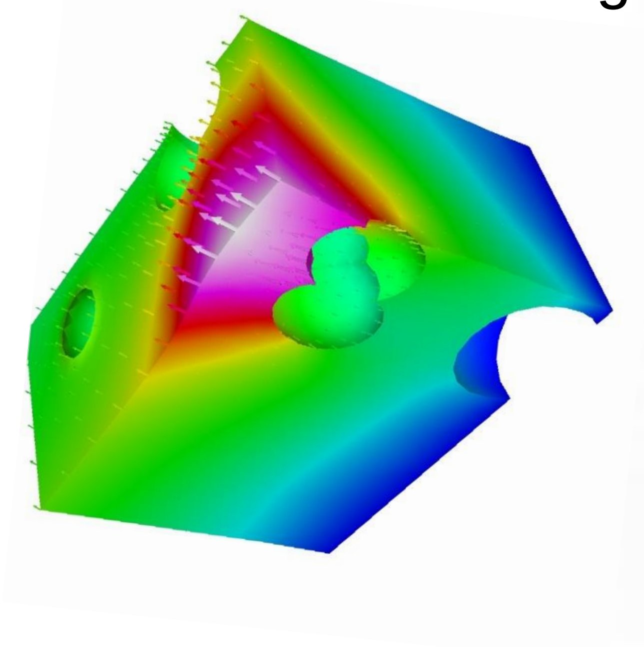
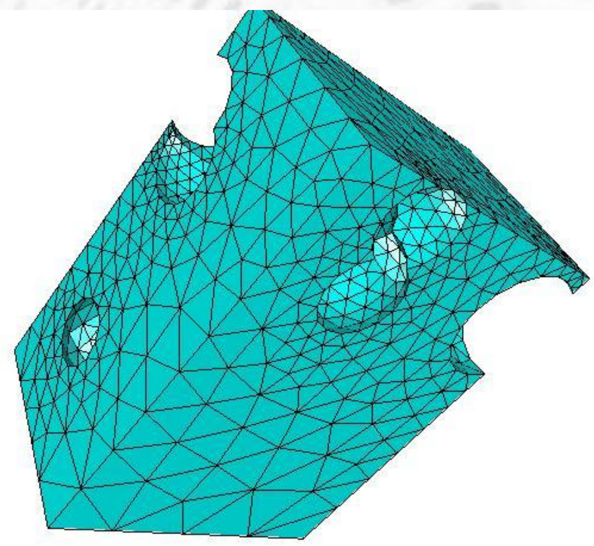
ElmerGUI demo example

How does your Swiss cheese deform?

A walk-through demonstration



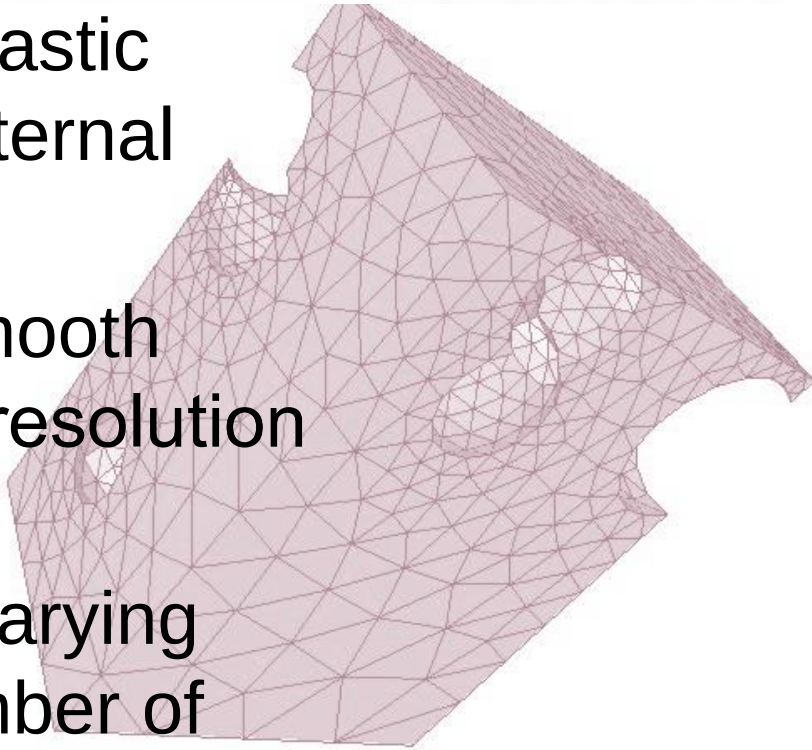


ElmerGUI Demo

- The material for this demo is on the USB stick
- If you have a working Elmer/ElmerGUI environment at hand, you can do the example at the same time, as we will go in a reasonably slow, yet steady pace
- BUT: we are not able to stop if you got seriously stuck
  - But, there is always time for short questions
- The example can be revisited (including our support) in the last hour of this tutorial

The Problem/Motivation

- A 20 x 20 x 40 cm block of cheese (assumed to be a linearly elastic material) is put under an external force
- **Linear equations** rather smooth solution→ increased mesh resolution only by geometry
- Testing on topologies with varying geometric complexity – number of voids may easily be altered

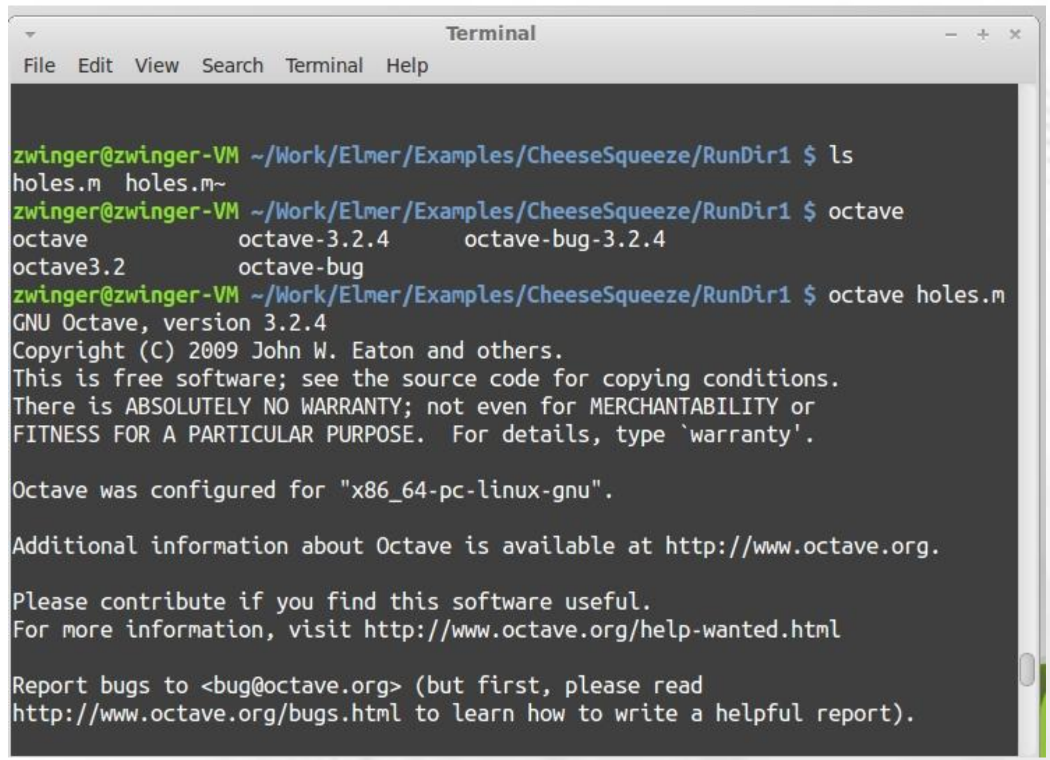


Work Flow

- Pre-processing
  - Create a random distribution of spherical voids (=holes) in a brick as a Netgen geometry file using Octave/Matlab script
  - Meshing geometry using Netgen
  - Checking mesh using ElmerGrid/Post
- Set-up
  - Importing mesh into ElmerGUI
  - Defining the case/project
- Run
- Post-processing

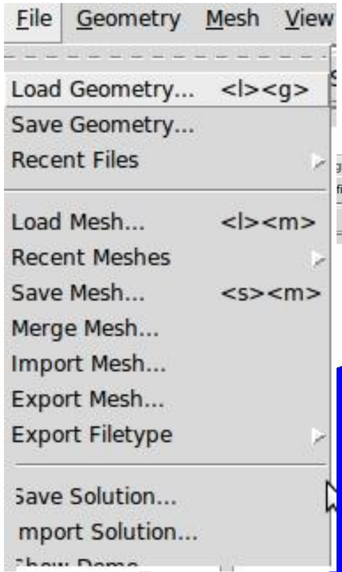
Pre-processing

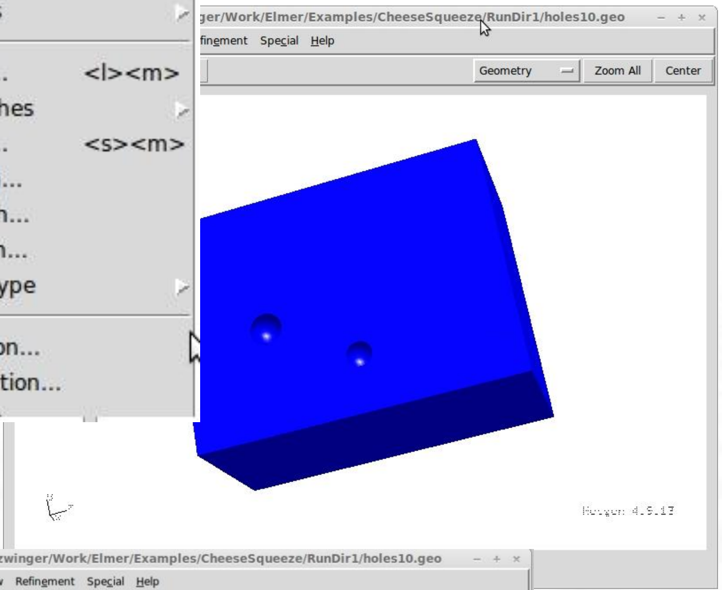
- Creating the geometry:
  - Copy file **`holes.m`** into a new directory
  - Apply the changes (change from 100 to 10 holes)
  - Run **`octave holes.m`**
  - That should create a file named **`holes10.geo`**

