Instructions for Direct FE^2 implementation of heterogeneous thin plate structures

The sample files are for an example in:

> Xu et al, 2022. Direct FE2 for concurrent multilevel modelling of heterogeneous thin plate structures. Computer Methods in Applied Mechanics and Engineering 392, 114658.

If you find the files useful / have used the files for your work, please cite the abovementioned paper. For academic use only. Thank you.

Note: the DFE^2 codes can only be successfully applied to shell element in the XY plane. If you want to apply to more complex cases, some minor changes may need to be made. It includes two steps. The first step is to construct the Direct FE^2 model. The purpose is to ensure that the micro RVE model is successfully placed at the Gauss points of the macro element. The second step is to establish MPCs between micro RVE model and the nodes of macro element. When these two steps are done, the Direct FE^2 model is ready.

1. First step: constructing the Direct FE^2 model

(1) Build the macro-scale model in ABAQUS as shown in Fig.1. Ensure that the macro-model is bigger than the RVE micro-structure.

(2) Mesh it as per requirement. These elements serve as the macro-element, see Fig.1.

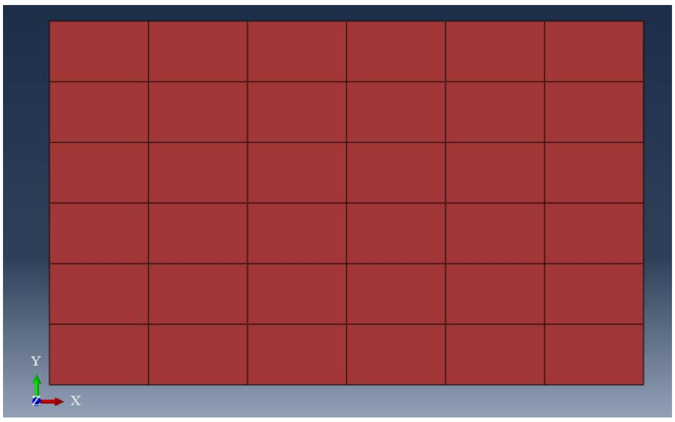


Fig.1 Macroscale shell element

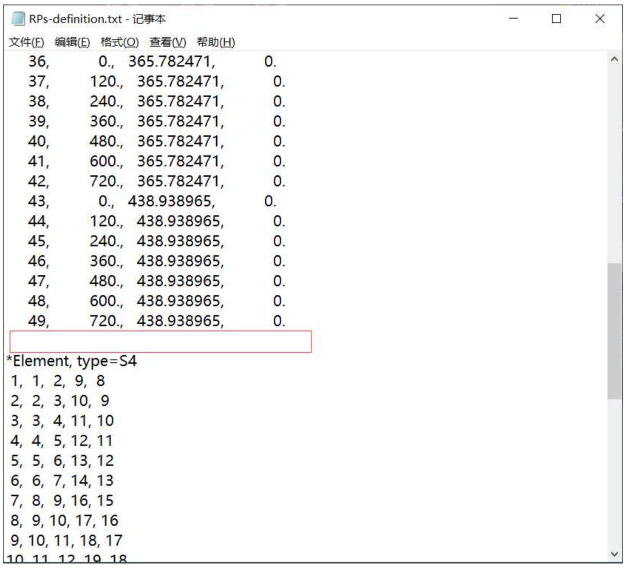
(3) Create an instance in the assembly module.

(4) Create a job and write the input file for the job.

(5) Copy the data under '*Node' and * '*Element, type=S4' (name will vary depending on the element type).

(6) Save as '.txt' (Eg: RPs-definition.txt) file.

(7) Ensure spacing between *Nodes and * Element as highlighted by the red line shown below.



RPs-definition.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
36,      0.,  365.782471,      0.
37,    120.,  365.782471,      0.
38,    240.,  365.782471,      0.
39,    360.,  365.782471,      0.
40,    480.,  365.782471,      0.
41,    600.,  365.782471,      0.
42,    720.,  365.782471,      0.
43,      0.,  438.938965,      0.
44,    120.,  438.938965,      0.
45,    240.,  438.938965,      0.
46,    360.,  438.938965,      0.
47,    480.,  438.938965,      0.
48,    600.,  438.938965,      0.
49,    720.,  438.938965,      0.

