Figure 2.3: Mesh

After we have the mesh we start to go through the Model menu from the top to bottom. In the Setup we choose things related to the whole simulation such as file names, time stepping, constants etc.

The simulation is carried out in 2-dimensional Cartesian coordinates.

`zzz` 2nd order bdf time stepping method is selected with `zzz` steps and with step size of `zzz` seconds.

`zzz` Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Setup

Simulation Type = Transient

Steady state max. iter = 20

Time Step Intervals = 200

Gravity = 0 -1 0 9.82

Apply

In the equation section we choose the relevant equations and parameters related to their solution.

`zzz` In this case we'll have one set of equations (named "Natural Convection") which consists of the heat equation and of the Navier-Stokes equation.

When defining Equations and Materials it is possible to assign to the bodies immediately, or to use mouse selection to assign them later. In this case we have just one body and therefore its easier to assign the Equation and Material to it directly. It is important to select the convection to be computed since that couples the velocity field to the heat equation.

`zzz` The system may include non-linear iterations of each equation and steady state iterations to obtain convergence of the coupled system. It is often a good idea to keep the number of non-linear iterations in a coupled case low. Here we select just one non-linear iteration for both equations.

For the linear system solvers we are happy to use the defaults. One may however, try out different preconditioners (ILU1,...) or direct Umfpack solver, for example.

`zzz` Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Equation

Name = Natural Convection

Apply to Bodies = 1

Heat Equation

Active = on

Add

OK

The Material section includes all the material parameters. They are divided into generic parameters which are direct properties of the material without making any assumptions on the physical model, such as the mass. Other properties assume a physical law, such as conductivities and viscosity.

`zzz` Discuss which materials are used, why they are used, and any interesting details.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Material
  Apply to Bodies = 1
  Material library
    Water (room temperature)
  General
    Reference Temperature = 293
  Add
  OK
```

A Body Force represents the right-hand-side of a equation. It is generally not a required field for a body.

zzz In this case, however, we apply the buoyancy resulting from heat expansion as a body force to the Navier-Stokes equation.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Body Force
  Name = Buoyancy
  Apply to Bodies = 1
  Navier-Stokes
    Boussinesq = on
  Add
  OK
```

Initial conditions should be given to transient cases, and probably are not needed for steady state solutions.

zzz In this case we choose a constant Temperature field and an small initial velocity that initializes the symmetry break.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Initial Condition
  Name = Initial Guess
  Heat Equation
    Temperature = 293
  Navier-Stokes
    Velocity 1 = 1.0e-9
    Velocity 2 = 0.0
```

Only one boundary condition may be applied to each boundary and therefore all the different physical BCs for a boundary should be grouped together.

zzz In this case the Temperature and Velocity. The side walls are assumed to be adiabatic.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
BoundaryCondition
  Name = Bottom
  Heat Equation
    Temperature = 293.5
  Navier-Stokes
```

```

    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
New

Name = Top
Heat Equation
    Temperature = 293
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
OK

```

The conditions may also be assigned to boundaries in the Boundary condition menu, or by clicking on each boundary with the mouse. Here we use the latter approach as that spares us of the need to know the indexes of each boundary.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

```

Model
Set boundary properties
    Choose Bottom -> set boundary condition Bottom
    Choose Top -> set boundary condition Top
    Choose Sides -> set boundary condition Sides
OK

```

For the execution ElmerSolver needs the mesh files and the command file. We have now basically defined all the information for ElmerGUI to write the command file. After writing it we may also visually inspect the command file.

```

Sif
Generate
Edit -> look how your command file came out

```

Before we can execute the solver we should save the files in a directory. The ElmerGUI project includes all the files needed to restart the case.

```

File
Save Project

```

After we have successfully saved the files we may start the solver.

```

Run
Start solver

```

A convergence view automatically pops up showing relative changes of each iteration. When there are some results to view we may start the postprocessor also.

```

Run
Start ParaView

```

Results

Due to the number of the time steps the simulation may take around zzz minutes.

