Linear Elasticity Tutorial 2D bar problem clamped at one end wile being pulled at the other end (Dirichlet–Neumann case)

Mohd Afeef Badri

Abstract

This document details a single tutorials of 'linear elasticity' module of PSD in a more verbos manner.

Similar simulation, as in the previous tutorial is presented in this section. We showcase the 2D bar problem simulation with one end clamped wile being pulled at the other end. Just like the previous simulation the body force is neglected. However, now the non clamped ends pull is approximated with Neumann force (aka. traction force) $\int_{\partial\Omega^h_{\mathrm{N}}}(\mathbf{t}\cdot\mathbf{v}^h)$. To simulate the pull we assume traction vector $\mathbf{t}=[t_x,t_y]=[10^9.,0]$ acting on the non clamped right end of the bar, i.e., force in x direction is 10^9 units. Here is how PSD simulation of this case can be performed. The same problem from previous tutorials 1 and 2 is used here, a bar 5 m in length and 1 m in width, and is supposed to be made up of a material with density $\rho = 8\times10^3$, Youngs modulus $E = 200\times10^9$, and Poissons ratio $\nu = 0.3$.

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 1 -tractionconditions 1 -postprocess u
```

the comandline flag -dirichletconditions 1, notifies to PSD that there is one Dirichlet border —the clamped end of the bar— in this simulation. And the flag -tractionconditions 1 notifies to PSD that there is one traction border —the right end of the bar— in this simulation.

To provide the clamped boundary condition $(u_1 = 0, u_2 = 0)$ set the variables Dbc0On 2, Dbc0Ux 0., and Dbc0Uy 0. in ControlParameters.edp. In the same file traction boundary conditions are provided via the variables Tbc0On 4 and Tbc0Tx 1.e9, which mean apply traction force $\mathbf{t}=[t_x,t_y]=[10^9.,0]$ on label number 4 (right) of the mesh. If user wishes to add traction force ,for instance $t_y = 100.$, simply add the missing macro macro Tbc0Tx 1.e9 //.

Step 2: Solving

Let us now use 5 cores to solve this problem. To do so enter the following command:

```
PSD_Solve -np 5 Main.edp -mesh ./../Meshes/2D/bar.msh -v 0
```

Notice, that this is the exact same command used in solving the previous bar problems from other sections, with only difference that we now use -np 5.

Note that for this simple problem, the bar mesh (bar.msh) has been provided in ../Meshes/2D/" folder, this mesh is a triangular mesh produced with Gmsh. Moreover detailing meshing procedure is not the propose of PSD tutorials. A user has the choice of performing their own meshing step and providing them to PSD in .msh[1] or .mesh format, we recommend using Salome or Gmsh meshers for creating your own geometry and meshing them.

Step 3: Postprocessing

Launch ParaView and have a look at the .pvd file in the PSD/Solver/VTUs_DATE_TIME folder.

[1]Please use version 2

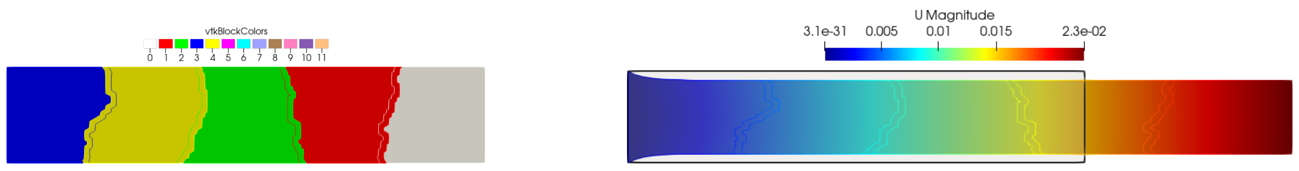

Figure 1: 2D bar results. Partitioned mesh (left) and 100X warped displacement field (right).

Note now in fig. 1 there are five subdomains in the partitioned mesh since five cores were used. Contrary to previous tutorial, as expected, we see that the right end of the bar which is being pulled now contract in y direction. This is due to the fact that there is no Dirichlet condition at this end now.