*Element, type=S4
1,  1,  2,  9,  8
2,  2,  3, 10,  9
3,  3,  4, 11, 10
4,  4,  5, 12, 11
5,  5,  6, 13, 12
6,  6,  7, 14, 13
7,  8,  9, 16, 15
8,  9, 10, 17, 16
9, 10, 11, 18, 17
10, 11, 12, 19, 18
```

Fig.2 Ensure spacing between *Nodes and * Element

(8) Update the headings (highlighted by green boxes) as per your .txt file in lines where keywords is defined as shown below. Update the f1 variable in the file as per the .txt file you have saved above. Update the folder in which you saved the above file in the os.chdir() command.

```python
os.chdir('D:\ABAQUS FILE\Benchmark\Python')
keyword=['*Node','*Element, type=S4'];
f1=open('RPs-definition.txt','r+')#input file name
```

(9) Create the RVE micro-structure (RVE, after scaling) as shown in Fig.3 and ensure the RVE is centred at the origin (0, 0) and define the center point of RVE (M0 (red mark), geometric set or node set). Note that M0 is not necessarily on the RVE

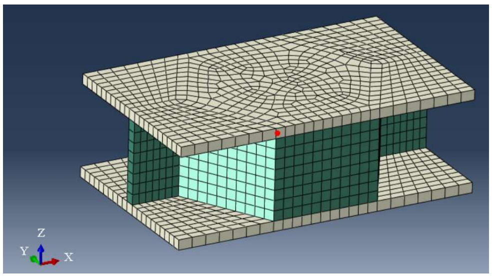


Fig.3 Micro-structure (M0)

(10) In order to control the rigid body displacement of the RVE, a point on the RVE must be constrained. Here, select M00 point on the panel, as shown in Fig.4.

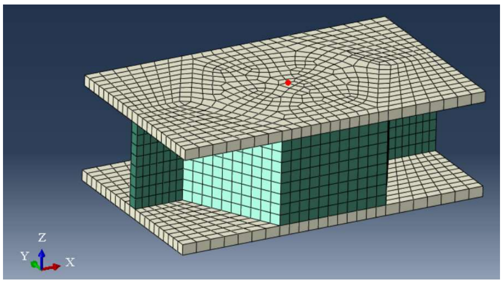


Fig.4 Micro-structure (M00)

(11) Update the value of position_z. The purpose is to move RVE vertically such that the z-coordinate of M0 is 0.

```
    print 'N=',i+1
    for ii in range(1,(len(GP_RVEs)+1)):
        N1,N2,N3,N4=tsi_eta_shape_fn(GP_RVEs[ii-1][0],GP_RVEs[ii-1][1]);
        print 'shape function',N1,N2,N3,N4
        print 'X_coordinate',X1,X2,X3,X4
        print 'Y_coordinate',Y1,Y2,Y3,Y4
        position_x=N1*X1+N2*X2+N3*X3+N4*X4
        position_y=N1*Y1+N2*Y2+N3*Y3+N4*Y4
        position_z=0
        print 'RVE_coordinate',position_x,position_y,position_z
```

(12) Run the ‘Direct-FE2-plate-element-set-assembly.py‘ python file in Python IDE (Spyder 2.7), not in ABAQUS. The following inputs are required.

