Applying Periodic Boundary Conditions in Finite Element Analysis

Weidong Wu, Joseph Owino

Department of Civil & Chemical Engineering, University of Tennessee Chattanooga

Ahmed Al-Ostaz, Liguang Cai

Department of Civil Engineering, The University of Mississippi

Abstract: *Periodic boundary conditions (PBC) are a set of boundary conditions that can be used to simulate a large system (i.e. bulk material) simply by modeling a finite Representive Volume Element (RVE). PBC has been favored among many researchers and practicing engineers in the study of various materials. Unfortunately it remains vague on how PBC should be applied properly in FEA packages such as Abaqus. In this article, we explicitly show the detailed procedures one could easily follow to define PBC in Abaqus through a simple example which is given in the forms of input file. A robust Matlab script which can be used to pair a large number of randomly distributed nodes on two opposite surfaces of a 3D RVE is also supplied to facilitate easy application of PBC.*

Key Words: *periodic boundary conditions, finite element analysis, Abaqus, constraint, Matlab*

1. Introduction

To study the properties of a bulk system such as a material, we run computer simulation as using molecular dynamic method to investigate the elastic properties of polymer. We could use a sufficient large simulation box such that few molecules are on the surface of such a "box", unfortunately this practice will need extremely extensive computational time and often it is impossible. On the other hand, a much smaller simulation box can be used, but most molecules would be near the edge of the simulation box. The solution to this dilemma is applying periodic boundary conditions.

Periodic boundary conditions are commonly applied in molecular dynamics, dislocation dynamics and materials modeling to eliminate the existence of surface and avoid huge amount of molecules or large size of simulation box. In periodic boundary conditions, an infinite lattice system is formed simply by repeating the simulation box throughout space (Figure 1). When a molecule leaves the box, one of its images will enter through the opposite face with exactly the same way and direction. The molecules in the simulation box will conserve and the system can be thought of as having no surface.

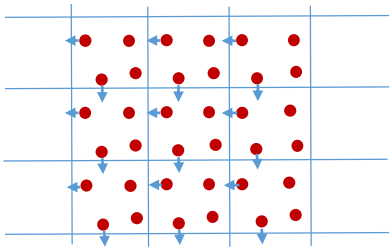

Figure 1: A 2-D periodic boundary condition cell

Composite material has appealing properties which is more and more widely used especially in aerospace industry. The Boeing 787 Dreamliner can be the best example of this claim. Finite element method is a time and cost effective numerical approach which has been applied to study composite materials with great success. (Ochoa and Reddy, 1992; Alfano and Crisfield, 2001). PBCs are favored by researchers for modeling the composite materials for a long time (Al-Ostaz, 1997; Jiang, 2001; Pegoretti et al, 2002; Wang, 2007; Melro, 2013). A randomly distributed inclusion-matrix composite RVE is shown in Figure 2. A strain controlled PBC may be specified for this RVE by the following equations (1)-(3):

$$u(x+L) = u(x) + \varepsilon^0 x \qquad \forall x \in B_\delta^2 \qquad (1)$$

$$u(x+L) = u(x) \qquad \forall x \in B_\delta^1 \qquad (2)$$

$$t(x+L) = -t(x) \qquad \forall x \in B_\delta^1 \qquad (3)$$

where u is the displacement at x, ε^0 is strained applied to the RVE, t is traction force and B_δ^1 represents the boundary B whose normal is along "1" direction.

Equation (1) represents kinematic boundary conditions, and B_δ^1 is subjected to periodic boundary condition. These are strain controlled PBC. Equation (1) can be simply understood as a strain ε^0 is applied to RVE as shown in the Figure 2.

Unfortunately, it has been unclear to many researchers how PBC may be properly defined in finite element modeling of composites. Question of how to define periodic boundary conditions during modeling process is very frequently asked but no concrete answer has been given on internet. Among many of the powerful FEA packages, Abaqus is a strong candidate which can be used for simulation of composite materials. In this article, we explicitly explain how PBC can be properly defined in Abaqus.

