Fracture mechanics Tutorials

Mohd Afeef Badri

Abstract

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

Parallel 2D

A two dimensional test is introduced. The problem of interest is the typical single notch square cracking test under tensile loading. A unit square with a pre existing crack is clamped at the bottom $u_1 = u_2 = 0$ (first boundary condition) and is loaded quasi-statically $u_2 = u_2 + \Delta u_2$ on its top surface till the crack propagates through its walls. So there are two Dirichlet conditions one on the top border and one on the bottom one.

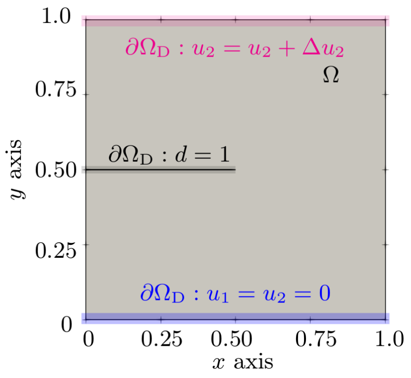

$\partial\Omega_\text{D} : u_2 = u_2 + \Delta u_2$
Ω
$\partial\Omega_\text{D} : d = 1$
$\partial\Omega_\text{D} : u_1 = u_2 = 0$
1.0
0.75
0.50
0.25
0
y axis
0
0.25
0.50
0.75
1.0
x axis

Figure 1: Domain of the single notch square cracking problem under tensile loading.

To model this test PSD provides hybrid phase-field modelling technique. We use ParaView post-processing of displacement u and phase-field d to visualise the cracking process. A PSD simulation is a two step process, with step one being the PSD_PreProcess :

```
PSD_PreProcess -dimension 2 -problem damage -model hybrid_phase_field \
-dirichletconditions 2 -postprocess ud
```

A note on flags.

- This is a two-dimensional problem, so we use the flag -dimension 2.
- This problem indeed falls under the category of damage-mechanics, hence the flag -problem damage .
- We wish to solve this problem by invoking the hybrid phase-field problem, which is signified by the flag -model hybrid_phase_field.
- Versed in the description above the problem contains two Dirichlet conditions, we signal this via the flag -dirichletconditions 2.
- Finally for this problem we use the flag -postprocess ud which enables post-processing of displacement u and damage (phase-field) d fields.

Once the step above has been performed, we solve the problem using four MPI processes, with the given mesh file tensile-crack.msh. This is step two of the PSD simulation PSD_Solve.

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

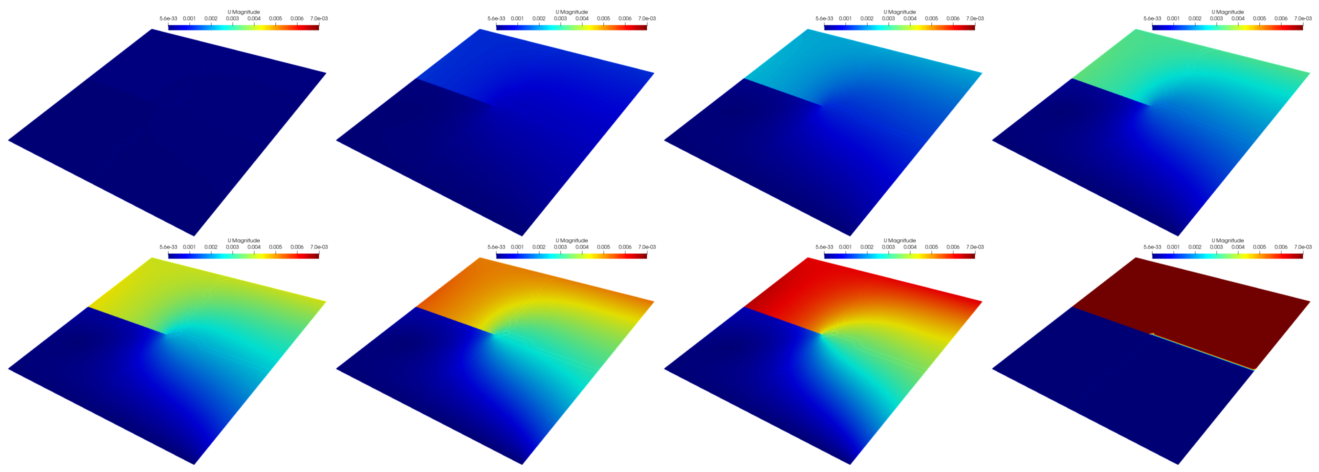


Figure 2: Finite element displacement visualised for the 2D problem with ParaView at different timesteps (quasi-statics). Time progresses from left to right in a row and top to bottom when comparing rows.