```
name of element_ connectivity .dat file?4x40_quad_full
name of Fe_square abaqus model generation .py file?4x40_quad_full
input 0 for full integration or 1 for reduced integration FE_2 element:0
Enter 1 for linear super elements or Enter 2 for quadratic super elements:2
Which model?Model-2
Which part?Part-1
```

(13) Input the file names for element connectivity file (needed for applying the MPCs for the second step) and FE^2 model generation python file when prompted.

(14) Update ‘0’ for full integration super element- 4 Gauss point RVEs (this case) or ‘1’ for reduced integration super element(1 Gauss point RVE at centre) as shown below when prompted by the code.

(15) Update 1 for linear interpolated super elements and 2 for quadratic interpolated super elements

(16) The code will prompt you to input Model name (Eg: Model-1) in which you want to create the FE2 model and part-name (Eg: Part-2) of the RVE. In the example above, Model-1 needs to contain Part-1. If this error message pops up-‘Part dimensionality does not match the dimensionality of the existing instances’- it means that your RVE and macro part are of different dimensionality (2D, 3D). Go to part definition and edit the dimensionality.

Note: After running the first Python script, two files will be generated here (.dat and .py).

(17) Run the .py script generated in the previous step in ABAQUS. The Direct FE^2 model can be seen in the assembly as shown in Fig.5, corresponding to the macro-scale part.

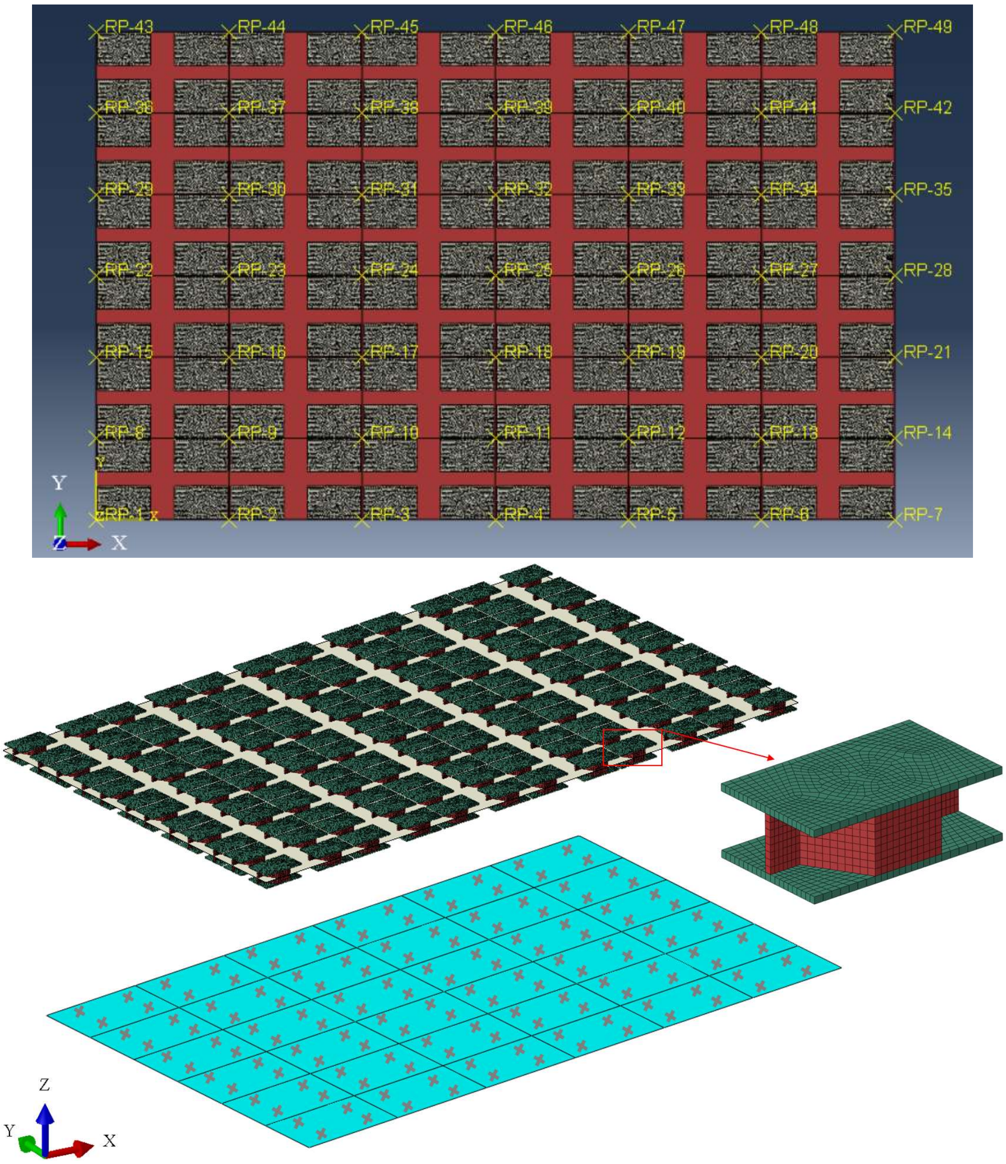


Fig.5 The Direct FE^2 model after constructing RVE

(18) If all the RVEs are displaced uniformly due to part not created at the origin (you can move it with a command. Please update the code if possible), calculate the components of the required translation for one of the RVE centres and translate all the RVE part instances accordingly.

(19) The first step is completed.

2. Second step: establishing the MPCs between micro RVE and macro element

(1) Update the "np.array" in 'Direct-FE2-Kirchhoff-Love -plate-element-set-PBC.py' based on the element connectivity file.dat in the first step. 15.