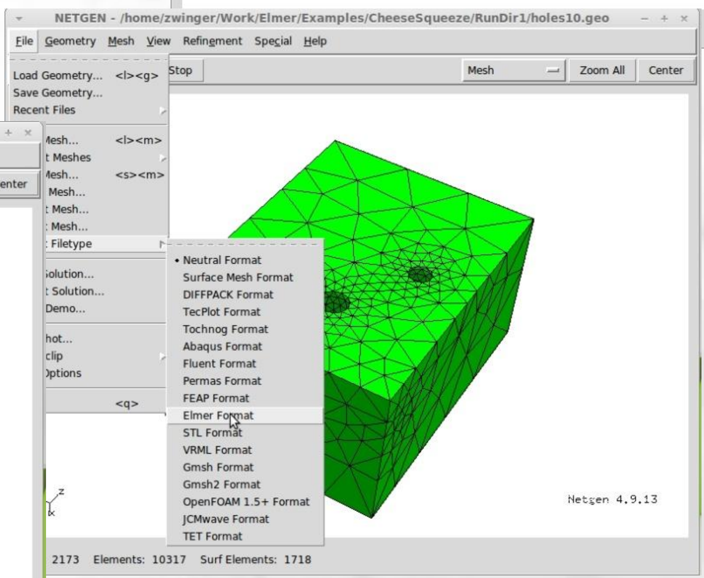


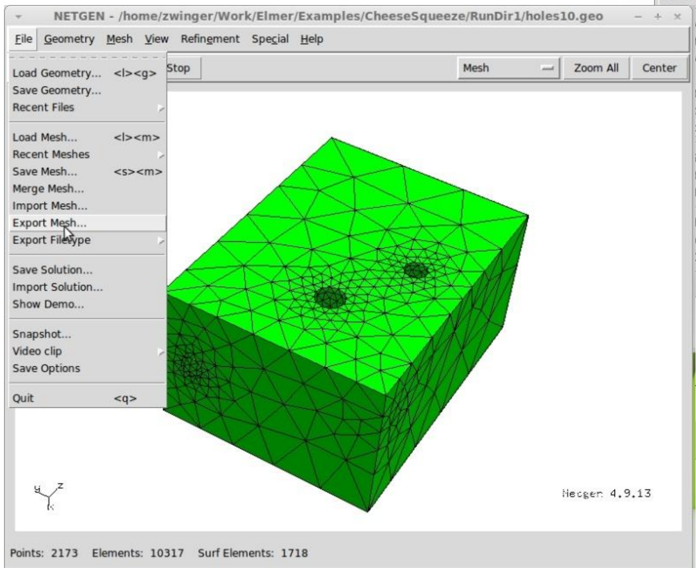
Pre-processing

- Creating the geometry:
  - Start **`netgen`**
  - Load **`holes10.geo`**
  - Click on **Generate Mesh**
  - Choose *Elmer* under **Export Filetype**
  - **Export Mesh**









Pre-processing

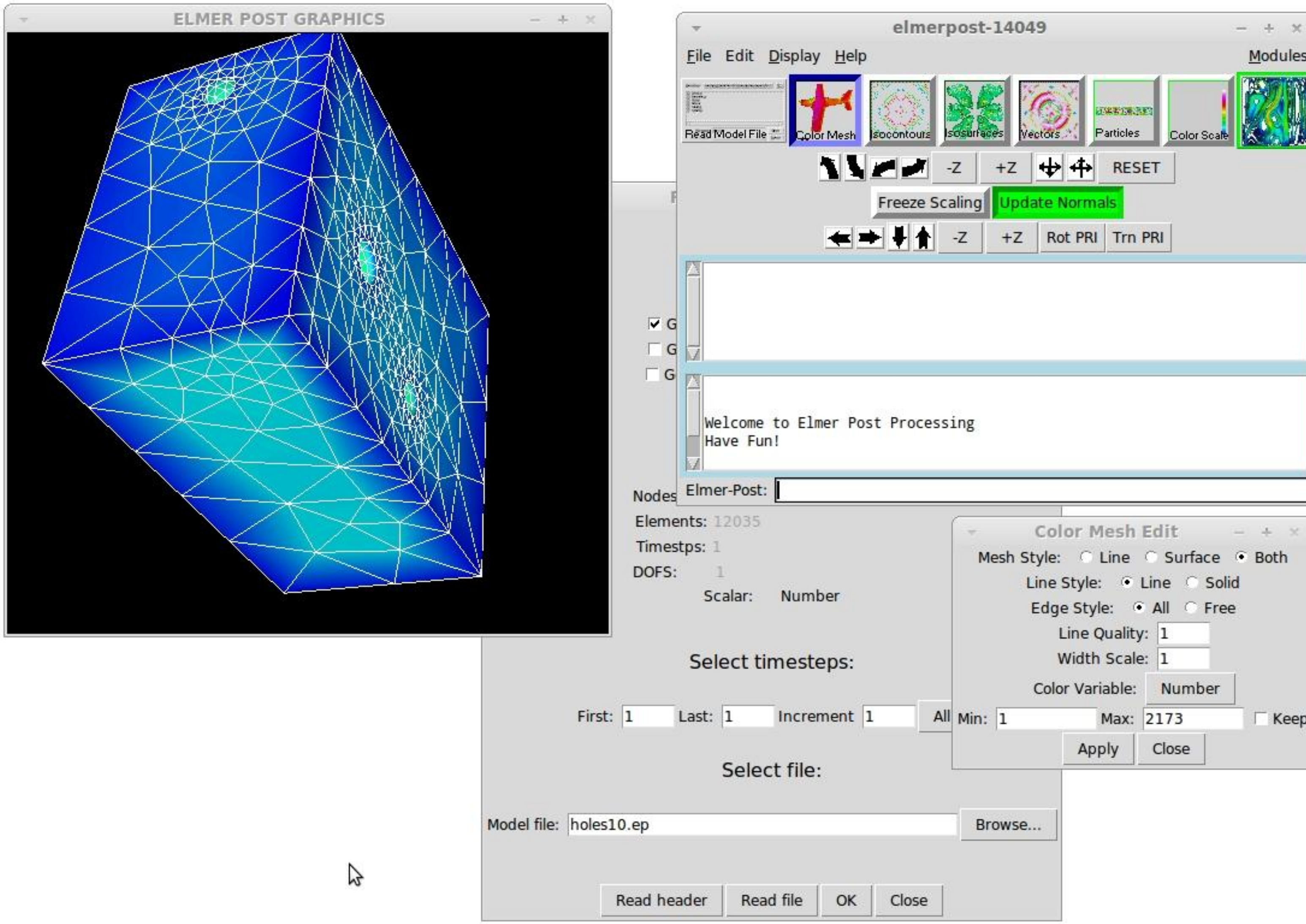
- The mesh is stored in the files

  `mesh.{header,nodes,elements,boundary}`

  - Copy them into a sub-directory **`holes10`**
- Create an ElmerPost output-file:

  `ElmerGrid 2 3 holes10`
- Launch **`ElmerPost`**
  - and load **`holes10.ep`**
  - Open Edit → Grouping … and check the different boundaries
  - Our relevant boundaries are 1-6 (the large sides)
  - Check the dimensions 4 x 4 x 2
    - Too large a piece of cheese, if meters

Pre-processing



Pre-processing

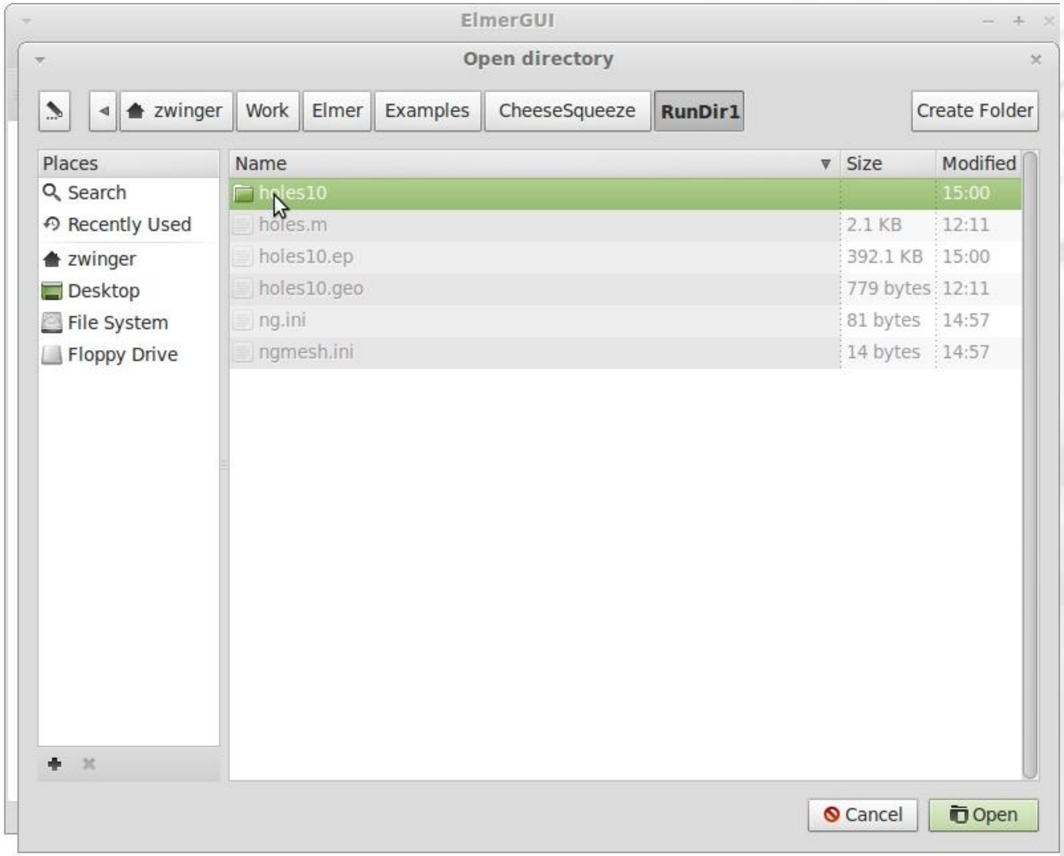
- Elmer does not assume any unit system
- Built-in material library parameters are in SI units
- User has to guarantee consistency between geometry and physical constants
- One possibility: Scale the mesh
  - unit length was 10 cm = 0.1 m

```
ElmerGrid 2 2 holes10 -scale 0.1 0.1 0.1 -out
holes10_scale
```

  - Now the mesh is in SI-units (meters) and the built-in material library (in SI units) could be used
- Here, we are using **internal scaling** provided by Elmer (see later in this tutorial)

