Guided examples using ELMER FEM

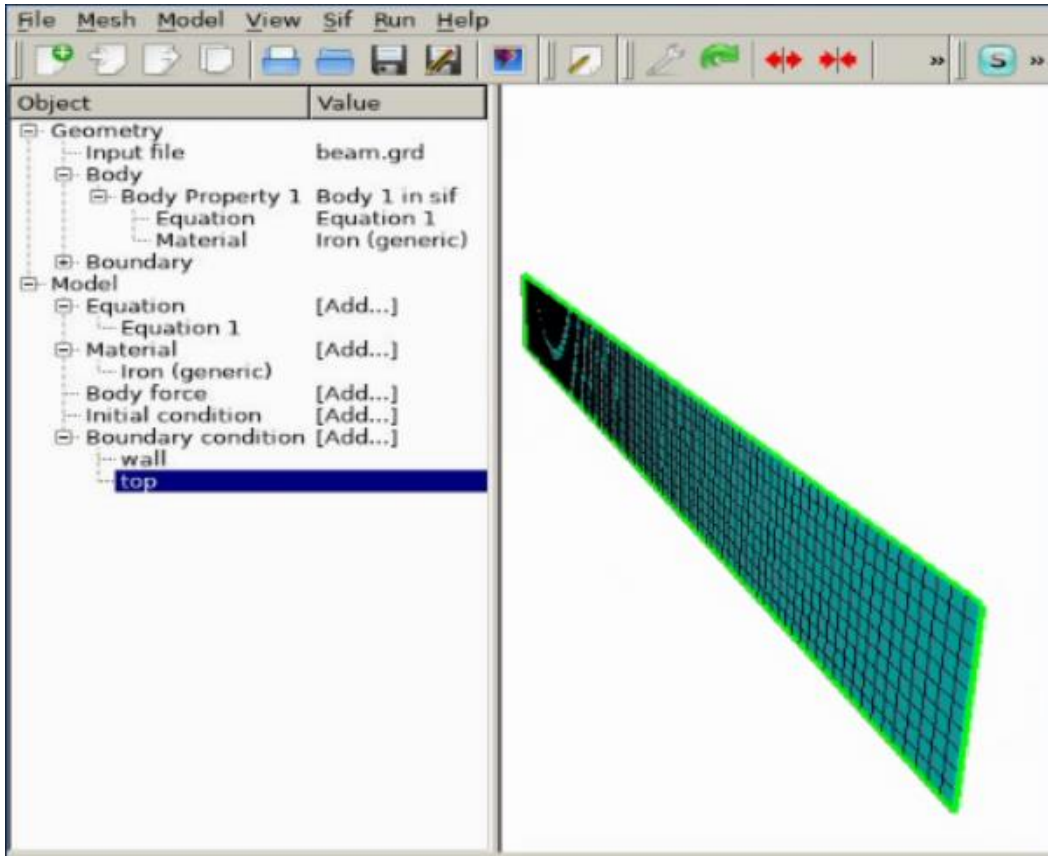
Presenter: Matic, Brank, Faculty of Mechanical Engineering, UL

Date: 12-03-2021

Quick software overview

- Elmer FEM
  - Finite element software package for multiphysical problems
  - Multiple modules within Elmer FEM:
    - Elmer Solver: Kernel of Elmer FEM, which performs FEM computations
    - Elmer Grid: Module for mesh preparation/conversion
    - Elmer GUI: Graphical user interface built around ElmerSolver with the intention to ease the case preparation and computation for users
  - Open source
  - Source code and documentation available at

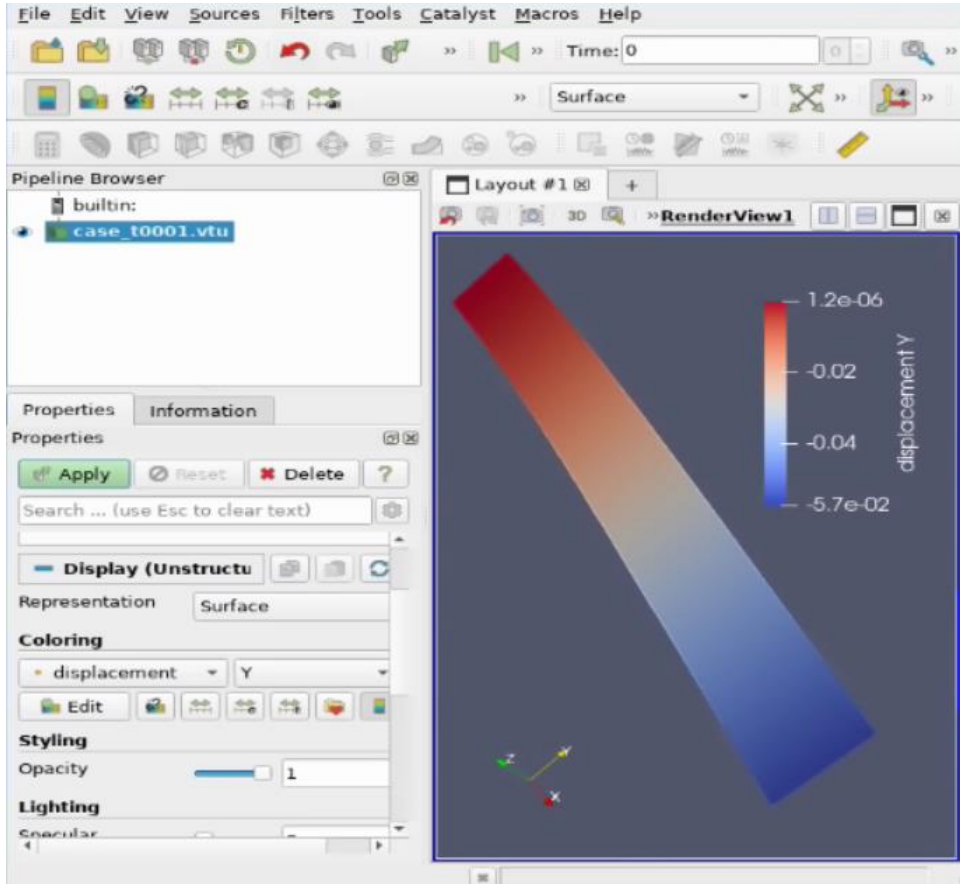
GitHub - ElmerCSC/elmerfem: Official git repository of Elmer FEM software



Quick software overview

- Paraview
  - multiple-platform application for interactive, scientific visualization
  - Open source
  - Many useful visualization capabilities:
    - Contours and isosurfaces for scalar and vector fields
    - Streamlines
    - Advanced data manipulation through Python
  - Support of variety of formats
    - VTK, VTU
    - CGNS
  - Source code and documentation available at

GitHub - Kitware/ParaView: VTK-based Data Analysis and Visualization Application



Software loading and running

- On Viz
  - Connect to Viz using NoMachine
  - Click bottom left icon on desktop to open the start menu (See figure)
  - Open terminal

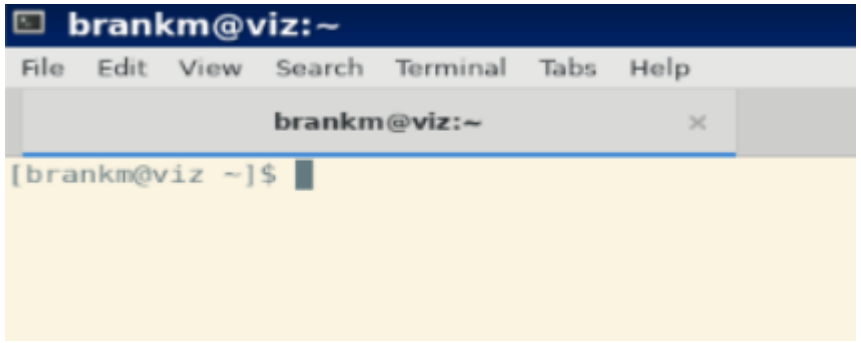


Fig: Terminal display

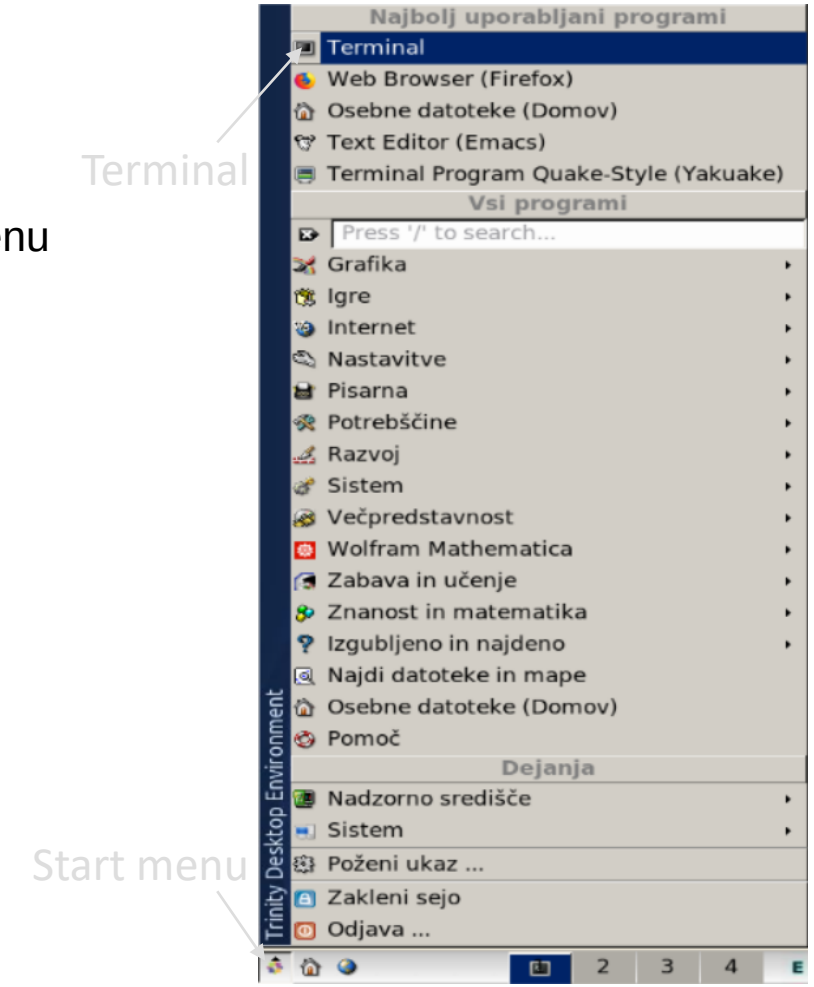


Fig: Start menu of trinity desktop.

