Computational Materials Science and Mechanics Laboratory (CMSML)

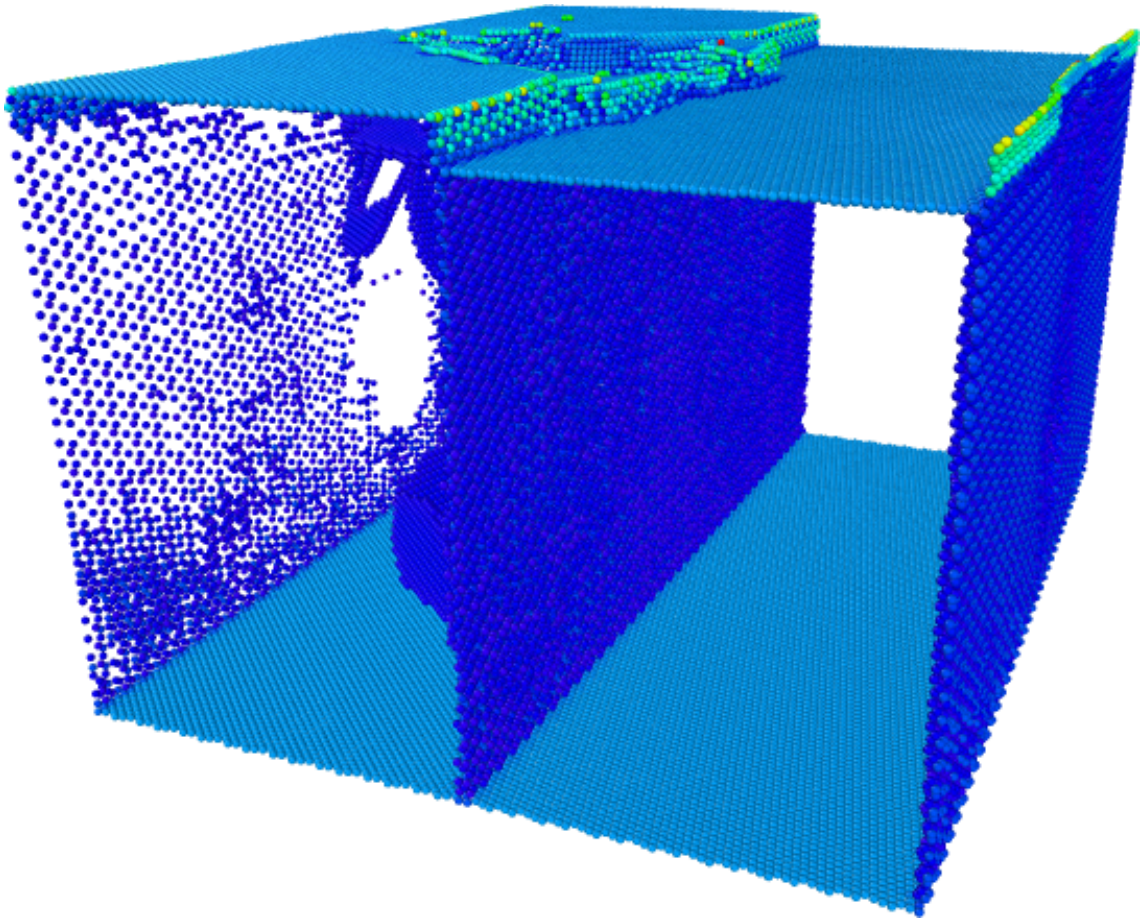
Department of Mechanical Engineering, Northern Illinois University

PI: Iman Salehinia

TUTORIAL:

GRAIN BOUNDARY EFFECT ON THE MECHANICAL RESPONSE OF COPPER BI-CRYSTAL UNDER NANOINDENTATION

Charles Gudbrandsen, Spring 2017



This tutorial is meant to be a step-by-step guide on how to use and implement LAMMPS code in order to conduct nano-indentation simulations on copper bicrystals. Various other software packages are also included. (Note: this tutorial assumes the windows operating system is being used)

List of Necessary Software:

The following programs are necessary in order to conduct these simulations:

PuTTY – a free and open source terminal emulator, serial console and network file transfer application. Download at: http://www.putty.org/

FileZilla – file transfer protocol software.
Download at: https://filezilla-project.org/ (download client version)

OVITO – a scientific visualization and analysis software for atomistic simulation.
Download at: https://ovito.org/

Microsoft word

Microsoft Excel

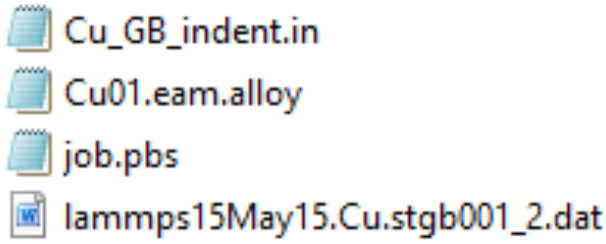
Notepad

You have probably noticed that LAMMPS software is missing from this list. This is because the simulations will be run on NIU's Gaea computing cluster, which already has the LAMMPS

software installed and ready for use. You can run the LAMMPS simulations on your own PC (which can be found at http://lammps.sandia.gov/), but this will not be covered in this tutorial.

Files needed for Nano Indentation:

The following files will also be needed (these will be found in the nano-indentation folder):



Indentation Code:

We will start with the first file Cu_GB_indent.in. This is the LAMMPS code that defines the parameters of the indentation simulation that we want to run. (Note: this code is easiest to edit in notepad) The entire code looks like this:

```
# SGB under nanoindentation
# Iman Salehinia, Washington State University, Dec. 2014
#special thanks to Mark Tschopp for providing the GB structures

# ---------- Setup Variables ----------------------

variable etol equal 1.0e-25
variable ftol equal 1.0e-25
variable maxiter equal 5000
variable maxeval equal 10000
variable latparam equal 3.615

# ---------- Initialize Simulation ----------------------

clear
units metal
dimension 3
boundary p p s
atom_style atomic

# ---------- Create Atomistic Structure ----------------------
```

```
lattice fcc ${latparam}
read_data lammps15May15.Cu.stgb001_2.dat # Garrett
replicate 2 1 29

#creating top and bottom partitions in order to trim the fat from the grains
region          3 block INF INF INF INF 102  INF units box
region          2 block INF INF INF INF INF  1.0 units box
region          7 block INF INF INF INF INF  2.0 units box

group           top    region 3
group           bottom region 2
group           base   region 7
group           inner subtract all bottom top


#creating the step in the material
#region         stepatoms block INF INF 0.01 INF 106 INF units box
#group          step region stepatoms

#delete_atoms group step
delete_atoms group top
delete_atoms group bottom

# ---------- Define Interatomic Potential ---------------------

pair_style eam/alloy
pair_coeff * * Cu01.eam.alloy Cu
neighbor 2.0 bin
neigh_modify delay 10 check yes

# ---------- Define Settings --------------------------------

compute csym all centro/atom fcc
compute eng all pe/atom

#this exports the data into the csp.all file
dump            101 all custom 1000 csp.all id type x y z c_csym c_eng

# ---------- Run Minimization ---------------------------

reset_timestep 0
thermo 10
#thermo_style custom step pe lx ly lz press pxx pyy pzz
min_style cg
minimize ${etol} ${ftol} ${maxiter} ${maxeval}
```

```
# ---------- totally fix all boundaries ----------------------

fix                 7 base setforce 0.0 0.0 0.0

# ---------- initial velocities ---------------------------

compute new inner temp
velocity inner create 10.00 4882748 temp new
#fix 10 all nve
fix 4 all nvt temp 10.0 10.0 0.01 drag 1.0
#fix 4 all temp/rescale 5 0.01 0.01 0.005 0.05

# ---------- relaxation -------------------------------

timestep        0.001
thermo          100
thermo_modify           temp new
thermo_style    custom step temp

run 5000

# ---------- indentation ------------------------------------------

variable z equal "162.5-0.1*step*dt"
fix                 5 all indent 100.0 sphere 50.7 40.0 v_z 60 units box

# ---------- Run with indenter ------------------------------------

timestep        0.001
thermo          10
thermo_modify           temp new
thermo_style    custom step temp f_5[3]

run 300000
```

This large amount of code may seem a bit overwhelming, but when it is broken down into its components it is rather simple. The main components or flow for nano indentation are as follows:

- Defining or Building the Model
- Performing Energy minimization of the system
- Relaxation of the system
- Running the simulation (in this case indentation)

So with these main components in mind, we will go through the code section by section.

Defining/Building the Model:

```
# SGB under nanoindentation
# Iman Salehinia, Washington State University, Dec. 2014
#special thanks to Mark Tschopp for providing the GB structures
```

This section is just the header and merely provides information on what the code if for and who wrote it initially.

Note: '#' instructs LAMMPS to ignore what comes after it on that line; much like '//' in C++

```
# ---------- Setup Variables ---------------------

variable etol equal 1.0e-25
variable ftol equal 1.0e-25
variable maxiter equal 5000
variable maxeval equal 10000
variable latparam equal 3.615
```

This section sets up the stopping parameters for energy minimization and the crystal lattice parameters.

- etol = stopping tolerance for energy (unitless)
- ftol = stopping tolerance for force (force units)
- maxiter = max iterations of minimizer
- maxeval = max number of force/energy evaluations
- latparam = length of lattice between atoms (this varies from material to materials so you will need to search for separate lattice parameters when switching from copper).

The first four variables terminate the system energy minimization when one of them is satisfied. More information on this can be found here: http://lammps.sandia.gov/doc/minimize.html

```
# ---------- Initialize Simulation ---------------------

clear
units metal
dimension 3
boundary p p s
atom_style atomic
```

This section preps the simulation space for the simulation that is to be run.

Clear deletes all of the atoms in the simulation space and resets all settings to their defaults.
http://lammps.sandia.gov/doc/clear.html

units metal defines the units for the input and output values of the simulation for metals it is:

- mass = grams/mole
- distance = Angstroms
- time = picoseconds
- energy = eV
- velocity = Angstroms/picosecond
- force = eV/Angstrom
- torque = eV
- temperature = Kelvin
- pressure = bars
- dynamic viscosity = Poise
- charge = multiple of electron charge (1.0 is a proton)
- dipole = charge*Angstroms
- electric field = volts/Angstrom
- density = gram/cm^dim

The different type of LAMMPS units can be found here:
http://lammps.sandia.gov/doc/units.html

dimension 3 tells LAMMPS that the simulation is going to be a 3D simulation
http://lammps.sandia.gov/doc/dimension.html

atom_style atomic defines the style of atoms that are to be used in the simulation
http://lammps.sandia.gov/doc/atom_style.html

```
# ---------- Create Atomistic Structure ---------------------

lattice fcc ${latparam}
read_data lammps15May15.Cu.stgb001_2.dat # Garrett
replicate 2 1 29

#creating top and bottom partitions in order to trim the fat from the grains
region          3 block INF INF INF INF 102  INF units box
region          2 block INF INF INF INF INF  1.0 units box
region          7 block INF INF INF INF INF  2.0 units box

group           top    region 3
group           bottom region 2
group           base   region 7
group           inner subtract all bottom top


#creating the step in the material
#region          stepatoms block INF INF 0.01 INF 106 INF units box
#group           step region stepatoms

#delete_atoms group step
delete_atoms group top
delete_atoms group bottom
```

In this section the copper bi-crystal system is created.

Lattice fcc ${latparam} – this tells LAMMPS that the system is using face center cubic lattice with a lattice spacing of 3.615 (which we defined in the Setup Variables Section).
http://lammps.sandia.gov/doc/lattice.html

