

A Tutorial on PhiPsi

Author: Fang Shi

Website: phipsi.top

Email: shifang@hyit.edu.cn

Updated on March 22, 2024

This quick tutorial familiarizes new users with the basics of using PhiPsi. We'll start with a simple two-dimensional project by generating input files using ANSYS or Abaqus, then creating the keywords file, carrying out the analysis using PhiPsi, and finally plotting contours with a Matlab-based post-processor or Paraview. You can interact with the programs as we walk through this tutorial.

Related files of this tutorial can be downloaded from:

<http://phipsi.top/manuals/tutorial.rar>

Keywords manual of PhiPsi can be accessed from:

http://www.phipsi.top/phipsi_keywords_manual.html

PhiPsi instruction manual can be accessed from:

<http://www.phipsi.top/README.html>

1. Description of the problem

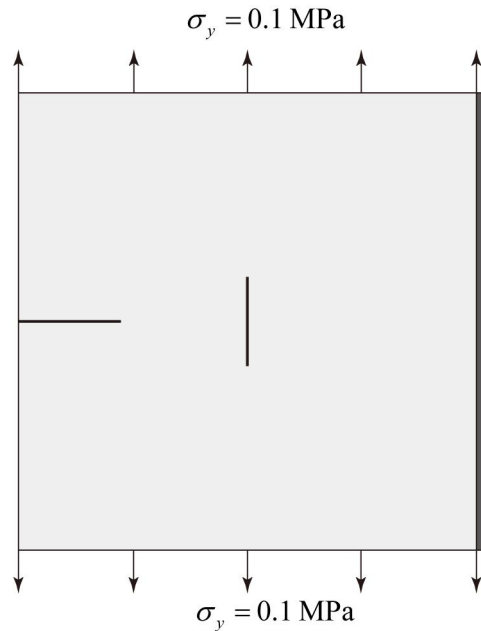


Figure. 1.1 This figure represents a plane-stress plate with two initial cracks (an edge crack and a vertical crack) under the action of tensile stress $\sigma_y = 0.1 \text{ MPa}$. The right edge is fixed. The width of the model is 1.55 m, and the height of the model is 1.85 m. The size of elements is $0.05 \times 0.05 \text{ m}$.

2. Pre-Processing

2.1 OPTION-1: generate input files using ANSYS

The name of the tutorial project we are going to present is “tutorial_example”. The input files (i.e., *tutorial_example.node*, *tutorial_example.elem*, *tutorial_example.boux*, *tutorial_example.bouy*, *tutorial_example.focx*, and *tutorial_example.focy*) that contain the information of node coordinates, element-node information, boundary conditions, and external forces are generated by a commercial finite element software ANSYS. Once the finite element model has been created in ANSYS (as shown in Figure 2.1), the input files can be automatically generated by running the macro file named *ANSYS2PhiPsi_2D.mac*. The macro file should be copied into the working directory of ANSYS so that it can be easily accessed and called. One thing has to be mentioned here is that forces and boundary constraints should be applied on nodes after meshing, rather than on lines or surfaces.

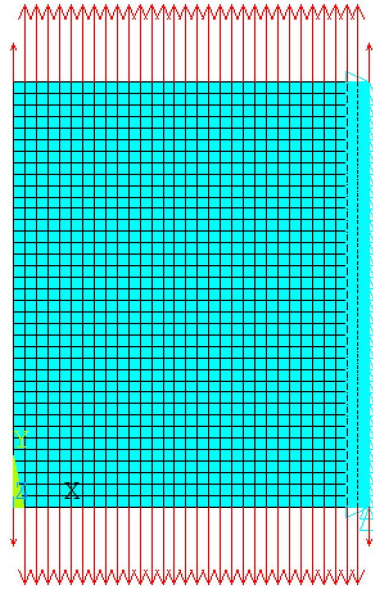


Figure. 2.1 The model created using ANSYS.