You may inspect the results with Paraview or with ElmerVTK.

zzz In Figure 5.4 the obtained temperature distribution is presented.

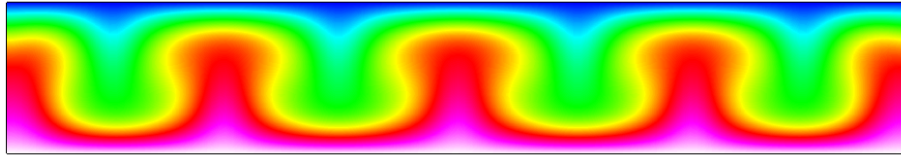


Figure 2.4: zzz Temperature distribution at 260 s.

zzz Extra task:

zzz If you have time you may try to solve the case with different parameters.

Tutorial 3

Voltage divider connected to a 2D stranded conductor

Directory: voltage_source

Transient Solvers: CircuitsAndDynamics, MagnetoDynamics2D

Harmonic Solvers: CircuitsAndDynamicsHarmonic, MagnetoDynamics2DHarmonic

Tools: Circuit Builder, ElmerGUI

Dimensions: 2D, Transient, and Harmonic

Author: Jonathan Velasco

Introduction

zzz This Template For Tutorials is designed to help make it easy to create a new tutorial using ElmerGUI. Having an outline of a tutorial should allow for simpler development of new tutorials, and will help keep the look and feel of new tutorials in unison with existing tutorials.

zzz Anywhere you see ‘zzz’, it means that section needs to be replaced. Any text that begins with ‘zzz’, should be deleted and replaced by the appropriate text for your tutorial.

zzz Think of ‘zzz’ as a spot where you should ‘fill in the blanks’.

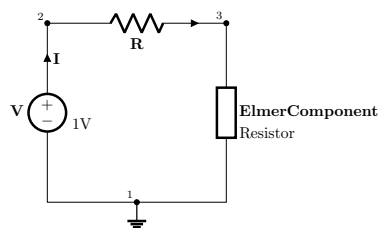


Figure 3.1: Voltage divider with a lumped resistor and FEM "resistor" component Schematics

Case definition

zzz Enter your case definition here.

The geometry for this tutorial looks like as shown in figure 5.2.

zzz Enter your geometry figure here.

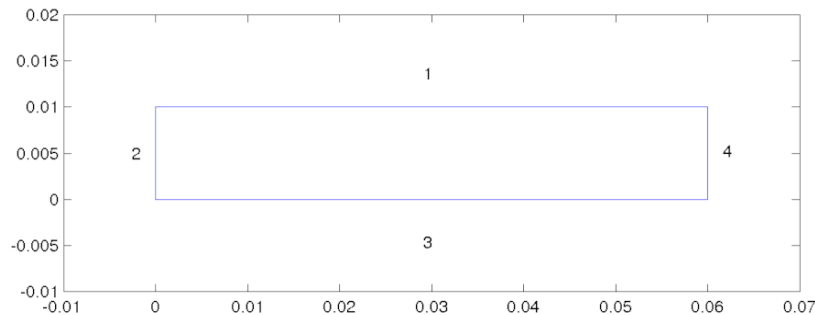


Figure 3.2: Geometry

ElmerGUI Equation Menu

zzz Note that we include both edf and edf-extra examples. If you don't need the edf-extra example, the first section can be deleted. It is suggested to keep the edf example, to show where the edf file comes from.

zzz The GUI definitions of zzz solver utilized in this tutorial are located in the `edf-extra` folder and thus need to be manually activated in the ElmerGUI.

zzz Note that the extra definition needs to be loaded first, before any other steps in building any tutorial. Once the extra definition has been loaded, and the ElmerGUI project has been saved, then the reference to the extra definition will be stored in the project, and you won't have to load the extra definition again.

```
File
  Definitions
    Append -> zzz.xml
  Close
```

The `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf-extra
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf-extra
```

We will also be using the zzz Solver, which is one of the default, pre-loaded GUI definitions, so it does not need to be manually activated. For reference, the `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf
```

