**Applying periodicity boundary conditions to a 2D plane strain finite element model using ABAQUS and MATLAB**

J. Michael Shockley, LaMCoS, INSA-Lyon, james.shockley@mail.mcgill.ca  
17 November 2015

Necessary MATLAB functions: **ReadandWriteINPFunction.m,** **EdgeElements.m**, **ReadDATSandbox.m, SelectCoords.m**

**Motivation and summary**

This document provides a brief description of a means by which to generate and execute a 2D plane strain finite element model of a unit cell Al2O3 particle embedded in a matrix of Al. Periodicity conditions, which directly tie the displacement of one side of the unit cell to the opposite side, are not normally necessary for simple tensile or compressive strains put on the unit cell. However, any imposed shear strains will result in far-field hydrostatic compression and tension unless periodicity conditions are applied. As this is not readily possible with ABAQUS, particularly for fine meshes, a MATLAB code was created to apply periodicity conditions to a model (specifically to the .inp file) generated using ABAQUS. The new .inp file can be run on any version of ABAQUS. The following flow chart shows the interaction between ABAQUS and MATLAB and outlines the contents of this document.

**A.** Define model

**B.** Generate mesh and

initial input file

**C.** Generate a new input file with periodicity conditions

**ReadandWriteINPFunction.m  
EdgeElements.m**

**E.** Optional: Open the .dat file to obtain node coordinates and stress/strain results

**ReadDATSandbox.m**

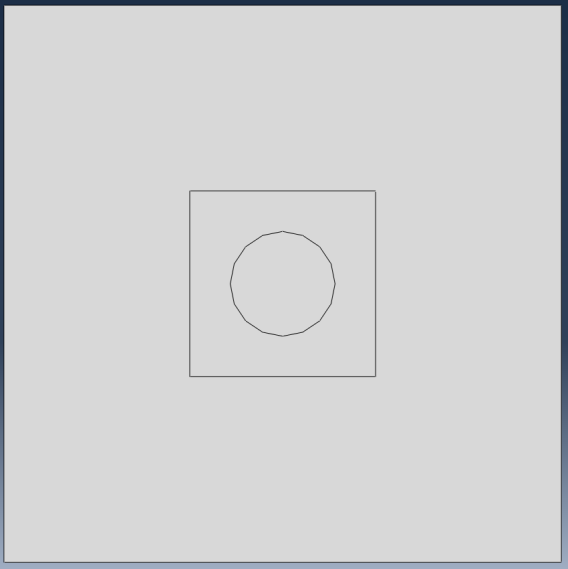
**SelectCoords.m**

**D.** Submit the job for analysis. The .odb file can be opened immediately afterward if desired.

**ABAQUS**

**MATLAB**

**A. Define model using ABAQUS CAE**



60

60

Partitions

Al2O3

Al

Al

(30, 30)

(-30, 30)

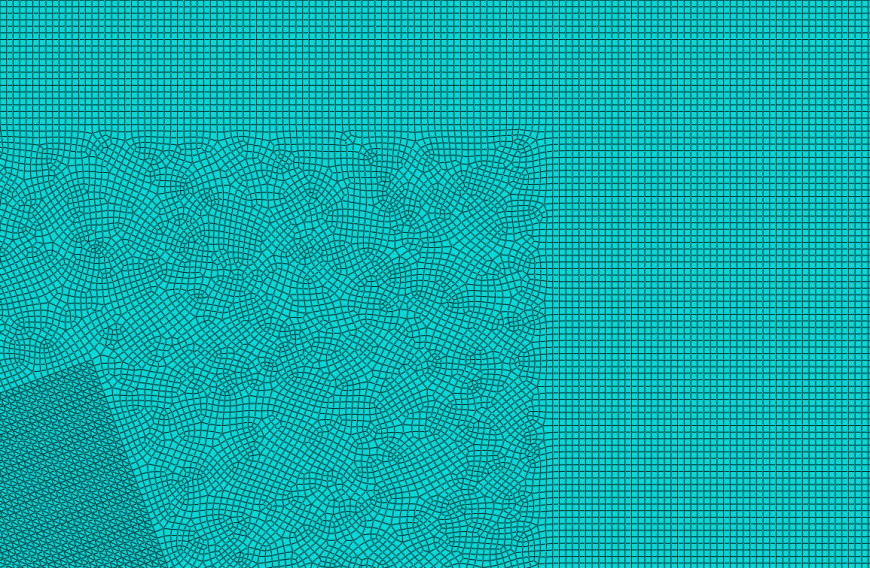
(-30, 30)

(-30, -30)

The MATLAB code will work best if the model created in ABAQUS CAE is square, 60 units by 60 units. Use partitions to create a particle of the desired morphology (circular, square, etc.) and also a square region of matrix material surrounding that particle. The use of partitions means that only one part (matrix + particle) is created, with perfect bonding at each interface.