The ANSYS APDL (i.e., ANSYS Parametric Design Language) file (*tutorial_example.apdl*, this file can be opened and edited by any text editor) of this tutorial has been given in the *tutorial* folder. The statement to call the macro file to generate the expected files can be found at the end of the APDL file, namely,

```
ANSYS2PhiPsi_2D.mac, 'tutorial_example'
```

After running the APDL file, six input files can be found in the ANSYS working directory. Create a new folder called *tutorial_example* on *disk D* (or any other disk, but the folder directory should be modified accordingly hereinafter) and copy the all six input files into it.

2.2 OPTION-2: generate input files using Abaqus

Use Abaqus to create the model, as shown in Figure 2.2. Then, save the model as *.inp (for example, *tutorial_example.inp*, this file has been presented in the *tutorial* folder) file.



Figure. 2.2 The model created using Abaqus.

Now, we will convert the Abaqus **.inp* file to input files of PhiPsi. Firstly, we should install the *Abaqus2PhiPsi* tool by double click the *Installer_web.exe* in the *Matlab_Runtime_Installer* folder after downloading and unzipping the *Abaqus2PhiPsi_Matlab* tool, which can be obtained from http://www.phipsi.top/source/Abaqus2PhiPsi_Matlab.rar.

Afterwards, run the *Abaqus2PhiPsi_Matlab* tool (*run as Administrator if necessary*), select the Abaqus inp file (*tutorial_example.inp* in this tutorial). The *Abaqus2PhiPsi_Matlab* tool will convert the inp file to six input files of PhiPsi, i.e., *tutorial_example.node*, *tutorial_example.elem*, *tutorial_example.boux*, *tutorial_example.bouy*, *tutorial_example.focx*, and *tutorial_example.focy*.

Create a new folder named *tutorial_example* on *disk D* (or any other disk, but the folder directory should be modified accordingly hereinafter) and copy the all six input files into it.

3. Create the keywords file

In PhiPsi, a keywords file defines information such as work directory, analysis type, coordinates of initial fractures, material parameters, and others data. The keywords file named *kw_tutorial_example.kpp* of this tutorial is listed as follows. The statement starts with “%” is comment. The keywords manual of PhiPsi can be found here:

[PhiPsi Keywords Manual.](#)

```
% Keywords file of PhiPsi.
% Written by Fang Shi, USTC.
% Data: October 21, 2016
```

```

% Website: phipsi.top
% Working directory.
*Work_Directory
D:\tutorial_example
% Filename of input files.
*Filename
tutorial_example
% Number of initial cracks.
*num_Crack
2
% Analysis type (Quasi-static).
*Key_Analysis_Type
1
% Plane strain.
*Key_Type_2D
2
% Contact checking (off).
*Key_Contact
0
% Linear system solver.
*Key_SLOE
5 % LAPACK
% Factor of propagation length (propagation length = factor * the average element size)
*Factor_Propagation
1.6
% Number of propagation steps.
*Num_Substeps
15
% Crack propagation criterion.
*CFCP
2
% Material(1-E,2-v,3-density,4-thick,5-St,6-KIc,7-Sc,etc.)
*Material_Para_1
20.0e9,0.2,2000.0,1.0,0.4e6,1.0e6,100.0e6,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
% Coordinates of initial cracks.
*CRACK_1
-0.1,0.925,0.35,0.925
*CRACK_2
0.725,1.075,0.725,0.775
% Allow crack 1 and crack 2 to propagate.
*Cracks_Allow_Propa
1,1

```

Copy the keywords file to root directory of disk *D*. At this point, we have six input files in *D:\tutorial_example* and a keywords file in *D:*.

4. Perform the analysis

Run *PhiPsi_Win64.exe* by double-clicking, when ask for the name of the keywords file, just type *D:\kw_tutorial_example.kpp* and press *Enter* to continue. During the analysis process, the generated results files could be found in *D:\tutorial_example*.