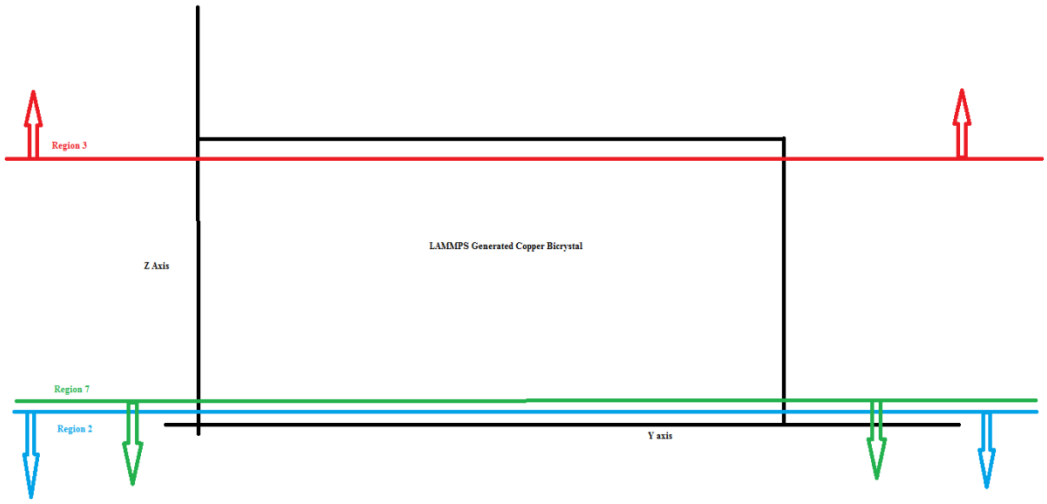
Read_data lammps15May15.Cu.stgb001_2.dat– this imports the basic grain model into LAMMPS. If you look at the files needed to run a LAMMPS simulation you will notice that data lammps15May15.Cu.stgb001_2.dat is one of the files. This file contains the data for building the basic bi-crystal. If you want to change the type of grain boundary or the angle of the grain boundary simply change the file name to the corresponding file name. (specifics about this file will be discussed in a later section. # Garrett is a comment that denotes the creator of the boundary.
http://lammps.sandia.gov/doc/read_data.html

Note: check the corresponding info files for specifics on the grain boundary properties.

Replicate 2 1 29 – multiplies the base model that was imported in by the previous line in order to scale up the size of the bicrystal. The general form of replicate is:
Replicate x y z
Where the values of x, y, and z duplicate/scale up the crystal by those multipliers; so in our case the gets 2 times longer in the x direction, 1 times longer in the y direction, and 29 times longer in the z direction.
http://lammps.sandia.gov/doc/replicate.html

```
#creating top and bottom partitions in order to trim the fat from the grains
region          3 block INF INF INF INF 102  INF units box
region          2 block INF INF INF INF INF  1.0 units box
region          7 block INF INF INF INF INF  2.0 units box
```

This section creates the regions for our bicrystal that will then be gathered into specific groups.



The general form for block regions is:
Region ID block xlo xhi ylo yhi zlo zhi units box
ID - user-assigned name for the region
xlo,xhi,ylo,yhi,zlo,zhi = bounds of block in all dimensions (distance units)
box = the geometry is defined in simulation box units
http://lammps.sandia.gov/doc/region.html

```
group           top    region 3
group           bottom region 2
group           base   region 7
group           inner subtract all bottom top
```

Our regions are then placed into specific groups that will then be manipulated or deleted.
The general form for group by region is:
Group ID region args
ID - user-assigned name for the region
region args = region-ID
http://lammps.sandia.gov/doc/group.html

```
#creating the step in the material
#region         stepatoms block INF INF 0.01 INF 106 INF units box
#group          step region stepatoms
```

This next part of the code (when not commented out with '#') would create a step at the grain boundary. The image below is what it looks like when a step is defined.

```
#creating the step in the material
region          stepatoms block INF INF INF -0.01 97 INF units box
group           step region stepatoms
```

#delete_atoms group step
delete_atoms group top
delete_atoms group bottom

This final part in the Create Atomistic Structure section removes the defined groups to give the final structure to the grain that will be indented. Deleting the top and bottom groups ensures flat surfaces for indenting and fixing. Also when not commented out the delete_atoms group step will remove the atoms in the defined group to create a step.
http://lammps.sandia.gov/doc/delete_atoms.html

```
# ---------- Define Interatomic Potential ---------------------

pair_style eam/alloy
pair_coeff * * Cu01.eam.alloy Cu
neighbor 2.0 bin
neigh_modify delay 10 check yes
```

This section defines the interatomic potential
Pair_style – sets the formulas that LAMMPS uses to compute pairwise interations
http://lammps.sandia.gov/doc/pair_style.html

pair_coeff – specifies the pairwise force field coefficients for one or more pairs of atom types. In our case we are importing these coefficients from Cu01.eam.alloy Cu, where Cu01.eam.alloy is the file of values being used and Cu defines the type of atom. Cu01.eam.alloy is also one of the files that must be included with the simulation. If you change the atom being used, the EAM file needs to be replaced with the appropriate embedded atomic model potential which can be found here: https://www.ctcms.nist.gov/potentials/
http://lammps.sandia.gov/doc/pair_coeff.html

neighbor 2.0 bin – sets the parameters that affect the building of pairwise neighbor lists. So 2.0 designates the atoms within 2.0 angstroms of the one being analyzed as a neighbor for pairwise force calculations.
http://lammps.sandia.gov/doc/neighbor.html

neigh_modify delay 10 check yes - This command also sets parameters that affect the building and use of pairwise neighbor lists.
http://lammps.sandia.gov/doc/neigh_modify.html

Note: The only line in this section that will probably be modified is the pair_coeff line for when the type of material is changed.

```
# ---------- Define Settings --------------------------------

compute csym all centro/atom fcc
compute eng all pe/atom

#this exports the data into the csp.all file
dump            101 all custom 1000 csp.all id type x y z c_csym c_eng
```

This section defines the type of data that we want to extract from the simulation. The first two lines are generally left as is (though fcc will be changed to bcc base on material). They compute the centrosymmetry parameter and the energy of the atoms that will be placed in our dump file.
http://lammps.sandia.gov/doc/compute.html

dump 101 all custom 1000 csp.all id type x y z c_csym c_eng
This line dumps the date for each atom (the x position, y position, z position, the calculated csym and energy of the atoms) into the csp.all file. This is performed every 1000 steps (so modify 1000 to increase or decrease the frequency of the data dumps.
http://lammps.sandia.gov/doc/dump.html

Energy Minimization of the Model:

```
# ---------- Run Minimization ---------------------------

reset_timestep 0
thermo 10
#thermo_style custom step pe lx ly lz press pxx pyy pzz
min_style cg
minimize ${etol} ${ftol} ${maxiter} ${maxeval}
```

This section tells the LAMMPS code to perform the minimization step

Reset_timestep 0 – places the timestep value back to zero
http://lammps.sandia.gov/doc/reset_timestep.html

thermo 10 – computes and prints the thermodynamic info for minimization every 10 time steps
http://lammps.sandia.gov/doc/thermo.html

