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.

Parallel 2D

The problem of interest is a single Dirichlet condition problem of soildynamics in 2D. For this problem we use Newmark-β time discretization. Additionally postrocessing is demanded for displacement, acceleration, and velocity (u, a, v).

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

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

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

Using ParaView for postprocessing the results that are provided in the VTUs... folder, results such as those shown in figure~1 can be extracted.

Parallel 3D

The problem of interest is a single Dirichlet condition problem of soildynamics in 3D. For this problem we use Newmark-β time discretization. Additionally postrocessing is demanded for displacement, acceleration, and velocity (u, a, v).

```
PSD_PreProcess -dimension 3 -problem soildynamics -dirichletconditions 1 -timediscretization newmark-beta \
-postprocess uav
```

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

```
PSD_Solve -np 3 Main.edp -mesh ./../Meshes/3D/soil.msh -v 0
```

Using ParaView for postprocessing the results that are provided in the VTUs... folder, results such as those shown in figure~2 can be extracted.

Parallel 2D with double couple

In the 2D problem above seismic sources was supplied on the border, in the current one the source is more realistic and comes from a double couple (point Dirichlet condition). The double couple boundary condition is a way to impose moments caused by faults that create earthquakes, here in this problem double couple is imposed using displacement based.

```
PSD_PreProcess -dimension 2 -problem soildynamics -model linear -timediscretization newmark-beta \
-useGFP -doublecouple displacement-based -postprocess uav
```

Once the step above has been performed, we solve the problem using two MPI processes, with the given mesh file soil-dc.msh.

```
PSD_Solve -np 2 Main.edp -v 1 -ns -nw -mesh ./../Meshes/2D/soil-dc.msh
```