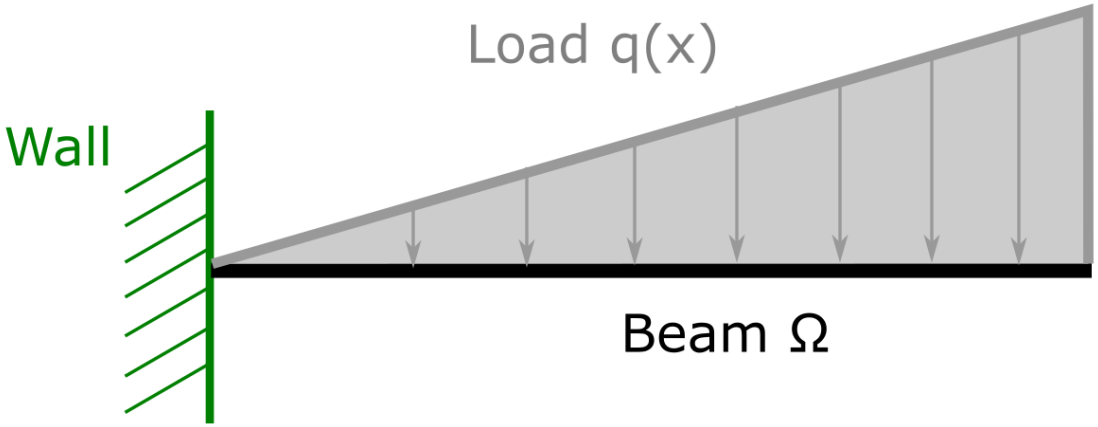
Software loading and running

- To open Elmer GUI:
  - To load Elmer into environment, type into terminal
    - `module load elmer/foss-2018b`
  - To run Elmer GUI
    - `ElmerGUI`

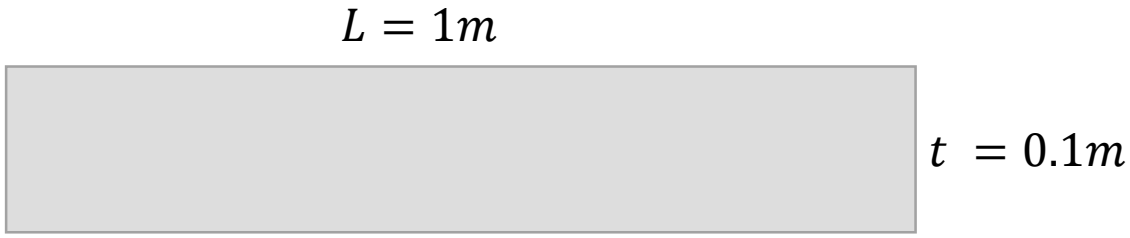
- To open ParaView
  - To load ParaView into environment, type into terminal
    - `module load ParaView/5.6.2-foss-2020b-mpi`
  - To run ParaView
    - `paraview`

Loaded elastic beam – 2D

- Case definition
  - 2D case
  - Homogeneous, elastic beam, defined on $x - y$ plane (length $1m$ and thickness $0.1m$)
  - Rigid support at the wall
  - Space dependent mechanical load, which grows linearly from 0 to $q_0 = 1e7N$

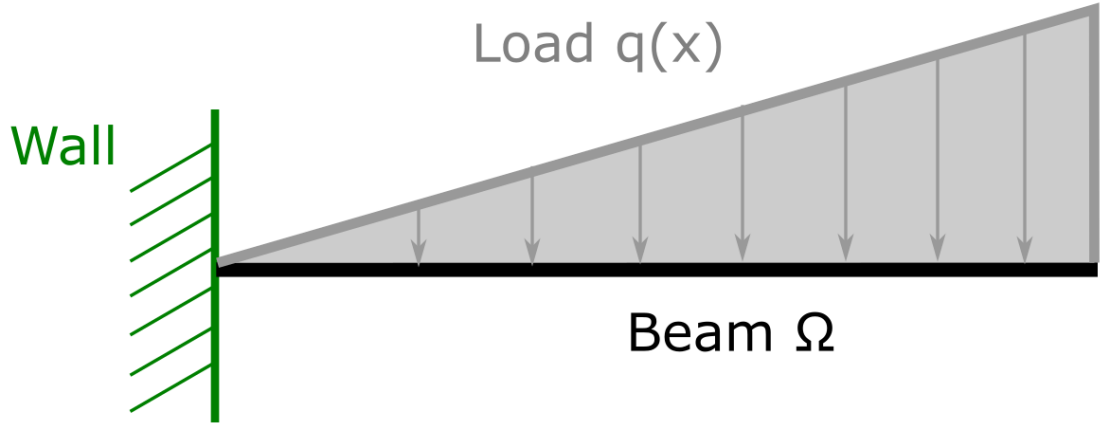


- Goal
  - Obtain the displacement of the beam

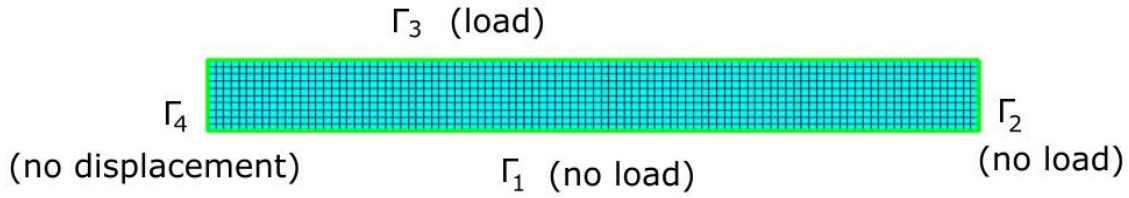


Loaded elastic beam – 2D

- Mathematical definition of the problem
  - Domain of computation:
    - Beam area Ω
  - Boundary conditions
    - Mechanical load Γ_3
    - No displacement at Γ_4
    - Zero load on Γ_1 and Γ_2
- Problem solution



$$\begin{cases} -\nabla\sigma = 0 & on\ \Omega \\ \sigma = \lambda tr[\varepsilon(u)]I + 2\mu\varepsilon(u) & on\ \Omega \\ u = 0 & on\ \Gamma_4 \\ \sigma n = 0 & on\ \Gamma_1\ and\ \Gamma\_2 \\ \sigma n = -q & on\ \Gamma_3 \end{cases}$$

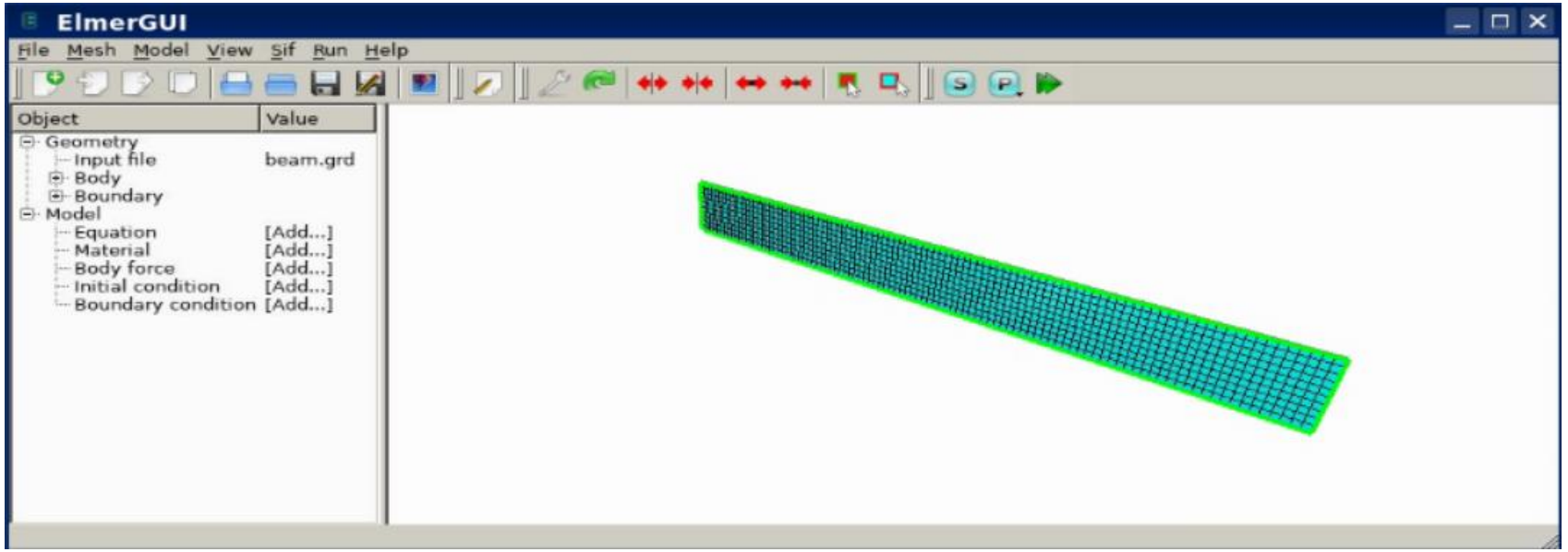


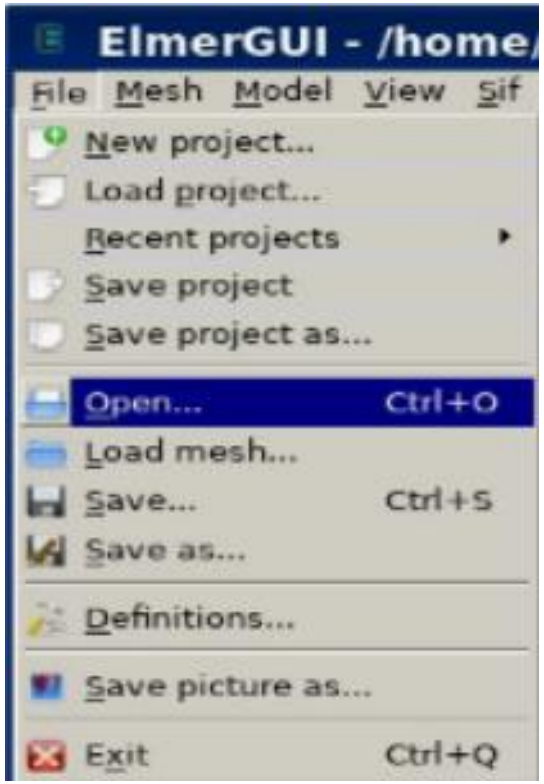
Case preparation - overview

1. Definition of domain of computation (Elmer GUI)
   - Mesh preparation/definition
2. Definition of type of physical problem (Elmer GUI)
   - Definition of equation to be solved
3. Definition of material properties (Elmer GUI)
4. Definition of boundary/initial conditions (Elmer GUI)
5. Computation (Elmer GUI)
6. Postprocessing of results (ParaView)

Case preparation

1. Definition of domain of computation (Elmer GUI)
   - Import mesh into Elmer GUI
   - `File->Open`
   - Navigate to your folder and select `beam.grd`
   - Use Mouse wheel/left button to rotate and zoom mesh

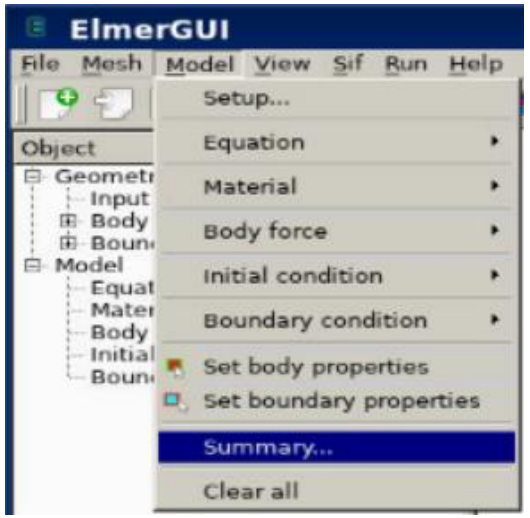


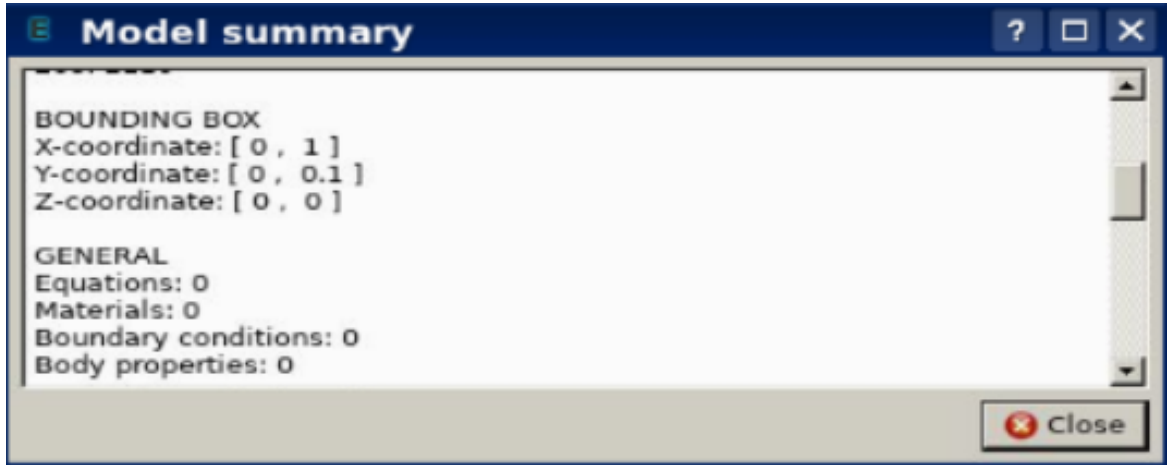


Case preparation

- Go to `Model->summary` to observe mesh information
- Check that mesh contains 3221 nodes
- How many surface elements are in the mesh?
- What is the type of the surface elements?
  - Triangles, quadrangles?
- Check Elmer documentation

