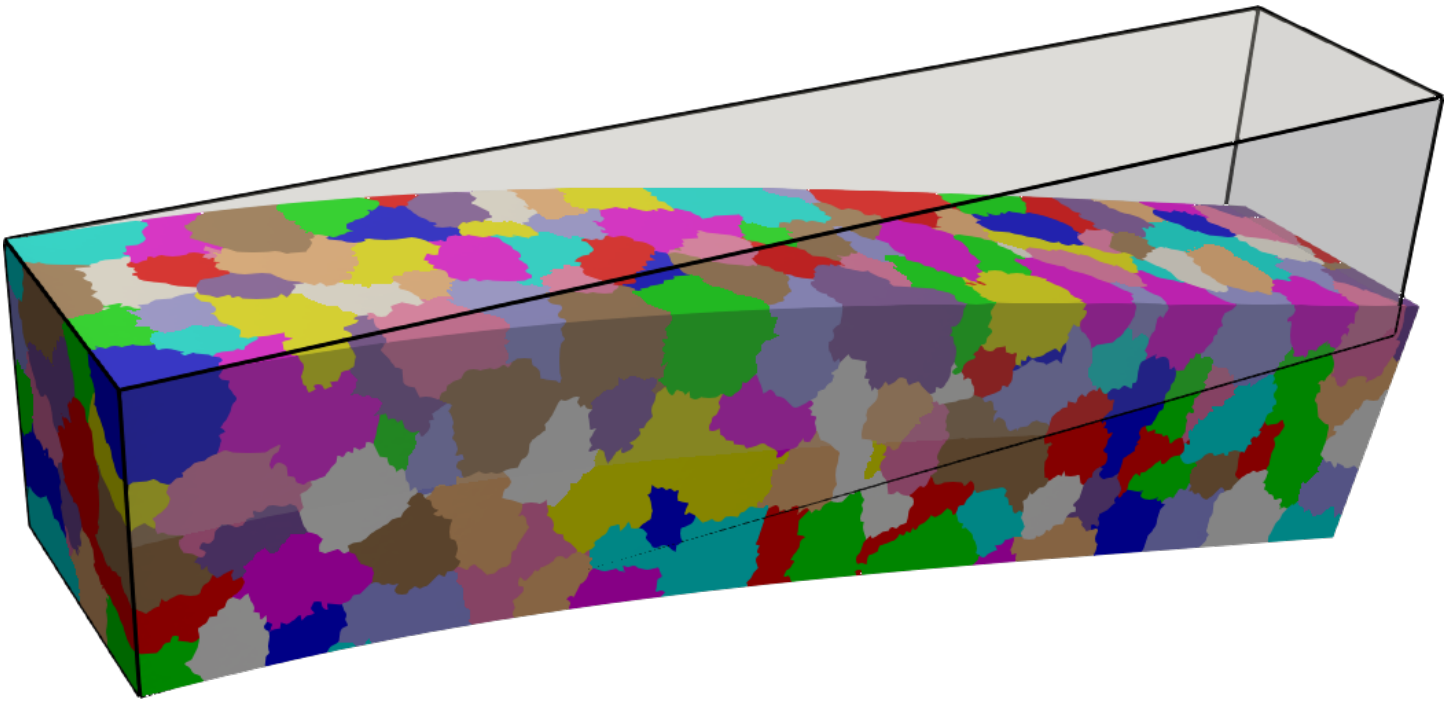
PSD 2.3: a Parallel Structural Dynamics solver



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 24,000 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 `.pvd` ,`.vtk` and `.vtu` files of the ParaView platform.

Installation

PSD is a cross-platform solver built to work with Linux platforms (Windows comping up soon). PSD has successfully been deployed on the following platforms, CentOS 7/8, Ubuntu 16.04/18.04/20.04, Raspberry Pi, Fedora 32, etc. Before installing PSD please ensure that you have the following dependencies installed on your OS.

Dependencies

- automake
- FreeFEM
- PETSc
- Gmsh
- C++
- git
- MFront (optional)
- MGIS (optional)
- gnuplot (optional)

Now that I have all the dependencies what next

- Go ahead and grab the latest copy of PSD. The code is hosted on GitLab repository.

```
git clone https://gitlab.com/PsdSolver/psd_sources.git PSD-Sources
```

Note: You can also use SSH protocol if your key has been added to the repo in that case use

```
git clone git@gitlab.com:PsdSolver/psd_sources.git
```

- Use automake PSD within the cloned folder

```
autoreconf -i
```

- Configure PSD within the cloned folder

```
./configure
```

Note: `./configure` will install PSD in `/usr/local/bin` and you would need sudo rights to perform installation, for non sudo users or for local install consider changing directory of installation. To change this directory use `--prefix=Your/Own/Path` with `./configure`. Remember to add `Your/Own/Path` to your `$PATH` variable, you can do so by `export PATH=$PATH:Your/Own/Path`. Also `./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

```
sudo make install
```

Note : You should not use `sudo` if you have used `--prefix` during the `./configure`

- Install PSD tutorials

```
make tutorials
```

Now you should have the PSD solver installed on your machine. Note that, the solver will be installed at `usr/bin` or `usr/local/bin` directories if you used `sudo make install` or else it will be in your `--prefix` location. The PSD tutorials are installed in `$HOME/PSD-tutorials`.