Solution procedure

The mesh is given in ElmerGrid format in file `zzz`, load this file.

```
File
  Open -> zzz
```

You should obtain your mesh and may check `Model Summary...` that it consists of `zzz` elements. Your mesh should look like as shown in figure 5.3

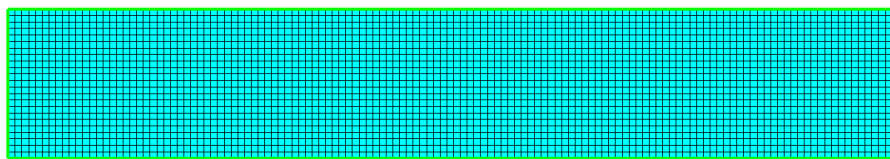


Figure 3.3: Mesh

After we have the mesh we start to go through the Model menu from the top to bottom. In the Setup we choose things related to the whole simulation such as file names, time stepping, constants etc.

The simulation is carried out in 2-dimensional Cartesian coordinates.

zzz 2nd order bdf time stepping method is selected with zzz steps and with step size of zzz seconds.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Setup

Simulation Type = Transient

Steady state max. iter = 20

Time Step Intervals = 200

Gravity = 0 -1 0 9.82

Apply

In the equation section we choose the relevant equations and parameters related to their solution.

zzz In this case we'll have one set of equations (named "Natural Convection") which consists of the heat equation and of the Navier-Stokes equation.

When defining Equations and Materials it is possible to assign to the bodies immediately, or to use mouse selection to assign them later. In this case we have just one body and therefore its easier to assign the Equation and Material to it directly. It is important to select the convection to be computed since that couples the velocity field to the heat equation.

zzz The system may include non-linear iterations of each equation and steady state iterations to obtain convergence of the coupled system. It is often a good idea to keep the number of non-linear iterations in a coupled case low. Here we select just one non-linear iteration for both equations.

For the linear system solvers we are happy to use the defaults. One may however, try out different preconditioners (ILU1,...) or direct Umfpack solver, for example.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Equation

Name = Natural Convection

Apply to Bodies = 1

Heat Equation

Active = on

Add

OK

The Material section includes all the material parameters. They are divided into generic parameters which are direct properties of the material without making any assumptions on the physical model, such as the mass. Other properties assume a physical law, such as conductivities and viscosity.

zzz Discuss which materials are used, why they are used, and any interesting details.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Material
  Apply to Bodies = 1
  Material library
    Water (room temperature)
  General
    Reference Temperature = 293
  Add
  OK
```

A Body Force represents the right-hand-side of a equation. It is generally not a required field for a body.

zzz In this case, however, we apply the buoyancy resulting from heat expansion as a body force to the Navier-Stokes equation.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Body Force
  Name = Buoyancy
  Apply to Bodies = 1
  Navier-Stokes
    Boussinesq = on
  Add
  OK
```

Initial conditions should be given to transient cases, and probably are not needed for steady state solutions.

zzz In this case we choose a constant Temperature field and an small initial velocity that initializes the symmetry break.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Initial Condition
  Name = Initial Guess
  Heat Equation
    Temperature = 293
  Navier-Stokes
    Velocity 1 = 1.0e-9
    Velocity 2 = 0.0
```

Only one boundary condition may be applied to each boundary and therefore all the different physical BCs for a boundary should be grouped together.

zzz In this case the Temperature and Velocity. The side walls are assumed to be adiabatic.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
BoundaryCondition
  Name = Bottom
  Heat Equation
    Temperature = 293.5
  Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
  Add
```

```

New

Name = Top
Heat Equation
    Temperature = 293
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
OK

```

The conditions may also be assigned to boundaries in the Boundary condition menu, or by clicking on each boundary with the mouse. Here we use the latter approach as that spares us of the need to know the indexes of each boundary.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

```

Model
Set boundary properties
    Choose Bottom -> set boundary condition Bottom
    Choose Top -> set boundary condition Top
    Choose Sides -> set boundary condition Sides
OK

```

For the execution ElmerSolver needs the mesh files and the command file. We have now basically defined all the information for ElmerGUI to write the command file. After writing it we may also visually inspect the command file.

```

Sif
Generate
Edit -> look how your command file came out

```

Before we can execute the solver we should save the files in a directory. The ElmerGUI project includes all the files needed to restart the case.

```

File
Save Project

```

After we have successfully saved the files we may start the solver.

```

Run
Start solver

```

A convergence view automatically pops up showing relative changes of each iteration. When there are some results to view we may start the postprocessor also.

```

Run
Start ParaView

```

Results

Due to the number of the time steps the simulation may take around zzz minutes.