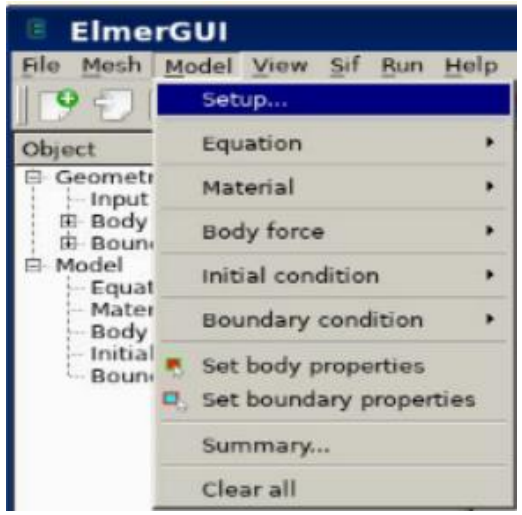
to see ID numbers for different elements

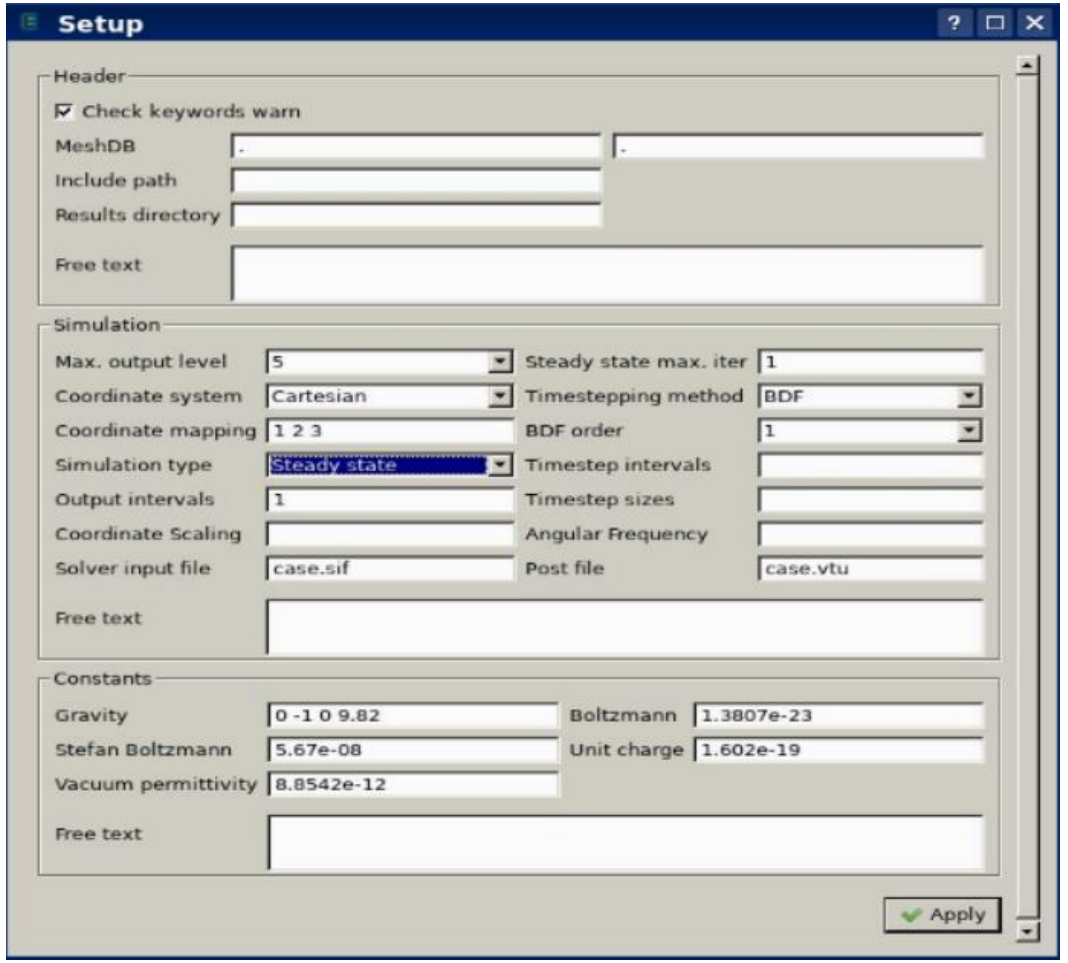




Case preparation

2. Definition of type of physical problem
   - In `Model->Setup...` one can define properties related to the simulation, such as results directory, input mesh directory, name of output file, constants, etc…
   - Make sure the `simulation type` is set to `Steady state`



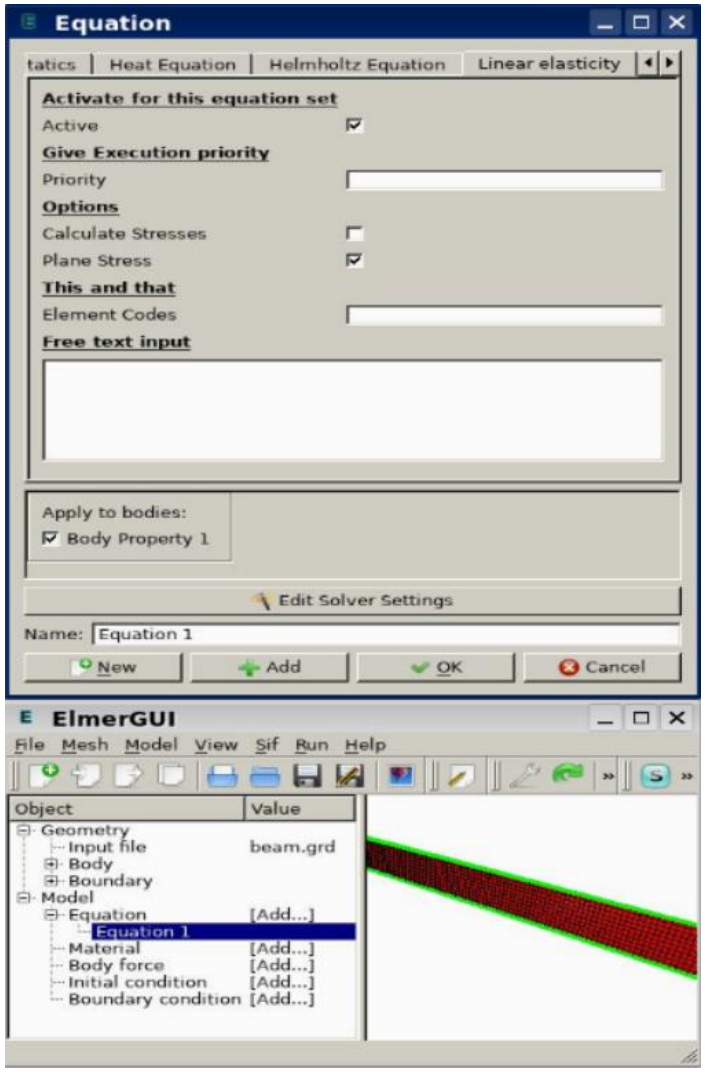


Case preparation

- In `Model->Equation->Add...` one can select equations to be solved during the computation. For the bending of beam, select `Linear elasticity tab`
- Tick checkboxes `Active` and `Plane stress`
- Assign the equation to the Mesh body

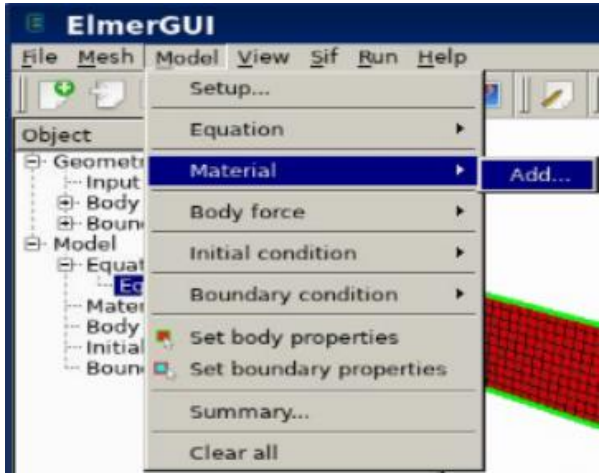
(Tick the box in `Apply to body`)

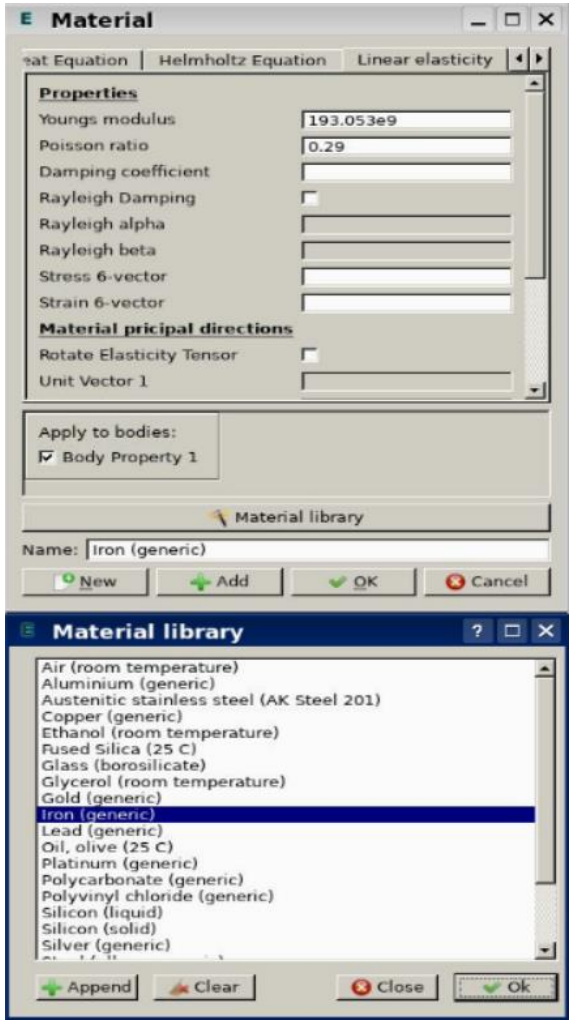
- In the end, click `OK` to accept new changes



Case preparation

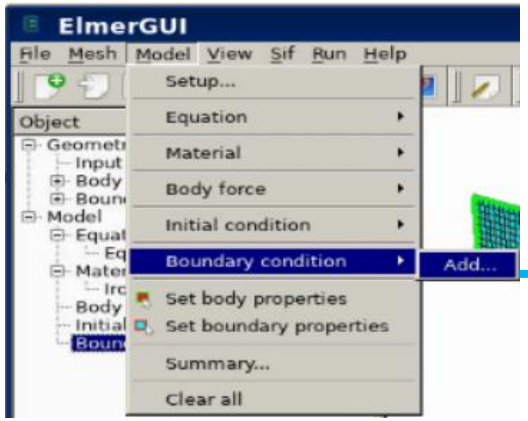
3. Definition of material properties
   - The material of the beam in our Study is iron with Poisson's ration of 0.29 and Young's modulus of $1.93 \cdot 10^9 N/m^2$
   - Click on `Model->Material->Add...`
   - Click on `Material library` and select `Iron(generic)`
   - Click `OK`
   - Tick box `Apply to bodies` to set the material to mesh.
   - Click `OK`

