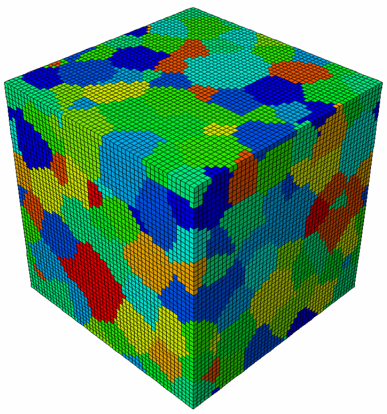
Solid Mechanics Research Group

ABAQUS Crystal Plasticity Guide



Latest Update: 13/09/2021

Contents

Figures

1 Introduction and Prerequisites

This document is designed to give a new user a brief tutorial on how to create a representative volume element (RVE) and run a crystal plasticity (CP) simulation in ABAQUS using the SMRG CP code. Before using this guide, the following files must be installed/downloaded in order to successfully complete CP simulations.

ABAQUS	Download from Company Portal	Finite Element Sotware
Dream3D	http://dream3d.bluequartz.net/	Software to generate 3D RVEs
MATLAB	Access through company portal or https://uk.mathworks.com/products/matlab/student.html	Used to run Dream3D-2-ABAQUS
Dream3D-2-ABAQUS	https://github.com/smrg-uob/Dream3d2Abaqus	Converts Dream3D microstructural data to ABAQUS model
Python	https://www.python.org	Used to run LengMorph
LengMorph	https://github.com/DylanAgius/LengMorph	Creates binary files to utilise with CP UMAT to incorporate length scale effects
ABQ_Lengthscale	https://github.com/smrg-uob/CP_UMATs	CP UMAT for ABAQUS that incorporates length scale effects
CP_recovery_rotation_grainsize	https://github.com/smrg-uob/CP_UMATs	CP UMAT with no length scale effects hence requires no binary files

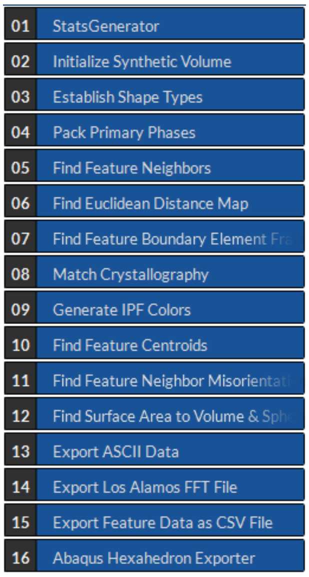
A compilation of these codes can be found alongside this file.

Locally installed ABAQUS CAE is useful to perform edits to the pre-generated RVE model, however is not so successful in running the simulation. This is due to later versions of ABAQUS/Intel fortran compiler not being able to compile the UMATs. Therefore, it is recommended to run simulations on the university's HPC such as BlueCrystal or BluePebble using modules abaqus-6.14 with the 2011_sp1.8.273 Intel fortran compiler.

2 RVE Generation Using Dream3D

Dream3D uses a statistics based approach to generate 3D RVE(s), which in this case will reflect the microstructure of austenitic stainless steel, 316H.

Dream3D uses a set of filters to form a pipeline to follow in order to generate the RVE. The recommended pipeline required is highlighted in Figure 1.



01 StatsGenerator
02 Initialize Synthetic Volume
03 Establish Shape Types
04 Pack Primary Phases
05 Find Feature Neighbors
06 Find Euclidean Distance Map
07 Find Feature Boundary Element Fr
08 Match Crystallography
09 Generate IPF Colors
10 Find Feature Centroids
11 Find Feature Neighbor Misorientat
12 Find Surface Area to Volume & Sph
13 Export ASCII Data
14 Export Los Alamos FFT File
15 Export Feature Data as CSV File
16 Abaqus Hexahedron Exporter