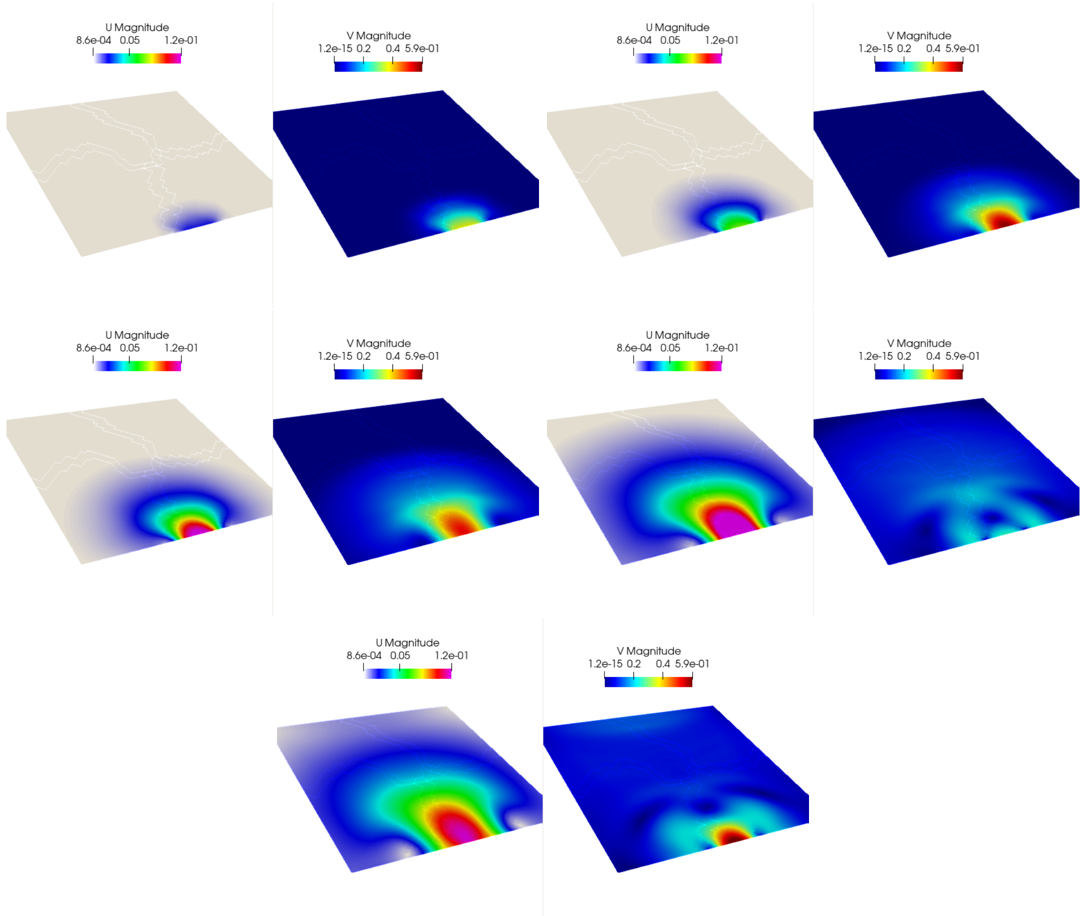

Figure 1: Finite element displacement and velocity fields visualized for the 2D problem with ParaView at different timesteps.

Using ParaView for postprocessing the results that are provided in the VTUs... folder, results such as those shown in figure~3 can be extracted.

Similarly try out the 3D problem. However take note that a the mesh ./../Meshes/2D/soil-dc.msh is not provided, so you will have to create your own mesh.

Parallel 3D with top-ii-vol meshing

Single Dirichlet at the bottom and using GFP.

```
PSD_PreProcess -dimension 3 -problem soildynamics -model linear -timediscretization newmark-beta \
-useGFP -top2vol-meshing -timediscretization newmark-beta -postprocess uav
```

```
PSD_Solve -np 4 Main.edp -v 0 -ns -nw
```

Parallel 3D with top-ii-vol meshing and double couple source

Single Dirichlet via double couple and using GFP. Double couple is displacement based.

```
PSD_PreProcess -dimension 3 -problem soildynamics -model linear -timediscretization newmark-beta \
-useGFP -top2vol-meshing -doublecouple displacement-based -postprocess uav
```

```
PSD_Solve -np 3 Main.edp -v 0 -ns -nw
```