You may inspect the results with Paraview or with ElmerVTK.

zzz In Figure 5.4 the obtained temperature distribution is presented.

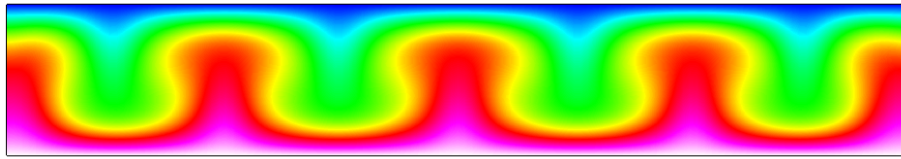


Figure 3.4: zzz Temperature distribution at 260 s.

zzz Extra task:

zzz If you have time you may try to solve the case with different parameters.

Tutorial 4

Current divider connected to a 2D stranded conductor

Directory: voltage_source

Transient Solvers: CircuitsAndDynamics, MagnetoDynamics2D

Harmonic Solvers: CircuitsAndDynamicsHarmonic, MagnetoDynamics2DHarmonic

Tools: Circuit Builder, ElmerGUI

Dimensions: 2D, Transient, and Harmonic

Author: Jonathan Velasco

Introduction

zzz This Template For Tutorials is designed to help make it easy to create a new tutorial using ElmerGUI. Having an outline of a tutorial should allow for simpler development of new tutorials, and will help keep the look and feel of new tutorials in unison with existing tutorials.

zzz Anywhere you see 'zzz', it means that section needs to be replaced. Any text that begins with 'zzz', should be deleted and replaced by the appropriate text for your tutorial.

zzz Think of 'zzz' as a spot where you should 'fill in the blanks'.

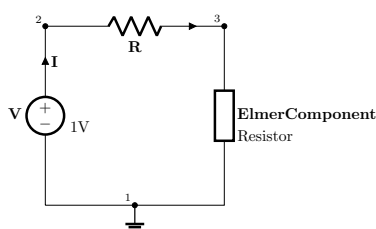


Figure 4.1: Current divider with a lumped resistor and FEM "resistor" component Schematics

Case definition

zzz Enter your case definition here.

The geometry for this tutorial looks like as shown in figure 5.2.

zzz Enter your geometry figure here.

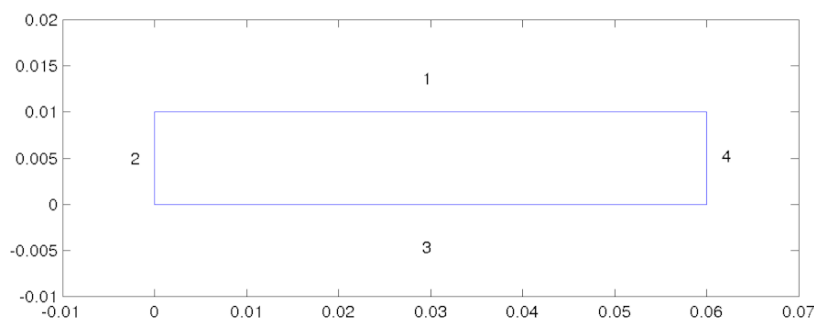


Figure 4.2: Geometry

ElmerGUI Equation Menu

zzz Note that we include both edf and edf-extra examples. If you don't need the edf-extra example, the first section can be deleted. It is suggested to keep the edf example, to show where the edf file comes from.

zzz The GUI definitions of zzz solver utilized in this tutorial are located in the `edf-extra` folder and thus need to be manually activated in the ElmerGUI.

zzz Note that the extra definition needs to be loaded first, before any other steps in building any tutorial. Once the extra definition has been loaded, and the ElmerGUI project has been saved, then the reference to the extra definition will be stored in the project, and you won't have to load the extra definition again.