Alternatively, you can run *PhiPsi_Win64.exe* directly in Windows Terminal as:

```
PhiPsi_Win64.exe -i "D:\kw_tutorial_example.kpp" -n 1
```

where *-i* specifies the keywords file, and *-n* specifies the number of CPU threads.

During the computation, the Terminal outputs some information, as shown in Figure 4.1. The tutorial example takes about 20 s using a computer with an Intel 8700K CPU.

```
Constructing global force vector...
Sum of globalF:  0.0000000000000000
Assembling K.....
Sum of globalK:  6077320424244.7275
Solving displacements.....
Dimension of Linear System:      2676
Solver: LAPACK
Range of displacement x:  -0.820147E-03 m to  0.156523E-03 m
Range of displacement y:  -0.140034E-02 m to  0.140034E-02 m
Max aperture of crack 1(mm):  2.7962186231876971
Saving disp...
Saving coordinates of calculation points...
Saving orientations of calculation points...
Saving apertures of calculation points...
Calculating stress of all nodes...
Max Von-mises stress of all nodes:  0.68324E+01 MPa
Max stress x of all nodes:  0.57577E+01 MPa
Min stress x of all nodes:  -0.75314E+01 MPa
Max stress y of all nodes:  0.42787E+01 MPa
Min stress y of all nodes:  -0.10672E+01 MPa
Saving stress of all nodes...
Calculating coordinates of Gauss points...
Saving coordinates of Gauss points...
Calculating displacements of Gauss points...
Saving displacement of Gauss points...
Calculating stress of all Gauss points...
Max Von-mises stress of Gauss points:  0.15265E+02 MPa
Saving stresses of Gauss points...
Saving SIFs...
Saving vtk file...
Checking propagation of each crack...
-- Max principal tensile stress of tip 1 of crack  2 is  6.6170893 MPa.
-- Tip 1 of crack  2 propagated!
-- Max principal tensile stress of tip 2 of crack  2 is  6.6170893 MPa.
-- Tip 2 of crack  2 propagated!

End time: 2024-03-23, 10:34:26
Total elapsed CPU time:  19 s, about  0.3167 mins
```

Figure 4.1 Terminal output of PhiPsi.

5. Post-Processing

5.1 OPTIN-1: post-processing using Matlab

The results file can be post-processed by using the open-sourced Matlab program. The program can be adopted to generate the deformation figures, stress contours and displacement contours, etc. The source codes of the post-processor can be download from <http://phipsi.top/downloads.html>. The main programs of the post-processor are *PhiPsi_2D_Post_1.m* and *PhiPsi_2D_Post_2.m*, and they are used to generate static

pictures and dynamic pictures, respectively. Usage can be found in the source codes. The Matlab script to perform the post-processing for this tutorial is [PhiPsi_2D_Post_1_for_tutorial_example.m](#).

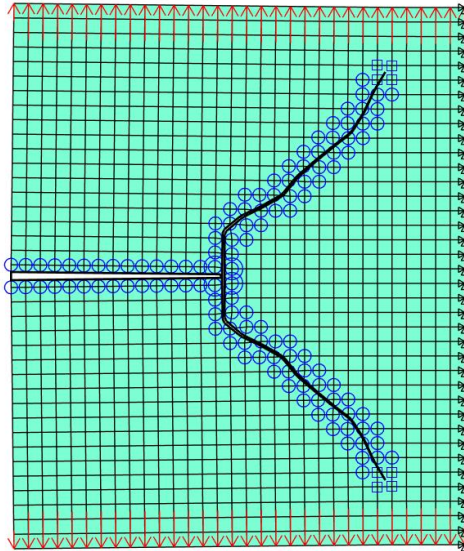


Figure. 5.1 Enriched nodes and the propagated cracks, post-processed using Matlab. The small circles represent the Heaviside enrichment nodes, the large circles represent the Junction enrichment nodes, and the boxes represent the tip enrichment nodes.