#thermo_style custom step pe lx ly lz press pxx pyy pzz – this can be used for trouble shooting. If there appears to be issues with the model, this data can be dumped to check for issues. (uncomment this line to use)
http://lammps.sandia.gov/doc/thermo_style.html

min_style cg – this defines the algorithm to use when the minimize command is performed
http://lammps.sandia.gov/doc/min_style.html

minimize ${etol} ${ftol} ${maxiter} ${maxeval} – this line tells LAMMPS to start the energy minimization of the system with the stop parameters that were defined from the beginning of the code. (note: ${} is used to reference the values assigned to variables)
http://lammps.sandia.gov/doc/minimize.html

```
# ---------- totally fix all boundaries --------------------

fix             7 base setforce 0.0 0.0 0.0
```

This section fixes the base of the model so that when the model is pressed on by the indenter it does not start moving. In this case region 7 is fixed in place.
http://lammps.sandia.gov/doc/fix.html

```
# ---------- initial velocities --------------------------

compute new inner temp
velocity inner create 10.00 4882748 temp new
#fix 10 all nve
fix 4 all nvt temp 10.0 10.0 0.01 drag 1.0
#fix 4 all temp/rescale 5 0.01 0.01 0.005 0.05
```

This section starts the motion of the atoms after the energy minimization. The motion will correspond with velocities of atoms at 10K. Note that the atoms that are given the velocities are those that are in the 'inner' group that we previously defined in the Create Atomistic Structure section.
http://lammps.sandia.gov/doc/velocity.html

Model Relaxation:

```
# ---------- relaxation --------------------------------

timestep        0.001
thermo          100
thermo_modify   temp new
thermo_style    custom step temp

run 5000
```

This section allows the model to relax after the velocity is given to the atoms. It allows the model to run for 5000 steps to let the atoms settle. It will also dump data every 100 steps.

```
# ---------- indentation -----------------------------------------

variable z equal "162.5-0.1*step*dt"
fix             5 all indent 100.0 sphere 50.7 40.0 v_z 60 units box
```

This section is the meat and potatoes of the nano indentation code. This defines the size, type, and position of the indenter to be used.

variable z equal "162.5-0.1*step*dt" – this defines the z coordinate of the center of the indenter. The 162.5 denotes the initial position of the indenter (height of the grain + radius of the indenter). This then moves lower as defined by -0.1*step*dt. Here step is the step is an integer that goes from 0-300,000 and increases in value everytime it runs through this loop (the step value goes up to the run value in the section below this). In that line 'dt' is defined by the timestep which is 0.001.
http://lammps.sandia.gov/doc/timestep.html

fix 5 all indent 100.0 sphere 50.7 40.0 v_z 60 units box
5 - user-assigned name
100.0 – force constant for the indenter surface
Sphere – shape of the indenter
50.7 – x coordinate for center of indenter
40.0 – y coordinate for center of indenter (changes the distance from the grain boundary with 0 being the location of the grain boundary)
V_z – z coordinate for center of indenter
60 – radius of the indenter
http://lammps.sandia.gov/doc/fix_indent.html

Note: when doing indentation on a lower step, simply make the y coordinate negative.

```
# ---------- Run with indenter -----------------------------------------

timestep        0.001
thermo          10
thermo_modify   temp new
thermo_style    custom step temp f_5[3]

run 300000
```

This last section defines the number of steps the indenter loop will run and the width of 'dt' by run and timestep respectively.

Thermo 10 – tells LAMMPS to dump the indenter data every 10 steps
Thermos_modify – sets options for how thermodynamic information is computed and printed by LAMMPS

http://lammps.sandia.gov/doc/thermo_modify.html

thermos_style custom step temp f_5[3] – tells LAMMPS to dump the step number, system temperature, and the force on the indenter in the z direction (f-force 5-indenter [3]-z direction) (note: f_5[1] and f_5[2] will output the force on the indenter in the x and y directions respectively)

EAM Potential File:

The second file needed for nano indentation simulation is the eam file (Cu01.eam.alloy) These files can be found at: https://www.ctcms.nist.gov/potentials/
The different potentials are updated in the indent code at:

```
# ---------- Define Interatomic Potential ---------------------

pair_style eam/alloy
pair_coeff * * Cu01.eam.alloy Cu
neighbor 2.0 bin
neigh_modify delay 10 check yes
```

A few key items can be found when this file is opened with Microsoft Word.

```
Cu EAM from Phys. Rev. B 63, 224106 (2001) in the LAMMPS setfl format.
Conversion by C. A. Becker from Y. Mishin files.
4 February 2009.  http://www.ctcms.nist.gov/potentials
           1 Cu
10000   0.12000000000000000E-02   10000  0.55067860000000000E-03
0.55067860000000000E+01
   29    0.6354600000E+02   0.3615000000E+01   fcc
  -0.2009087334761683E-06
```

The top of the file contains the information on how and when it was made. More importantly it contains the lattice parameter, circled in red, and the crystal structure, circled in blue.

Note: do not modify this file.

Job.pbs File:

This is the third file needed for nano indentation. It contains the instructions for the GAEA computing cluster to execute the indentation code. This file can be viewed and modified with notepad. This is what the job.pbs code looks like:

```
#!/bin/bash
#PBS -N STGB001_2
#PBS -A mdmech15
#PBS -u z136839
#PBS -l nodes=1:ppn=12,walltime=08:00:00
#PBS -W umask=022
#PBS -j oe
#PBS -m abe
#PBS -M z1234567@students.niu.edu

cd $PBS_O_WORKDIR

#source /usr/share/Modules/init/bash
module add cuda/cuda-7.5
module add intelmpi-2015

mpirun -n 12 /data1/local/bin/lmp_gaea < Cu_GB_indent.in >
stgb001_2dump.txt
```

-N – this is the name that will appear when you check the status of the simulation

-u – this is the user name that will appear when you check the status of the simulation and is the owner/initiator of the simulation

-l nodes=1:ppn=12,walltime=08:00:00 – this defines the number of nodes in the GAEA cluster to use. Each node has 12 cores so if nodes=2 then ppn=24 (you should only need to ever use 1 node). Walltime specifies the amount of time you want GAEA to run the simulation. The simulation will either terminate when it runs though all 300,000 steps defined by run in the indentation code, or if it exceeds the wall time. (if it exceeds walltime simply increase the time until the simulation terminates from completing the number of steps).

-M – this is the e-mail address that GAEA will send messages to. These messages let you know when the simulation starts and when it ends.