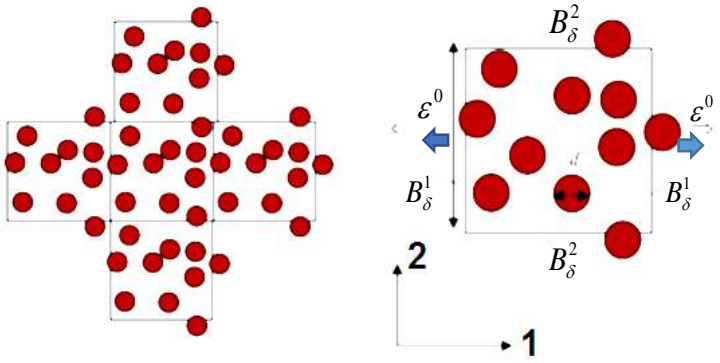

Figure 2. A randomly distributed inclusion-matrix composite RVE (Al-Ostaz et al. , 2007)

2. Applying PBC in FEA

2.1 Constraint equations in Abaqus

We apply PBC through linear constraint in Abaqus (Abaqus 6.12 user manual). Multi points may be constrained by a general linear combination of nodal variables (such as displacements at different nodes), the summation of the product of a coefficient and the corresponding nodal variable is equal to zero. Specifically, we define a general linear homogeneous equation

$$A_1 u_i^P + A_2 u_j^Q + ... + A_N u_k^R = 0 \quad (4)$$

where R is node, k is degree of freedom, i.e. 1, 2, or 3 which represent x, y ,z directions, A_N is a constant coefficient that define the relative motion of nodes. In abaqus, we use the command:

```
* Equation
```

to define the above general linear constraints of N points. The command line data is shown in Figure 3. The data lines in Figure 3 below *Equation may be contained in a separate input file i.e. file.inp, a command line

```
*Equation, input=file
```

is then included in the model input file.

2.2 Dummy node

In order to apply PBC using constraint equations described above, one abstract concept of "dummy node" is introduced in Abaqus. We rewrite Equation (4) by replacing zero on the right side of the equation (4) by a nonzero value $\hat{u}$

$$A_1 u_i^P + A_2 u_j^Q + ... + A_N u_k^R = \hat{u} \quad (5)$$

where $\hat{u}$ is a prescribed value such as a strain or displacement.

The prescribed value $\hat{u}$ will be applied through a dummy node, Z, which is not attached to any other part in model. Though the dummy node will not be connected to any part in a model, a reference point with arbitrary coordinates should be defined to represent the dummy node. The dummy node can be specified as a boundary condition with a value $\hat{u}$ at a certain direction. One has to define a load step in order to apply this value as a boundary condition since in initial step, users are not allowed to specify a nonzero displacement value as boundary.