File

Definitions

Append -> `zzz.xml`

Close

The `zzz.xml` definition file is, by default, located in Linux in:

`$ELMER_HOME/share/ElmerGUI/edf-extra`

and in Windows, located in:

`C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf-extra`

We will also be using the zzz Solver, which is one of the default, pre-loaded GUI definitions, so it does not need to be manually activated. For reference, the `zzz.xml` definition file is, by default, located in Linux in:

`$ELMER_HOME/share/ElmerGUI/edf`

and in Windows, located in:

`C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf`

Solution procedure

The mesh is given in ElmerGrid format in file `zzz`, load this file.

File

Open -> `zzz`

You should obtain your mesh and may check `Model Summary...` that it consists of `zzz` elements. Your mesh should look like as shown in figure 5.3

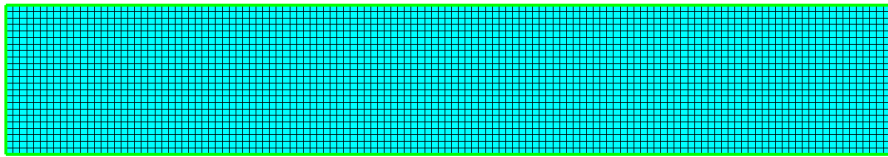


Figure 4.3: Mesh

After we have the mesh we start to go through the Model menu from the top to bottom. In the Setup we choose things related to the whole simulation such as file names, time stepping, constants etc.

The simulation is carried out in 2-dimensional Cartesian coordinates.

zzz 2nd order bdf time stepping method is selected with zzz steps and with step size of zzz seconds.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Setup

```
Simulation Type = Transient
Steady state max. iter = 20
Time Step Intervals = 200
Gravity = 0 -1 0 9.82
```

Apply

In the equation section we choose the relevant equations and parameters related to their solution.

zzz In this case we'll have one set of equations (named "Natural Convection") which consists of the heat equation and of the Navier-Stokes equation.

When defining Equations and Materials it is possible to assign to the bodies immediately, or to use mouse selection to assign them later. In this case we have just one body and therefore its easier to assign the Equation and Material to it directly. It is important to select the convection to be computed since that couples the velocity field to the heat equation.

zzz The system may include non-linear iterations of each equation and steady state iterations to obtain convergence of the coupled system. It is often a good idea to keep the number of non-linear iterations in a coupled case low. Here we select just one non-linear iteration for both equations.

For the linear system solvers we are happy to use the defaults. One may however, try out different preconditioners (ILU1,...) or direct Umfpack solver, for example.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

Equation

```
Name = Natural Convection
Apply to Bodies = 1
Heat Equation
Active = on
Add
OK
```

The Material section includes all the material parameters. They are divided into generic parameters which are direct properties of the material without making any assumptions on the physical model, such as the mass. Other properties assume a physical law, such as conductivities and viscosity.

zzz Discuss which materials are used, why they are used, and any interesting details.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Material
  Apply to Bodies = 1
  Material library
    Water (room temperature)
  General
    Reference Temperature = 293
  Add
  OK
```

A Body Force represents the right-hand-side of a equation. It is generally not a required field for a body.

zzz In this case, however, we apply the buoyancy resulting from heat expansion as a body force to the Navier-Stokes equation.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Body Force
  Name = Buoyancy
  Apply to Bodies = 1
  Navier-Stokes
    Boussinesq = on
  Add
  OK
```

Initial conditions should be given to transient cases, and probably are not needed for steady state solutions.