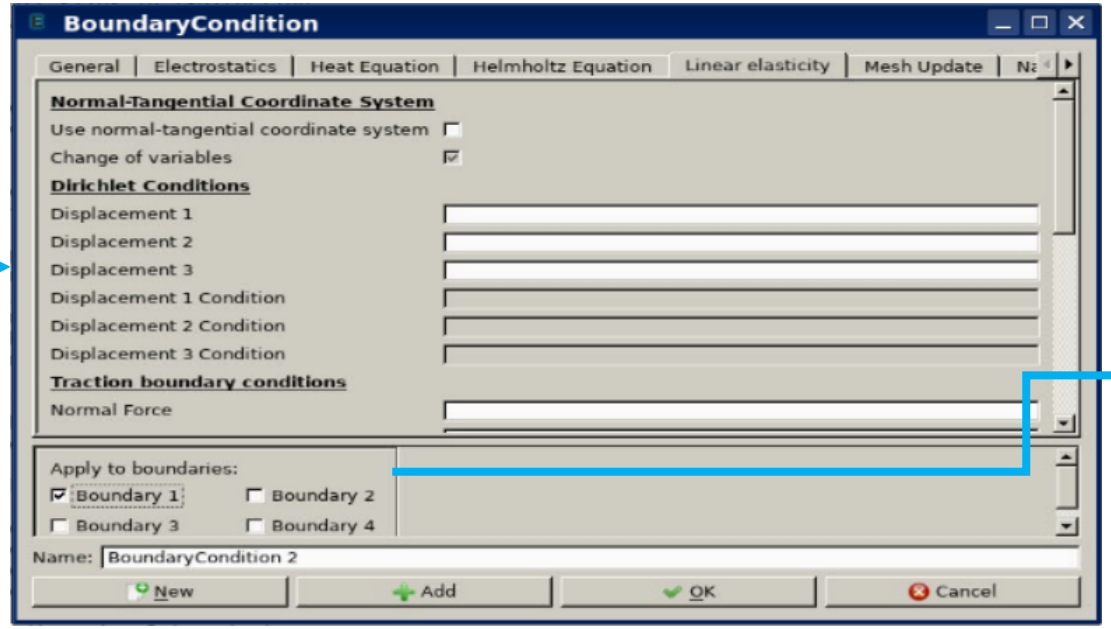


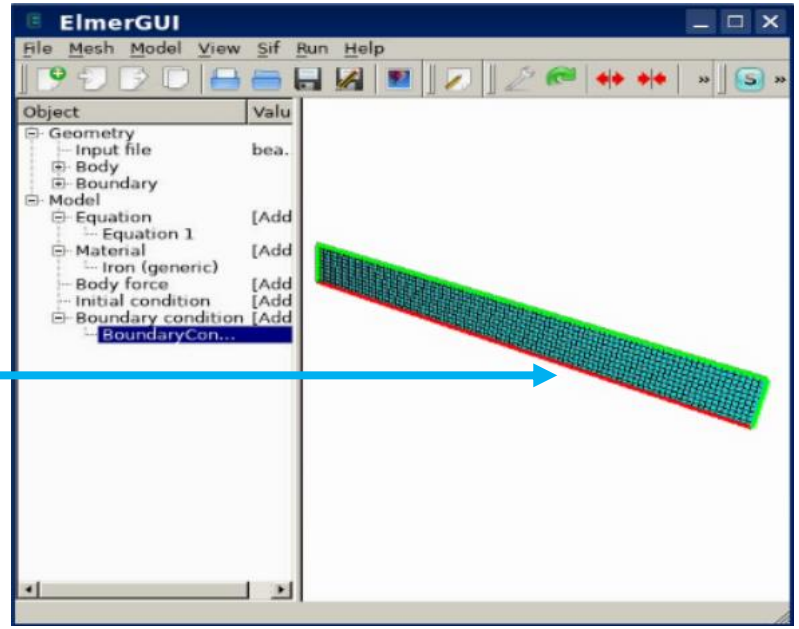


Case preparation

4. Definition of boundary conditions
   - To define a boundary condition, click on `Model->Boundary condition->Add...`
   - Navigate to `Linear elasticity`
   - Here, we can define displacements and loads on the mesh boundaries







Case preparation

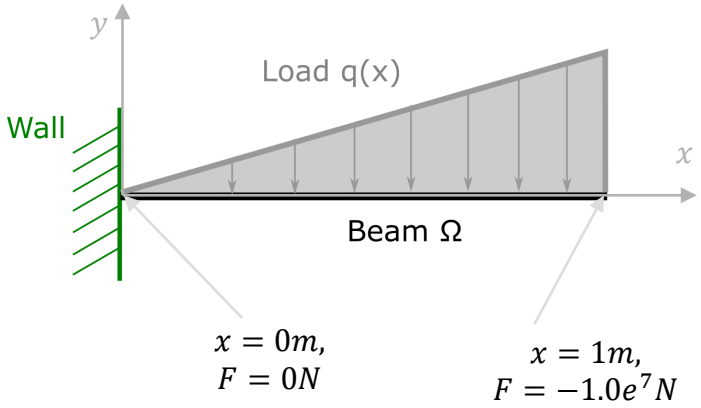
- Definition of wall boundary condition
  - Set `Displacement 1` and `Displacement 2` to 0. Identification number 1 refers to x-coordinate and 2 refers to y-coordinate
  - Select appropriate Edge on the Mesh
  - Set name of boundary condition to `Wall`
  - Click `OK`
- Definition of mechanical load $q(x)$
  - Create a new boundary condition (`Model->Boundary condition->Add…`)
  - Navigate to `Linear elasticity`
  - Now we have to define linear load that **points in** y-direction and is **linearly increasing along** x-direction. To do that:
    - Go to `Force 2` (remember, 2 refers to y-direction)
    - The variable to specify: `Variable Coordinate 1; Real; 0 0; 1 -1.0e7; End`
  - Set boundary condition name to `Load`

Case preparation

- Explanation of `Force 2` definition
  - The semicolon in Elmer FEM specifies the definition of new keyword
  - The first keyword (`Variable Coordinate 1`) specifies that `Force 2` is changing its value along coordinate 1 (remember, 1 refers to x-direction)
  - Then the user can define a table of x (coordinate), F (force) values. So the next keywords look like

```
Real;
0 0;
1 -1.0e7;
End
```

  - Keyword `Real` defines the type of numbers in a table
  - Keyword `0 0` specifies that at $x = 0m$, the force is $F = 0N$
  - Keyword `1 -1.0e7` specifies that at $x = 1m$, the force is $F = -1.0e^7 N$
  - Keyword `End` specifies end of table

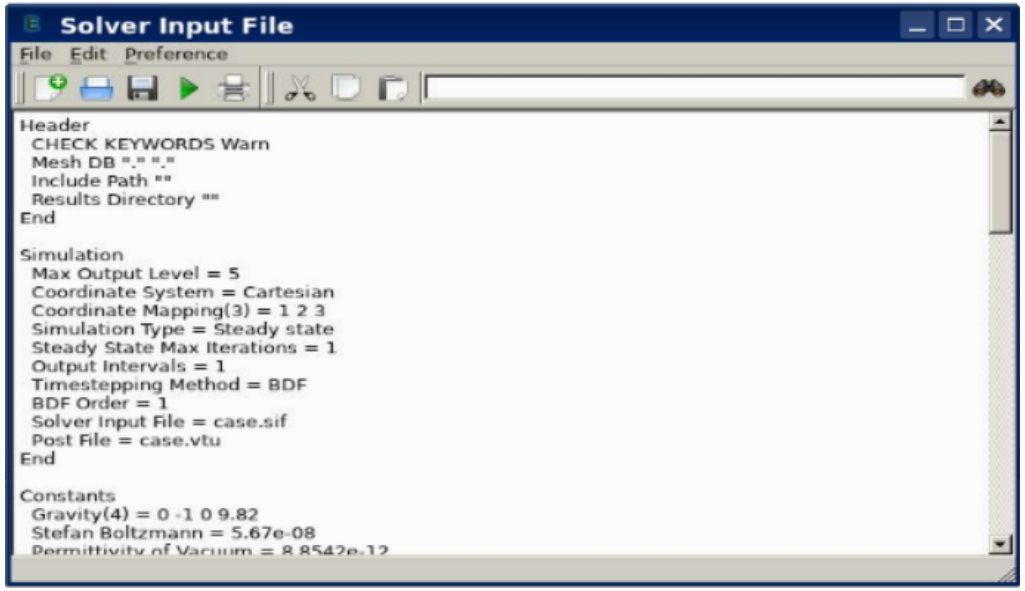


Case preparation

- In case of elasticity solver, the Mesh boundaries are automatically set to 0 load
- Thus it is not necessary to define zero force on remaining two edges
- As an exercise, one can set the forces at these two edges to 0

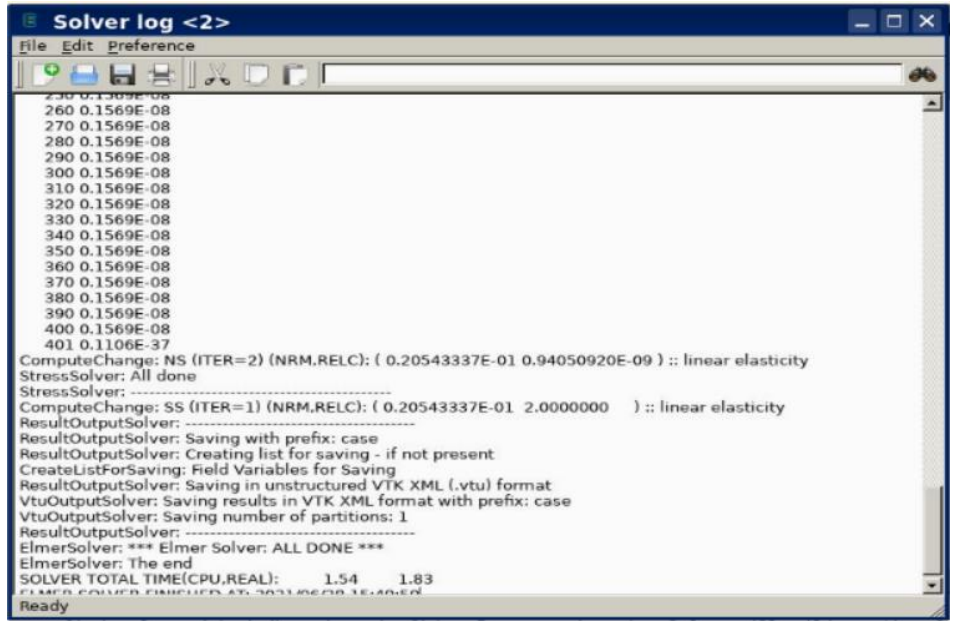
Running the case

5. Computation
   - Before running the case, one can observe the solver input file (extension .sif)
   - SIF file is a file, generated by Elmer GUI.
   - The file contains the case set-up with all the parameters specified by the user in Elmer GUI
   - This file is then taken as input by the Elmer kernel (ElmerSolver), which then computes the solution based on these parameters
   - Click on `Sif->Generate` and then `Sif->Edit`…
   - Here you can manually change properties in the editor