**B. Generate mesh and initial input file using ABAQUS CAE**

Any desired level of mesh refinement (fine, coarse) may be used. Furthermore, *the mesh must be perfectly symmetrical on the left and right edges*. To accomplish this, the medial axis meshing algorithm should be used for the outer part of the matrix. In the area immediately surrounding the particle, an advancing front algorithm should be used to account for the irregular shape of the particle. **Use CPE3 (3-node linear plane strain triangle) elements for the particle and CPE4R (4-node bilinear plane strain quadrilateral, reduced integration, hourglass control) for the matrix.** It is important to use these element types for the matrix and particle because the MATLAB code will assign material properties accordingly.



**Bulk of the matrix:**

**Medial axis algorithm**

**CPE4R**

**Matrix region around reinforcement:**

**Advancing front algorithm**

**CPE4R**

**Reinforcement: Medial** **axis algorithm**

**CPE3**

Once the mesh is generated, create the .inp file by submitting the job on CAE. The job will be aborted because no steps were be defined, but this is not a problem because the only point of doing this is to obtain the .inp file.

**C. Generate a new input file with periodicity conditions using MATLAB**

The MATLAB code goes by the name of “**ReadandWriteINPFunction.m**” and is divided into four parts. The function “**EdgeElements.m**” must be present in the directory as well, but if it is lost, there is a copy of its code written in the comments of “**ReadandWriteINPFunction.m**.” The following are the parts of the “**ReadandWriteINPFunction.m**” code:

**ReadandWriteINPFunction.m part 1:** The .inp file is opened and the node and element coordinates are extracted. The user inputs the desired amount of compressive deformation and shear deformation according to the number of units in the ABAQUS model. **Importantly, the amount of shear deformation is adjusted slightly by the program from the user input value** according to the amount of compressive deformation in order to maintain the same degree of strain for a given amount of shear deformation. A maximum of 1 unit of deformation is possible because excessive element deformation and divergence arise at higher levels of deformation.

Strain calculation of compression only:  
1 unit of compressive deformation / 60 units height = 1.7 % compressive strain

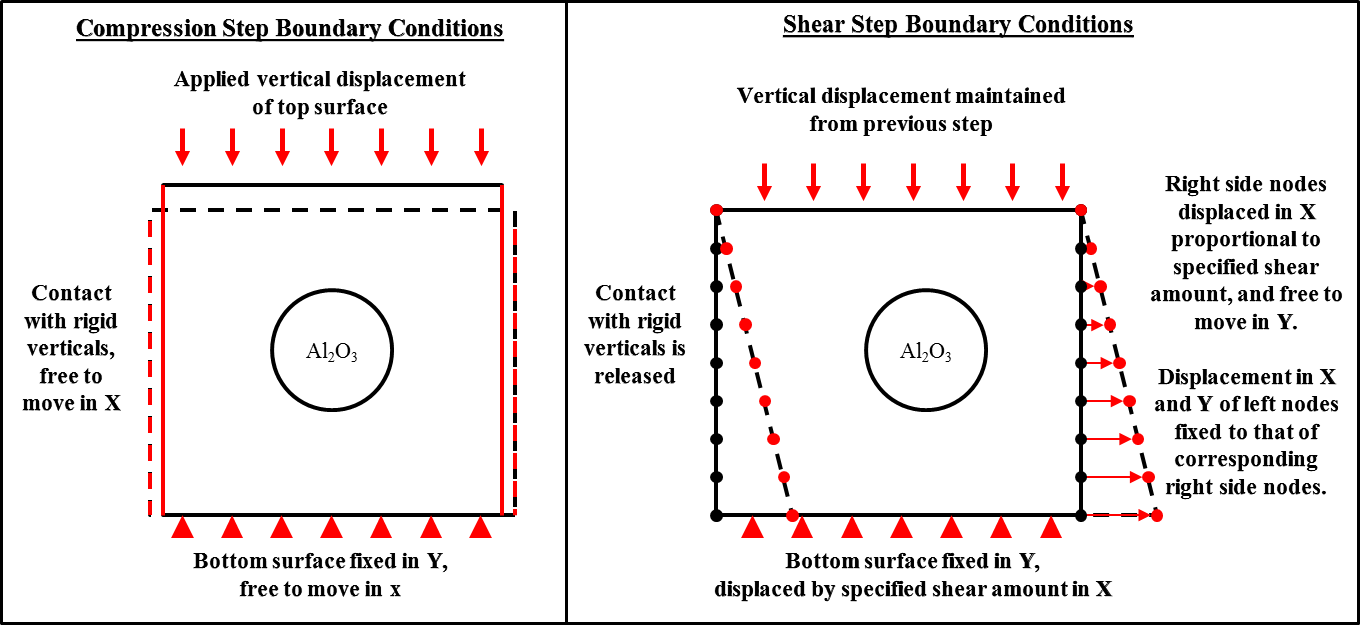
Strain calculation of shear only:  
1 unit of shear deformation / 60 units height = 1.7 % shear strain