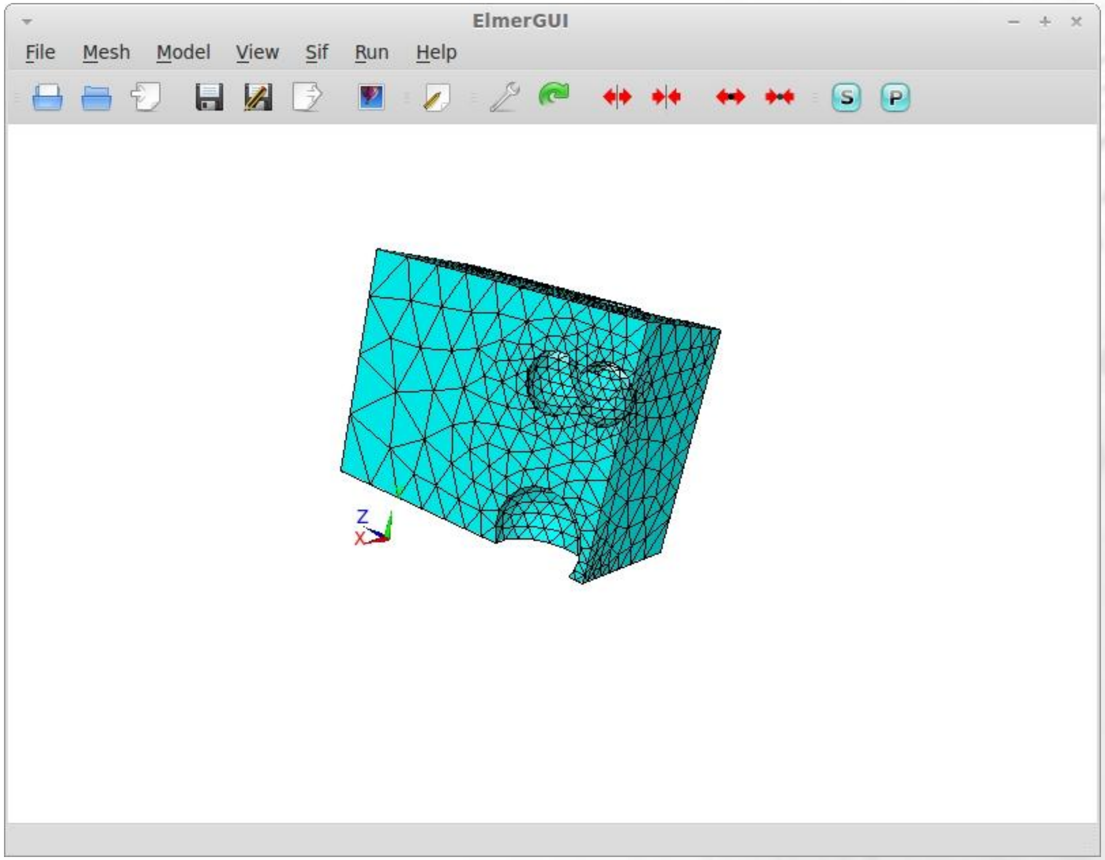
Set-up

- Launch **`ElmerGUI`**
- **File → Load Mesh**: **`holes10`**



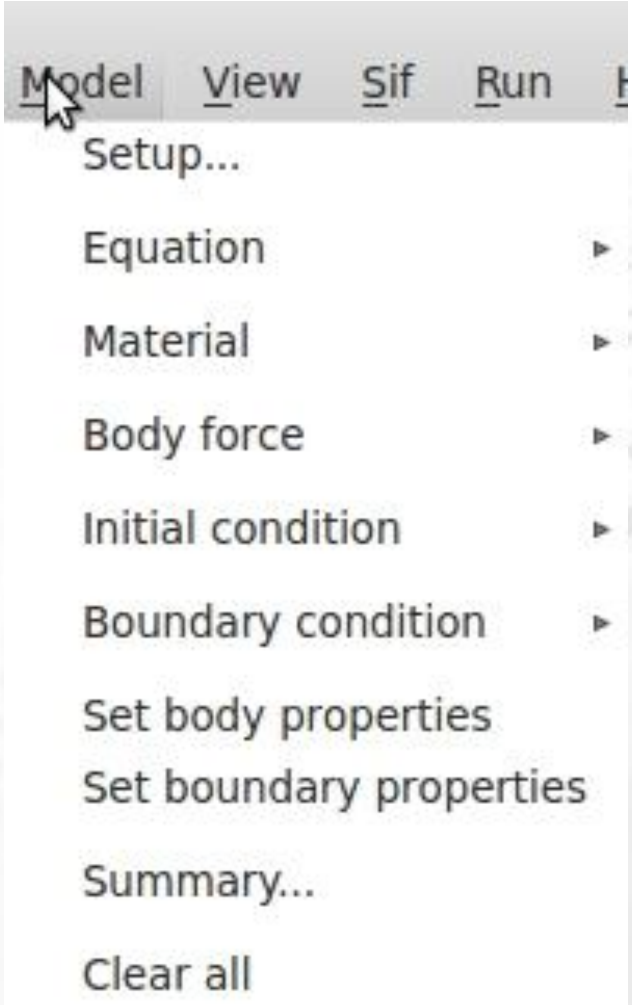
Set-up

- Launch **`ElmerGUI`**
- **File → Load Mesh**: **`holes10`**



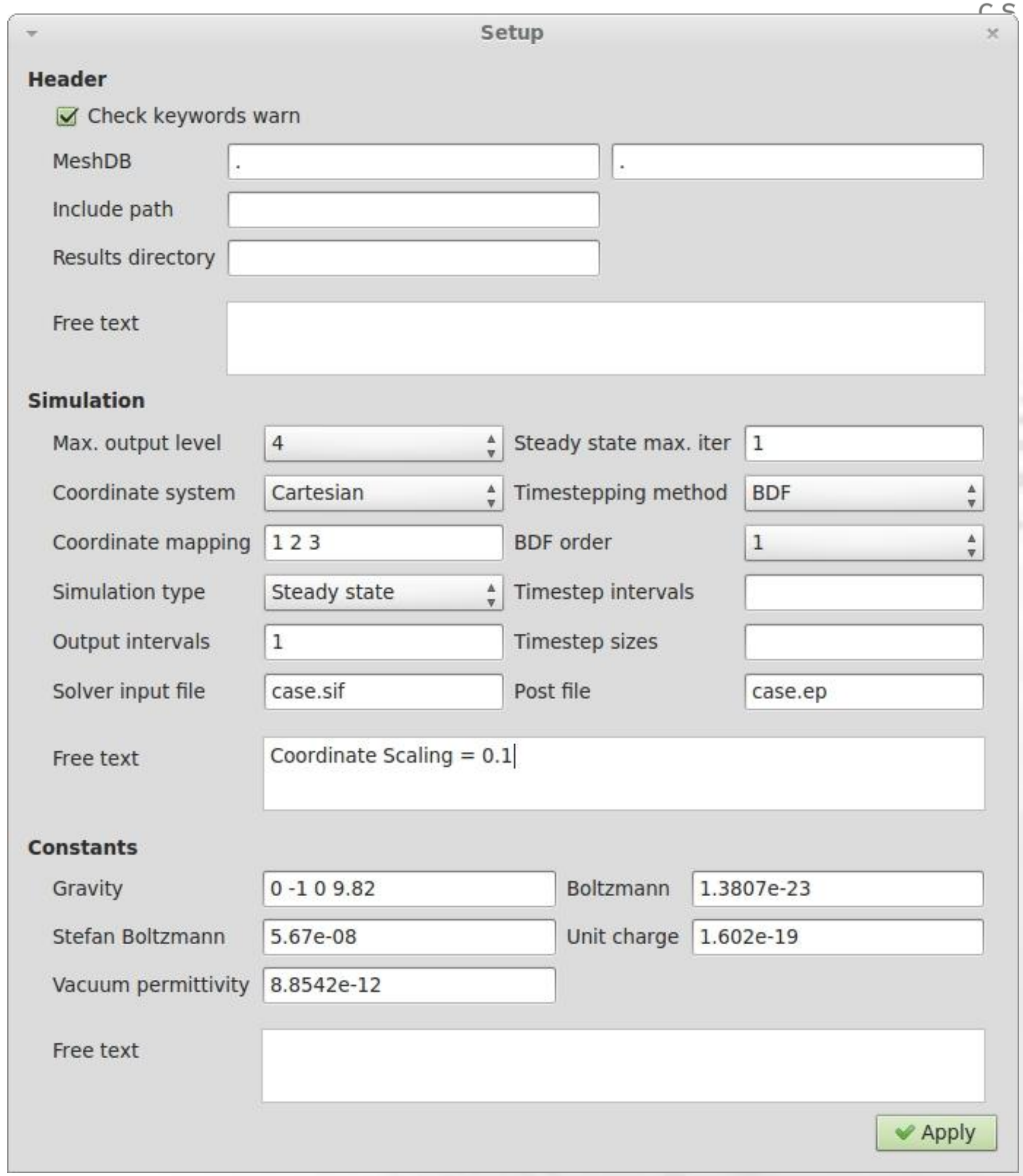
Set-up

- All following steps are sequentially ordered in the menu **Model**
- They reflect the different section in the solver input file that is created for the solver step



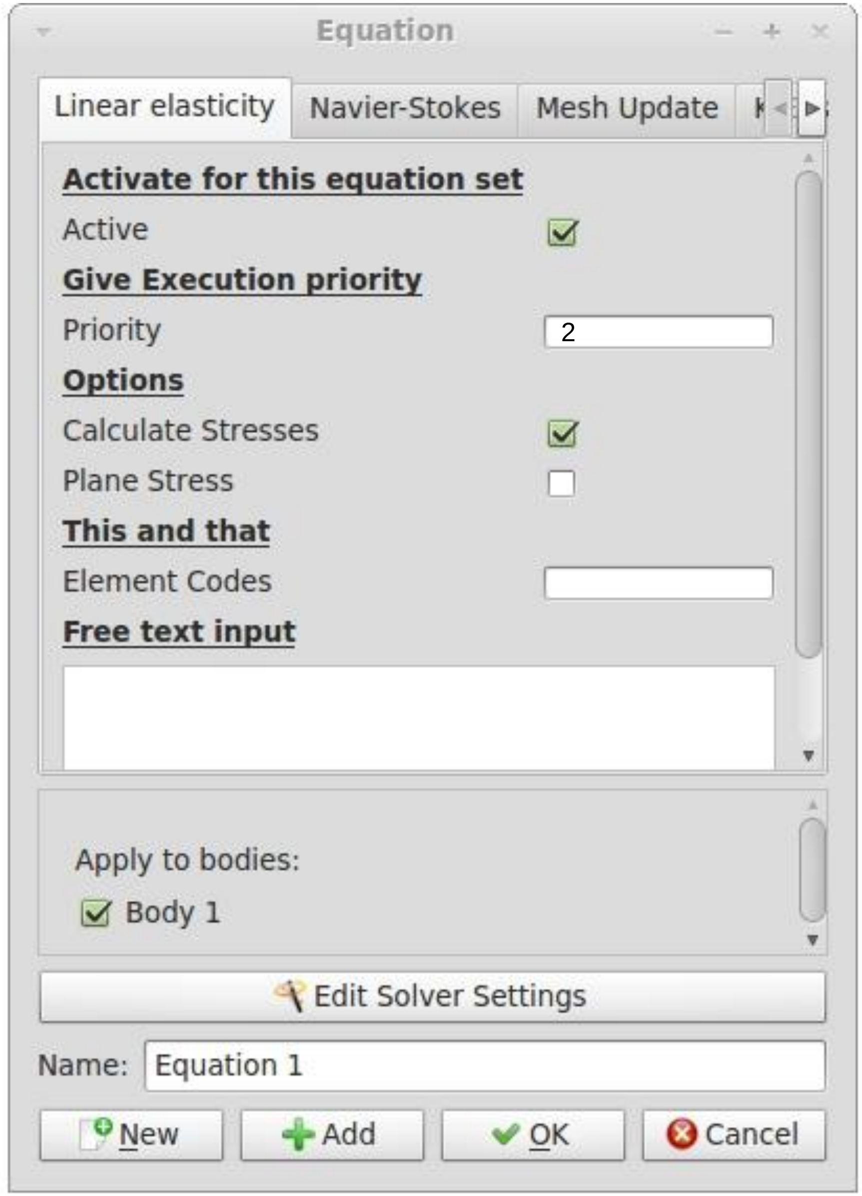
Set-up

- **Model → Setup**
  - Basic setup of the simulation
    - Header for mesh
    - Simulation
    - Constants
  - Possibility to scale input mesh