zzz In this case we choose a constant Temperature field and an small initial velocity that initializes the symmetry break.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Initial Condition
  Name = Initial Guess
  Heat Equation
    Temperature = 293
  Navier-Stokes
    Velocity 1 = 1.0e-9
    Velocity 2 = 0.0
```

Only one boundary condition may be applied to each boundary and therefore all the different physical BCs for a boundary should be grouped together.

zzz In this case the Temperature and Velocity. The side walls are assumed to be adiabatic.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
BoundaryCondition
  Name = Bottom
  Heat Equation
    Temperature = 293.5
  Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
  Add
```

```

New

Name = Top
Heat Equation
    Temperature = 293
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
OK

```

The conditions may also be assigned to boundaries in the Boundary condition menu, or by clicking on each boundary with the mouse. Here we use the latter approach as that spares us of the need to know the indexes of each boundary.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

```

Model
Set boundary properties
    Choose Bottom -> set boundary condition Bottom
    Choose Top -> set boundary condition Top
    Choose Sides -> set boundary condition Sides
OK

```

For the execution ElmerSolver needs the mesh files and the command file. We have now basically defined all the information for ElmerGUI to write the command file. After writing it we may also visually inspect the command file.

```

Sif
Generate
Edit -> look how your command file came out

```

Before we can execute the solver we should save the files in a directory. The ElmerGUI project includes all the files needed to restart the case.

```

File
Save Project

```

After we have successfully saved the files we may start the solver.

```

Run
Start solver

```

A convergence view automatically pops up showing relative changes of each iteration. When there are some results to view we may start the postprocessor also.

```

Run
Start ParaView

```

Results

Due to the number of the time steps the simulation may take around zzz minutes.

You may inspect the results with Paraview or with ElmerVTK.

zzz In Figure 5.4 the obtained temperature distribution is presented.

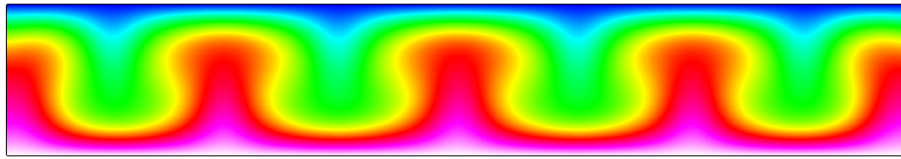


Figure 4.4: zzz Temperature distribution at 260 s.

zzz Extra task:

zzz If you have time you may try to solve the case with different parameters.

Tutorial 5

Three-dimensional closed-coil connected to sinusoidal current TEAM Workshop Problem 7

Directory: current_source

Transient Solvers: CircuitsAndDynamics, MagnetoDynamics2D

Harmonic Solvers: CircuitsAndDynamicsHarmonic, MagnetoDynamics2DHarmonic

Tools: Circuit Builder, ElmerGUI

Dimensions: 2D

Simulation Type: Transient, and Harmonic

Author: Jonathan Velasco

Introduction

zzz This Template For Tutorials is designed to help make it easy to create a new tutorial using ElmerGUI. Having an outline of a tutorial should allow for simpler development of new tutorials, and will help keep the look and feel of new tutorials in unison with existing tutorials.

zzz Anywhere you see 'zzz', it means that section needs to be replaced. Any text that begins with 'zzz', should be deleted and replaced by the appropriate text for your tutorial.

zzz Think of 'zzz' as a spot where you should 'fill in the blanks'.

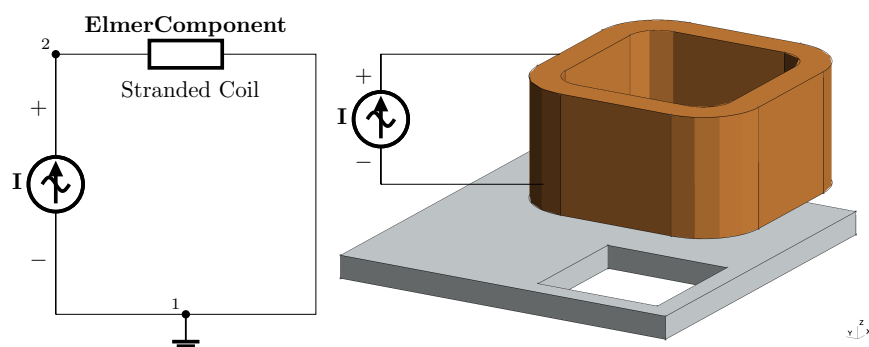


Figure 5.1: TEAM 7 with sinusoidal current source

Case definition

zzz Enter your case definition here.