Strain calculation compression + shear:

1 unit of compressive deformation / 60 units height = 1.7 % compressive strain  
1 unit of shear deformation / 59 units height \* (59/60) = 1.7 % shear strain

**ReadandWriteINPFunction.m part 2:** The periodicity conditions are calculated according to the amount of compression and shear deformation the user has input. Periodicity in this context fixes the displacement of the node on the left to that of the corresponding node on the right.

**ReadandWriteINPFunction.m part 3:** The new input file is written. The boundary conditions are as follows (see diagram below). In the first step (compression), the bottom surface is fixed in Y but permitted to move in X, and the top surface is displaced in Y according to the amount of compressive deformation specified by the user. The left and right sides are fixed in a frictionless contact to rigid vertical surfaces which are permitted to move in X but not Y. Each pair of corresponding nodes on the left and right side is grouped into a node set along with a freely moving dummy node not connected to anything in the model, and the displacement of each set is fixed by the following equation:



In the second step (shear), the top surface is fixed in place in X and Y. The bottom surface is displaced (as a velocity over the step time of 1 due to this being a relative, not absolute displacement) in X according to the amount of shear deformation specified by the user. The contact with the rigid verticals is released and each node on the right is displaced in X (again, as a velocity) the amount calculated for the periodicity conditions. The dummy nodes for this step are fixed in place, forcing the displacement of each node on the right to be the same as its corresponding node on the left.

The .odb output file is very large (~1 GB) so instead the a select set of results are written to the \*.dat file, consisting of the coordinates of each node and the hydrostatic pressure (PRESS), von mises stress (MISES) and plastic equivalent strain (PEEQ). These outputs are called in the input file, and as desired, could be changed. One option is to replace them all with “S” and calculate the desired stress components from the S11, S22, S33, and S12 values.

The new input file will show up in the working directory and takes the form shown in Appendix 1.

**ReadandWriteINPFunction.m part 4**: Write a launcher file for the LaMCoS computational cluster. The file will show up in the working directory. The launcher file takes the form shown in Appendix 2.

**D. Submit the job to ABAQUS for analysis**

Using ABAQUS Command or your local computational cluster, submit the job. Once the analysis is complete, the .odb file can be opened as desired. However, in many cases this file is very large (~1 GB) and if a computational cluster is used, the file transfer times can be excessive. Therefore the final stress/strain information can be extracted from the .dat file instead, which is described in the next section.

**E. Optional: Open the .dat file using MATLAB**

The .dat file contains output information in three tables of the following form:

THE FOLLOWING TABLE IS PRINTED FOR ALL ELEMENTS WITH TYPE CPE3 AVERAGED AT THE NODES

NODE FOOT- MISES PRESS PEEQ AC YIELD

NOTE

THE FOLLOWING TABLE IS PRINTED FOR ALL ELEMENTS WITH TYPE CPE4R AVERAGED AT THE NODES

NODE FOOT- MISES PRESS PEEQ AC YIELD

NOTE

THE FOLLOWING TABLE IS PRINTED FOR ALL NODES

NODE FOOT- COOR1 COOR2

NOTE

By running “**ReadDATSandbox.m”,** information in the tables is extracted and organized such that each node is listed with its post-deformation coordinates, von Mises, hydrostatic pressure, and PEEQ value. Then, this information is written to individual text files with the form specified below, which can be plotted in a spreadsheet-based graphing program (such as Origin).