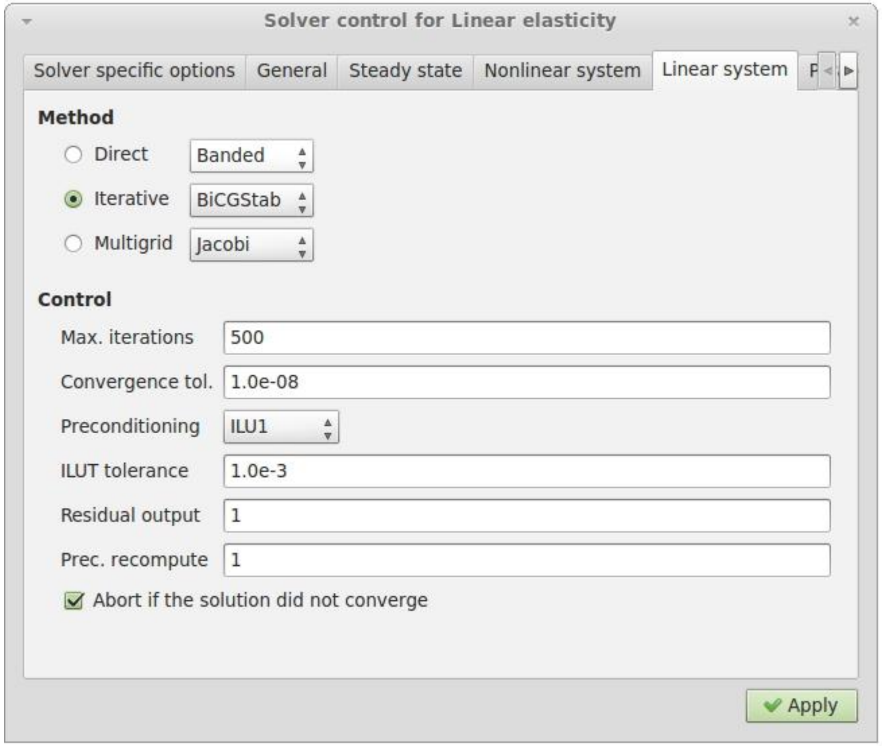
Set-up

- **Model → Equation**
  - definition of physical models (=Solvers) for the simulation
- Toggle **Linear elasticity**
  - **Priority** to **2**
  - **Edit Solver Settings**



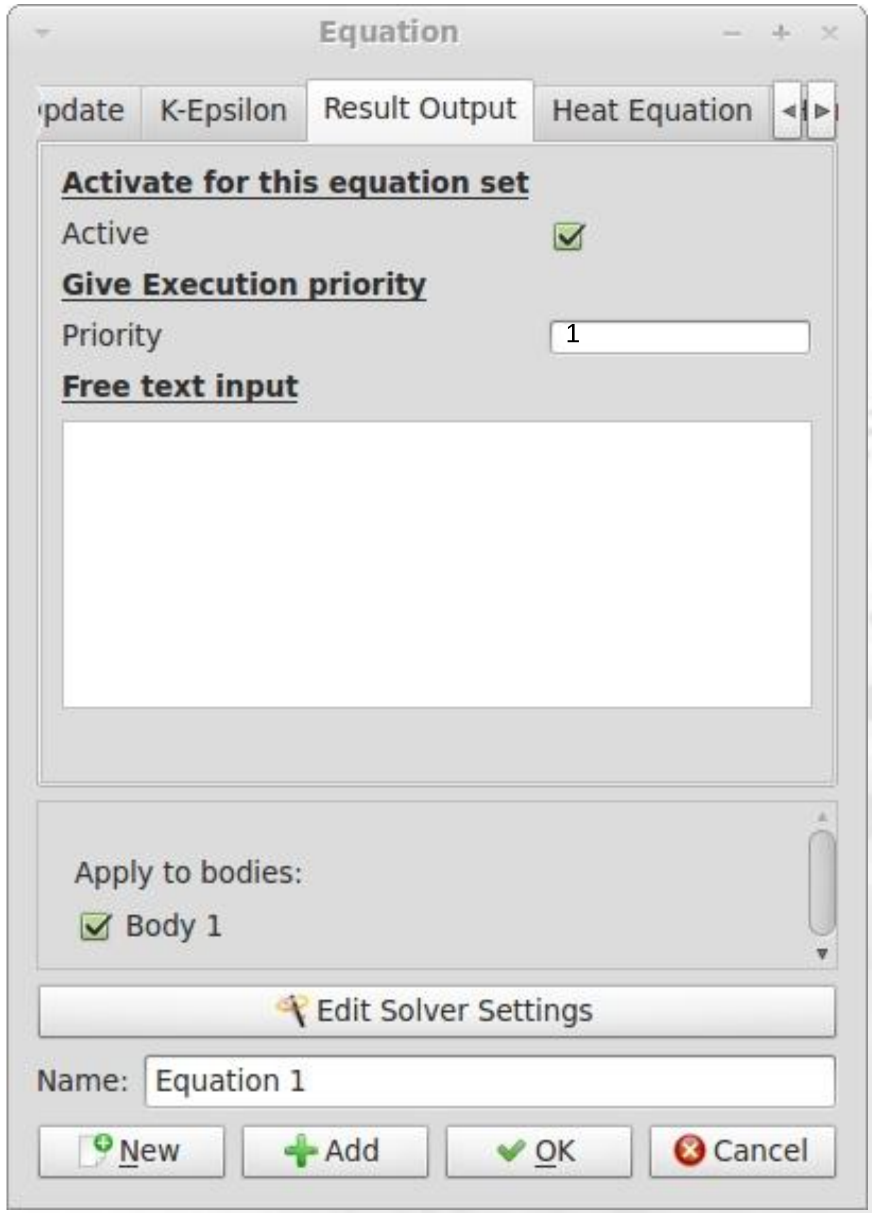
Set-up

- **Linear system**
- **Nonlinear system**:
  - Reduce **Max. Iterations** to 1 (linear problem!)



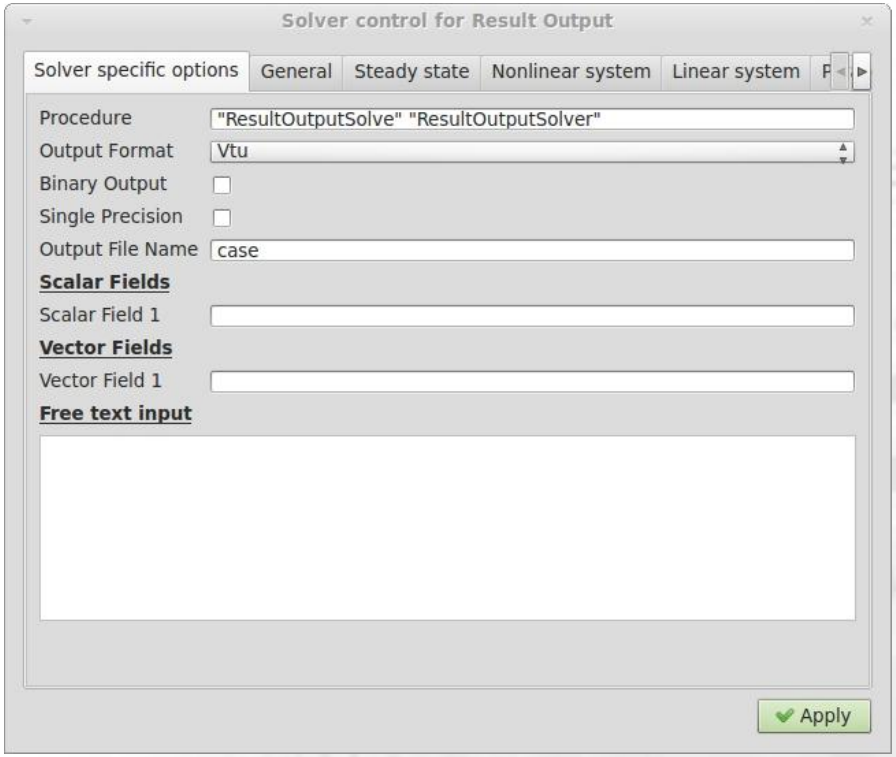
Set-up

- Toggle **Result Output**
  - **Priority** to **1**
  - **Edit Solver Settings**