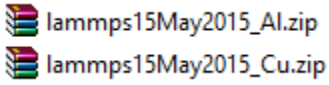
```
mpirun -n 12 /data1/local/bin/lmp_gaea < Cu_GB_indent.in >
stgb001_2dump.txt
```

- this section tells GAEA to run LAMMPS. The value after –n should correspond to the ppn value in the previous line (-l line). <file to execute> is the file that GAEA will execute (in this case < Cu_GB_indent.in>). Stgb001_2dump.txt is the file name that all of the dumps from the indentation code will go to.

Note: I only mentioned the lines the lines that would need modification. Do not modify the others or modify at your own risk.

Lammps15MAY15.XXXXXXXX.dat File:

The last file needed for the nano indentation is the .dat file. This file contains the information for the grain boundary that is to be studied. This is the data that gets replicated and manipulated in the Create Atomistic Structure section of the indent code. These can be extracted from the following zip files:



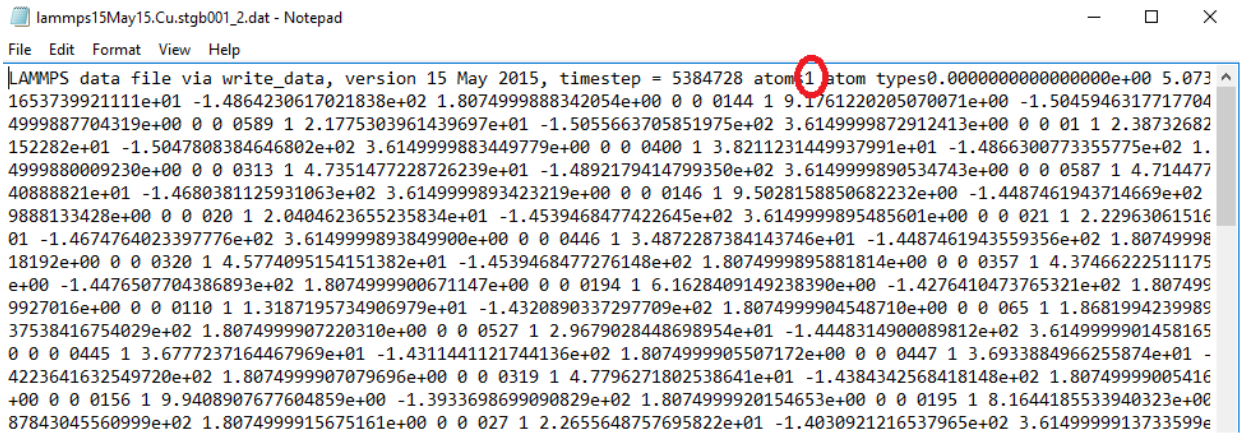
Note: make sure the name of the file is in the indentation code.

The contents can be viewed by opening the file with Microsoft Windows. The part that is most important to use is at the top:

```
LAMMPS data file via write_data, version 15 May 2015, timestep = 538

4728 atoms
1 atom types

0.0000000000000000e+00 5.0738942999999999e+01 xlo xhi
-1.5229104102122105e+02 1.5229104102122105e+02 ylo yhi
0.0000000000000000e+00 3.6150000000000002e+00 zlo zhi

Atoms # atomic

225 1 3.8317556914741937e+00 -1.5032971412376273e+02
```

The dimensions circled in blue are the values that get scaled up during the replicate command (so replicated 2 1 29 makes the crystal go from a thickness of 3.615A to 104.835A).

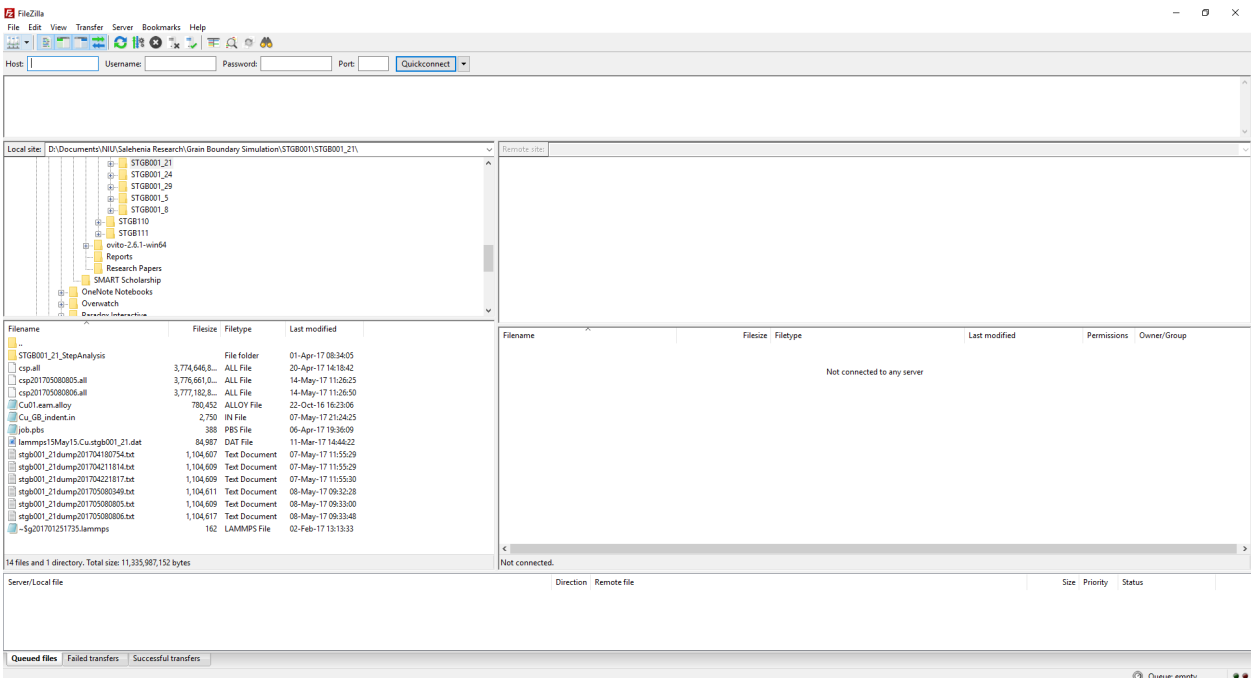
The part circled in red is the number of types of atoms being used (since we are only using copper there is only one type of atom. This value has to match the number of different atoms used in the indenter code; otherwise the simulation will not work. To edit this file open it with notepad and change the atom type value:



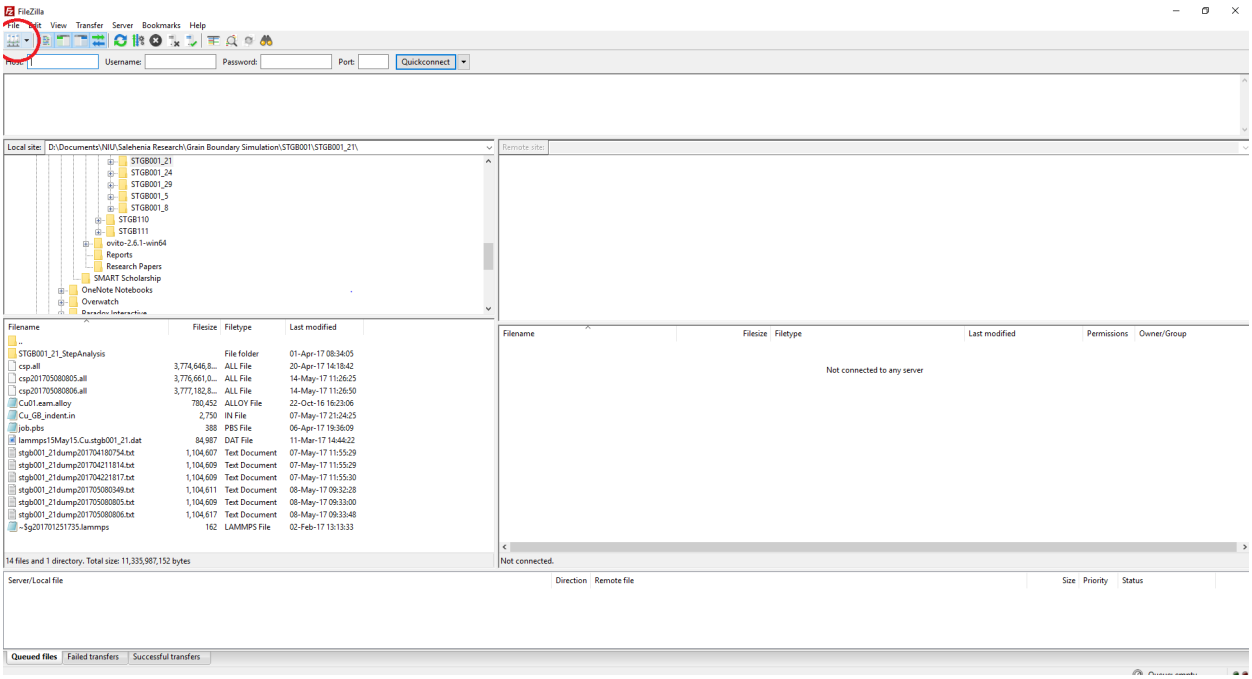
Reopen the file with Microsoft Word to verify it was changed.

Upload Files to GAEA:

Now that you have the four necessary files prepped and ready to run. It is time to upload them to the GAEA server. Start by opening FileZilla.

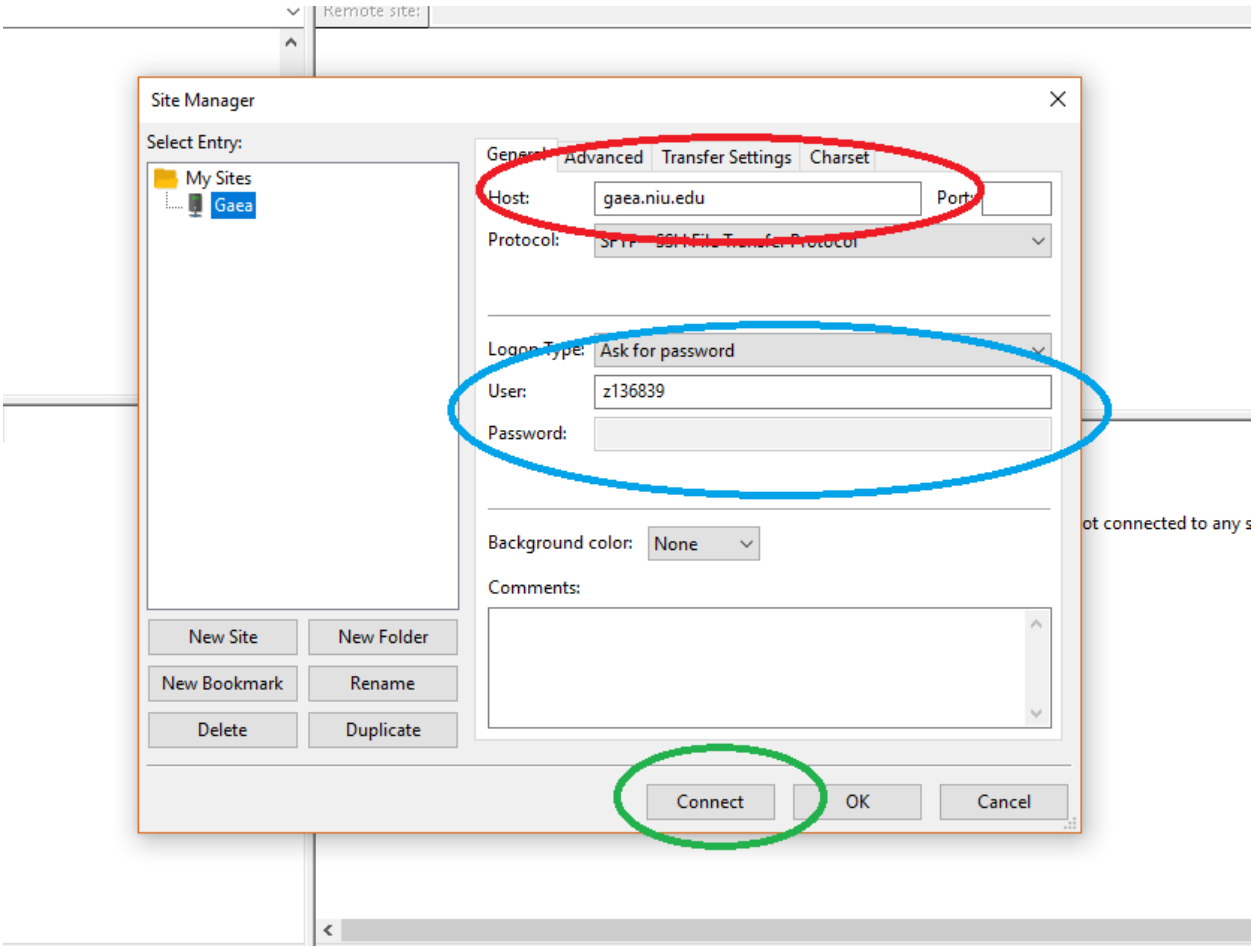


The windows on the left will display the files on your pc, and the windows on the right will display the files of the server you are connected to. So now we need to connect to a server; click the circled button:

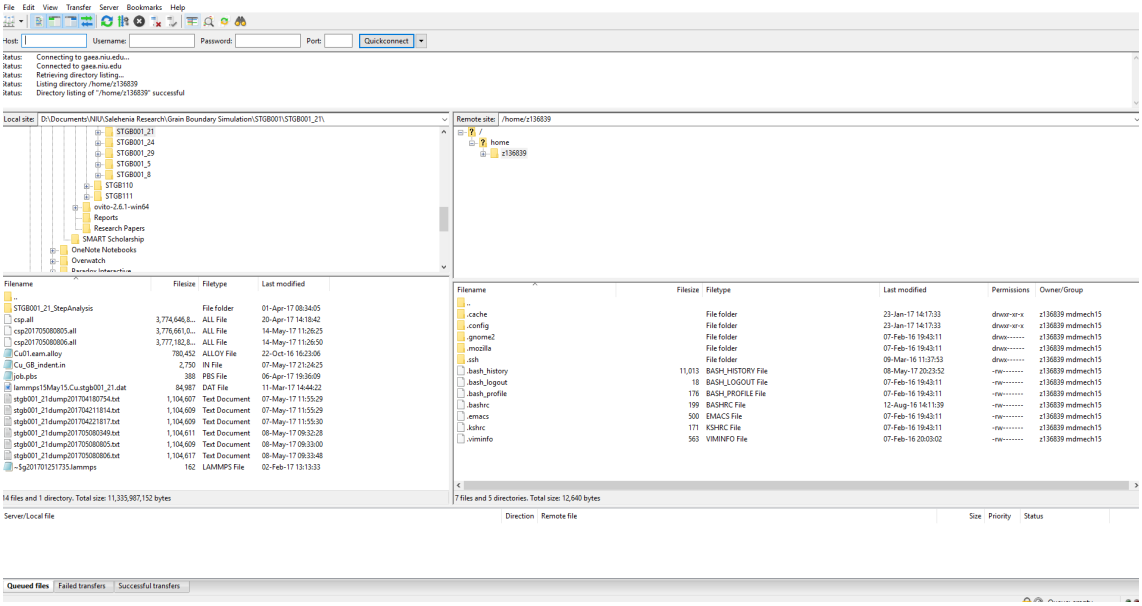


This is the button close up:

You will then get a popup window:

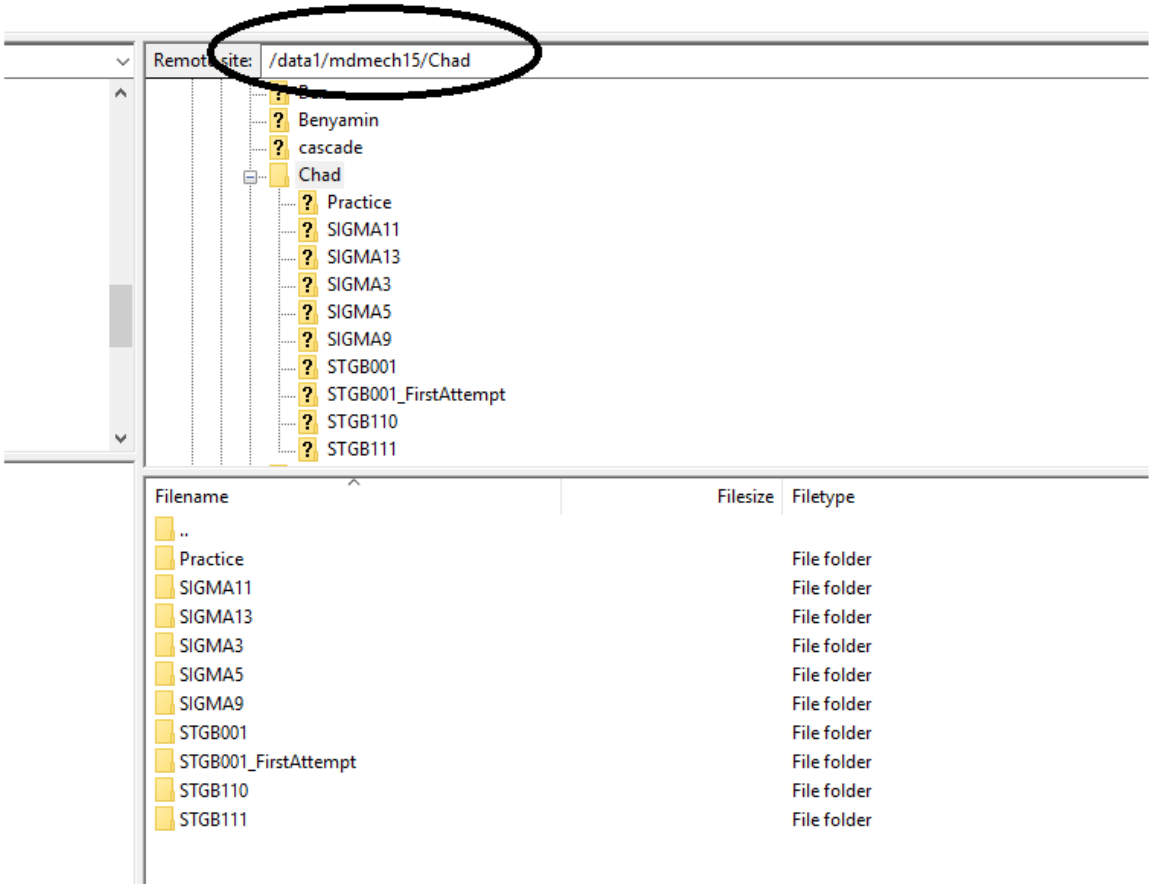


Enter in the host name: gaea.niu.edu
Enter your user name and password (this will be provided by the GAEA system administrators, see Dr. Salehinia for details)
Then click connect. You should now see the files on the GAEA server.