“~ -Mises.txt”

COOR1, COOR2, MISES

“~ -Press.txt”

COOR1, COOR2, PRESS

“~ -PEEQ.txt”

COOR1, COOR2, PEEQ

“~ -Triax.txt”

COOR1, COOR2, PRESS/MISES

**Appendix 1**: Sample ABAQUS input file generated by the MATLAB script.

\*HEADING

\*NODE

(coordinates redacted)

\*ELEMENT, TYPE=CPE3  
(coordinates redacted)

\*ELEMENT, TYPE=CPE4R

(coordinates redacted)  
\*ELSET  
(various element sets are defined as necessary)  
\*NSET  
(various node sets are defined as necessary including the nodes on the top side, bottom, left, right, dummy nodes, etc.)  
\*SURFACE  
(slave and master surfaces are defined for the contact between the rigid verticals)  
\*SOLID SECTION, ELSET=REINFORCEMENT-SET, MATERIAL=AL2O3

\*MATERIAL, NAME=AL2O3

(material parameters are defined here for the Al2O3)  
\*SOLID SECTION, ELSET=MATRIX-SET, MATERIAL=AL

\*MATERIAL, NAME=AL

(material parameters are defined here for the Al)  
\*Surface, type=SEGMENTS, name=m\_Surf-left

START, -30., 35.

LINE, -30., -35.

\*Rigid Body, ref node=LEFTREFPT, analytical surface=m\_Surf-left

\*Surface, type=SEGMENTS, name=m\_Surf-right

START, 30., -35.

LINE, 30., 35.

\*Rigid Body, ref node=RIGHTREFPT, analytical surface=m\_Surf-right

\*Surface Interaction, name=IntProp-sansfriction

1.,

\*Friction

0.,

\*Surface Behavior, no separation, pressure-overclosure=HARD

\*Contact Pair, interaction=IntProp-sansfriction, type=SURFACE TO SURFACE

S\_SURF-LEFT, m\_Surf-left

\*Contact Pair, interaction=IntProp-sansfriction, type=SURFACE TO SURFACE

S\_SURF-RIGHT, m\_Surf-right

\*Equation

3

LEFTNODE1606,1,1, RIGHTNODE2206,1,-1, 368393,1,-1

(additional equations redacted)  
\*STEP, NLGEOM=YES

\*STATIC

0.01, 1., 1e-05, 1.

\*BOUNDARY

TOPNSET, 2,2, -1.0

BOTHREFPTS, 2,2,

\*BOUNDARY, type=VELOCITY

BOTNSET, 2,2,

\*OUTPUT, FIELD, VARIABLE=PRESELECT

\*OUTPUT, HISTORY

\*ENERGY PRINT

\*END STEP

\*STEP

\*STATIC

0.01, 1., 1e-05, 1.

\*MODEL CHANGE, TYPE=CONTACT PAIR, REMOVE

S\_SURF-LEFT, m\_Surf-left

\*MODEL CHANGE, TYPE=CONTACT PAIR, REMOVE

S\_SURF-RIGHT, m\_Surf-right

\*Boundary, type=VELOCITY

(the shear displacement of the bottom surface and each node on the side is entered here, and the dummy nodes are fixed in place)  
\*NODE PRINT, FREQUENCY=999  
COORD  
\*EL PRINT, POSITION=AVERAGED AT NODES, FREQUENCY=999  
MISES, PRESS, PEEQ   
\*OUTPUT, FIELD, VARIABLE=PRESELECT, FREQUENCY=999  
\*ELEMENT OUTPUT  
MISES, PRESS, PEEQ  
\*OUTPUT, HISTORY  
\*ENERGY PRINT  
\*END STEP

**Appendix 2**: LaMCoS cluster launcher file.

#!/bin/bash  
#PBS -l nodes=1:ppn=4:compute,walltime=24:00:00,mem=2gb  
#PBS -N JMS-Job-Model-d0-periodicity\_Comp\_(filled in by MATLAB)\_Shear\_(filled in by MATLAB)  
#PBS -j oe  
# Receive email at beginning and end of job :  
#PBS -M firstname.lastname@insa-lyon.fr  
#PBS -m abe  
# Set Intel compilers for user subroutines  
module load intel/11.1.080  
# Load Abaqus module to define executable path  
module load abaqus/6.11.2  
cd $PBS\_O\_WORKDIR  
INPNAME=Job-Model-d0-periodicity\_Comp\_( filled in by MATLAB)\_Shear\_( filled in by MATLAB)  
JOBNAME=JMS-Job-Model-d0-periodicity\_Comp\_( filled in by MATLAB)\_Shear\_( filled in by MATLAB)  
abaqus input=$INPNAME.inp job=$JOBNAME cpus=${PBS\_NP} interactive