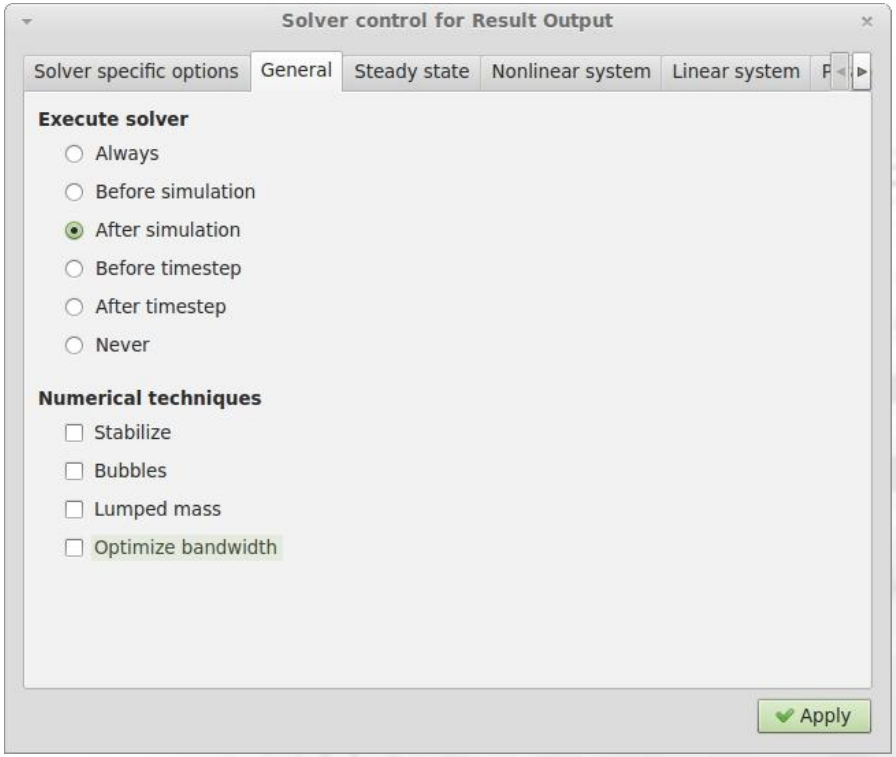
Set-up

- **Linear system**
- **Solver specific options**



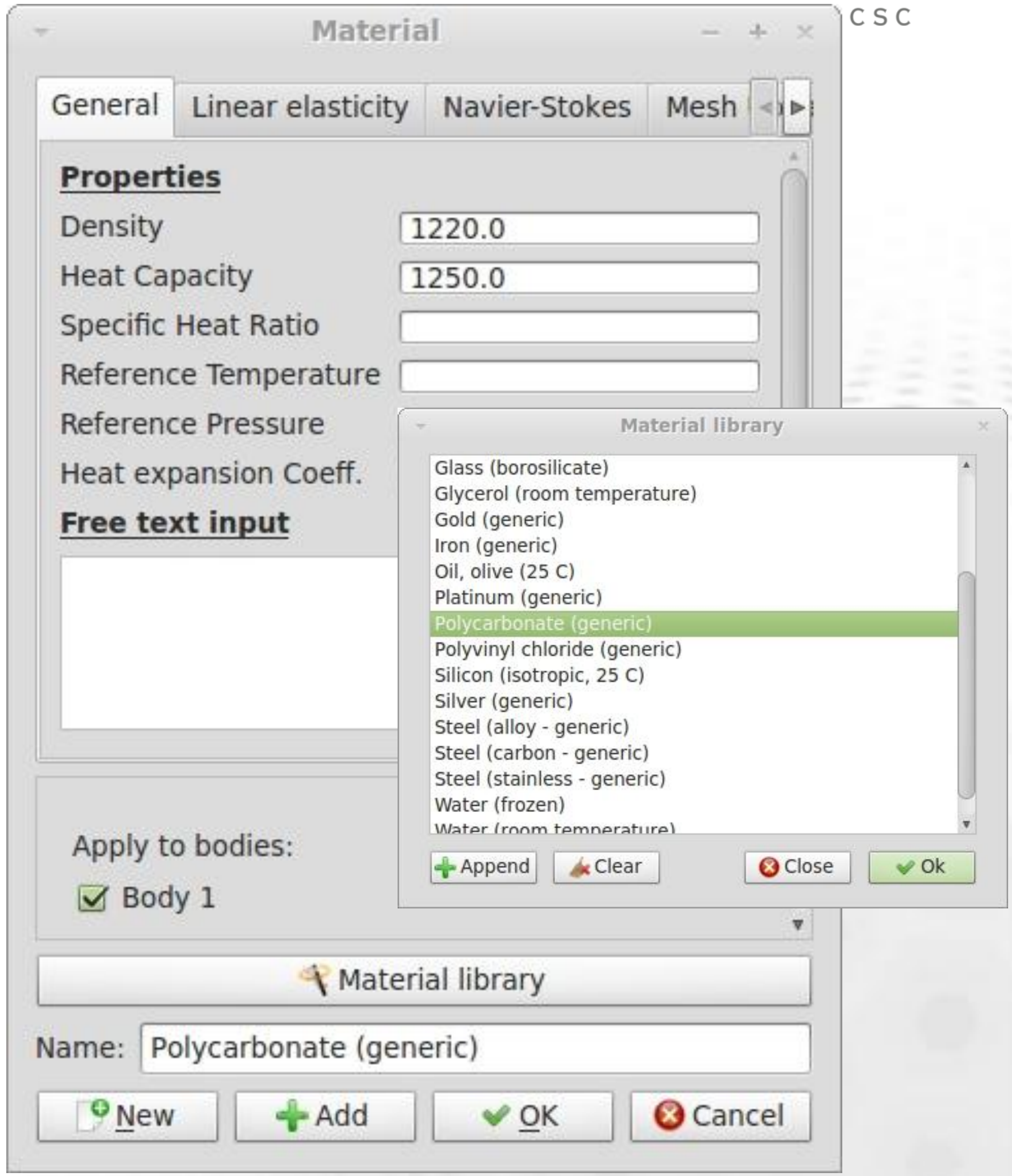
Set-up

- **Linear system**
- **Solver specific options**
- **General:**
  - Toggle **after simulation**
  - Why?: Results to be saved when converged



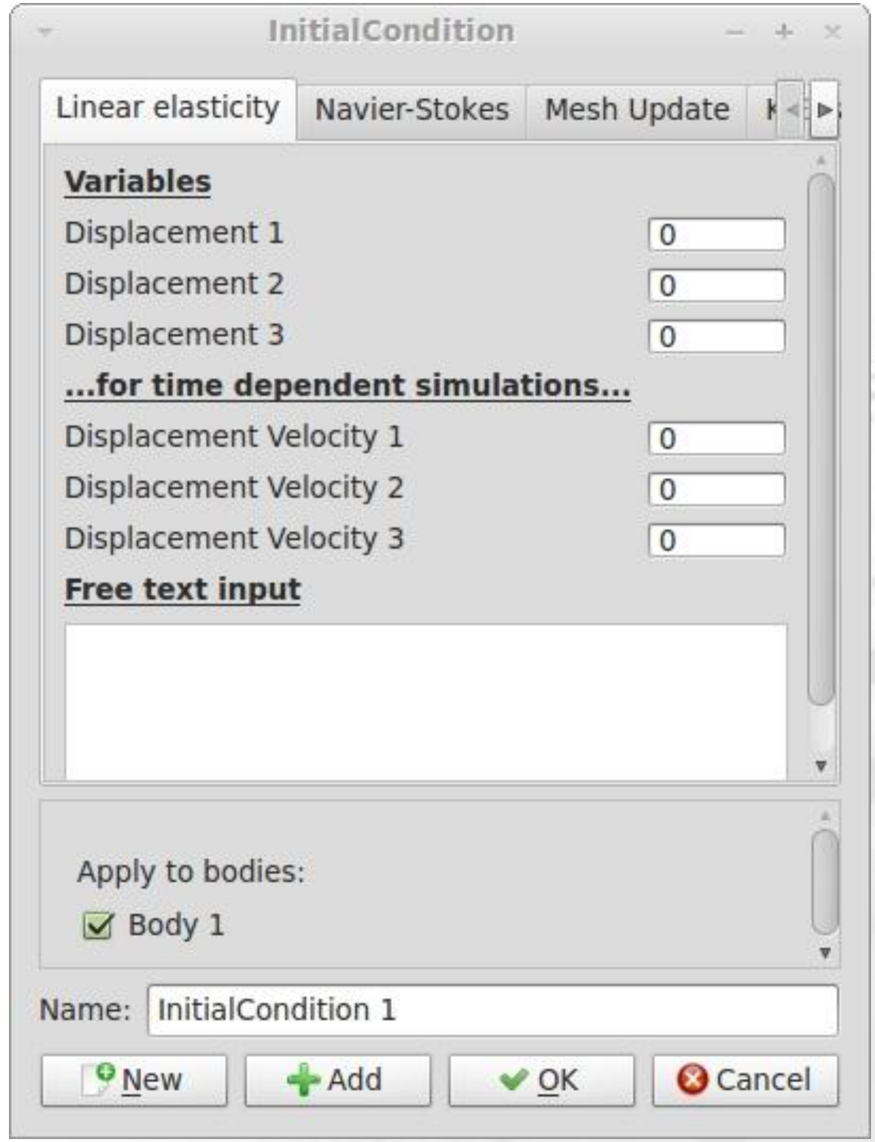
Set-up

- **Model → Material**
  - definition of physical properties for the simulation
- **Material Library**
  - **Polycarbonate (generic)**



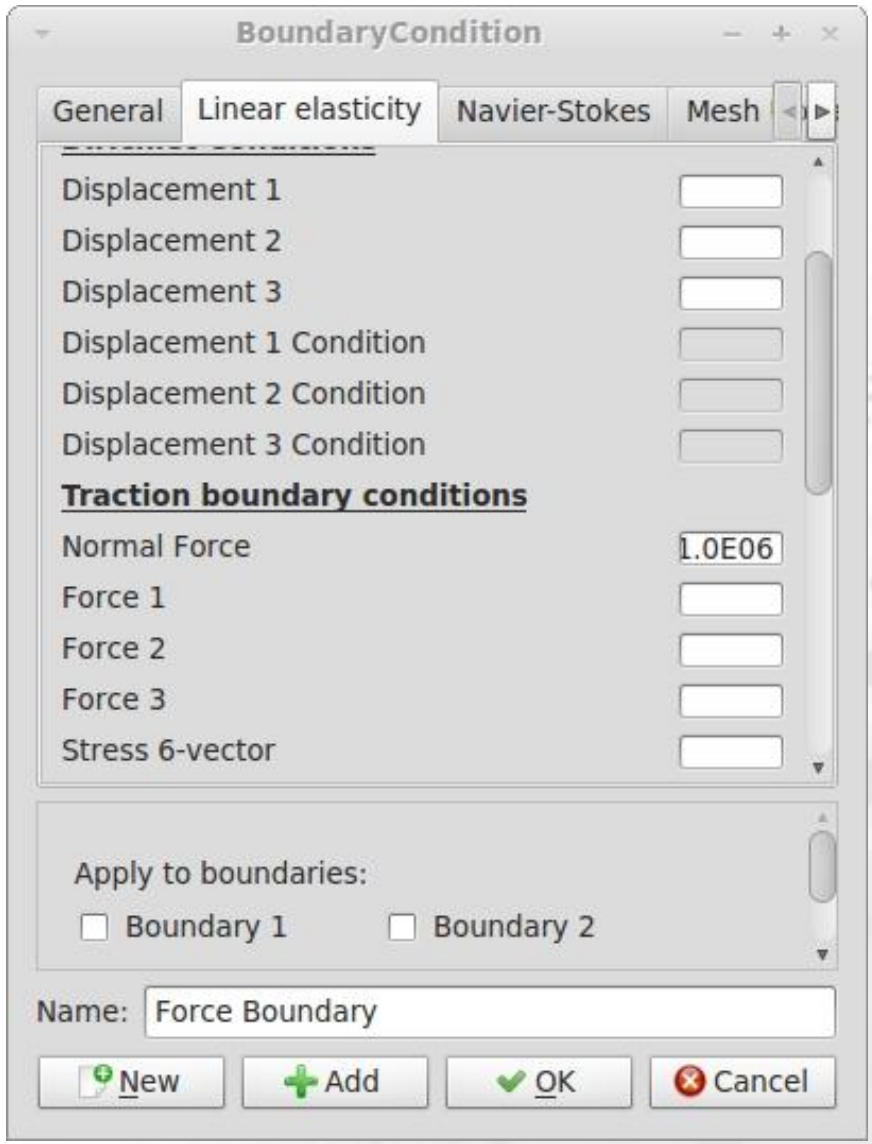
Set-up

- **Model → Initial Condition**
  - initialization of variable values for the simulation
- **Linear elasticity**
  - All variables to zero



