# Linear Elasticity Tutorial 3:

In this tutorial we showcase the 2D bar problem simulation with one end clamped wile being pulled at the other end. Body force is neglected and the non clamped ends pull is approximated with Dirichlet displacement $u_1 = 1$. If this simulation is compared to the previous one from tutorial 1 ans tutorial 2, the only difference now is that no body force is applied and an additional Dirichlet condition is applied at the free end of the bar. Here is how PSD simulation of this case can be performed. he bar $5 \times 1$ m$^2$ in area is made up of material with $\rho = 8 \times 10^3$, $E = 200 \times 10^9$, and $\nu = 0.3$.

## Step 1: Preprocessing

First step in a PSD simulation is PSD preprocessing , at this step you tell PSD what kind of physics, boundary conditions, approximations, mesh, etc are you expecting to solve.

In the terminal `cd` to the folder `\home\PSD-tutorials\linear-elasticity` . Launch `PSD_PreProcess` from the terminal, to do so run the following command.

```
PSD_PreProcess -problem linear-elasticity -dimension 2 -dirichletconditions 2 \
-postprocess u
```

After the `PSD_PreProcess` runs successfully you should see many `.edp` files in your current folder.

*What do the arguments mean ?* `-problem linear-elasticity` means that we are solving linear elasticity problem, `-dimension 2` means it is a 2D simulation; `-dirichletconditions 2` says we have two Dirichlet border; and `-postprocess u` means we would like to have ParaView post processing files.

In comparison to preprocessing from other two tutorials (tutorial 1 and 2), notice that the body force flag `-bodyforceconditions 1` is missing. This is due to the fact that for this problem we assume null body force. `-dirichletconditions 2`, which notifies to PSD that there are two Dirichlet borders in this simulation i) the clamped end and ii) the pulled ends of the bar. To provide these Dirichlet conditions of the two ends in `ControlParameters.edp` set the variables `Dbc0On 2`, `Dbc0Ux 0.`, and `Dbc0Uy 0.` signifying the clamped end ($u_x = 0, u_y = 0$ on mesh label 2) and `Dbc1On 4`, `Dbc1Ux 1.`, and `Dbc1Uy 0.` signifying the pulled end ($u_x = 1, u_y = 0$ on label `4`). Note that here at border `4` we have explicitly set $u_2 = 0$ this means the bar is not allowed to shrink (compress) in $y$ direction, however you might wish to allow the bar to compress. For such a simulation simply use `Dbc1On 4` and `Dbc1Ux 1.`, and remove the term `Dbc1Uy 0.` therefor asking PSD not to apply constrain in $y$ direction on the pulled end.

Just like the previous tutorial the input properties $E, \nu$ should be mentioned in `ControlParameters.edp`, use `E = 200.e9`, and `nu = 0.3;`. The volumetric body force condition is mentioned in the same file via variable `Fbc0Fy -78480.0`, i.e ( $\rho * g = 8.e3 * (-9.81) = -78480.0$). One can also provide the mesh to be used in `ControlParameters.edp` , via `ThName = "../Meshes/2D/bar.msh"` (*note that mesh can also be provided in the next step*) .In addition variable `Fbc0On 1` has to be provided in order to indicate the volume (region) for which the body force is acting, here `1` is the integer volume tag of the mesh.

## Step 2: Solving

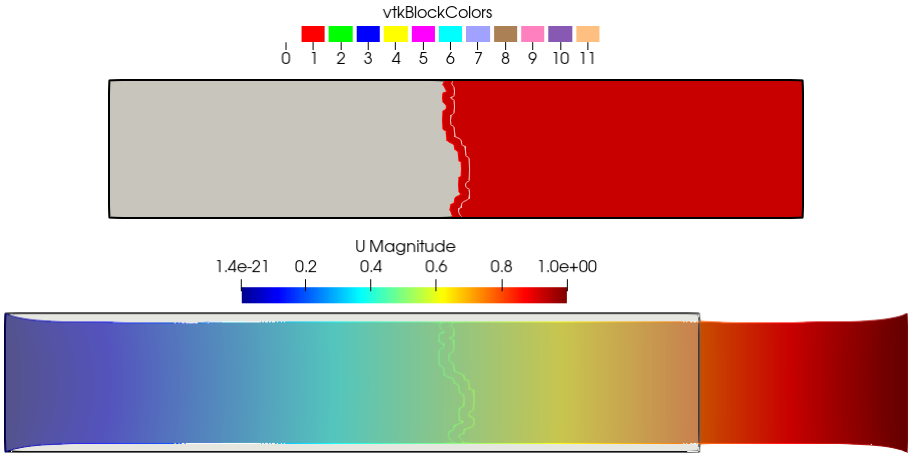
As PSD is a parallel solver, let us use 2 parallel processes to solve this 2D bar case. To do so enter the following command:

```
PSD_Solve -np 2 Main.edp -mesh ./../Meshes/2D/bar.msh -v 0
```

Here `-np 2` denote the argument used to enter the number of parallel processes (MPI processes) used while solving. `-mesh ./../Meshes/2D/bar.msh` is used to provide the mesh file to the solver. `-v 0` denotes the verbosity level on screen. `PSD_Solve` is a wrapper around `FreeFem++` or `FreeFem++-mpi`. Note that if your problem is large use more cores. PSD has been tested upto 13,000 parallel processes and problem sizes with billions of unknowns, surely you will now need that many for the 2D bar problem.

## Step 3: Postprocessing

PSD allows postprocessing of results in ParaView. After the step 2 mentioned above finishes. Launch ParaView and have a look at the `.pvd` file in the `VTUs_DATE_TIME` folder.




You are all done with your 2D linear-elasticty simulation.

# What else should you try to become an advanced user

- On the non clamped Dirichlet condition try removing the constrain on $y$ direction `Dbc1Uy 0.` simply comment this line. Rerun the problem and compare to the results above.
- Try running the 3D problem. Keep in mind to rerun the `PSD_PreProcess` with `-dimension 3` flag and using the appropriate mesh via `-mesh` flag with `-PSD_Solve` . It goes without saying you will need to adjust the Dirichlet border labels in `ControlParameters.edp`.
- Optionally try using `-withmaterialtensor` flag with `PSD_PreProcess` , and run the simulation. You are encouraged to have a look at `ControlParameters.edp` and `VariationalFormulations.edp` file produced with `-withmaterialtensor` flag and without this flag.
- Add `-sequential` flag to `PSD_PreProcess` for sequential solver, but remember to use `PSD_Solve_Seq` instead of `PSD_Solve` and no `-np` flag.

```
PSD_PreProcess -problem linear-elasticity -dimension 2 -sequential \
-bodyforceconditions 1 -dirichletconditions 2 -postprocess u
```

```
PSD_Solve_Seq Main.edp -mesh ./../Meshes/2D/bar.msh -v 0
```

- You are encouraged to time your the PSD solver and see if you have considerable gains when using more processes in parallel PSD or when comparing a a sequential solver with a parallel one. To time the solver use `-timelog` flag during `PSD_PreProcess`.
- You are encouraged to use more complex meshes for this same problem, but do not forget to update the `ControlParameters.edp` file.