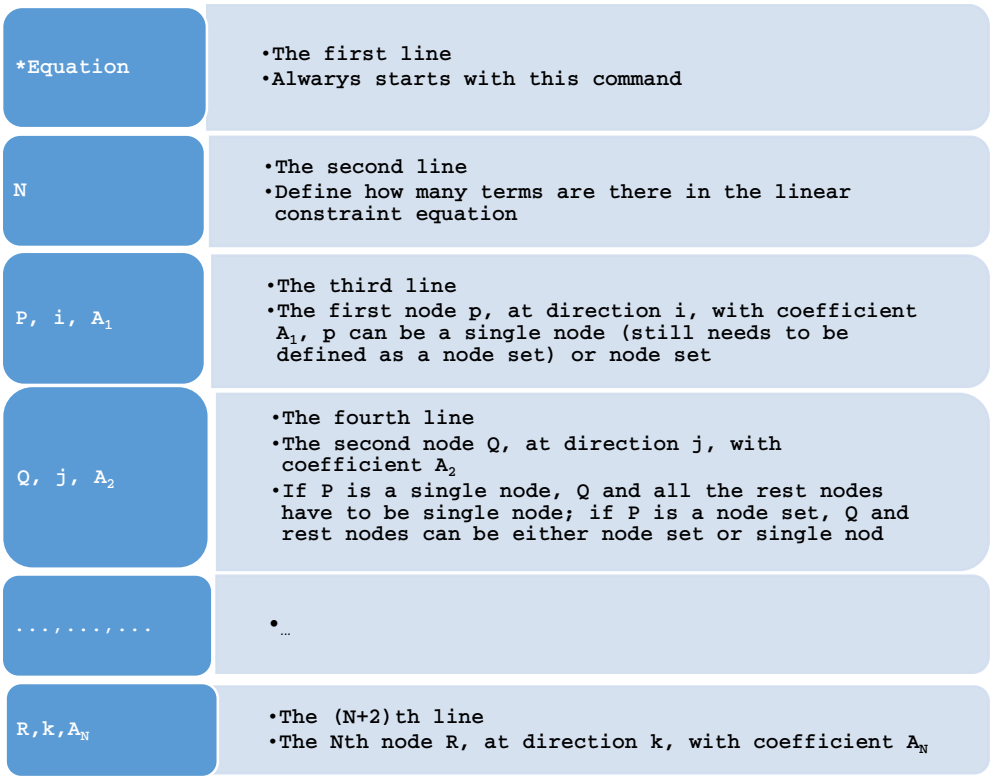

Figure 3. Definition of *equation

2.3 An example to define PBC in Abaqus

A one by one (1×1) square RVE is subjected to the following PBCs:

$$u_1^{Left} - u_1^{Right} = 0.01 \qquad (6)$$

$$u_2^{Top} - u_2^{Bottom} = 0 \qquad (7)$$

We use “Left” to represent node set “Left” which includes nodes “ 1,7,13,19,25,31”, “Right” is the node set which has nodes “6, 12, 18, 24, 36”,“ Top” is the node set which includes nodes “ 32, 33, 34, 35” and “ Bottom” is a node set having nodes “2, 3, 4, 5”.

In order to apply the constraint equations (6) and (7), one should follow the steps below:

Step 1:

Introducing a dummy node Z and rewrite the equation

$$A_1 u_i^P + A_2 u_j^Q + ... + A_N u_k^R = \hat{u} \qquad (8)$$

as

$$A_1 u_i^P + A_2 u_j^Q + ... + A_N u_k^R - \hat{u}_m^Z = 0 \qquad (9)$$

which means applying a displacement value $\hat{u}$ to a dummy node Z at direction m.

Specifically for our simple example:

$$u_1^{Left} - u_1^{Right} = 0.01 \qquad (10)$$

will be rewritten as

$$u_1^{Left} - u_1^{Right} + u_1^Z = 0 \qquad (11)$$

where

$$u_1^Z = -0.01 \qquad (12)$$

Equation (11) is used to constrain degree of freedom 1 at node Z (1000 for this example) to 0.01. Equation (12) requires defining a boundary value in history modal (i.e. assigning a displacement value “-0.01” to dummy node 1000 in load module within a load step other than in initial step)

Step 2:

Add the following command lines in **modal data**, it is vitally important to put these command lines in an appropriate place (refer to Appendix A of sample input file)

```
*Equation
3                    **   equation has 3 terms
Left, 1, 1           **   left surface node set, dof =1, coeff. = 1
Right, 1, -1         **   right surface node set, dof =1, coeff. = -1
1000, 1, 1           **   dummy node Z=1000, dof =1, coeff. =1

*Equation
3                    **   equation has 3 terms
Top, 2, 1            **   left surface node set, dof =2, coeff. = 1
Bottom, 2, -1        **   right surface node set, dof =2, coeff. = -1
2000, 2, 1           **   dummy node Z=2000, dof =2, coeff. =1
```

One needs to make sure to define a big enough dummy node number such that it does not have conflict with any other node numbers in his model. Numbers 1000 and 2000 are used in this example.

Step 3:

Prescribe boundary conditions for dummy node Z (e.g. “1000” and “2000” in here).