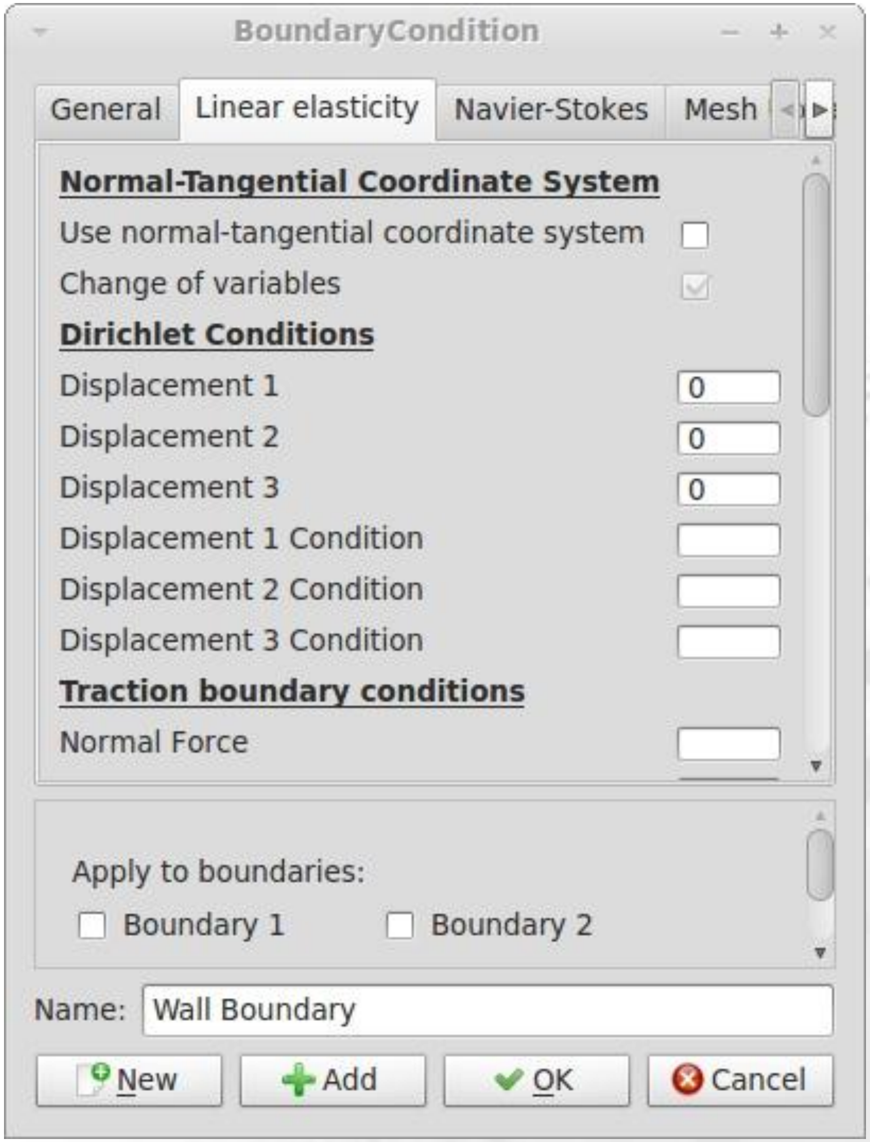
Set-up

- **Model → Boundary Condition**
  - definition of variable values at boundaries for the simulation
  - Usually multiple
  - Different names
- **Linear elasticity**
  - Force boundary
  - **Add** + **New**



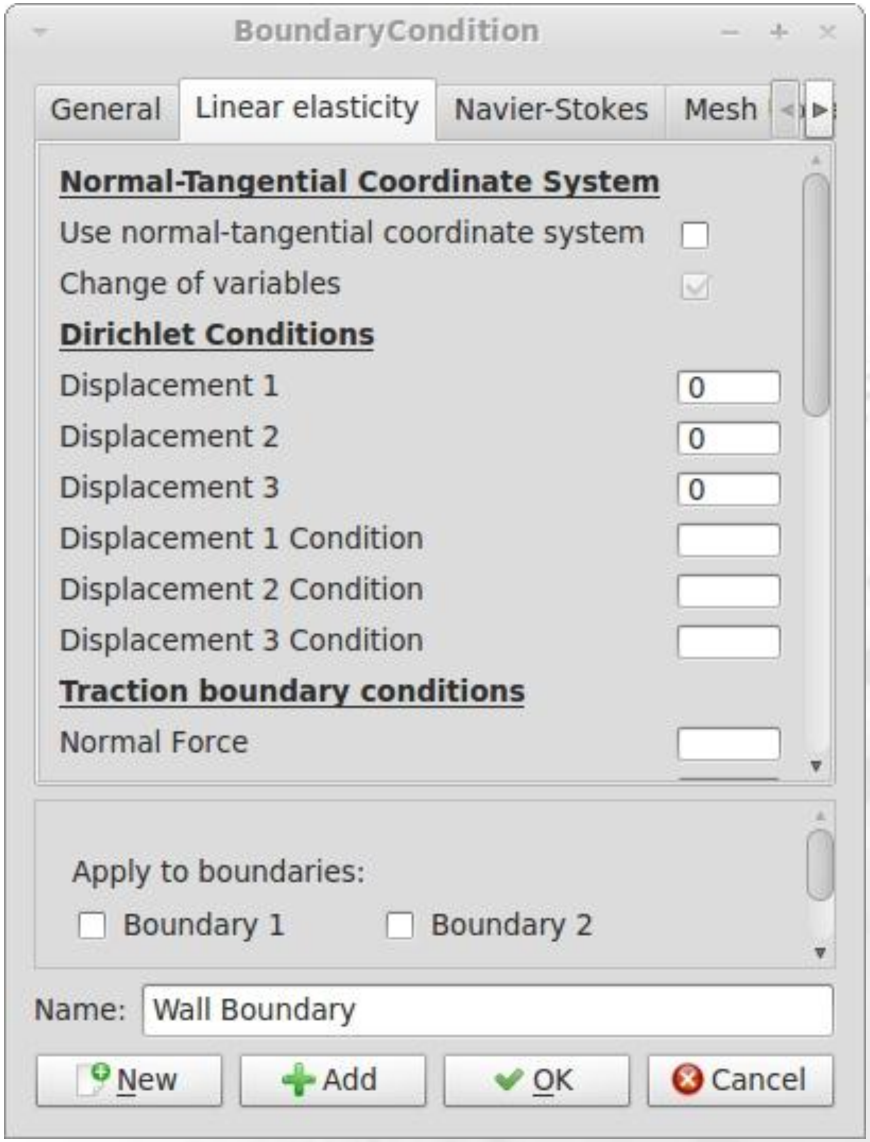
Set-up

- **Linear elasticity**
  - Wall boundary
  - **Add** + **New**



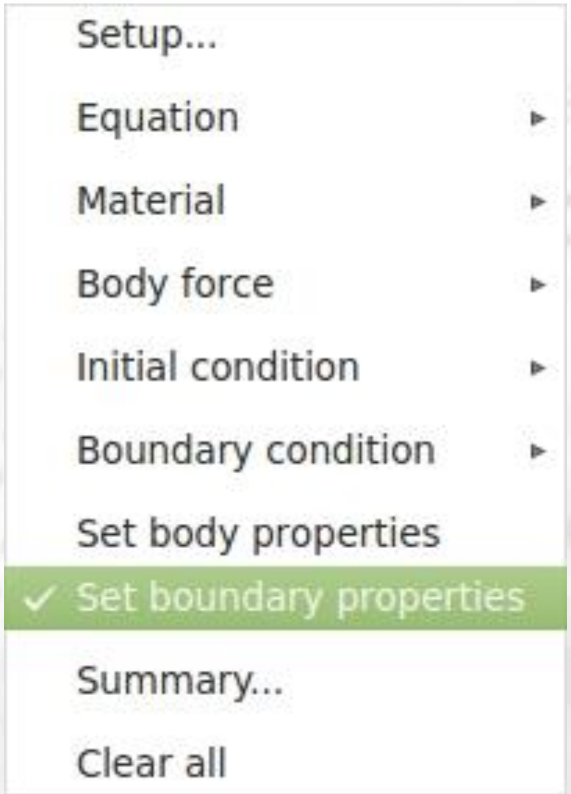
Set-up

- **Linear elasticity**
  - Wall Boundary
  - **Add + OK**



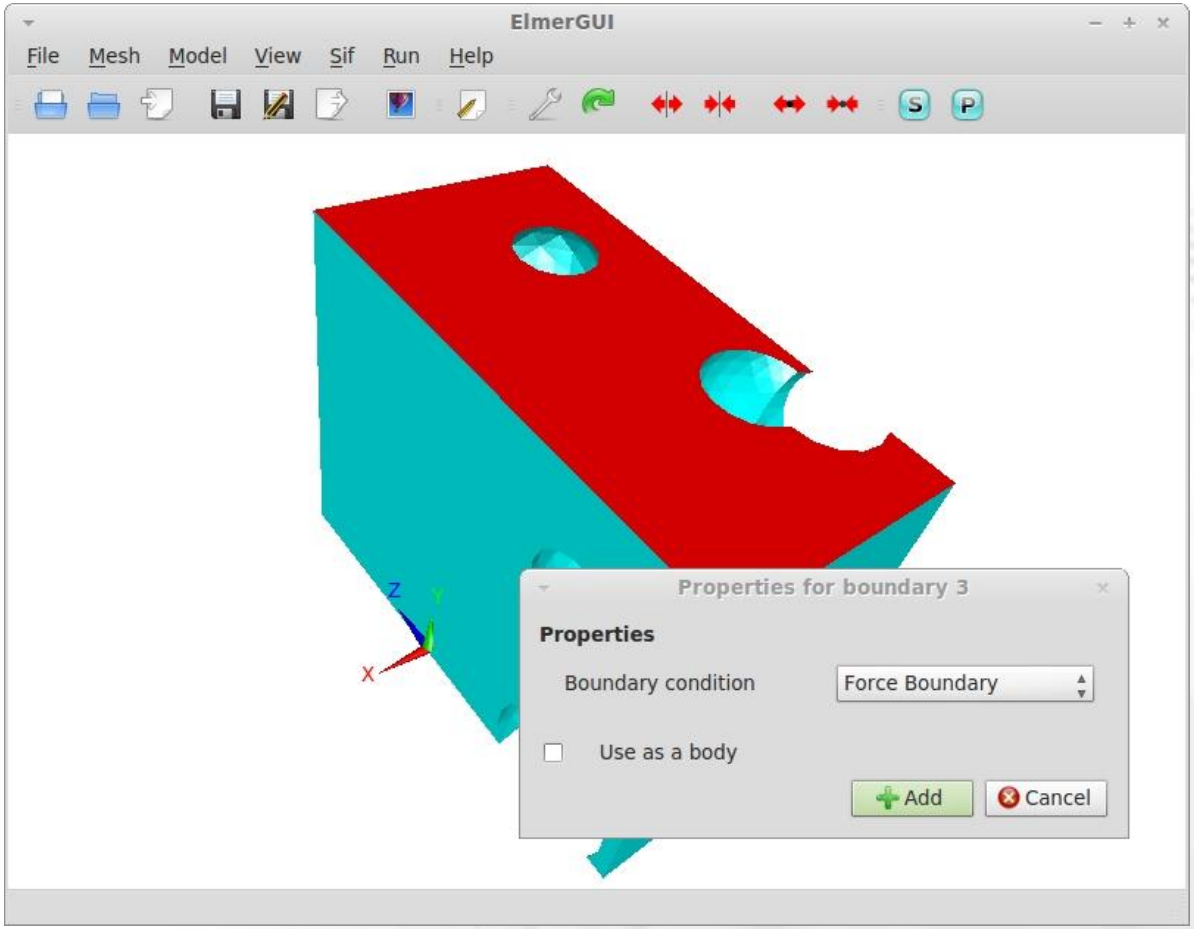
Set-up

- In general it is difficult to know the boundary number in the mesh → assign manually
- **Model → Set boundary properties**
- Then double-klick on specific boundary (gets highlghted)



