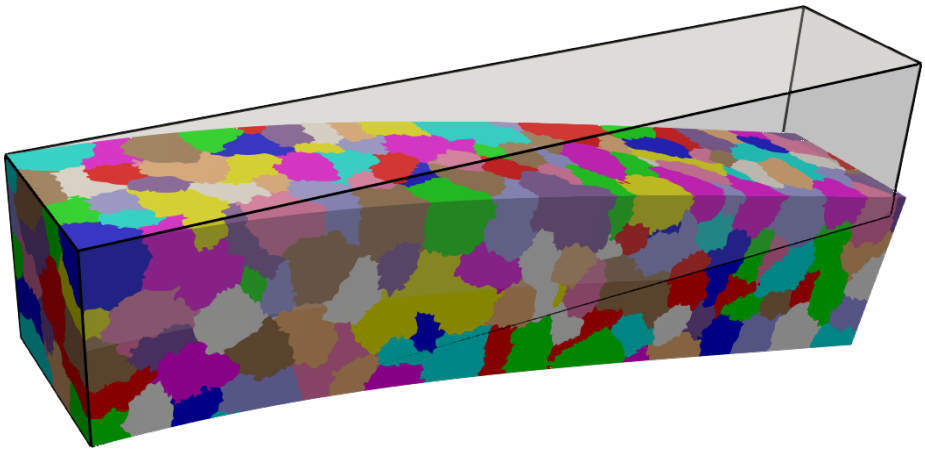
PSD: a Parallel Structural Dynamics solver



PSD is a finite elements-based solid mechanics solver with capabilities of performing High Performance Computing (HPC) simulations with billions of unknowns. The kernel of PSD is wrapped around FreeFEM for finite element discretization, and PETSc for linear algebra/Preconditioning. PSD solver contains straightforward supports for *static* or *dynamic* simulations with *linear* and *nonlinear* solid mechanics problems. Besides these *hybrid-phase field fracture mechanics* models have also been incorporated within PSD. For dynamics the *genralized-α model* for time discretization is used, this models enable straightforward use of Newmark-β, central difference, or HHT as time discretization. PSD uses sate-of-the art domain-decomposition paradigm via *vectorial finite elements* for parallel computing and all solvers are proven to scale quasi-optimally. PSD has proven scalabilty uptill 13,000 cores with largest problem solved containing over 5 Billion unknowns.

Besides the parallel suite, PSD also includes a sequential solver which does not require PETSc.

PSD works for two and three dimensional problems only. Unstructured meshes (triangular for 2D and tetrahedral for 3D) are supported in MEDIT's `.mesh` format or Gmsh's `.msh` format. PSD post processing is done via `.vtk` and `.vtu` files of the ParaView platform.

Installation procedure

PSD is built to work with Linux platforms. (Windows comping up soon)

Dependencies

- automake
- FreeFEM
- PETSc (optional)

- Gmsh
- gnuplot (optional)
- C++
- git

Now that I have all the dependencies what next

- Go ahead and grab the latest copy of PSD. The code is hosted on CEA's internal git repository.

```
git clone https://codev-tuleap.intra.cea.fr/plugins/git/hpcseism/freefem.git PSD-Sources
```

- Autoconf PSD within the cloned folder

```
autoreconf -i
```

- Configure PSD within the cloned folder

```
./configure
```

Note: `./configure` will install PSD in `$HOME/PSD` to change this directory use `--prefix=Your/Own/Path` with `./configure`. *Note*: `./configure` will try to look for installation of FreeFEM and Gmsh in `usr/bin/` or `usr/local/bin/` directories. If you have these packages installed in some other directory this should be specified during `./configure` by using flags `--with-FreeFEM=` and `--with-Gmsh=`. For example, if FreeFEM is installed at `home/FreeFem/bin` and Gmsh in `home/Gmsh/bin` then one should use

```
./configure --with-FreeFEM=home/FreeFem/bin  --with-Gmsh=home/Gmsh/bin
```

- Make PSD directives

```
make
```

- Install PSD

```
make install
```

Now you should have the solver at `$HOME/PSD`. To use the solver please go to `$HOME/PSD`.

Additional FreeFEM tweak for brittle fracture mechanics

Note that this procedure is only recommended if you are interested in using PSD for brittle fracture problems. In your FreeFEM source files (installation) go to `src/femlib/fem.cpp` , in this file replace the lines of code

```
R seuil=hm/splitmax/4.0;
```

by the following

```
R seuil=hm/splitmax/4.0/1000.0;
```

Using procedure

The parallel/sequential solvers can be run on Linux platforms and can be used with command line options to control the solver. In order to make your choice of physics, command line options need to be typed right into the bash.