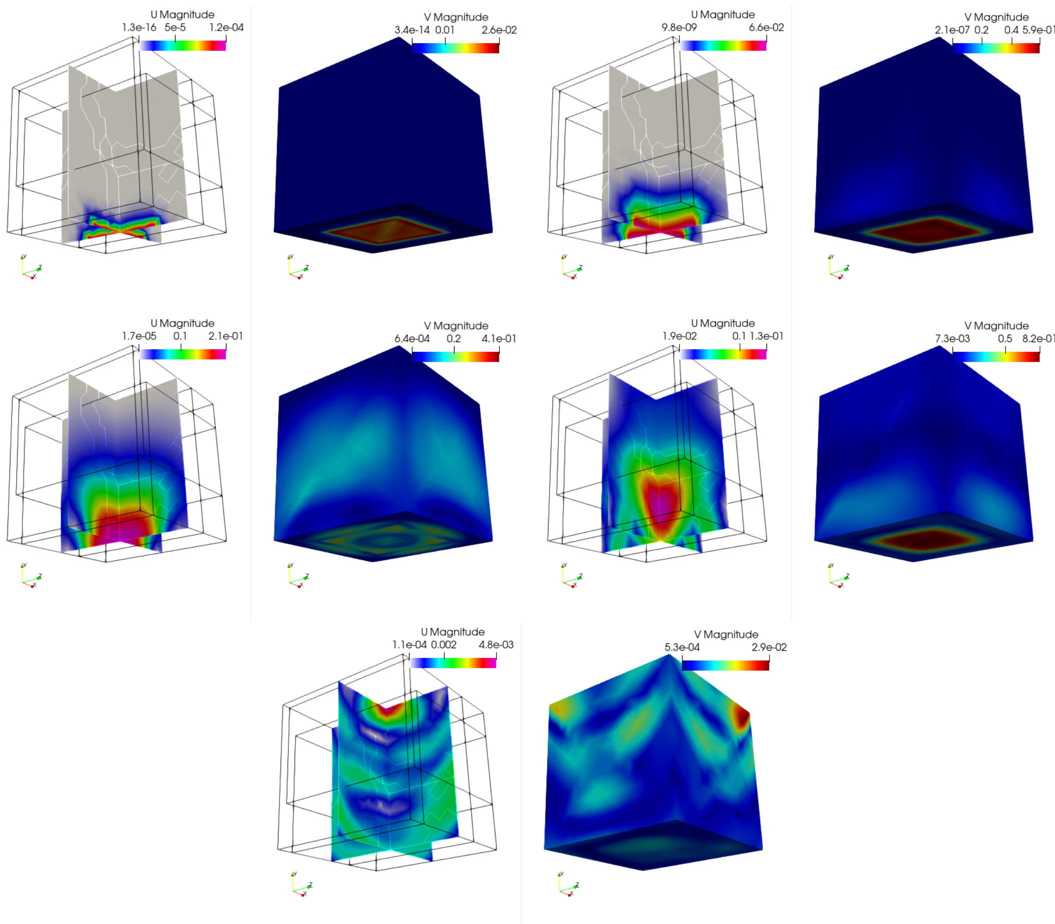

Figure 2: Finite element displacement and velocity fields visualized for the 3D problem with ParaView at different timesteps.

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.

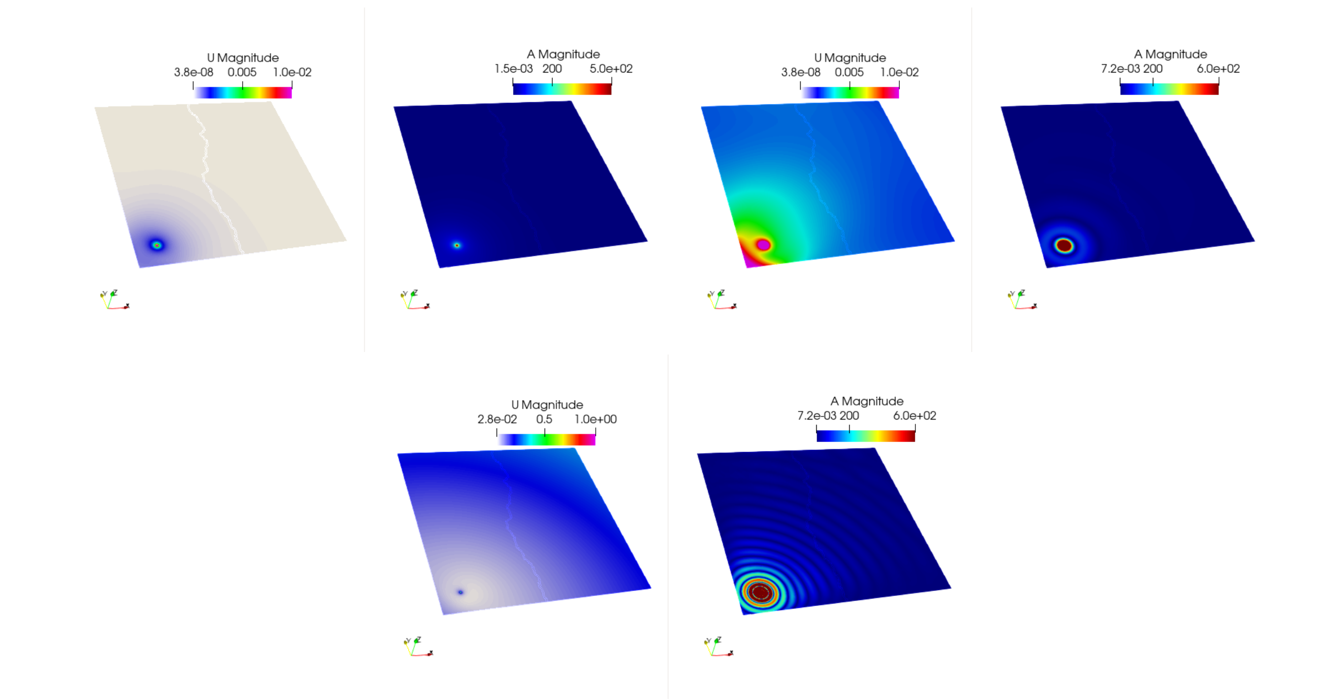

Figure 3: Finite element displacement and acceleration fields visualized for the 2D problem with ParaView at different timesteps.

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.

```
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.