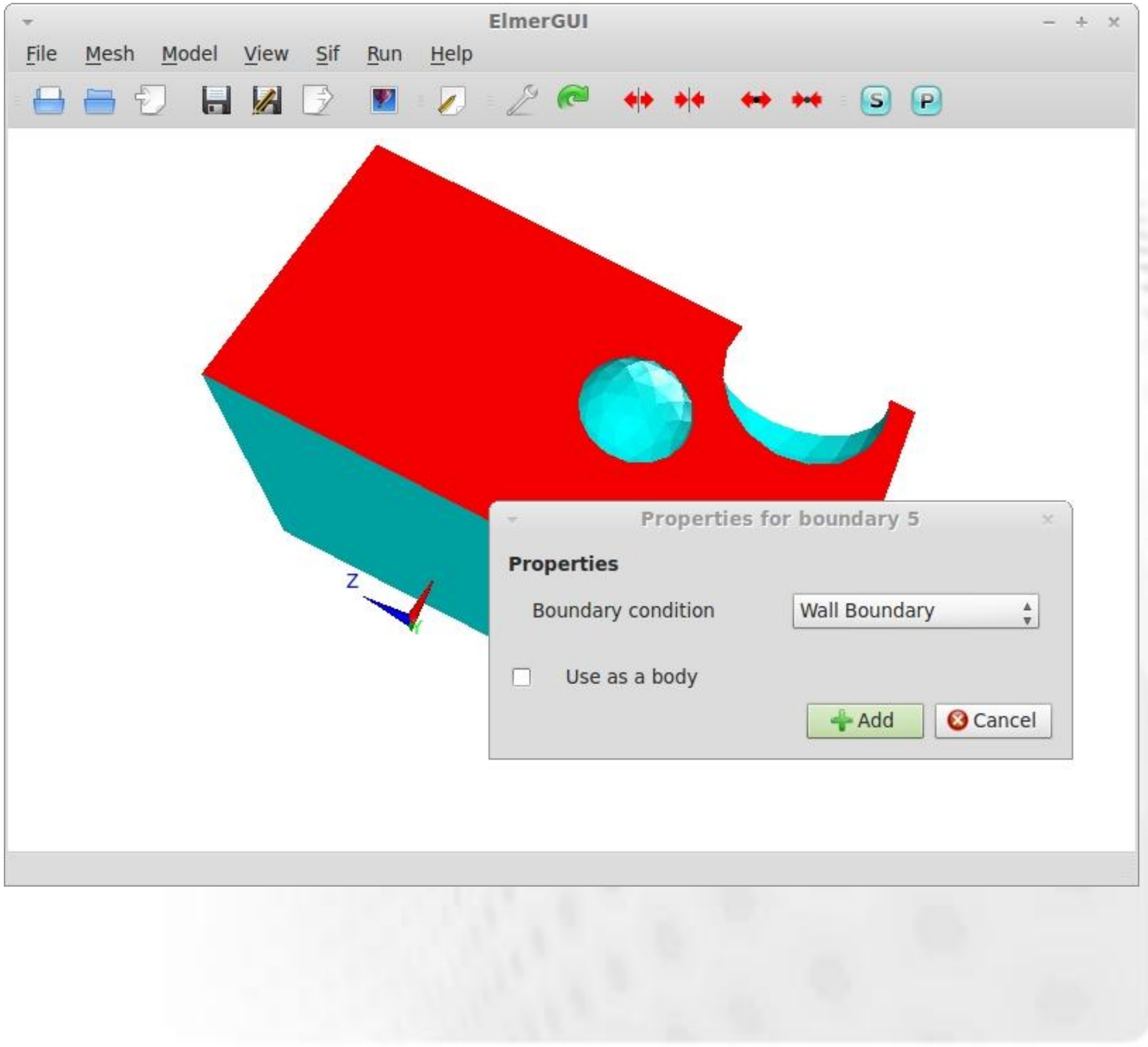
Set-up

- Assigning force boundary
- Press **+Add**
- Rotate to lower boundary



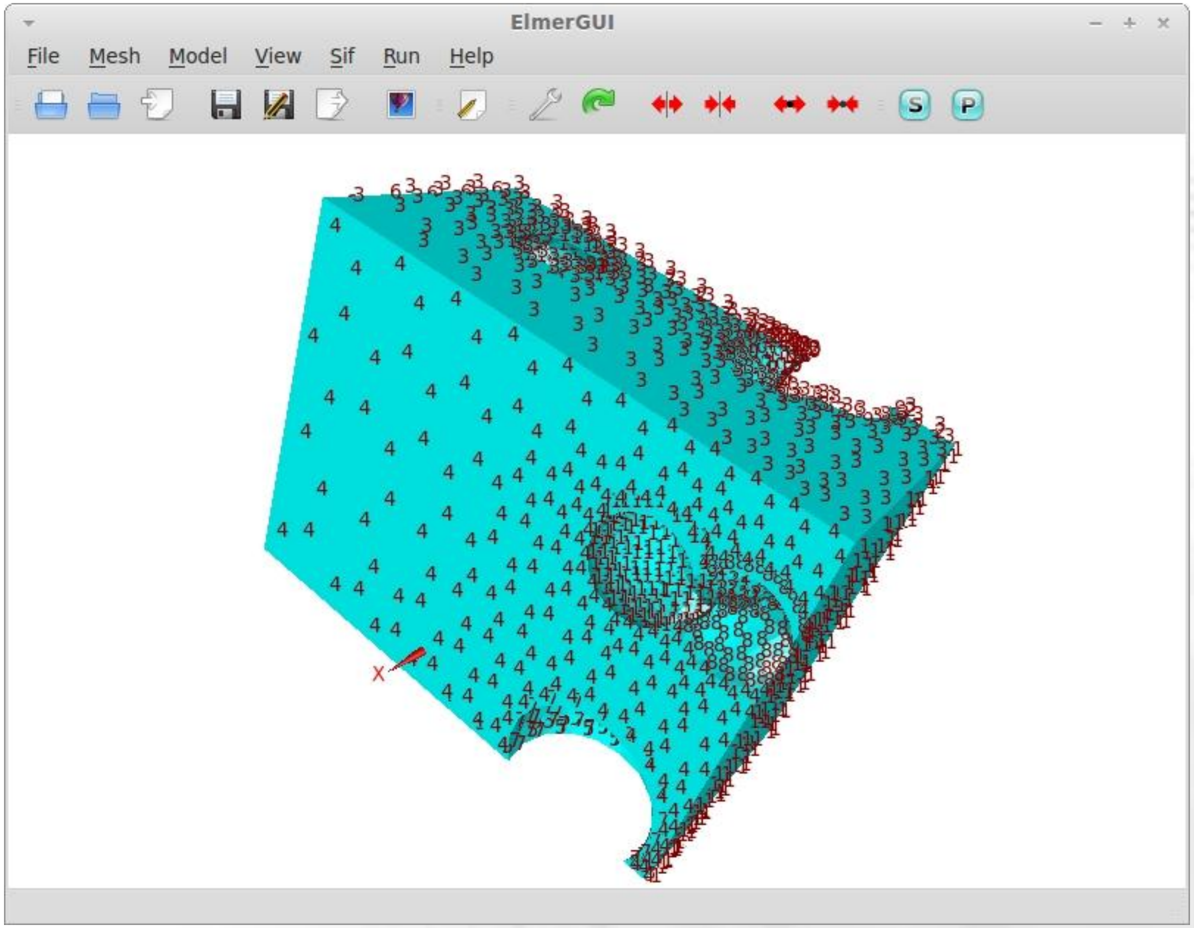
Set-up

- Assigning wall boundary
- Press **+Add**
- Toggle off **Model → Set boundary properties**



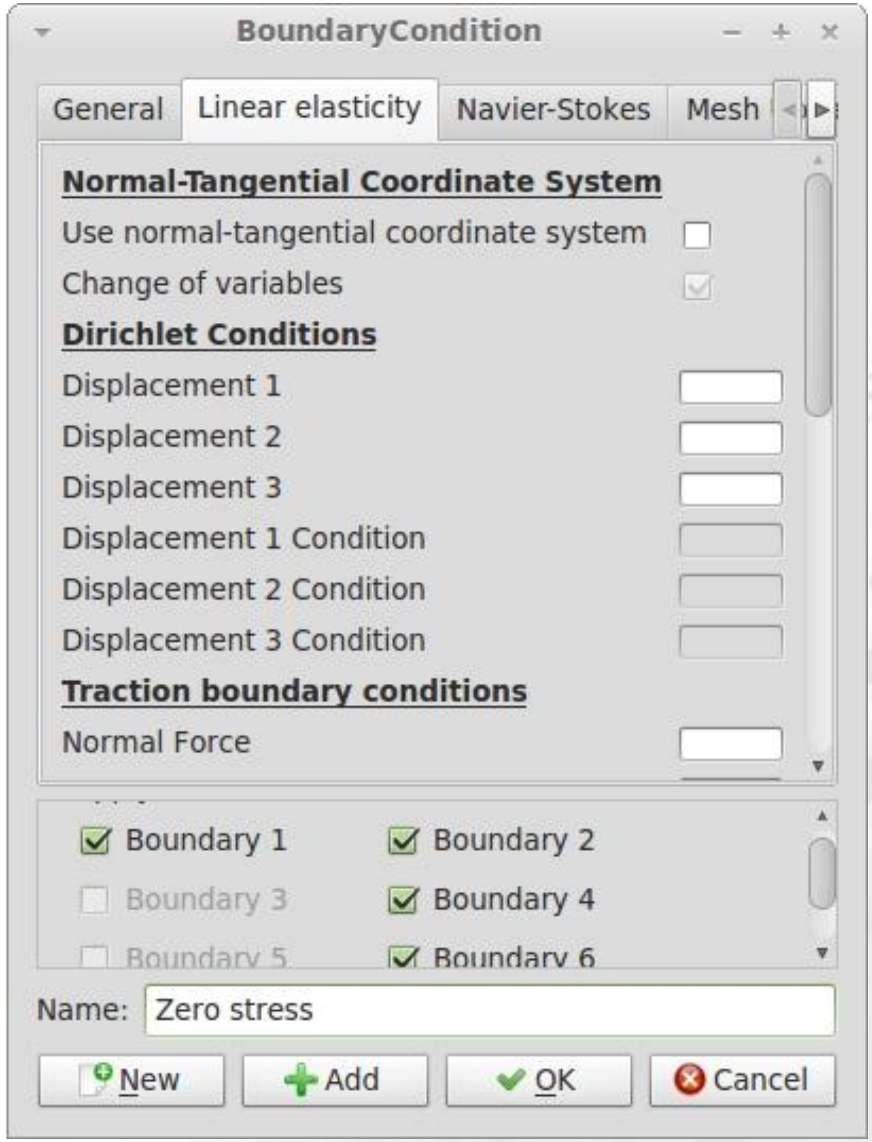
Set-up

- Visualization of boundary ID in ElmerGUI:
  - **View→ Numbering → Boundary index**