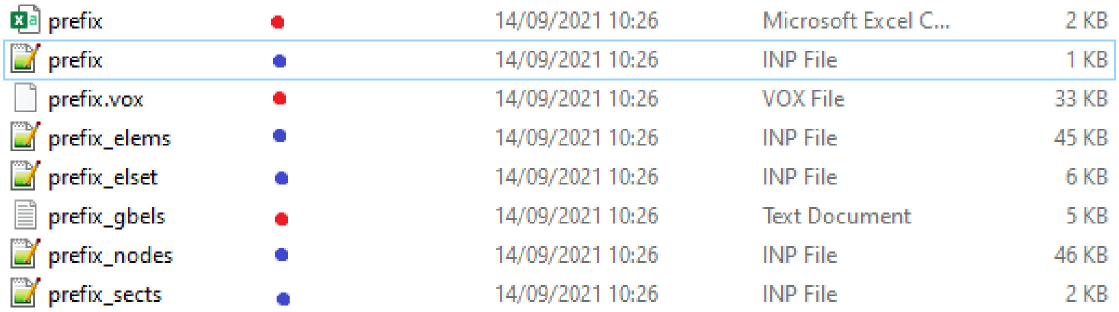
Figure 1: Recommended Dream3D Pipeline.

This pipeline can also be found in the repository alongside this file, titled 'Pipeline_RVE.json'. Upon creating/loading the correct pipeline, a few alterations must/may be completed.

- Under 'Initialise Synthetic Volume' There is an option to alter the dimensions of the RVE if you wish to do so. Ticking the 'Estimate Number of Features' box will provide an approximation of the number of grains within the RVE.
- For all export filters (13-16), ensure an appropriate file location has been selected and there is a consistent prefix label used, this will save time later.
- Under 'Export Los Alamos FFT File', ensure the file type is saved as .vox otherwise the MATLAB script will not be able to convert this into an ABAQUS model.
- If CP_recovery_rotation_grainsize is to be used, 'Abaqus Hexahedron Exporter' can be removed from the pipeline. This step generates the files used to create the length scale binary files hence is not needed for this UMAT.

When a satisfactory pipeline is prepared, press ‘Start Pipeline’ in the lower left of the screen which should produce the required files.

Upon successful completion of the pipeline, three or eight files will be generated. These are shown in Figure 2 (Prefix ‘prefix’). Red markers indicate files that will always be produced, blue markers indicate those produced by hexahedron exporter for use with ABQ_Lengthscale, excluding prefix.inp.



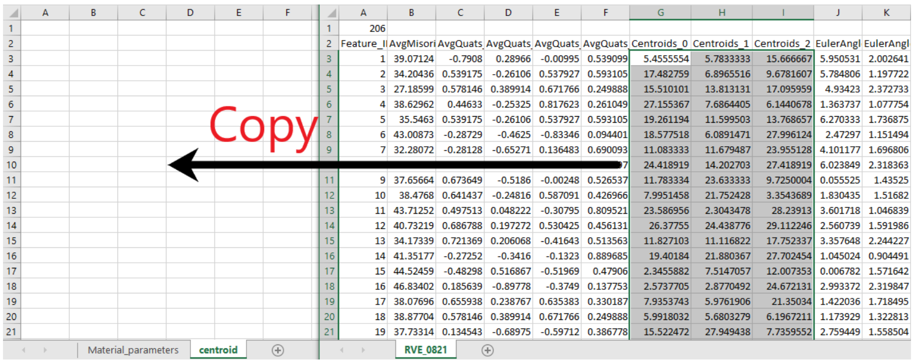
prefix	🔴	14/09/2021 10:26	Microsoft Excel C...	2 KB
prefix	🔵	14/09/2021 10:26	INP File	1 KB
prefix.vox	🔴	14/09/2021 10:26	VOX File	33 KB
prefix_elems	🔵	14/09/2021 10:26	INP File	45 KB
prefix_elset	🔵	14/09/2021 10:26	INP File	6 KB
prefix_gbels	🔴	14/09/2021 10:26	Text Document	5 KB
prefix_nodes	🔵	14/09/2021 10:26	INP File	46 KB
prefix_sects	🔵	14/09/2021 10:26	INP File	2 KB

Figure 2: Files produced from Dream3D pipeline. Red markers indicate files that will always be produced, Blue markers indicate those produced by hexahedron exporter for use with ABQ_Lengthscale.

3 Converting Dream3D Microstructure to ABAQUS model

Before running the Dream3D-2-Abaqus code, a file must be modified. Within the 'Dream3d2Abaqus' folder there is an excel document, 'inputfile_info.xslx'. The first sheet is 'Material_parameters' which contains specific parameters to 316H, details of which can be found on the CP_UMATs GitHub repository. The second sheet, 'centroid', is empty and requires centroid values from the previously generated CSV file.