To perform a simulation three steps are involved

Step 1: Providing the mesh

A valid finite element mesh needs to be provided to the solver. PSD accepts only unstructured meshes: triangular for 2D and tetrahedral for 3D. These meshes need to be in MEDIT's `.mesh` format or Gmsh's `.msh` format. Once you have your mesh please copy the mesh to `Meshes` folder of your installed PSD directory. The 2D meshes need to be separated by the 2D ones.

Note: There is a native mesher `bmag` within FreeFEM which could also be used. However, it can only be used to generate simple geometries. In this case no need to copy the mesh to the `Meshes` folder.

Step 2: Setting up the solver

Now that the mesh has been loaded. Its time to set up the solver. Open the `terminal` window at the location of the solver, i.e., `$HOME/PSD/Solver.` Then run the following command in the `terminal`.

```
./PSD_PreProcess [Options-PSD]
```

Via the command line options you will embed the physics within the solver. This step generates a bunch of `.edp` files which are native to FreeFEM and additionally prints out instructions on what to do next. You then need to open and edit couple of these files via your favorite text editor, which could be `vim`, `gedit` ,`Notepad++`, etc. To facilitate the edit process for your will have to go through the instructions printed on the terminal.

For example to generate a sequential 2D elasticity solver for a problem with body force and one Dirichlet border use

```
/PSD_PreProcess -dimension 2 -bodyforceconditions 1 -dirichletconditions 1
```

Step 3: Launching the solver

Now you are all set to run your simulation. To do so you will need to do the run the following in the `terminal`:

if you complied a parallel PSD version

```
PSD_Solve -np $N Main.edp -v 0 -nw
```

if you complied a sequential PSD version

```
PSD_Solve Main.edp -v 0 -nw
```

- In the parallel command **$N** is an `int` value, i.e., number of processes that you want to use for performing the simulation in parallel.
- Additional flag `-wg` may be required while launching the solver, this is in case debug mode is on.

Sit back and be amazed at the speed of the solver.

[Options-PSD] explained

These are a set of commandline flags/options that control your simulation. You can think of it as a way to talk to the solver. Here is a table that lists out some of the options that are available (for full list see documentation).

Option	Type	Comment
`-help`	`[bool]`	To activate helping message on the terminal.
`-debug`	`[bool]`	To activate debug mode with a wait plot. Development flag.
`-useGFP`	`[bool]`	To activate use of GoFastPlugins. A suite of C++ plugins.
`-useRCM`	`[bool]`	To activate mesh level renumbering via Reverse Cuthill Mckee.
`-timepvd`	`[bool]`	To activate time pvd plotting for ParaView
`-pipegnu`	`[bool]`	Use to activate realtime pipe plotting using gnuplot.
`-timelog`	`[bool]`	To activate time logging the different phases of the solver.
`-supercomp`	`[bool]`	Use when using a super computer without Xterm. support .
`-bodyforce`	`[bool]`	To activate volumetric source term (body force).
`-vectorial`	`[bool]`	To use vectorial finite element method.
`-sequential`	`[bool]`	To solve via a sequential solver.
`-energydecomp`	`[bool]`	To activate energy decomposition, only for phase-field.
`-doublecouple`	`[bool]`	To activate double couple source for soildynamics.
`-dirichletconditions`	`[int]`	Number of Dirchlet boundaries.
`-bodyforceconditions`	`[int]`	Number of regions acted upon by bodyforce.
`-tractionconditions`	`[int]`	Number of Neumann/traction bounders.
`-parmetis_nworker`	`[int]`	Number of parallel workers used by ParMetis for partiotining.
`-lagrange`	`[int]`	Lagrange order used for FE spaces. 1 for P1 or 2 for P2.
`-dimension`	`[int]`	Accepts values 2 or 3. Use 3 for 3D. and 2 for 2D problem.
`-nonlinearmethod`	`[string]`	Nonlinear method type. Use “Picard” or “Newton-Raphsons”.

Option	Type	Comment
`-postprocess`	`[sting]`	To communicate what to postprocess “u”, “v”, “a”, “uv” , “ud”, “ua”, “d”, “ud”, or “uav”.
`-problem`	`[sting]`	Interested problem. Use “linear-elasticity”, “damage”, “elastodynamics”, or “soildynamics”.
`-partitioner`	`[sting]`	Mesh partitioner could be “metis” “parmetis” or “scotch”.

To report bugs, issues, feature-requests contact:

- **mohd-afeef.badri@cea.fr**
- **mohd-afeef.badri@hotmail.com**
- **mohd-afeef.badri@etu.univ-nantes.fr**