

Linear Elasticity Tutorial 1 PSD simulation of bar problem bending under own body weight

Mohd Afeef Badri

Abstract

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

To showcase the usage of linear elasticity, we shall discuss here an example of a 2D bar which bends under its own load (body weight). The bar 5 m in length and 1 m in width, and is supposed to be made up of a material with density $\rho = 8 \times 10^3$, Youngs modulus $E = 200 \times 10^9$, and Poissons ratio $\nu = 0.3$. Figure 1 shows this bar considered in this tutorial.

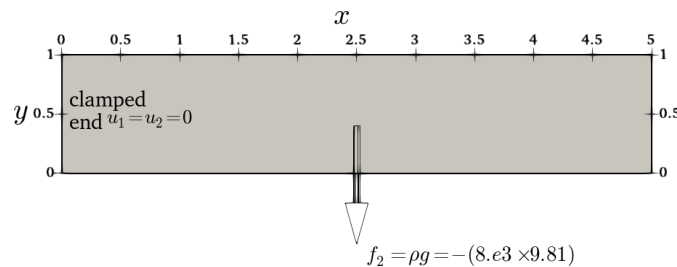


Figure 1: The 2D clamped bar problem.

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. PSD is a TUI (terminal user interface) based application, so the user needs to use the terminal (command-line) to communicate to PSD.

In the terminal `cd` to the folder `/home/PSD-tutorials/linear-elasticity`¹. Launch `PSD_PreProcess` with some falsgs from the terminal, to do so run the following command.

```
1 PSD_PreProcess -problem linear_elasticity -dimension 2 -bodyforceconditions 1 \  
2 -dirichletconditions 1 -postprocess u
```

What do the arguments mean ?

- `-problem linear_elasticity` means that we are solving a linear elasticity problem;
- `-dimension 2` means it is a 2D simulation;
- `-bodyforceconditions 1` with applied body force (body weight) acting on the domain;
- `-dirichletconditions 1` says we have one Dirichlet border (clamped end);
- `-postprocess u` means we would like to have ParaView post processing files.

After the `PSD_PreProcess` runs successfully you should see many `.edp` files in your current folder. You will now have to follow an edit cycle, where you will provide PSD with some other additional information about your simulation that you wish to perform - in this case 2D linear elasticity bending under its own body weight.

¹Note that one can perform these simulation in any folder provided that PSD has been properly installed. We use `/home/PSD-tutorials/linear-elasticity` for simplicity, once the user is proficient a simulation can be launch elsewhere.

At this stage the input properties of Young's modulus and Poisson's ratio (E, ν) can 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"`². 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. Dirichlet boundary conditions are also provided in [ControlParameters.edp](#). To provide the clamped boundary condition the variables `Dbc0On 2`, `Dbc0Ux 0.`, and `Dbc0Uy 0.` are used, which means for Dirichlet border `2` (`Dbc0On 2`) where `2` is the clamped border label of the mesh Dirichlet constrain is applied and `Dbc0Ux 0.`, `Dbc0Uy 0` i.e., the clamped end condition ($u_x = u_y = 0$).

Please 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`³ or `.mesh` format, we recommend using Salome or Gmsh meshers for creating your own geometry and meshing them.

Step 2: Solving

As PSD is a parallel solver, let us use 4 cores to solve the 2D bar case. To do so enter the following command:

```
1 PSD_Solve -np 4 Main.edp -mesh ../Meshes/2D/bar.msh -v 0
```

This will launch the PSD simulation.

Here `-np 4` (number of processes) denote the argument used to enter the number of parallel processes (MPI processes) used by PSD while solving. `-mesh ../Meshes/2D/bar.msh` is used to provide the mesh file to the solver⁴. `-v 0` denotes the verbosity level on screen. `PSD_Solve` is a wrapper around `FreeFem++-mpi`. Note that if your problem is large use more cores. PSD has been tested upto 24,000 parallel processes (on the French Joliot-Curie supercomputer) and problem sizes with billions of unknowns, surely you will not 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...` folder. Using ParaView for postprocessing the results that are provided in the `VTUs...` folder, results such as those shown in fig. 2 can be extracted.



Figure 2: The 2D clamped bar problem: partitioned mesh and displacement field visualization in ParaView.

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

2D bar is ok, but what about 3D ?

In PSD a 3D simulation follows the same logic as a 2D one, in the preprocessing step. Imagine the same problem as above, however now the geometry is 3D with length 5 m and cross sectional area 1 m \times 1 m. Indeed all what changes for this simulation is the geometry (consequently the mesh) and the dimension of the problem, these two changes will be handled by (`-dimension` and `-mesh`) arguments.

²Note that mesh can also be provided in the next step i.e, Step 2: solving.

³Please use version 2

⁴`-mesh` argument is not needed if the user has indicated the right mesh in [ControlParameters.edp](#) file

The preprocessing step now becomes:

```
1 PSD_PreProcess -problem linear_elasticity -dimension 3 -bodyforceconditions 1 \  
2 -dirichletconditions 1 -postprocess u
```

compared to the 2D problem, note that all what has changed **-dimension 3** instead of **-dimension 2**.

Solving step remains exactly the same with except **-mesh** flag now pointing towards the 3D mesh of the bar.

```
1 PSD_Solve -np 4 Main.edp -mesh ../Meshes/3D/bar.msh -v 0
```

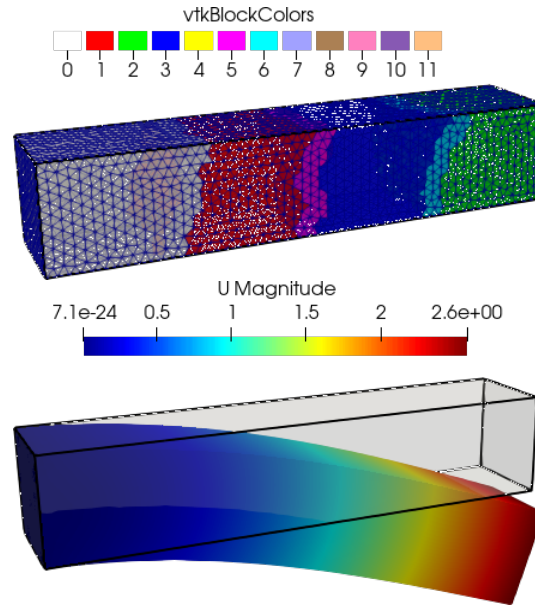


Figure 3: The 3D clamped bar problem: partitioned mesh and displacement field visualization in ParaView.

Finally, using ParaView for postprocessing the results that are provided in the **VTUs...** folder, results such as those shown in fig. 3 can be extracted.