Copy the numerical values (do not copy table headers) of the three centroid columns from the CSV file to the first three columns of sheet 'centroid' in 'inputfile_info'. This is shown in Figure 3 Now this is complete, the MATLAB code can be utilised.



	A	B	C	D	E	F	G	H	I	J	K
1	206										
2	Feature_I	AvgMisori	AvgQuats_	AvgQuats_	AvgQuats_	AvgQuats_	Centroids_0	Centroids_1	Centroids_2	EulerAngl	EulerAngl
3	1	39.07124	-0.7908	0.28966	-0.00995	0.539099	5.4555554	5.7833333	15.666667	5.950531	2.002641
4	2	34.20436	0.539175	-0.26106	0.537927	0.593105	17.482759	6.8965516	9.6781607	5.784806	1.197722
5	3	27.18599	0.578146	0.389914	0.671766	0.249888	15.510101	13.813131	17.095959	4.93423	2.372733
6	4	38.62962	0.44633	-0.25325	0.817623	0.261049	27.155367	7.6864405	6.1440678	1.363737	1.077754
7	5	35.5463	0.539175	-0.26106	0.537927	0.593105	19.261194	11.599503	13.768657	6.270333	1.736875
8	6	43.00873	-0.28729	-0.4625	-0.83346	0.094401	18.577518	6.0891471	27.996124	2.47297	1.151494
9	7	32.28072	-0.28128	-0.65271	0.136483	0.690093	11.083333	11.679487	23.955128	4.101177	1.696806
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7	24.418919	14.202703	27.418919	6.023849	2.318363
11	9	37.65664	0.673649	-0.5186	-0.00248	0.526537	11.783334	23.633333	9.7250004	0.055525	1.43525
12	10	38.4768	0.641437	-0.24816	0.587091	0.426966	7.9951458	21.752428	3.3543689	1.830435	1.51682
13	11	43.71252	0.497513	0.048222	-0.30795	0.809521	23.586956	2.3043478	28.23913	3.601718	1.046839
14	12	40.73219	0.686788	0.197272	0.530425	0.456131	26.37755	24.438776	29.112246	2.560739	1.591986
15	13	34.17339	0.721369	0.206068	-0.41643	0.513563	11.827103	11.116822	17.752337	3.357648	2.244227
16	14	41.35177	-0.27252	-0.3416	-0.1323	0.889685	19.40184	21.880367	27.702454	1.045024	0.904491
17	15	44.52459	-0.48298	0.516867	-0.51969	0.47906	2.3455882	7.5147057	12.007353	0.006782	1.571642
18	16	46.83402	0.185639	-0.89778	-0.3749	0.137753	2.5737705	2.8770492	24.672131	2.993372	2.319847
19	17	38.07696	0.655938	0.238767	0.635383	0.330187	7.9353743	5.9761906	21.35034	1.422036	1.718495
20	18	38.87704	0.578146	0.389914	0.671766	0.249888	5.9918032	5.6803279	6.1967211	1.173929	1.322813
21	19	37.73314	0.134543	-0.68975	-0.59712	0.386778	15.522472	27.949438	7.7359552	2.759449	1.558504

Figure 3: Copy the three centroid columns from the CSV file to the 'centroid' sheet in 'inputfile_info'.

Open MATLAB and point the directory to the 'Dream3d2Abaqus' folder. Move the .vox file created earlier and the edited 'inputfile_info' spreadsheet into this directory. Run the following command in MATLAB, of course change the prefix of the filename where appropriate.

dream2abq('RVE_0821.vox','RVE_0821.inp')

Be patient with the code, it can take anywhere from seconds to minutes to complete the conversion. Once it has complete, you will receive notification within MATLAB where it will print the amount of time it took to run. There should now be an ABAQUS input file located within the directory.

4 Creating Binary Files for ABQ_Lengthscale UMAT

Presuming python is installed, the first step is to install the LengMorph tool.

- Download LengMorph from the GitHub link provided
- Navigate to the ‘PyLengMorph’ folder using the cd command
- Initialise Python
- Enter command ‘*pip install .*’

The tool should now be installed.