Add the following command lines in history data, which means we need to define a load step first, then specify the magnitude of displacement for the dummy node in that load step.

```
*Boundary
1000, 1, 1, -0.01
** at dummy node 1000 prescribed boundary conditions
**from DOF 1 to DOF 1 as "-0.01"
*Boundary
2000, 2, 2, 0
```

2.4 Algorithm to generate paired nodes

It is nontrivial to retrieve all the nodes on the PBC target surfaces and save them in a text or dat file. After that, we use a Matlab script (Appendix B) to find the matching nodes on two opposite surfaces as “Left” and “Right” in Figure 4. The algorithm which could be used for the above purpose is:

1. Read the data file which includes all the x, y, z coordinates for all the target nodes on the left and right sides of the three-dimensional RVE (Figure 4). Save the node numbers and corresponding coordinates in a cell array.
2. Sort the nodes (including node numbers, x, y, and z coordinates at the same time, node # and coordinates have to be “tied” together) on both sides according to their Y coordinates.
3. Compute the pairwise distance between the node sets “Left” and “Right” and save the distances in a matrix.
4. Find the minimum distance between the first point on the left side and a certain point on the right side, fetch the node numbers of the two points with the minimized distance and save them in a matrix.
5. Move to the next node on the left side surface and repeat step 4 until all the nodes on the left side are read.
6. Retrieve the paired nodes and incorporate them into Abaqus input file to define “Equations”.

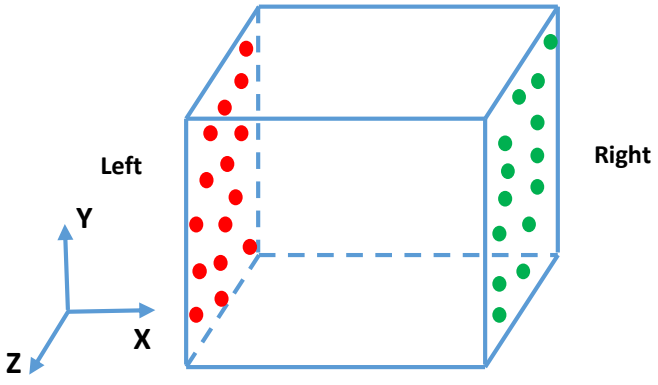

Figure 4. A 3D RVE with nodes on left and right side surfaces

3. Results and Conclusions

Figure 5 shows the stress contour results when a RVE is subjected to a strain controlled PBC (refer to Appendix A for Abaqus input file). The advantage of applying PBC to material study is to avoid large simulation box thus reduce the computational time dramatically. Applying PBC in a 3D RVE with many nodes can be tedious. With the script included in this work (Appendix B and C), one may easily find paired nodes which are needed to define constrain equations in Abaqus.