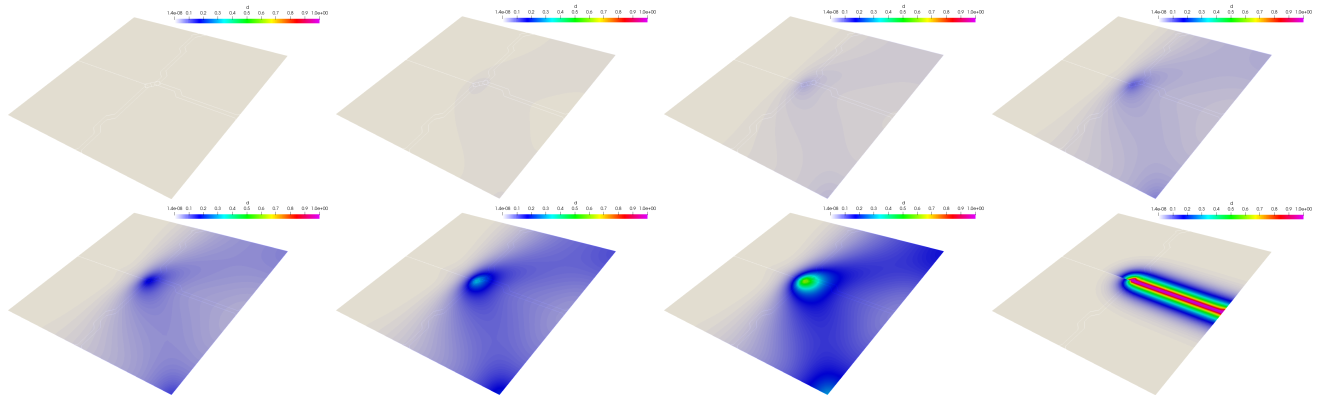
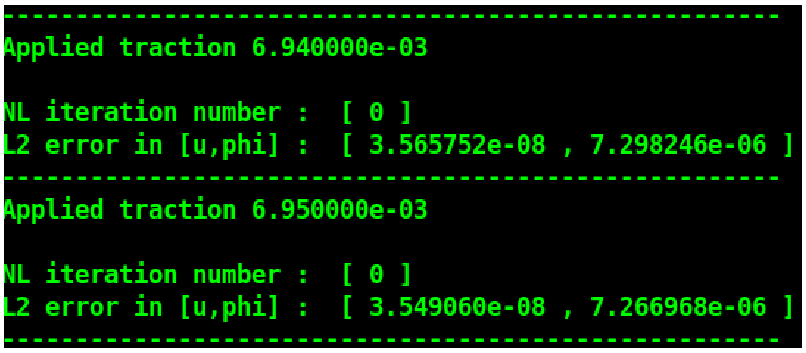


Figure 3: Finite element damage visualised for the 2D problem with ParaView at different timesteps (quasi-statics). Time progresses from left to right in a row and top to bottom when comparing rows.



```
--------------------------------------------------
Applied traction 6.940000e-03

NL iteration number :  [ 0 ]
L2 error in [u,phi] :  [ 3.565752e-08 , 7.298246e-06 ]
--------------------------------------------------
Applied traction 6.950000e-03

NL iteration number :  [ 0 ]
L2 error in [u,phi] :  [ 3.549060e-08 , 7.266968e-06 ]
--------------------------------------------------
```

Figure 4: Applied traction, non-linear iterations to convergence, and residual being casted onto the terminal shell.

Figures 2 and 3 present the finite element displacement and damage field, which enable us to visualise the cracking of the square plate.

While this test runs, you will see on your screen the amount of traction updated, non-linear iterations taken to converge per-quasi-time-step and residue of u and d. See figure 4 that shows the screenshot of the terminal while the test was running. In order to construct your own test case try editing the ControlParameters.edp file

Parallel 3D

```
PSD_PreProcess -dimension 3 -problem damage -model hybrid_phase_field \
-dirichletconditions 2
```

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

Parallel 2D and calculate reactionforce

```
PSD_PreProcess -dimension 2 -problem damage -model hybrid_phase_field \
-dirichletconditions 2 -getreactionforce -reactionforce stress_based
```

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

Parallel 3D and calculate reactionforce

```
PSD_PreProcess -dimension 3 -problem damage -model hybrid_phase_field \
-dirichletconditions 2 -getreactionforce -reactionforce stress_based
```

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

Exercise 1

Optionally try changing -reactionforce stress_based to -reactionforce variational_based for changing the method to extract reaction force, note that stress based method is way faster.

Exercise 2

Optionally try using -useGFP flag with PSD_PreProcess optimized solver

Exercise 3

Add -sequential flag to PSD_PreProcess for sequential solver, but remember to use PSD_Solve_Seq instead of PSD_Solve

Advanced Exercise 1

try the -vectorial flag for vectorial finite element method

Advanced Exercise 2

try the -energydecomp flag for using split of tensile energy

Advanced Exercise 3

try using -constrainHPF flag for using the constrain condition in hybrid phase field model