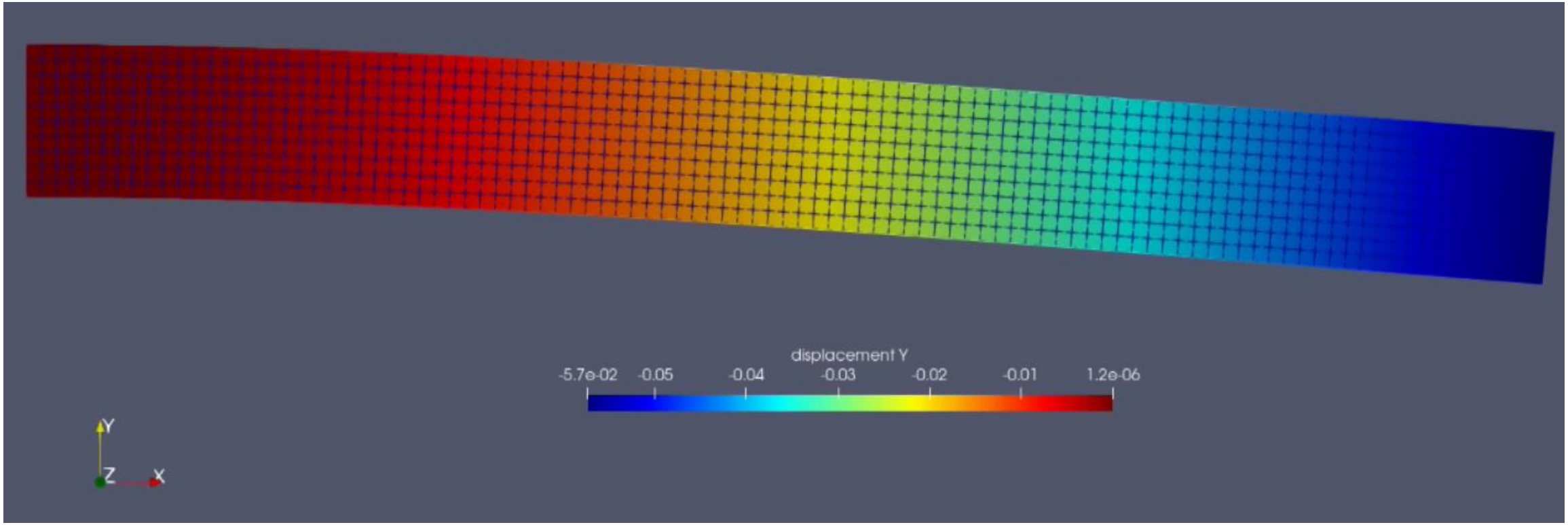
Running the case

- To run the case, click "`Save and run`" button (see figure)
- This button will then ask you to select a project folder. Then it will save the changes and run the case.
- Two windows open:
  - Convergence monitor
  - Solver log



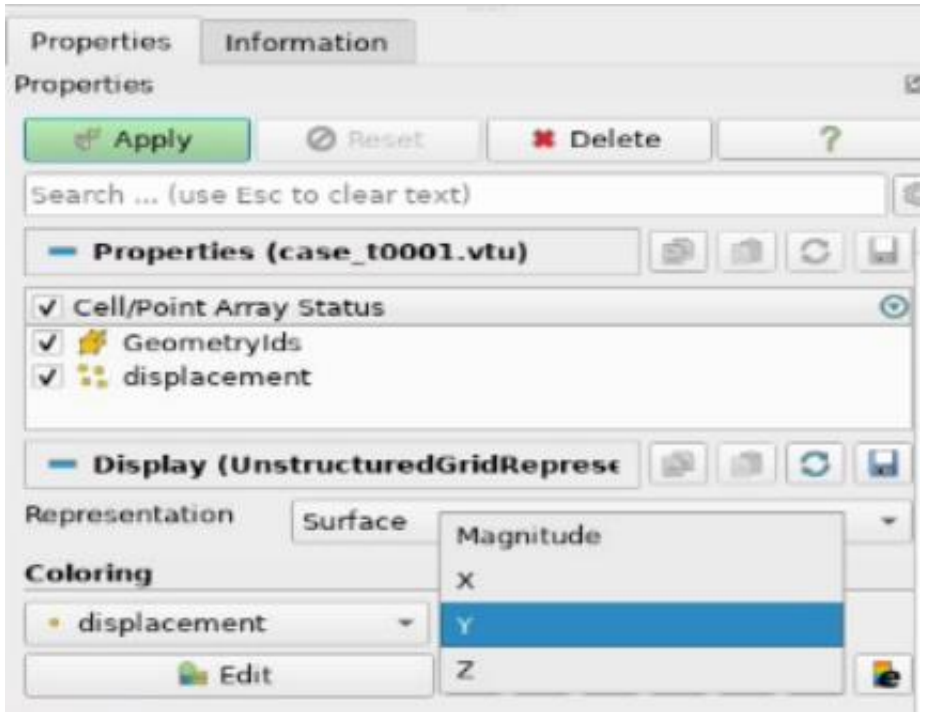
Post-processing

- Use mouse wheel/right button to rotate and zoom result

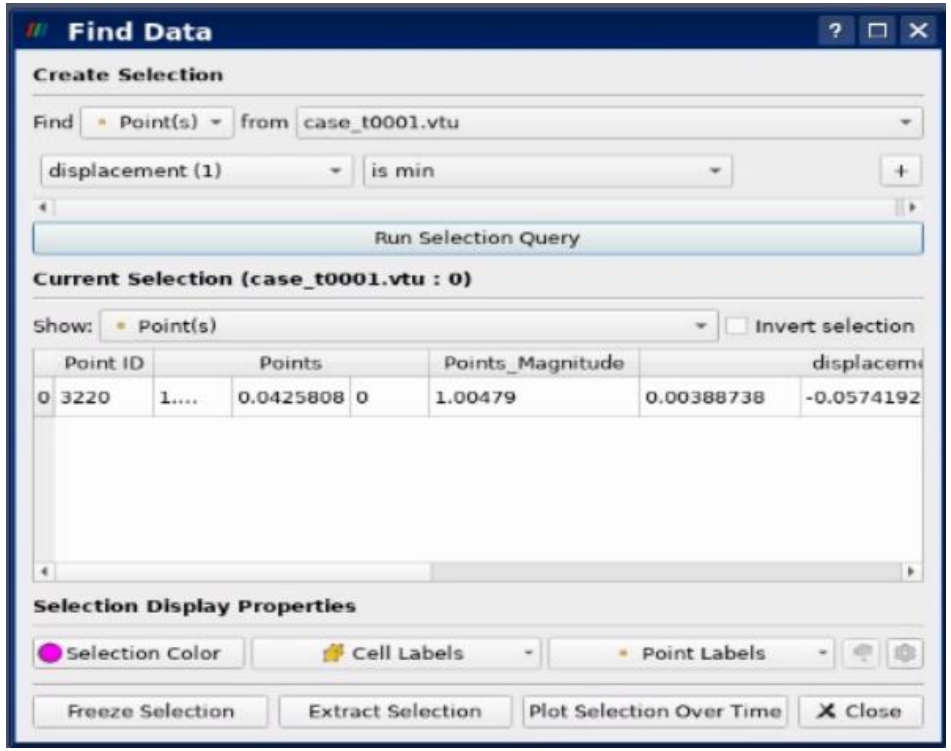


Post-processing

- To change coordinate of displacement, navigate to `Properties->coloring->displacement`



- To get the maximum displacement, navigate to `Edit->Find data...`
- In the dialog, select `Point(s)` and set `displacement(1)` to "`is min`"

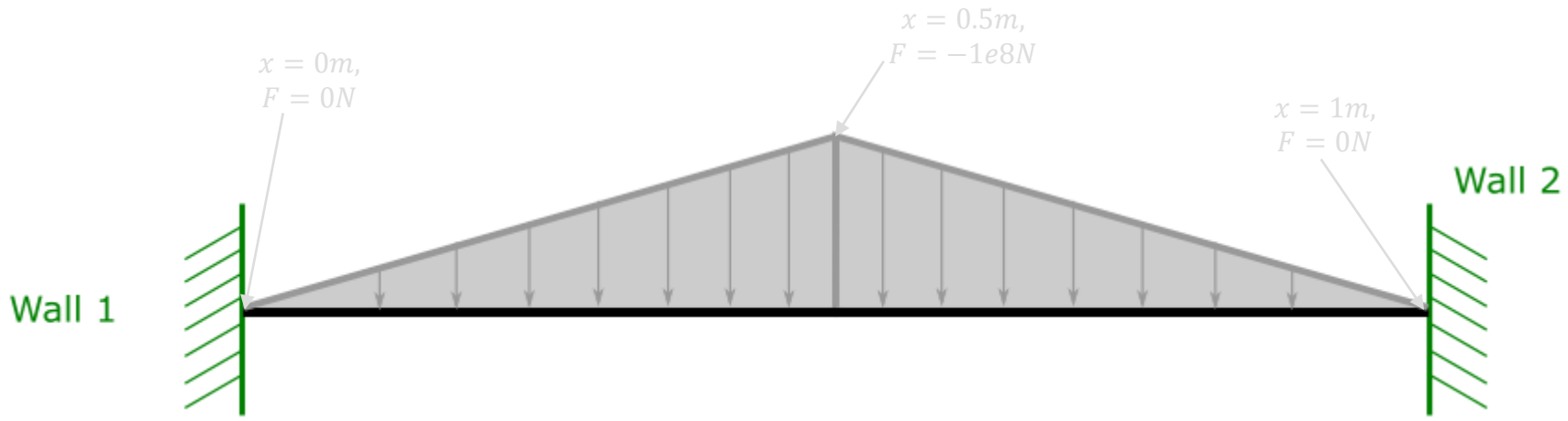


Exercise 1

- Change maximum load on the right edge (at $x = 1m$) to $F = -1.0e^{8}N$ and run the case again.
- Change the load again to $F = -1.0e^{9}N$. Run the case and observe the maximum displacement with increase of load

Exercise 2

- Add a new wall boundary condition to the right side of the beam (where $x = 1m$, set displacement in x and y direction to 0)
- Modify the load at the top to have the following dependency in x-direction. What is the maximum displacement in y-direction?

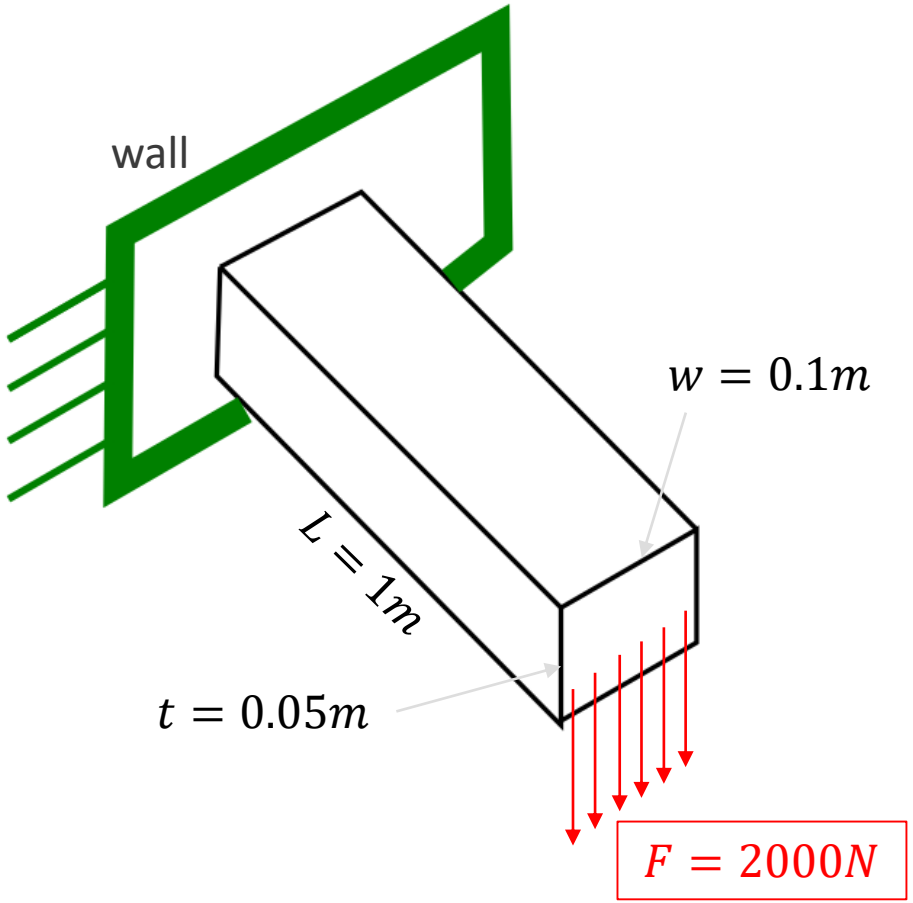


Exercise 3

- Transient loading
  - Change `Simulation Type` to `Transient (Setup->simulation type)`
  - Set the `Time Stepping Method` to `bdf`
  - Set the `Time step intervals` to 200
  - Set the `Time Step Sizes` to $1e^{-4}$

Loaded elastic beam – 3D

- Case definition
  - A homogeneous, elastic beam is rigidly supported on one end. On the other end, the force is 2000N due to an attached object. The weight of the beam itself is also included as additional load.
  - The length of the beam is 1m, thickness is 0.05 m and width is 0.1m
  - Young's modulus is $10 \cdot 10^9 N/m^2$ and Poisson's ratio is 0.37
  - Density of the material is $550 kg/m^2$
- Goal
  - Calculate displacement field of the beam
  - Obtain the location and value of the maximum displacement
  - Calculate stress field