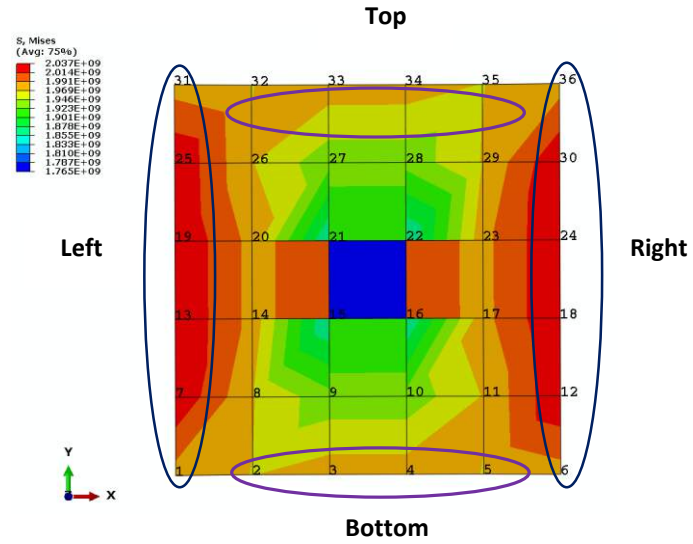

Figure 5. An 1×1 RVE model and stress contour with strain controlled PBC

4. References

1. Abaqus Users Manual, Version 6.12-1, Dassault Systémes Simulia Corp., Providence, RI.
2. Alfano, G. and Crisfield, M. A., "Finite Element Interface Models for the Delamination Analysis of Laminated Composites: Mechanical and Computational Issues," International Journal for Numerical Methods in Engineering, no. 50, 2001.
3. Al-Ostaz, A., Diwakar, A., and Alzebdeh, K., "Statistical Model for Characterizing Random Microstructure of Inclusion-matrix Composites," J Mater Sci., no. 42, 2007
4. Al-Ostaz, A. and I. Jasiuk, "Crack Initiation and Propagation in Materials with Randomly Distributed Holes, Engineering Fracture Mechanics, no. 58,1997
5. Bulatov V., Rhee, M., and Cai, W., "Periodic Boundary Conditions for Dislocation Dynamics Simulations in Three Dimensions," Mat Res. Soc. Symp. no. 653, 2001
6. Jiang, M., Alzebdeh,K., and Ostoja-Starzewski,M. "Scale and Boundary Conditions Effects in Elastic Properties of Random Composites", no.148, 2001

7. Leach A. (2001) Molecular Modelling: Principles and Applications 2nd , Prentice Hall
8. Melro, A.R., Camanho, P.P., Andrade Pires, F.M., and Pinho, S.T., "Micromechanical Analysis of Polymer Composites Reinforced by Unidirectional Fibres: Part II – Micromechanical analyses," International Journal of Solids and Structures, no. 50, 2013.
9. Ochoa, O.O. and Reddy, T.N. Finit Element Analysis of Composite Laminates, Springer, 1992.
10. Pegoretti,A. , Fambri,L., Zappini,G., Bianchetti,M., Finite Element Analysis of a Glass Fibre Reinforced Composite Endodontic post, Biomaterials, no. 23, 2002.
11. Periodic boundary conditions, http://en.wikipedia.org/w/index.php?title=Periodic_boundary_conditions&oldid = 586216413 (last visited Jan. 1, 2014)
12. User manual for the Wasser program, Boston University Center for Polymer Studies, Boston, MA.
13. Wang, X.F., Wang, X.W., Zhou, G.M. and Zhou, C.Z. "Multi-scale Analysis of 3D Woven Composite Based on Periodicity Boundary Conditions", Journal of Composite Materials, No. 41, 2007.