The geometry for this tutorial looks like as shown in figure 5.2.

zzz Enter your geometry figure here.

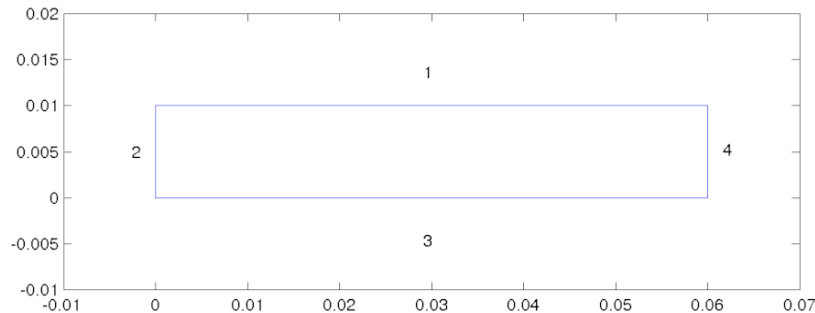


Figure 5.2: Geometry

ElmerGUI Equation Menu

zzz Note that we include both edf and edf-extra examples. If you don't need the edf-extra example, the first section can be deleted. It is suggested to keep the edf example, to show where the edf file comes from.

zzz The GUI definitions of zzz solver utilized in this tutorial are located in the `edf-extra` folder and thus need to be manually activated in the ElmerGUI.

zzz Note that the extra definition needs to be loaded first, before any other steps in building any tutorial. Once the extra definition has been loaded, and the ElmerGUI project has been saved, then the reference to the extra definition will be stored in the project, and you won't have to load the extra definition again.

```
File
  Definitions
    Append -> zzz.xml
  Close
```

The `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf-extra
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf-extra
```

We will also be using the zzz Solver, which is one of the default, pre-loaded GUI definitions, so it does not need to be manually activated. For reference, the `zzz.xml` definition file is, by default, located in Linux in:

```
$ELMER_HOME/share/ElmerGUI/edf
```

and in Windows, located in:

```
C:/Program Files/Elmer 9.0-Release/share/ElmerGUI/edf
```

Solution procedure

The mesh is given in ElmerGrid format in file `zzz`, load this file.

File

Open -> zzz

You should obtain your mesh and may check `Model Summary...` that it consists of zzz elements. Your mesh should look like as shown in figure 5.3

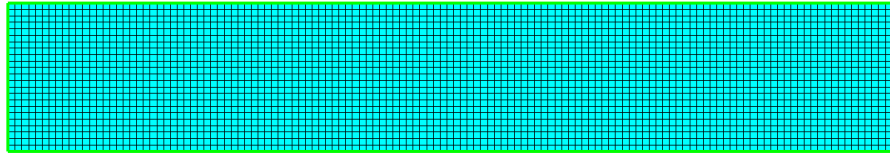


Figure 5.3: Mesh

After we have the mesh we start to go through the `Model` menu from the top to bottom. In the `Setup` we choose things related to the whole simulation such as file names, time stepping, constants etc.

The simulation is carried out in 2-dimensional Cartesian coordinates.

zzz 2nd order bdf time stepping method is selected with zzz steps and with step size of zzz seconds.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

`Model`

`Setup`

`Simulation Type = Transient`

`Steady state max. iter = 20`

`Time Step Intervals = 200`

`Gravity = 0 -1 0 9.82`

`Apply`

In the equation section we choose the relevant equations and parameters related to their solution.

zzz In this case we'll have one set of equations (named "Natural Convection") which consists of the heat equation and of the Navier-Stokes equation.

When defining Equations and Materials it is possible to assign to the bodies immediately, or to use mouse selection to assign them later. In this case we have just one body and therefore its easier to assign the Equation and Material to it directly. It is important to select the convection to be computed since that couples the velocity field to the heat equation.

zzz The system may include non-linear iterations of each equation and steady state iterations to obtain convergence of the coupled system. It is often a good idea to keep the number of non-linear iterations in a coupled case low. Here we select just one non-linear iteration for both equations.

For the linear system solvers we are happy to use the defaults. One may however, try out different preconditioners (ILU1,...) or direct Umfpack solver, for example.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

`Model`

`Equation`

`Name = Natural Convection`

`Apply to Bodies = 1`

`Heat Equation`

`Active = on`

`Add`

`OK`

The Material section includes all the material parameters. They are divided into generic parameters which are direct properties of the material without making any assumptions on the physical model, such as the mass. Other properties assume a physical law, such as conductivities and viscosity.

zzz Discuss which materials are used, why they are used, and any interesting details.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Material
  Apply to Bodies = 1
  Material library
    Water (room temperature)
  General
    Reference Temperature = 293
  Add
  OK
```