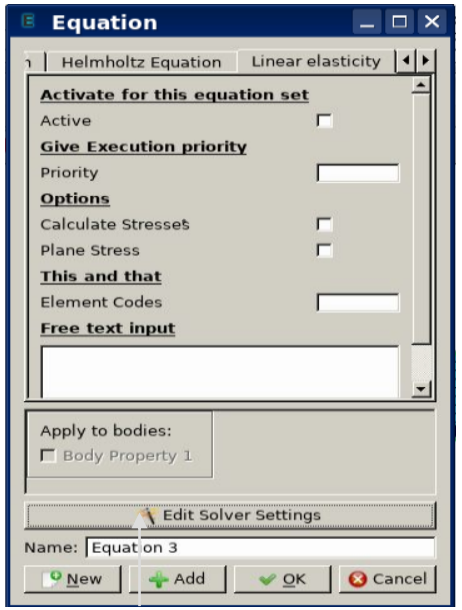


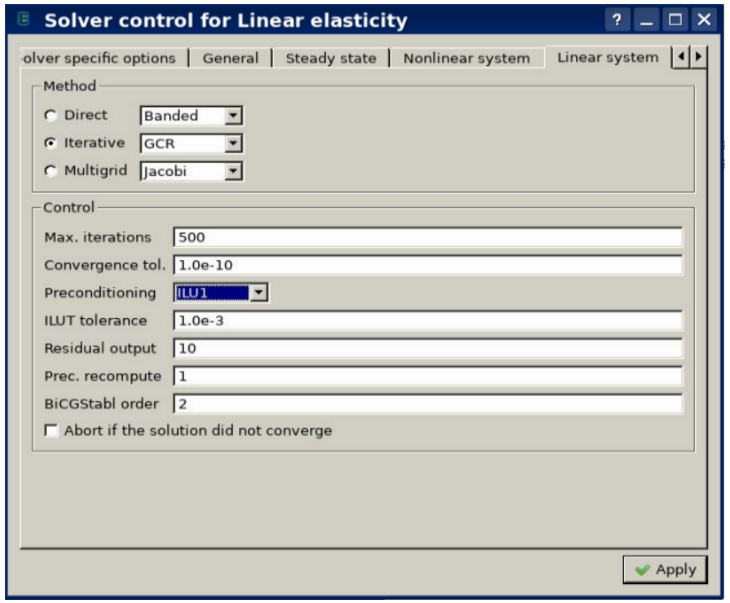
Loaded elastic beam – 3D

1. Definition of domain of computation
   - Import mesh into Elmer GUI
   - Navigate to your folder and select `beam3d.grd`
   - Use Mouse wheel/left button to rotate and zoom mesh
   - Verify that mesh consists of 6073 nodes and of 1200 quadratic hexahedral elements

Loaded elastic beam – 3D

2. Definition of type of physical problem
   - Set the `simulation type` to `Steady state`
   - Set equation to `Linear elasticity`
   - In `Linear elasticity`, set checkbox next to `Active`
   - We also want to compute stresses as a post-processing step, so set checkbox of `Calculate Stresses`
   - Click on `Edit solver settings`:
     - check `Iterative method` and set it to `GCR`
     - Set `Preconditioning` to `ILU1`





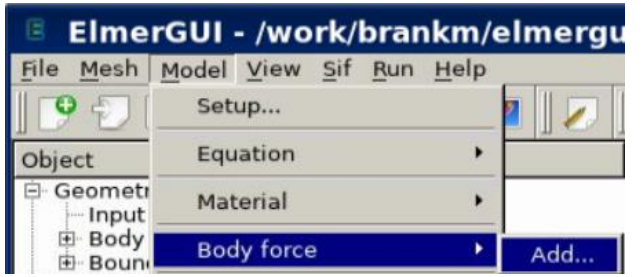
Edit Solver Settings

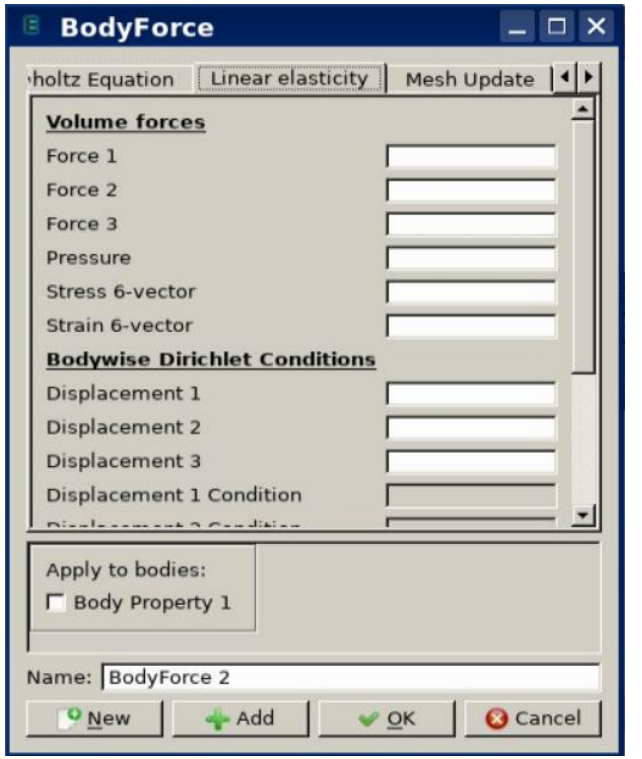
Loaded elastic beam – 3D

3. Definition of material properties
   - Create a new material and define:
     - Density,
     - Poisson’s ratio and
     - Young’s modulus
   - Assign the material to 3D beam

Loaded elastic beam – 3D

4. Definition of boundary conditions
   - Define zero displacement on left face that lies in x-z plane (at the wall – refer to figure on slide 24)
   - Note that second boundary condition distributes the load of 2000N uniformly on the area of $t \cdot w = 0.05m \cdot 0.1m = 5.0e^{-3}m^2$
   - The weight of the beam should be defined through `Body Force`
     - Navigate to `Model->Body force->Add...->Linear elasticity`
     - Set the force in negative y-direction through mass of beam and gravitational acceleration
     - Apply the body force to the mesh body



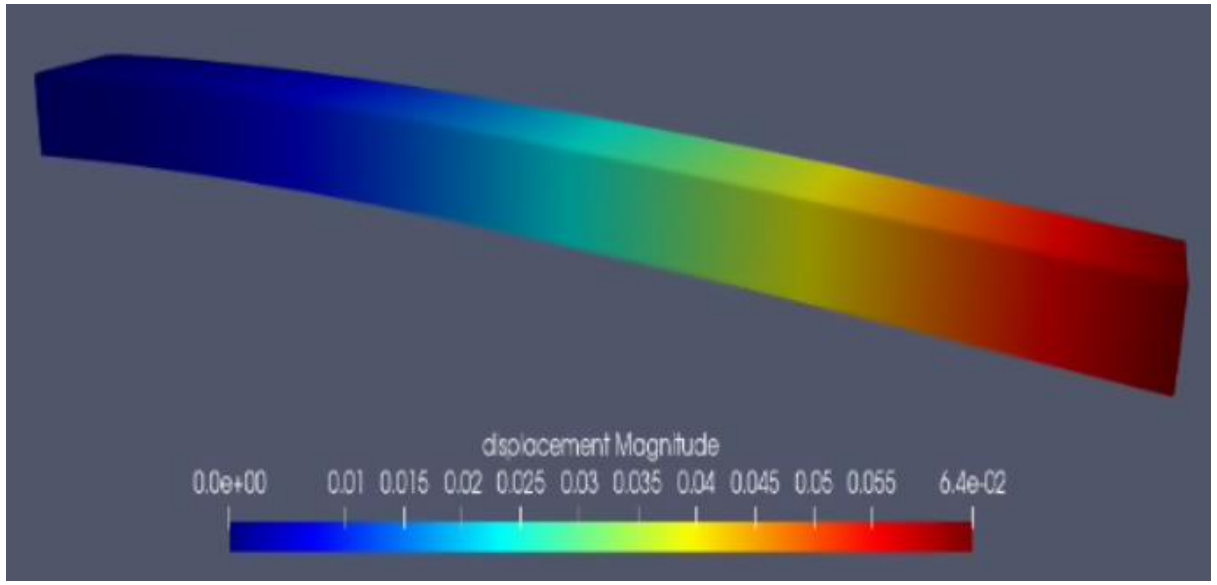


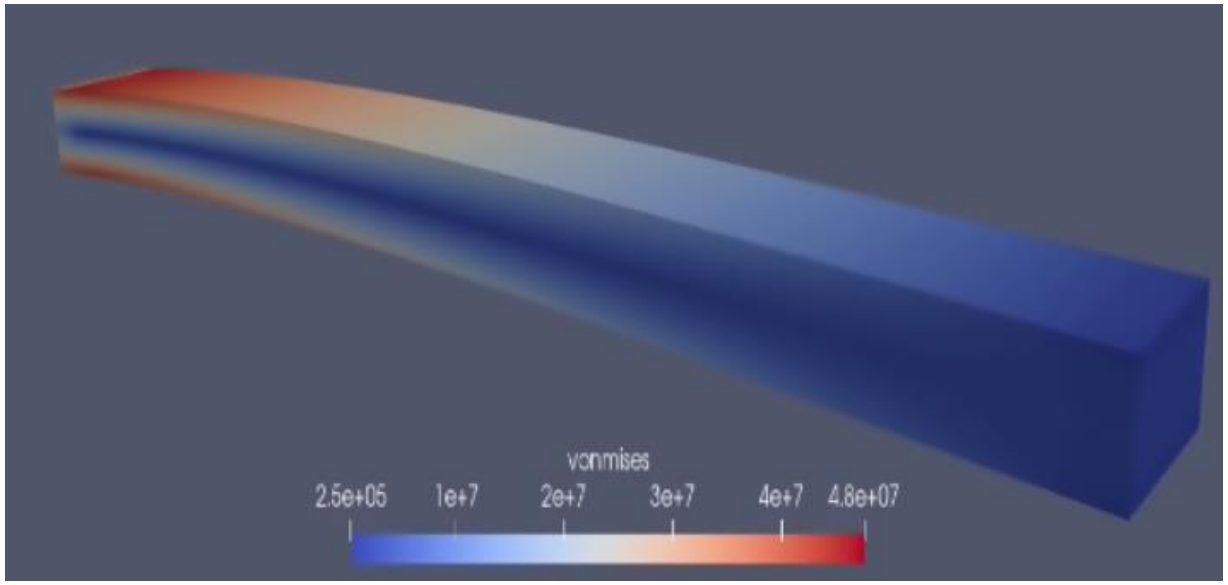
Loaded elastic beam – 3D

5. Computation
   - Generate and check SIF file
   - Save the case and run it

Loaded elastic beam – 3D

5. Post processing
   - What is the maximum displacement and where is its location
   - Plot displacements and Von mises stresses





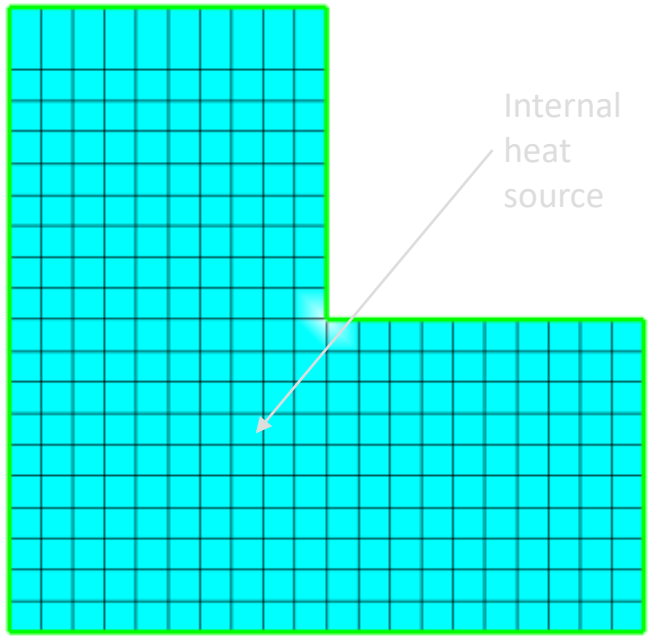
Exercise

Gravity in x-direction

- The beam should be more rigid if the beam is orientated differently
- Change the direction of gravity and force of attached object in the negative x-direction

