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

Mohd Afeef Badri

Abstract

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

Similar simulations, 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. Contrary to simulation in the previous tutorial, the clamped end just restricts x movement, i.e, $u_x = 0$. Just like simulation from the previous tutorial the body force is neglected. Just like simulation in the previous tutorial , the non clamped ends pull is approximated with Neumann 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. 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 $\nu = 0.3$. Here is how PSD simulation of this case can be performed.

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 \
-dirichletpointconditions 1 -postprocess u
```

Additional flag -dirichletpointconditions 1 now appears, this notifies to PSD that there is one Dirichlet point boundary condition. Edit the ControlParameters.edp to communicate the desired point boundary conditions, set the variables Pbc0Ux 0. and Pbc0Uy 0. to specify $u_x = 0, u_y = 0$, and variable PbcCord = [[0. , 0.]]; to specify the point coordinates $(x, y) = (0, 0)$. Via the flags we specified that -dirichletconditions 1, i.e., there is one Dirichlet border. To provide the Dirichlet condition ($u_x = 0$) set the variables Dbc0On 2 and Dbc0Ux 0. in ControlParameters.edp. PSD understands that 4 is the mesh border label on which Dirichlet is applied and ($u_x = 0$) is the condition to be applied.

Step 2: Solving

Let us now use 6 cores to solve this problem. To do so enter the following command:

```
PSD_Solve -np 6 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 6.

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.

[1] Please use version 2

Step 3: Postprocessing

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

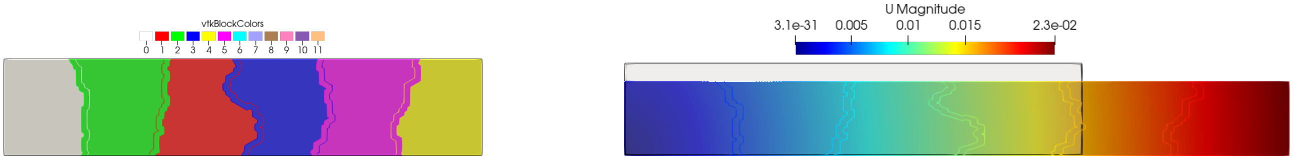

Figure 1: 2D bar results. Partitioned mesh (left) and 100X warped displacement field (right).

Note now in fig. 1 there are six subdomais in the partitioned mesh. As expected, we see that the right and the left end of the bar which is being pulled now contract in y direction, and the bar elongates in x direction.