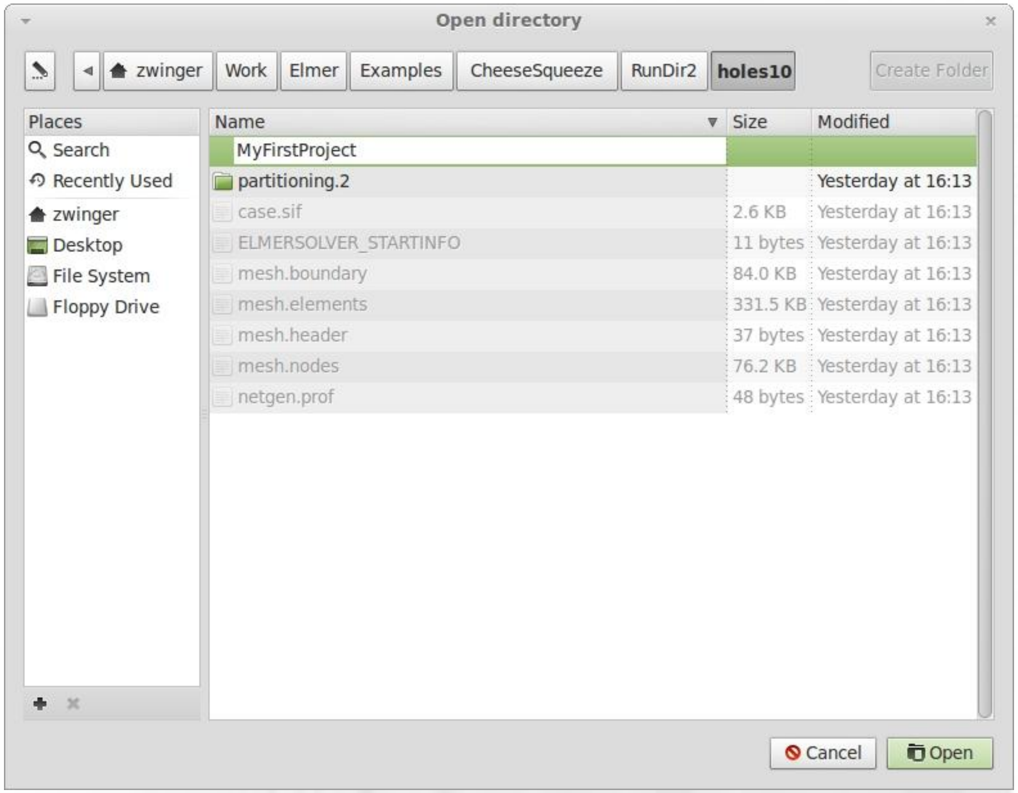
Set-up

- **Model →Boundary condition → Add…**
- **Linear elasticity**
  - Zero stress
  - Toggle remaining boundaries with ID ≤ 6
  - **Add + OK**



Set-up

- Finnishing the setup:
  - **SIF → generate**
  - Save the project:
    - Either by **File → →Save project**
    - Or by the symbol in the task bar
    - Create new folder
  - Save the files:
    - Either by **File →Save**
    - Or by the symbol in the task bar

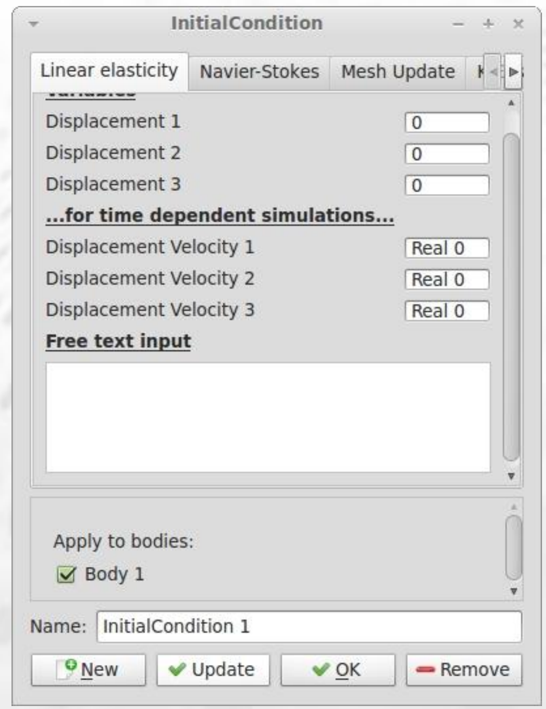


Run

- Run the case:
  - Either **Run→start solver** or
  - The symbol then will change color
  - A log window will occur
  - … and display an error message:

```
ERROR:: Model Input:  Unknown specifier: [0]
ERROR:: Model Input:  In section: [initial condition 1]
ERROR:: Model Input:  For property name:[displacement velocity 3]
```

  - Problem: *Displacement Velocity* not in Keyword Database
  - Re-open Initial Condition 1
  - Cast the values with *Real*
  - **Update** and **OK**
  - **SIF →generate** and

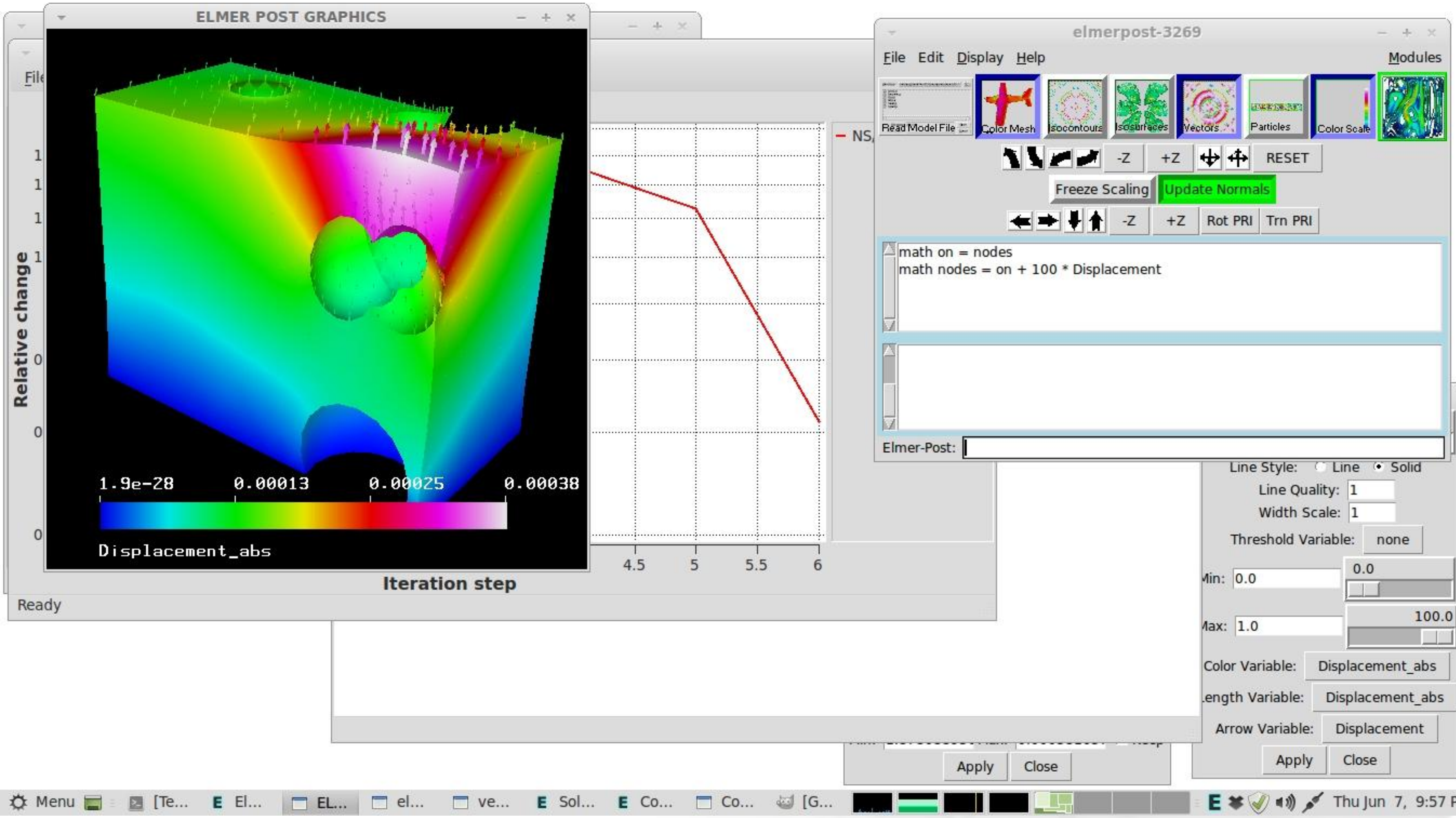


Post-processing

- Two post processors:
    1. Internal VTK based: **Run → Postprocessor (VTK)**
    2. Externally linked ElmerPost (legacy postprocessor of Elmer):
        - Either **Run → Start postprocessor** or 
        - Changes color if active 
- ElmerPost manipulations:

```
math on = nodes
math nodes = on + 100 * Displacement
```

Post-processing



Further steps

- Why is the cheese not squeezed, but pulled?
  - Mind, that surface normal (that defines the direction of normal force) by definition is pointing outwards.
- Task: change force to correct sign and re-run case
  - Remember to create the SIF and save it, before re-running