```
nodal_connectvity=np.array([[9, 8, 1, 2],
[10, 9, 2, 3],
[11, 10, 3, 4],
[12, 11, 4, 5],
[13, 12, 5, 6],
[14, 13, 6, 7],
[16, 15, 8, 9],
[17, 16, 9, 10],
[18, 17, 10, 11],
[19, 18, 11, 12],
[20, 19, 12, 13],
[21, 20, 13, 14],
[23, 22, 15, 16],
[24, 23, 16, 17],
[25, 24, 17, 18],
[26, 25, 18, 19],
[27, 26, 19, 20],
[28, 27, 20, 21],
[30, 29, 22, 23],
[31, 30, 23, 24],
[32, 31, 24, 25],
[33, 32, 25, 26],
[34, 33, 26, 27],
[35, 34, 27, 28],
[37, 36, 29, 30],
[38, 37, 30, 31],
[39, 38, 31, 32],
[40, 39, 32, 33],
[41, 40, 33, 34],
[42, 41, 34, 35],
[44, 43, 36, 37],
[45, 44, 37, 38],
[46, 45, 38, 39],
[47, 46, 39, 40],
[48, 47, 40, 41],
[49, 48, 41, 42]])
```

(2) Run the 'Direct-FE2-Kirchhoff-Love -plate-element-set-PBC.py' (Appendix 2) python script **in ABAQUS**. Here are some parameters that need to be revised according to your model. For this case, the details are shown in Fig.6. It is worth emphasizing that the general fixed Macro thickness value is 1, and then input in the RVE thickness according to the volume scaling factor.

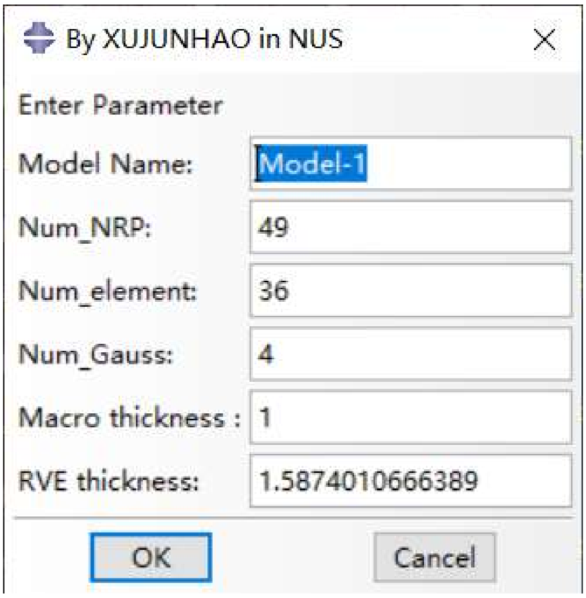


Fig.6 Some parameters need to be modified

(3) The second step is completed.

Note: please be careful with the tolerance value in Direct-FE2-Kirchhoff-Love -plate-element-set-PBC.py. This value depends on the quality of RVE mesh. For this case, the tolerance value is equal to 0.1. Generally, this value can be set very small (i.e. 10e-5).

Appendix

1. Direct-FE2-plate-element-set-assembly.py
2. Direct-FE2-Kirchhoff-Love -plate-element-set-PBC.py
3. RPs-definition.txt
4. Direct FE2 for aluminum honeycomb sandwich panel.inp