Place the following six files into a folder titled ‘data’:

- prefix.csv
- prefix_elems.inp
- prefix_elset.inp
- prefix_gbels.inp
- prefix_nodes.inp
- prefix_sects.inp

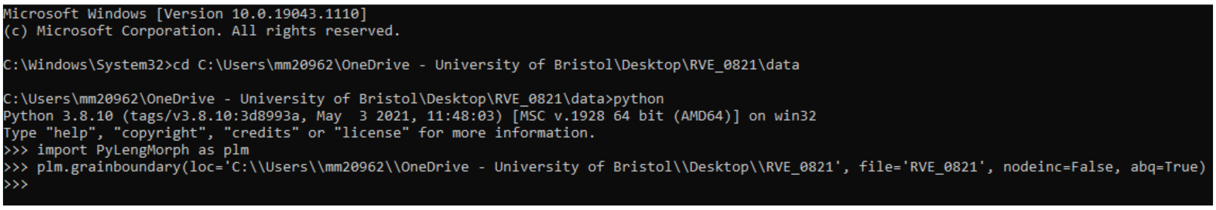
Now follow these next steps to generate the binary files

- Open a command prompt window
- Navigate into the data folder previously created using the cd command
- Initialise python
- Enter command ‘*import PyLengMorph as plm*’ (capitals are important)
- Enter command ‘*plm.grainboundary(loc=’\\data \\directory’, file=’prefix’, nodeinc=False, abq=True)*’ The location pathway must have double backslashes and must navigate to the folder prior to the data folder created

An example of this can be found in Figure 4.

This should have generated five new binary (.bin) files and two .inc files within the newly created data folder. These are:

- xvalues.bin
- yvalues.bin



```
Microsoft Windows [Version 10.0.19043.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Users\mm20962\OneDrive - University of Bristol\Desktop\RVE_0821\data

C:\Users\mm20962\OneDrive - University of Bristol\Desktop\RVE_0821\data>python
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import PyLengMorph as plm
>>> plm.grainboundary(loc='C:\\Users\\mm20962\\OneDrive - University of Bristol\\Desktop\\RVE_0821', file='RVE_0821', nodeinc=False, abq=True)
>>>
```

Figure 4: Example commands to generate binary files.

- zvalues.bin
- el_centroid.bin
- boundfeat.bin
- orien.inc
- param_array.inc

These files are needed in order for the ABQ_Lengthscale UMAT to run.

5 Editing generated ABAQUS Job

The first step is to modify the MATLAB generated model to the desired specifications. This can be achieved by importing the input file to ABAQUS CAE. The following list contains a few points to consider when editing.

- Based on material specification, the units of load and time are in MPa and seconds respectively.
- Using pressure on a surface is a loading condition which is known to succeed.
- Due to the nature of an RVE, periodic boundary conditions should be appropriately applied to best replicate the repeating unit nature. Symmetry conditions should be applied to three faces sharing the same corner. An example is highlighted in Figure 5.

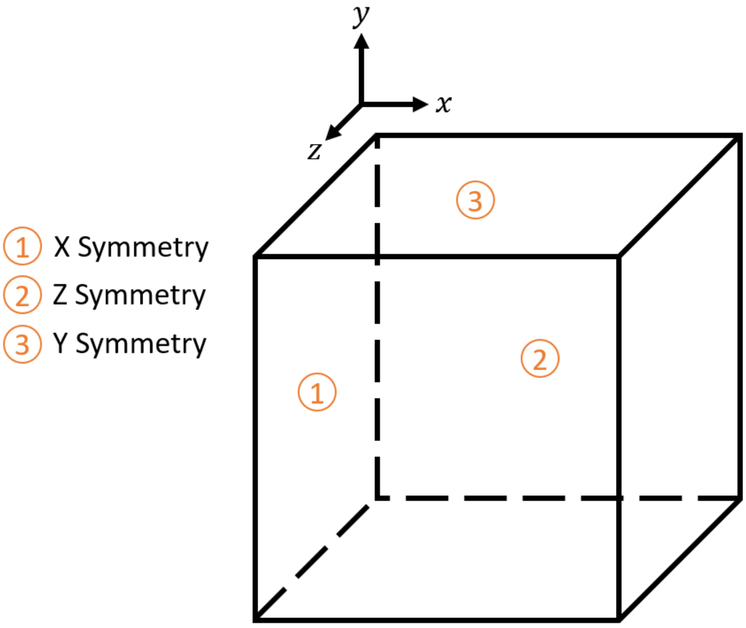
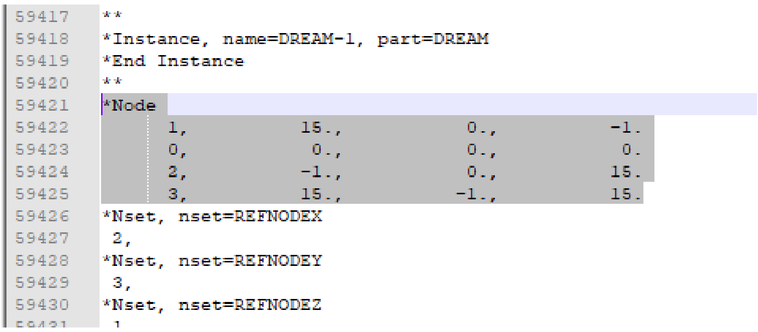