Heat equation

- Case definition
  - A structure in L-shape is heated by internal source with magnitude of $1W/m^3$. The density of the structure is $1kg/m^3$ and the heat conductivity is $1W/mK$. The temperature of boundaries of the structure is set to 0.
- Goal
  - Obtain temperature distribution in the structure



$T_{boundary} = 0℃$

Heat equation

- Mathematically the problem to be solved is

$$\begin{cases} -k\Delta T = \rho f & on\ \Omega \\ T = 0 & on\ \Gamma \end{cases}$$

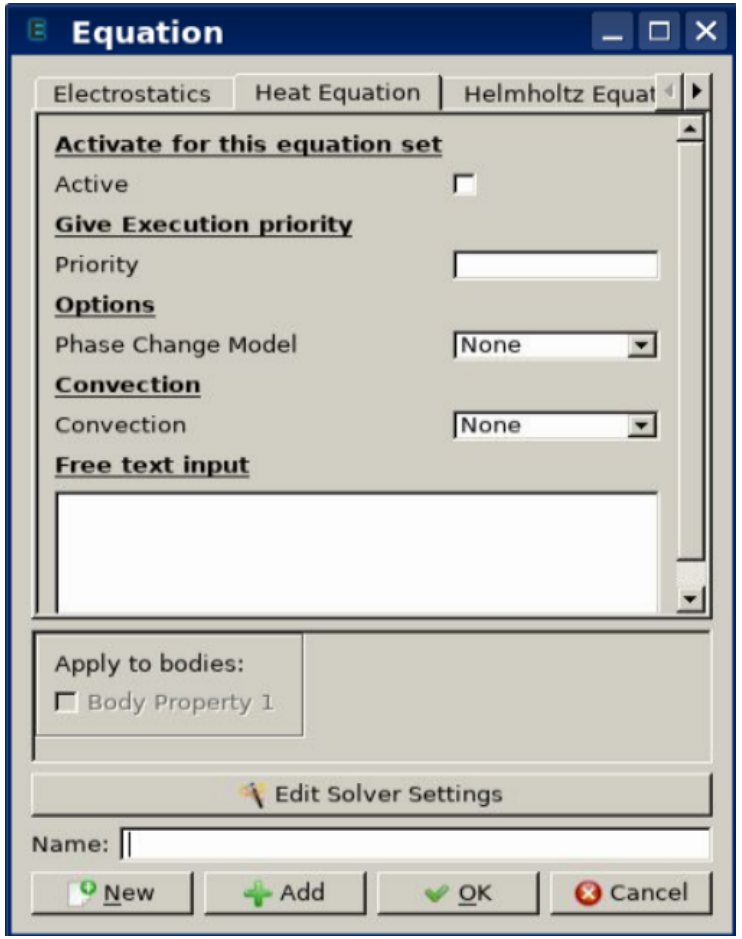
- Parameters
  - k … heat conductivity
  - T … temperature
  - f … heat source

Heat equation

1. Definition of domain of computation
   - Import mesh into Elmer GUI
   - Navigate to your folder and select `angle.grd`
   - Use Mouse wheel/left button to rotate and zoom mesh
   - Verify that mesh was successfully imported and that it consists of 341 nodes and of 300 bilinear elements

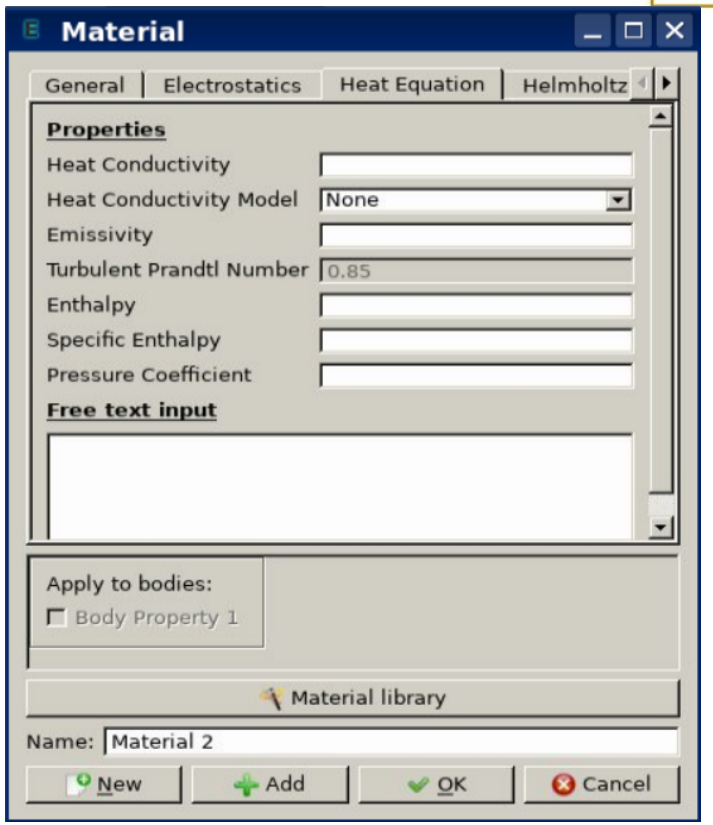
Heat equation

2. Definition of type of physical problem
   - Set the `simulation type` to `Steady state`
   - Set equation to `Heat equation`
   - In `Heat equation` check checkbox next to `Active`



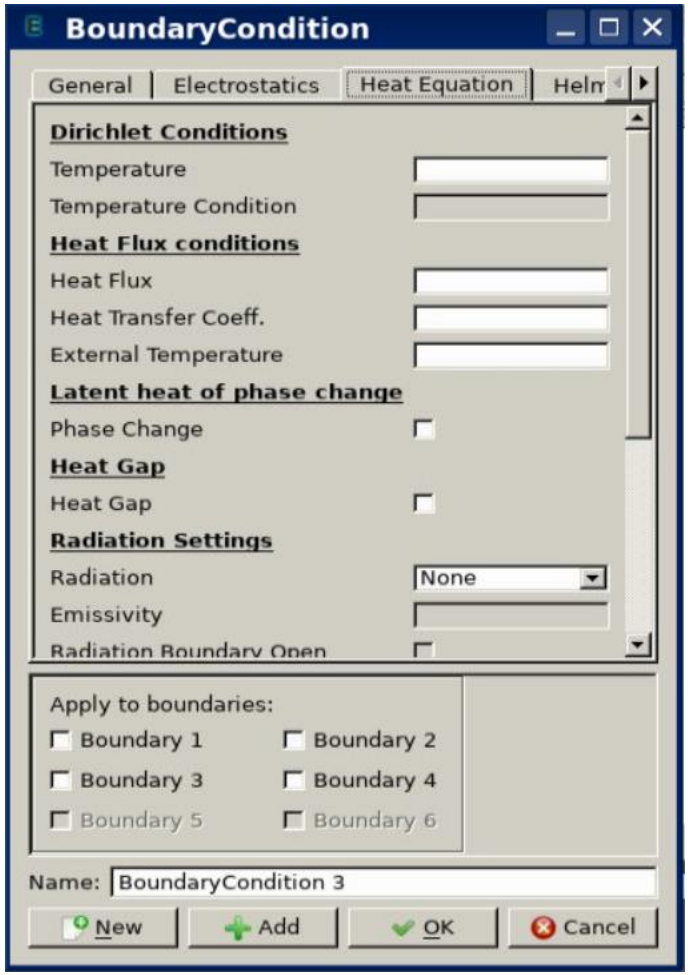
Heat equation

3. Definition of material properties
   - Create a new material and define:
     - Density,
     - Heat conductivity (in Heat)
   - Assign the material to the body



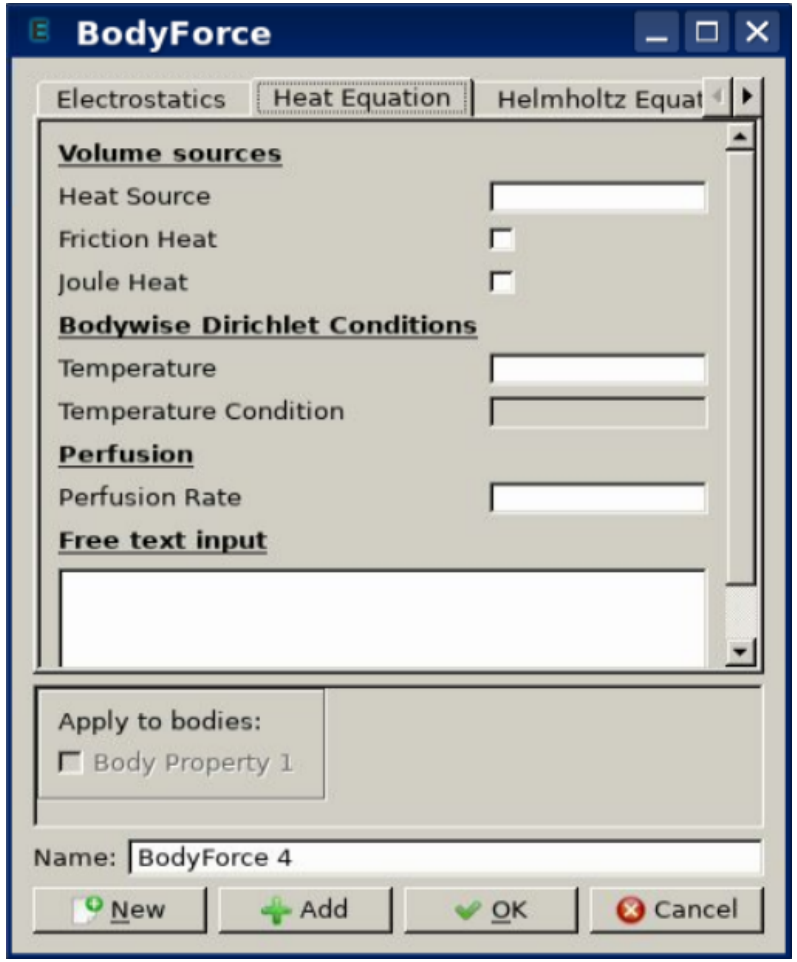
Heat equation

4. Definition of boundary conditions
   - Define zero temperature on boundaries
   - Temperature is defined under `Dirichlet conditions`
   - Apply temperature to all boundaries



Heat equation

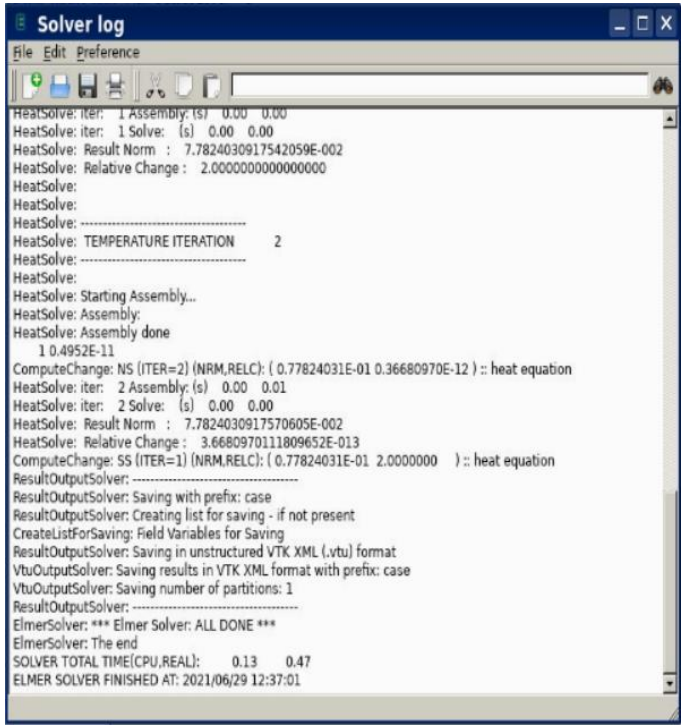
4. Definition of boundary conditions
   - Define volumetric heat source in the structure
   - A Body force represents the right-hand side of the equation which in this case represents the heat source
   - Navigate to `Body force`
   - Set `Heat Source` to `1` and apply the condition to a body, then click `OK`