5. Appendix

A. Sample Input file to apply PBC in Abaqus: pbc.inp

```
** This is to use a 1*1 RVE to demonstrate how to apply PBC in Abaqus
** Units: Metric-Meter-Pa
*Heading
** Job name: PBC Model name: pbc
** Generated by: Abaqus/CAE 6.11-1
*Preprint, echo=No, model=YES, history=YES, contact=NO
**
** PARTS
**
*Part, name=RVE
*End Part
**
*Part, name=dummy-LR
*End Part
**
*Part, name=dummy-TB
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=RVE-1, part=RVE
*Node
      1, -0.5, -0.5
      2, -0.3, -0.5
      3, -0.1, -0.5
      4,  0.1, -0.5
      5,  0.3, -0.5
      6,  0.5, -0.5
      7, -0.5, -0.3
      8, -0.3, -0.3
```

```
      9, -0.1, -0.3
     10,  0.1, -0.3
     11,  0.3, -0.3
     12,  0.5, -0.3
     13, -0.5, -0.1
     14, -0.3, -0.1
     15, -0.1, -0.1
     16,  0.1, -0.1
     17,  0.3, -0.1
     18,  0.5, -0.1
     19, -0.5,  0.1
     20, -0.3,  0.1
     21, -0.1,  0.1
     22,  0.1,  0.1
     23,  0.3,  0.1
     24,  0.5,  0.1
     25, -0.5,  0.3
     26, -0.3,  0.3
     27, -0.1,  0.3
     28,  0.1,  0.3
     29,  0.3,  0.3
     30,  0.5,  0.3
     31, -0.5,  0.5
     32, -0.3,  0.5
     33, -0.1,  0.5
     34,  0.1,  0.5
     35,  0.3,  0.5
     36,  0.5,  0.5
*Element, type=CPS4R
 1,  1,  2,  8,  7
 2,  2,  3,  9,  8
 3,  3,  4, 10,  9
 4,  4,  5, 11, 10
 5,  5,  6, 12, 11
 6,  7,  8, 14, 13
 7,  8,  9, 15, 14
 8,  9, 10, 16, 15
 9, 10, 11, 17, 16
10, 11, 12, 18, 17
11, 13, 14, 20, 19
12, 14, 15, 21, 20
13, 15, 16, 22, 21
14, 16, 17, 23, 22
15, 17, 18, 24, 23
16, 19, 20, 26, 25
17, 20, 21, 27, 26
18, 21, 22, 28, 27
19, 22, 23, 29, 28
20, 23, 24, 30, 29
21, 25, 26, 32, 31
22, 26, 27, 33, 32
23, 27, 28, 34, 33
24, 28, 29, 35, 34
25, 29, 30, 36, 35
*Nset, nset=_PickedSet2, internal, generate
  1,  36,   1
*Elset, elset=_PickedSet2, internal, generate
  1,  25,   1
```

```
** Section: Section-1
*Solid Section, elset=_PickedSet2, material=steel
*End Instance
**
** Defining two dummy nodes 1000, 2000 to apply prescribed boundary condition values
**
*Instance, name=dummy-LR-1, part=dummy-LR
*Node
1000,           -10.,           10.,           0.
**This dummy node can be arbitrary
*Nset, nset=dummy-LR-1-RefPt_, internal
1000,
*End Instance
**
*Instance, name=dummy-TB-1, part=dummy-TB
*Node
2000,           10.,           0.,           0.
**This dummy node can be arbitrary
*Nset, nset=dummy-TB-1-RefPt_, internal
2000,
*End Instance
**
** Define nset "Set-dummy-LR" and "Set-dummy-TB" for the two dummy nodes
**
*Nset, nset=Set-dummy-LR, instance=dummy-LR-1
 1000,
*Nset, nset=Set-dummy-TB, instance=dummy-TB-1
 2000,
*Nset, nset=Left, instance=RVE-1, generate
  1,  31,   6
*Nset, nset=Right, instance=RVE-1, generate
  6,  36,   6
*Nset, nset=Top, instance=RVE-1, generate
  32,  35,   1
*Nset, nset=Bottom, instance=RVE-1, generate
  2,  5,  1
**Start defining constraints using equation
*Equation
3
Left, 1, 1.,Right, 1, -1.,Set-dummy-LR, 1, 1.
*Equation
3
Top, 2, 1.,Bottom, 2, -1.,Set-dummy-TB, 2, 1.
*End Assembly
**
** MATERIALS
**
*Material, name=steel
*Elastic
 2e+11, 0.3
**
**
** STEP: Step-apply-constraint
**
** Create a load step to apply a prescribed disp value to dummy node
*Step, name=Step-apply-constraint
**This step is used to apply constraint to dummy nodes
*Static
```

```
1., 1, 1e-05, 1.
** Apply prescribed boundary conditions as displacement or strain to dummy nodes
** BOUNDARY CONDITIONS
**
** Name: BC-LR Type: Displacement/Rotation
*Boundary
Set-dummy-LR, 1, 1, -0.01
** Name: BC-TB Type: Displacement/Rotation
*Boundary
Set-dummy-TB, 2, 2, 0
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

B. Matlab script to find paired nodes between two opposite surfaces of a three-dimensional (3D) RVE : pbc.m