Figure 5: Example application of periodic boundary conditions.

Once the model has been altered to the desired standard, create a job and write the input file in the working directory. Due to some additional nodes being created as a result of an error in the code, a specific line in the input file must be deleted.

Open the job in a text editor and search for term ‘*Node’, eventually this should bring you to a line with a matrix of numbers below it, one row of which should be zeroes. Delete this line of zeroes and save the file. An example of this section of code is shown in Figure 6.



```
**
*Instance, name=DREAM-1, part=DREAM
*End Instance
**
*Node
      1,          15.,           0.,          -1.
      0,           0.,           0.,           0.
      2,          -1.,           0.,          15.
      3,          15.,          -1.,          15.
*Nset, nset=REFNODEX
 2,
*Nset, nset=REFNODEY
 3,
*Nset, nset=REFNODEZ
 1
```

Figure 6: Example section of code displaying the zero line to delete.

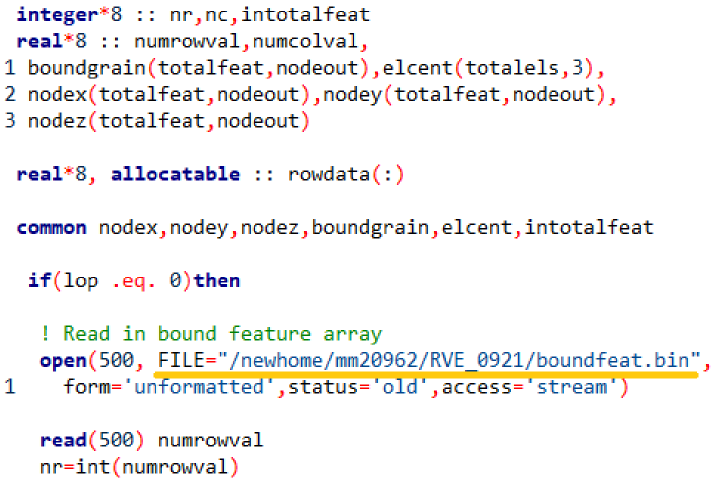
The model is now ready to run. Unless the correct ABAQUS and Intel Fortran Compiler are installed locally, it is recommended to run the simulation on a HPC, on which the specific versions can be defined easily.

6 Running job on BlueCrystal

To run the CP simulation on BlueCrystal follow these steps. To gain access to BlueCrystal log a ticket with IT services to provide you with a login. For introductory information on using BlueCrystal, please read the following guide.