A Body Force represents the right-hand-side of a equation. It is generally not a required field for a body.

zzz In this case, however, we apply the buoyancy resulting from heat expansion as a body force to the Navier-Stokes equation.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Body Force
  Name = Buoyancy
  Apply to Bodies = 1
  Navier-Stokes
    Boussinesq = on
  Add
  OK
```

Initial conditions should be given to transient cases, and probably are not needed for steady state solutions.

zzz In this case we choose a constant Temperature field and an small initial velocity that initializes the symmetry break.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
Initial Condition
  Name = Initial Guess
  Heat Equation
    Temperature = 293
  Navier-Stokes
    Velocity 1 = 1.0e-9
    Velocity 2 = 0.0
```

Only one boundary condition may be applied to each boundary and therefore all the different physical BCs for a boundary should be grouped together.

zzz In this case the Temperature and Velocity. The side walls are assumed to be adiabatic.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

Model

```
BoundaryCondition
```



```
Name = Bottom
Heat Equation
    Temperature = 293.5
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
New

Name = Top
Heat Equation
    Temperature = 293
Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
Add
OK
```

The conditions may also be assigned to boundaries in the Boundary condition menu, or by clicking on each boundary with the mouse. Here we use the latter approach as that spares us of the need to know the indexes of each boundary.

zzz Every menu selection section must be revised for each tutorial. These examples show the proper format and typical content.

```
Model
Set boundary properties
    Choose Bottom -> set boundary condition Bottom
    Choose Top -> set boundary condition Top
    Choose Sides -> set boundary condition Sides
OK
```

For the execution ElmerSolver needs the mesh files and the command file. We have now basically defined all the information for ElmerGUI to write the command file. After writing it we may also visually inspect the command file.

```
Sif
Generate
Edit -> look how your command file came out
```

Before we can execute the solver we should save the files in a directory. The ElmerGUI project includes all the files needed to restart the case.

```
File
Save Project
```

After we have successfully saved the files we may start the solver.

```
Run
Start solver
```

A convergence view automatically pops up showing relative changes of each iteration.
When there are some results to view we may start the postprocessor also.

```
Run
Start ParaView
```

Results

Due to the number of the time steps the simulation may take around zzz minutes.

You may inspect the results with Paraview or with ElmerVTK.

zzz In Figure 5.4 the obtained temperature distribution is presented.

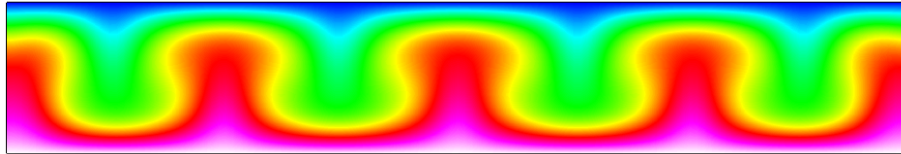


Figure 5.4: zzz Temperature distribution at 260 s.

zzz Extra task:

zzz If you have time you may try to solve the case with different parameters.