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;
```

Additional Installation steps for experts and developers

- Check if installation is correct

```
make check
```

- Configure PSD with MFront support

```
./configure --prefix=$HOME/Install/local --with-mgis=$HOME/Install/local/mgis --with-mfront=$HOME/Install/loca
```

Here in this example we assume that MFront, MGIS, Gmsh, and FreeFEM are installed in `$HOME/Install/local` directory, if that is not the case for you please improvise or you can always ask for help from Linux geeks around.

- Update PSD to the latest version. If you would like to update your old PSD source to a new one. Go to your`PSD-Sources` folder and

```
git pull origin master && ./reconfigure &&  make &&  make install
```

- If you would like to install a developers copy of PSD

```
make install-devel
```

A quick sneak-peek of a typical PSD simulation

PSD is a TUI (terminal user interface) based finite element solver. Parallel or sequential PSD simulations can run on Linux platforms. Command line options (flags) which user enters are used to control the PSD solver. In order to make your choice of physics, model, mesh, etc., command line options need to be typed right into the bash.

A typical PSD simulation is performed in three steps.

Step 1: Setting up the solver

Its time to set up the PSD 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 favourite 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 2: 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_Seq 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.

Step 3: Result visualization Final step is to have a look at the results of the simulation. PSD can provides output results in the form of plots, finite element fields of interest, etc. ParaView's pvd, vtu, and pvtu files are used for postprocessing. ASCII data files that to trace certain quantities of interest like reaction forces, kinetic energies, etc can also be outputted.

PSD flags 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). It is advised to print these and have them around when performing a PSD simulation.

Flag	Type	Comment
Boolean flags		These flags accept values *1/0/yes/no/on/off/true/false* and are used to activate or deactivate any functionality of PSD.
`-help`	`[bool]`	To activate helping message on the terminal. Gives description and list of available flags.
`-debug`	`[bool]`	To activate live plot while PSD runs. Development flag.
`-useGFP`	`[bool]`	To activate use of GoFastPlugins. A suite of C++ plugins.
`-useRCM`	`[bool]`	Activate mesh level renumbering: Reverse Cuthill Mckee.
`-pipegnu`	`[bool]`	Use to activate real time 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
`-useMfront`	`[bool]`	Activate Mfornt interface for PSD.

Flag	Type	Comment
-bodyforce	[bool]	To activate volumetric source term (body force).
-vectorial	[bool]	To use vectorial finite element method.
-pointprobe	[bool]	To postprocess point fields.
-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.
-constrainHPF	[bool]	To use constrain condition in hybrid phase-field model.
-top2vol-meshing	[bool]	Activate top-ii-vol point source meshing for soil-dynamics.
-getreactionforce	[bool]	Activate routine for extraction reactions at surface.
-plotreactionforce	[bool]	Activate realtime pipe plotting using GnuPlot.
-withmaterialtensor	[bool]	Activate material tensor for building stiffness matrix.
-crackdirichletcondition	[bool]	To activate pre-cracked surface Dirichlet.
Integer flags		These flags accept a integer value followed by the flag itself. These integer values are used in PSD simulations for various definitions.
-dirichletpointconditions	[int]	Number of Dirichlet points.
-dirichletconditions	[int]	Number of Dirichlet boundaries.
-bodyforceconditions	[int]	Number of regions acted upon by bodyforce.
-tractionconditions	[int]	Number of Neumann/traction boundaries.
-parmetis_worker	[int]	Number of parallel workers used by ParMetis for partitioning.
-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.
String flags		These flags accept a string value followed by the flag itself. These string values are used in PSD simulations for various definitions.
-mesh	[sting]	Provide mesh to be solved by PSD_Solve.
-timediscretization	[sting]	Time discretization type. Use “generalized_alpha” or “newmark_beta” or “hht_alpha” or “central_difference”
-nonlinearmethod	[sting]	Nonlinear method type. Use “Picard” or “Newton_Raphsons”.
-reactionforce	[sting]	Reaction force calculation method “stress_based” or “variational_based”.
-doublecouple	[sting]	Soil dynamics double couple. Use “force_based” or “displacement_based”.
-postprocess	[sting]	To communicate what to postprocess “u”, “v”, “a”, “uv” , “ud”, “ua”, “d”, “ud”, or “uav”.
-partitioner	[sting]	Mesh partitioner could be “metis” “parmetis” or “scotch”.
-problem	[sting]	Interested problem. Use “linear_elasticity”, “damage”, “elastodynamics”, or “soildynamics”.
-model	[sting]	Interested model. Use “hybrid_phase_field” or “Mazar”.

To report bugs, issues, feature-requests contact:

- **mohd-afeef.badri@cea.fr**
- **mohd-afeef.badri@hotmail.com**