```
%%pbc.m
%%This script is to find out the pairs of nodes on two opposite surfaces
% of a 3D RVE which can be used to define Periodic Boundary Conditions
% by using "Equations" in FEA package Abaqus
% Authors: Weidong Wu, Joseph Owino, University of Tennessee Chattanooga
% Ahmed Al-Ostaz, The University of Mississippi,Oxford MS
% Liguang Cai, Comau, Inc. Southfield MI
%%
% First the program will read the coordinates of the points on the two
% sufraces in a text file
clear all;
fileID = fopen('coordinates.txt');
formatSpec = '%s';
N = 8;
% reads file data, using the formatSpec N times
% c_h: cell header
c_h = textscan(fileID,formatSpec,N,'delimiter','|');
% Read coordinates for nodes on the two opposite surfaces
% Save them in a cell array whose first and fourth columns are node #
% rest columns are x,y,z coordinates
% c_cord
c_cord = textscan(fileID,'%d %f %f %f %d %f %f %f');
fclose(fileID);
%%
% Turn cell array which stored coordinates info. for points on left and
% right side of RVE into a sorted matrix
% Initialize matrix
cordMatrix=[];
for i=1:N
c1_cell=c_cord(1,i);
c1_elem=c1_cell{1,1};
```

```
cordMatrix(:,i)=c1_elem;
end
% Sort the matrix by the third column-Y coordinates
% sortedMatrixByLy-sorted matrix by left y coordinates
sortedMatrixByLy=sortrows(cordMatrix, 3);
% sortedMatrixByRy-sorted matrix by right y coordinates
sortedMatrixByRy=sortrows(cordMatrix, 3);
%%
% pairwise distance between left and right side sets of points
% # of points on Left side and right side do NOT have to the the same
Left=sortedMatrixByLy(:,1:4);
Right=sortedMatrixByRy(:,5:8);
% Fetch the x,y,z coordinates of left side points
LC=Left(:,2:4);
% Fetch the x,y,z coordinates of right side points
RC=Right(:,2:4);
% Compute all the distances between points on left and right side
% i.e. left has M points, right has N points, size of D matrix is M*N
D = pdist2(LC,RC);
%%
% Find the minimum distance value in each row of D and
% return the corresponding indices
DD=D;
[Sml,ind] = min(DD,[],2);
for j=1:size(DD,1)
[Sml(j),ind(j)] = min(DD(j,:),[],2);
% Replace the value in the same column of ind(j) by a very large number
% eg.999999 in here to avoid duplicat indice (i.e. the same point on one
% side used more than once)
DD(:,ind(j))=999999;
end
% Based on the returned indices find the paired points on left and right
% sides which has minimum distances
% The paired nodes then can be incorporated into FEA package Abaqus input file to
% define Periodic Boundary Conditions by using "Equations" in Abaqus
% pn:parid nodes
pn=[Left(:,1)  Right(ind,1)];
```

C. Sample coordinates for the points (nodes) on the two opposite surfaces of a 3D RVE: coordinates.txt

L_Node_No	x	y	z	R_Node_No	x	y	z
1047	8.1688	17.6658	8.25E-01	323	20.4995	17.989401	8.24E-01
1048	8.1647	17.5807	8.21E-01	324	20.4956	18.0783	8.29E-01
1050	8.1662	17.680201	9.16E-01	325	20.4991	17.989401	9.07E-01
1051	8.1555	17.578501	9.15E-01	326	20.495701	18.0783	9.05E-01
1279	8.1657	17.4765	8.22E-01	565	20.498501	17.884399	8.24E-01
1281	8.1569	17.476	9.15E-01	566	20.495899	17.884001	9.04E-01
1519	8.1657	17.374001	8.22E-01	803	20.497801	17.782	8.22E-01
1521	8.157	17.374001	9.15E-01	804	20.4965	17.7813	9.15E-01
1757	8.1657	17.271999	8.22E-01	1045	20.4963	17.68	8.22E-01
1759	8.157	17.271999	9.15E-01	1046	20.5068	17.6803	9.15E-01

6. Acknowledgment

The first two authors would like to thank their colleagues Drs. Fomunung and Onyango for sharing their opinions and their support.