Fracture mechanics Tutorials - Tensile cracking of a pre-cracked plate a 2D example of PSD parallel solver

Mohd Afeef Badri

Abstract

This document details some tutorials of 'fracture mechanics' module of PSD.

A two dimensional test is introduced. The problem of interest is the typical single notch square plate 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.

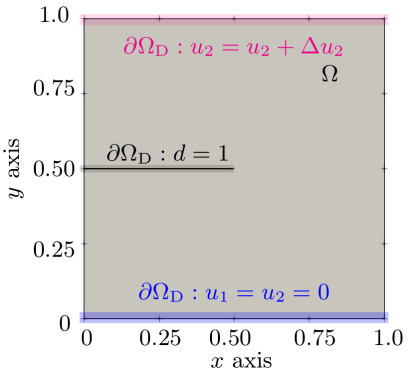

Figure 1: Domain of the single notch square cracking problem under tensile loading.

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
```

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

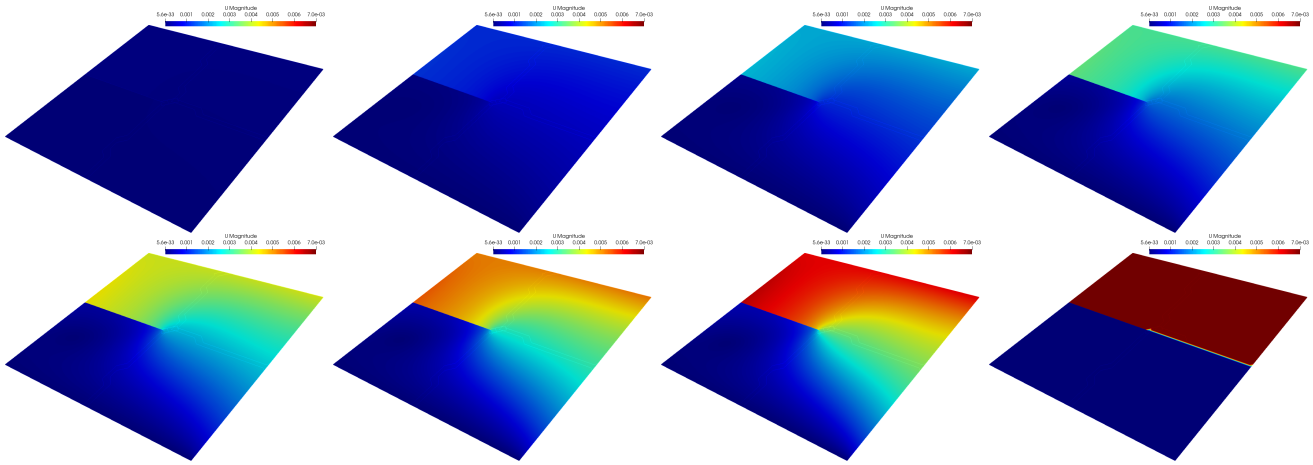

Figure 2: Finite element displacement visualised for the 2D problem with ParaView at different timesteps (quasi-statics). Time progresses from left to right in a row and top to bottom when comparing rows.

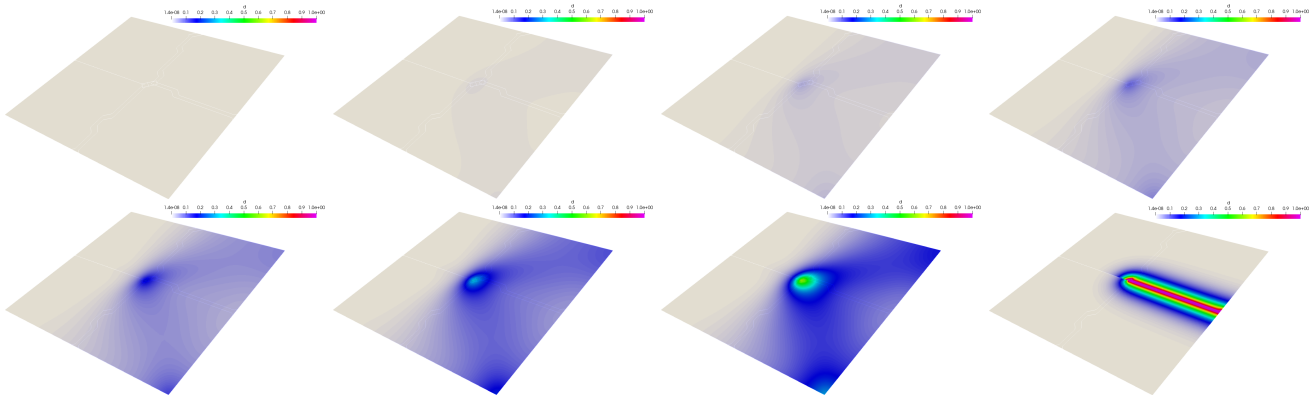

Figure 3: Finite element damage visualised for the 2D problem with ParaView at different timesteps (quasi-statics). Time progresses from left to right in a row and top to bottom when comparing rows.

```
-----------------------------------------------------
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.

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.