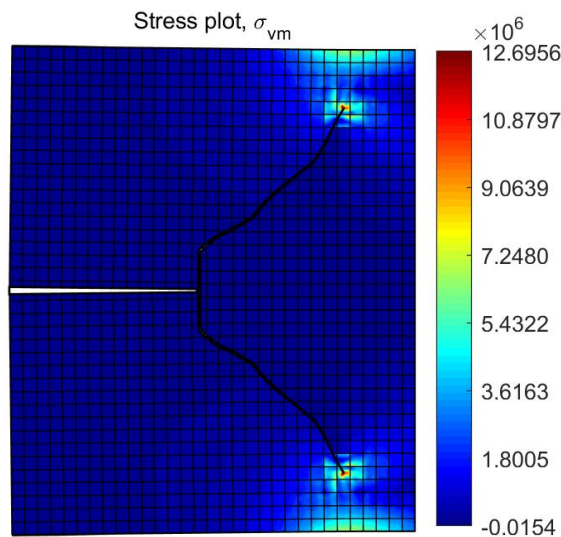


Figure. 5.2 von Mises stress contour, post-processed using Matlab.

5.2 OPTIN-2: post-processing using Paraview

Alternatively, users can post-process the results through the automatically generated *.vtk files ([tutorial_example_00001.vtk](#) - [tutorial_example_00015.vtk](#) and [tutorial_example_CRACK_00001.vtk](#) - [tutorial_example_CRACK_00015.vtk](#) in this

tutorial) using Paraview (<https://www.paraview.org/>).

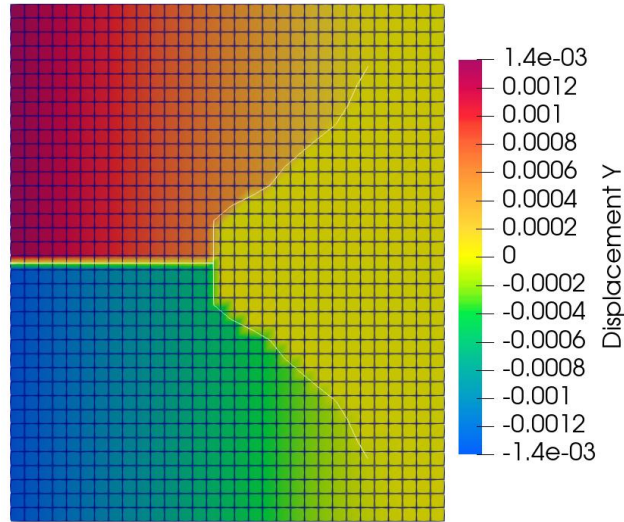


Figure. 5.3 Displacement contour in *Y*-direction, post-processed using Paraview.

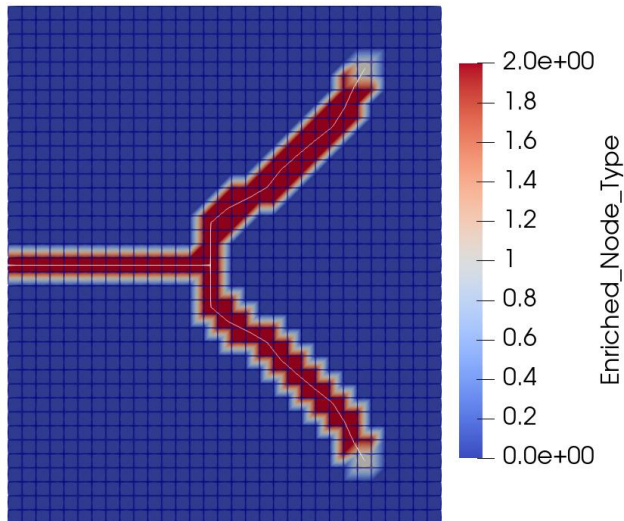


Figure. 5.4 Enriched element type, post-processed using Paraview.