Soildynamics Tutorials for PSD

Mohd Afeef Badri

Abstract

This document details some tutorials of soildynamics module of PSD. These tutorials are not verbose, but does instead give a kick start to users/developers for using PSD's soildynamics module.

Exercise 1

You are encouraged to try out sequential PSD solver, to do so used add -sequential flag to PSD_PreProcess step and run the solver with PSD_Solve_Seq instead of PSD_Solve. For example, the PSD sequential solver workflow for the first 2D example in this tutorial would be:

```
PSD_PreProcess -dimension 2 -problem soildynamics -dirichletconditions 1 -timediscretization newmark_beta \
-postprocess uav -sequential
```

Once the step above has been performed, we solve the problem using PSD_Solve_Seq, with the given mesh file soil.msh.

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

Try it out for other problems of this tutorial.

Exercise 2

For soildynamic problems with double couple source, the double couple source can be introduced into the solver either by displacement-based operator – providing displacements at the double couple points that will be converted to moments – or by force-based operators – providing forces at the double couple points that will be converted to moments. In the tutorials above we already tried displacement-based way of introducing double couple source by using -doublecouple displacement_based. You are encouraged to try out the force-based double couple source by using -doublecouple force_based.

Exercise 3

You are encouraged to try out timelogging and find out if the code (parallel/sequential) is any faster when we use Newmark-β or Generalized-α. Read the documentation for other types of time discretizations that can be performed with PSD, try each one out with -timelog and compare.

Exercise 4

PSD comes with additional set of plugins/functions that are highly optimized for performing certain operations during solving. These operations are handled by GoFast Plugins (GFP) kernel of PSD (optimize C++ classes/templates/structures), by default this functionality is turned off and not used. You are encouraged to try out using GFP functions in a solver by using -useGFP flag flag to PSD_PreProcess For example, the PSD solver workflow for the first 2D example in this tutorial would be:

```
PSD_PreProcess -dimension 2 -problem soildynamics -dirichletconditions 1 -timediscretization newmark\_beta \
-postprocess uav -useGFP
```

Once the step above has been performed, we solve the problem using, with the given mesh file soil.msh.

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

Try it out for other problems of this tutorial. -useGFP should lead to a faster solver, it might be a good idea to always use this option. To go one step further, use -timelog flag and determine if you have some speed up.