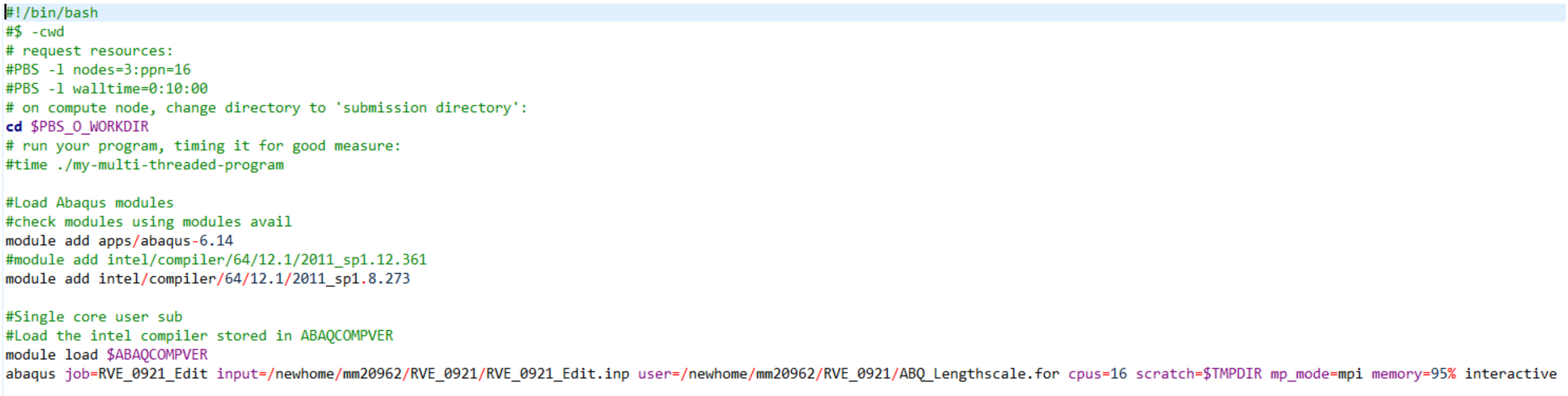
https://www.acrc.bris.ac.uk/acrc/pdf/bc-user-guide.pdf

- Login to BlueCrystal using an SFTP client and create a folder for this job.
- Copy, into the new directory, all the files generated in the previous section if utilising ABQ_Lengthscale, the edited input file, the shell file found in the repository alongside this guide and the desired UMAT.
- If using ABQ_Lengthscale, open it with a text editor and change the binary files directories at the beginning of the file. Each directory must be the full directory and single forward-slashes between each folder.
- Once the UMAT has been edited, open the shell file in a text editor and edit the job name, input file name and UMAT name at the end of the file. The line that adjusts the walltime is found at the start of the shell file.
- Screenshots of example UMAT and shell files can be found in Figures 7 and 8 respectively.
- The job is now ready to run. Login to BlueCrystal using an SSH client. Change directory to the job folder and enter command '*qsub BC_Run.sh*'. The job should now be submitted. Please refer to the aforementioned guide in this section for commands on how to check the status of a job.



```
 integer*8 :: nr,nc,intotalfeat
 real*8 :: numrowval,numcolval,
1 boundgrain(totalfeat,nodeout),elcent(totalels,3),
2 nodex(totalfeat,nodeout),nodey(totalfeat,nodeout),
3 nodez(totalfeat,nodeout)

 real*8, allocatable :: rowdata(:)

 common nodex,nodey,nodez,boundgrain,elcent,intotalfeat

  if(lop .eq. 0)then

   ! Read in bound feature array
   open(500, FILE="/newhome/mm20962/RVE_0921/boundfeat.bin",
1    form='unformatted',status='old',access='stream')

   read(500) numrowval
   nr=int(numrowval)
```

Figure 7: Beginning of ABQ_Lengthscale UMAT, first directory that needs altering is underlined.



```
#!/bin/bash
#$ -cwd
# request resources:
#PBS -l nodes=3:ppn=16
#PBS -l walltime=0:10:00
# on compute node, change directory to 'submission directory':
cd $PBS_O_WORKDIR
# run your program, timing it for good measure:
#time ./my-multi-threaded-program

#Load Abaqus modules
#check modules using modules avail
module add apps/abaqus-6.14
#module add intel/compiler/64/12.1/2011_sp1.12.361
module add intel/compiler/64/12.1/2011_sp1.8.273

#Single core user sub
#Load the intel compiler stored in ABAQCOMPVER
module load $ABAQCOMPVER
abaqus job=RVE_0921_Edit input=/newhome/mm20962/RVE_0921/RVE_0921_Edit.inp user=/newhome/mm20962/RVE_0921/ABQ_Lengthscale.for cpus=16 scratch=$TMPDIR mp_mode=mpi memory=95% interactive
```

Figure 8: Functioning shell file to run CP job.