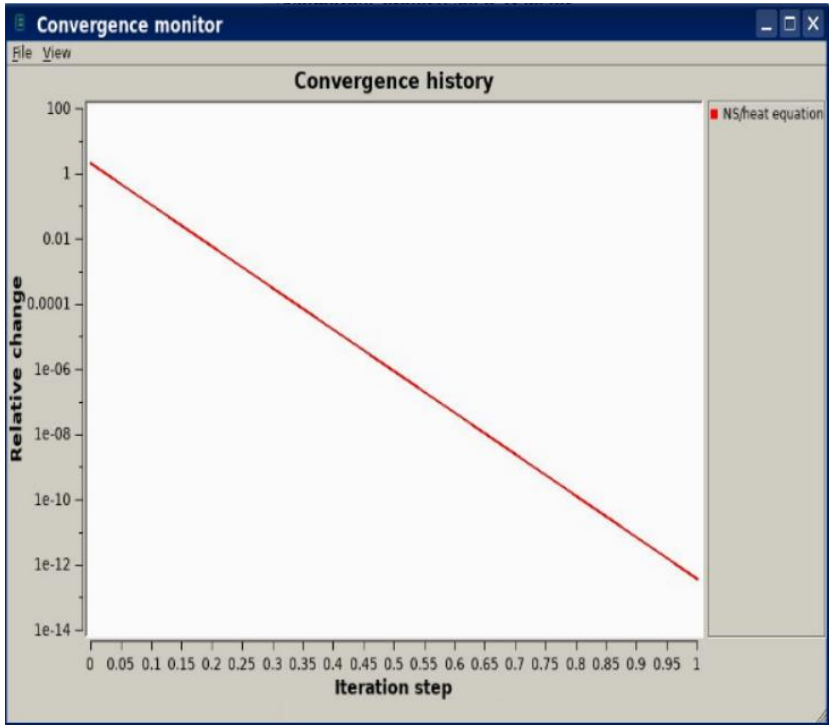


Heat equation

5. Computation

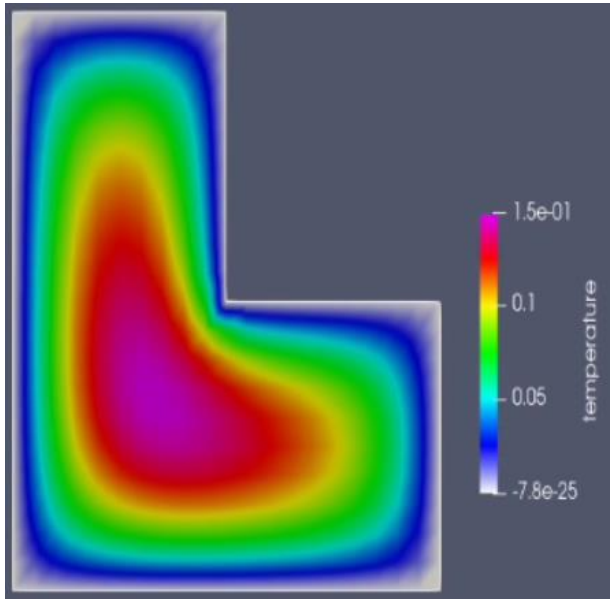
- Generate and check SIF file
- Save the case and run it

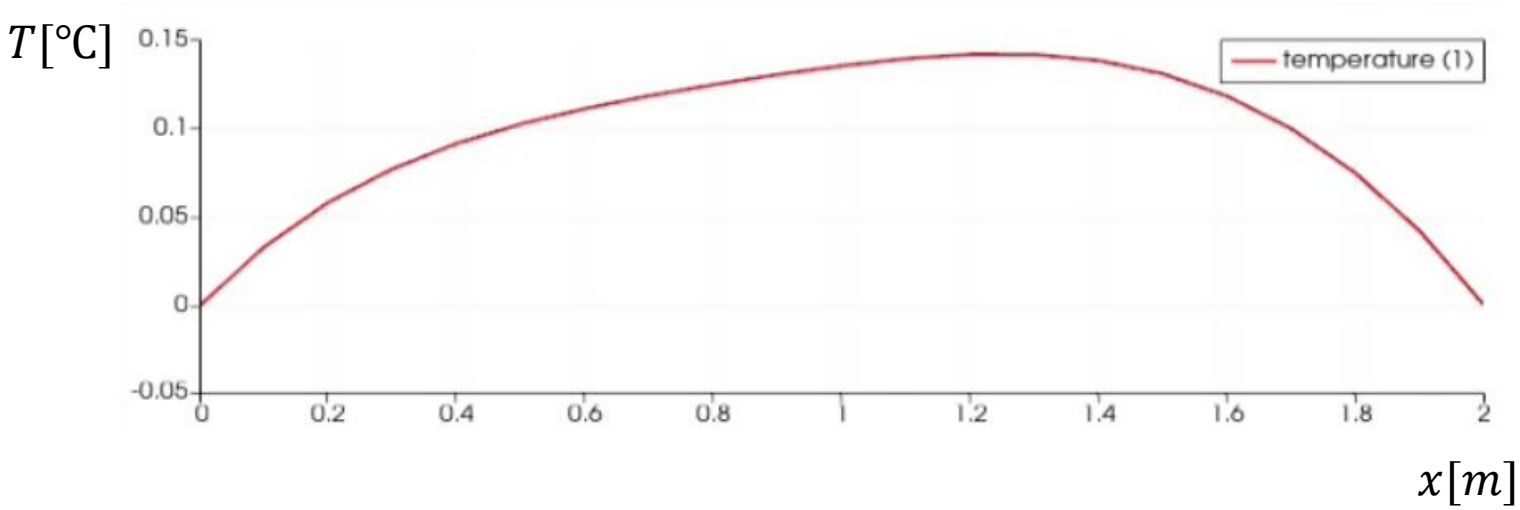




Heat equation

6. Post-processing
   - Open the result in ParaView
   - What is the value of maximum temperature
   - Plot temperature as a function of y-coordinate at $x = 0.5m$
     - Go to `Filters->Alphabetical->Plot On Intersection Curves`
     - In `Properties`, specify the intersection plane and then click `Apply`





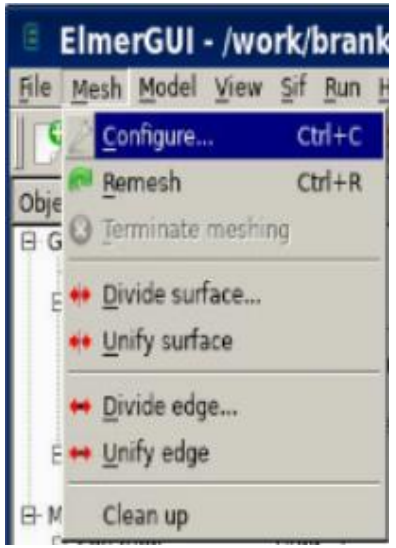
Exercise

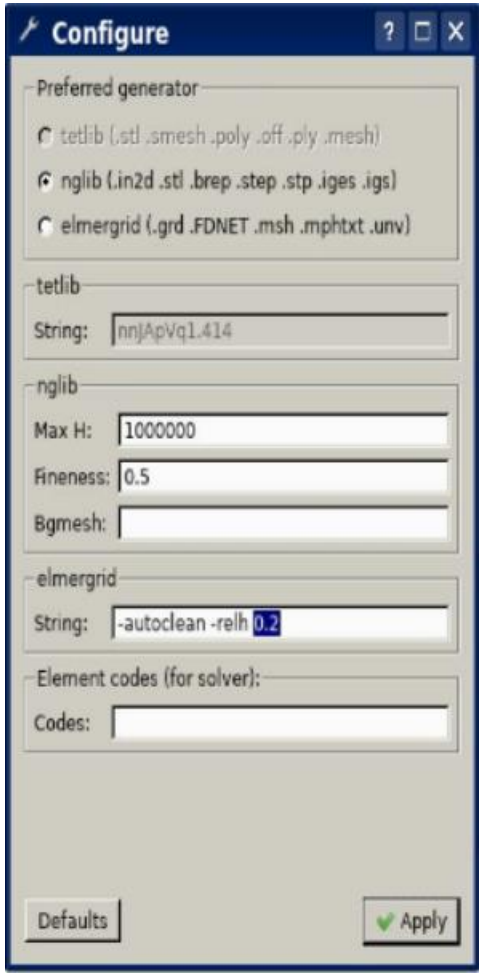
6. Post-processing

- Examination of computational time vs. number of mesh nodes
- Check output of Solver Log and read second line from bottom to get the CPU time (see below)

```
ResultOutputSolver: ----------
ElmerSolver: *** Elmer Solver: ALL DONE ***
ElmerSolver: The end
SOLVER TOTAL TIME(CPU,REAL):       0.39       0.64
ELMER SOLVER FINISHED AT: 2021/06/29 13:25:53
```

- Let's increase the mesh density (i.e. increase the number of nodes and elements) and rerun the case
- Navigate to `Mesh->Configure…`
- In `elmergrid->String:`, change `-autoclean -relh 1.0` to `-autoclean -relh 0.6`
- `Click Apply`
- Click `Mesh->Remesh` to recompute mesh





Exercise

- Run the case with denser mesh and observe the time difference
- Try to increase the density even more, for example change the value of 0.6 to 0.1
- Compare CPU times vs. number of mesh nodes

Overview of commands

- `elmerf` ... Command to compile user defined Fortran routines
- `ElmerGrid` ... Command to generate and convert mesh data
- `ElmerGUI` ... Command to run the user interface
- `ElmerSolver` ... Command that reads SIF file and mesh data and performs FEM calculations
- `ElmerSolver_mpi` ... Command that performs parallel FEM calculations

Overview of commands

`elmerf`

- Users can define their own functions in Fortran that are then passed to ElmerSolver
- User defined functions are defined in files with `.f90` extension
- In SIF file, the reference to this file can be defined in boundary condition block, material block,...
- ElmerSolver then calls this function by reading reference in SIF file

Overview of commands

- Let’s examine the zero temperature boundary condition in Heat transfer case
- In SIF file, the boundary condition block is defined as:

```
Boundary Condition 1
  Target Boundaries(1) = 1
  Name = "zero_temp"
  Temperature = 0
End
```

- Here, the temperature of the whole target boundary is set to 0.
- To call an external function, delete the `Temperature = 0` line and add these two lines

```
Temperature = Variable Coordinate 2
Real Procedure "define_temp" "defineTemp"
```

- The first line means that temperature will be a variable of y-coordinate
- The second line means that we are calling a function `"defineTemp"` that is located in `define_temp.f90` file.

Overview of commands

- File `define_temp.f90` should look like this:

```
FUNCTION defineTemp(Model, elmer_node, t, y) RESULT(elmer_temperature)
USE DefUtils
TYPE(Model_t) :: Model
TYPE(Nodes_t) :: Nodes, EdgeNodes
INTEGER :: elmer_node, i
REAL(KIND=dp) :: t, y
.
. $ Do your own calculations
.
END FUNCTION defineTemp
```

Overview of commands

The name of the function is `defineTemp`

- Input parameters are:
  - `Model` -> structure of ElmerSolver, it contains different parameters that we can access during function execution
  - `elmer_node` -> ID of the node that is currently being processed
  - `t`-> current timestep
  - `y`-> variable specified in SIF file, in our case y-coordinate
- Output parameter is temperature, as defined in SIF file
- Inside this function we can now write different routines and specify our own boundary condition

Thank you for you attention!

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, United Kingdom, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Switzerland, Turkey, Republic of North Macedonia, Iceland, Montenegro