Instructions to run an Adaptive Direct FE2 analysis on Abaqus using Python Scripts

Macroscale input file

This input file describes the macroscale problem to be solved. It can be generated using Abaqus CAE and should contain the following information:

1. Macroscale structure’s mesh (only CPS4 for current scripts, minor modifications required for other element types).
2. Material properties need not be assigned.
3. Macroscale boundary conditions, applied using nodal sets.
4. Any interactions between macroscale entities.

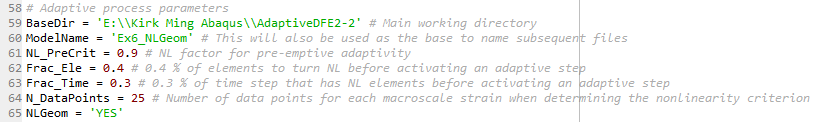
Microscale input file

This input file describes the microscale RVE of the macroscale structure to be analysed. It can be generated using Abaqus CAE and should contain the following information:

1. Microscale RVE’s mesh, with one Instance placed in the Assembly (only CPS4 for current scripts, minor modifications required for other element types).
2. Material properties, assigned using element sets.
3. Any interactions between microscale entities.

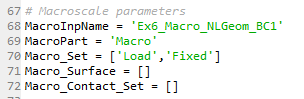
Python scripts

Python scripts are used to automate the Adaptive Direct FE2 analysis. In addition to the two input files above, some information is required for the central control script (‘DFE2\_Adaptive\_Main.py’):



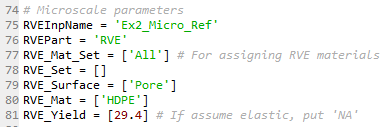
Adaptive process parameters

1. Line 59 – working directory where the scripts and user-provided input files are located. The files generated for the adaptive analysis will also be placed there.
2. Line 60 – name for the model or adaptive analysis, will merely be used to name the analysis files.
3. Line 61 – pre-emptive factor for adapting, refer to Section 4.5.1 of the paper.
4. Line 62 – , fraction of macroscale elements to turn nonlinear before an adaptive process is triggered, refer to Section 4.5.2 of paper. Set to a very small non-zero number to immediately trigger the adaptive process when new nonlinear macroscale elements are present.
5. Line 62 – , fraction of load/time to elapse with nonlinear macroscale elements present before the adaptive process is triggered, refer to Section 4.5.2 of paper. Set to 1.0 if a simple adaptivity is desired, where the full macroscale problem is solved to completion to estimate all possible nonlinear macroscale elements first, before the adaptive process is triggered.
6. Line 64 – number of data points used for each volume averaged strain when obtaining the nonlinear transition criterion.
7. Line 65 – geometrically nonlinear analysis, either ‘YES’ or ‘NO’.



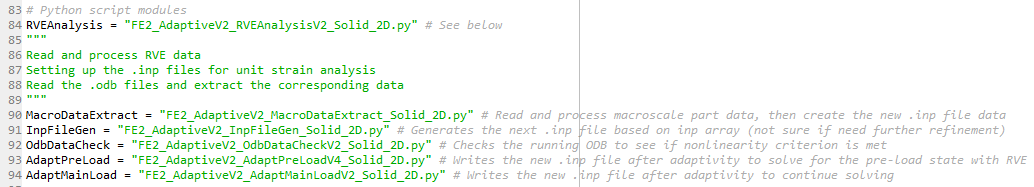
Macroscale input parameters

1. Line 68 – name of user-provided macroscale input file without the .inp suffix
2. Line 69 – name of macroscale part
3. Line 70 – name of macroscale sets, including those where boundary conditions are applied
4. Line 71 – name of macroscale surface, if any
5. Line 72 – name of macroscale sets to application of contact or other interactions, if any



Microscale input parameters

1. Line 75 – name of microscale input file without the .inp suffix
2. Line 76 – name of microscale part
3. Line 77 – name of microscale sets for assigning material properties
4. Line 78 – name of other microscale sets, if any
5. Line 79 – name of microscale surface, if any
6. Line 80 – name of microscale materials, in the same order as the sets in Line 77
7. Line 81 – yield stress (or other critical nonlinear value) for each microscale material, in the same order as the sets in Line 77. If a material is assumed to be always elastic, put ‘NA’.



Python scripts modules for subfunctions (refer to Appendix A in the paper)

1. Line 84 – name of Python script to perform RVE unit strain linear analysis
2. Line 90 – name of Python script to initialise the main multiscale analysis
3. Line 91 – name of Python script to write Abaqus .inp files
4. Line 92 – name of Python script to perform nonlinearity check
5. Line 93 – name of Python script to perform phase I of the adaptive process
6. Line 94 – name of Python script to perform phase II of the adaptive process

Initialising the analysis

Once both user-provided input files and the central control script is ready, the Adaptive Direct FE2 analysis can be initiated. This is done by executing the central control script using the following command:

execfile(‘DFE2\_Adaptive\_Main.py’)

The execution is recommended to be done on Abaqus CAE nogui for improved computational efficiency. The analysis will then proceed by itself. It will provide updates as it proceeds to the next stage, which will also be recorded in a time log file.