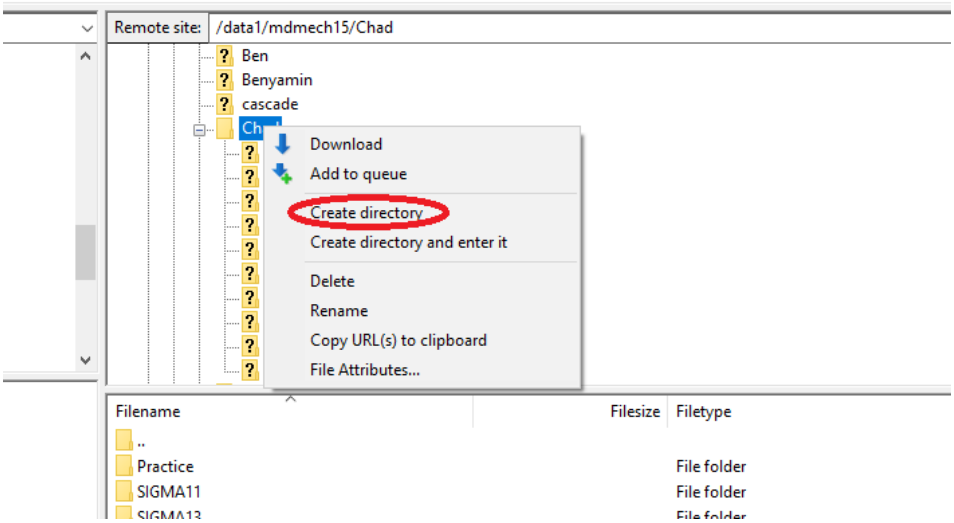


Navigate the server files to find where you will be running the simulations.

/ > data1 > mdmech14> ‘your name’

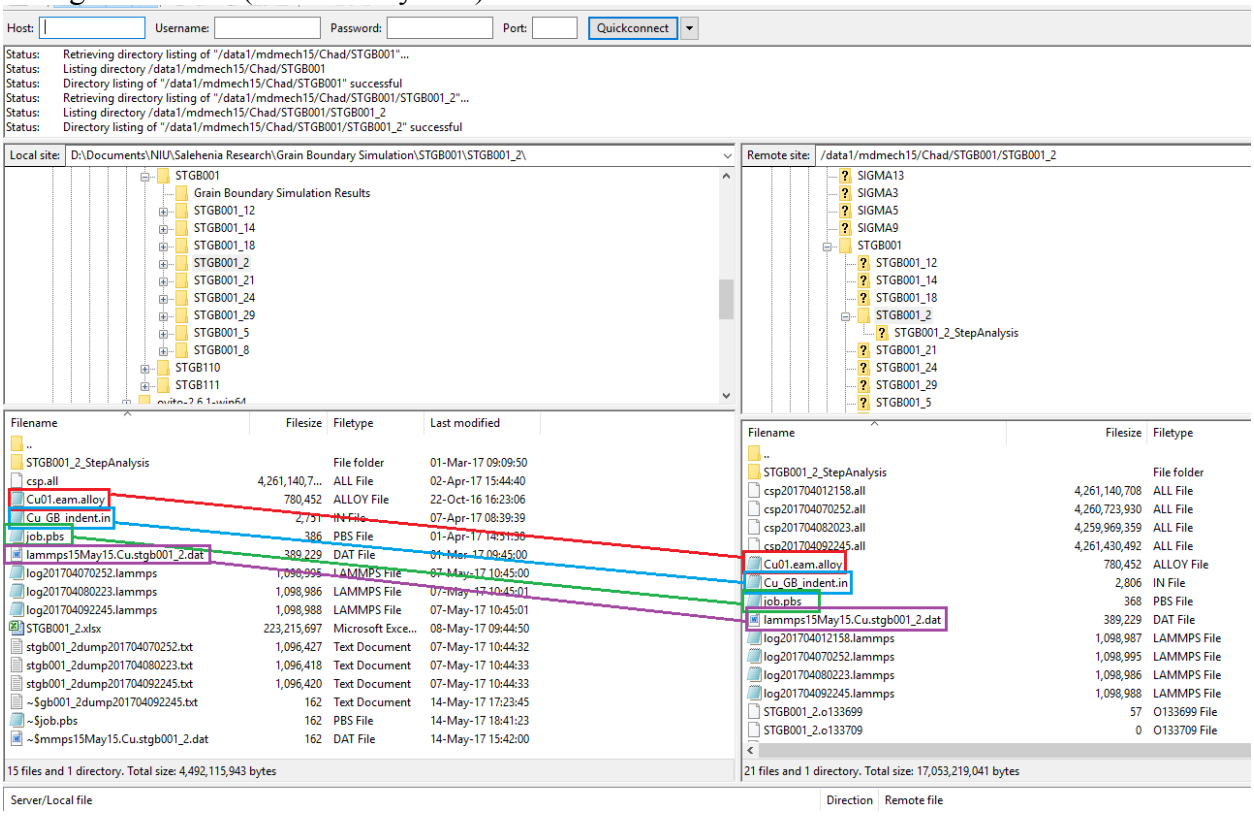


You can create a new folder to dump the files needed for the simulation by:
Right click file (on server) > create directory

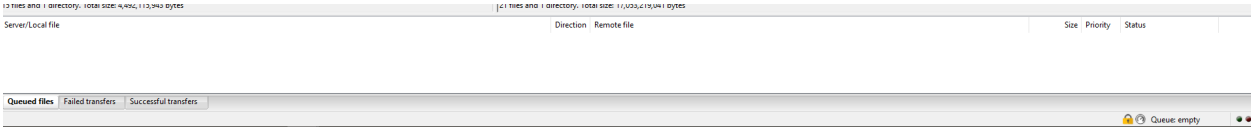


NOTE: Organization is critical! You will be running many simulations so plan your folder management and file naming system carefully.

Now it is time to transfer the files. Simply navigate the files in the left window (the files on your pc) to find the files you would like to transfer. Then double click on the files to transfer them to the right windows (the server system).



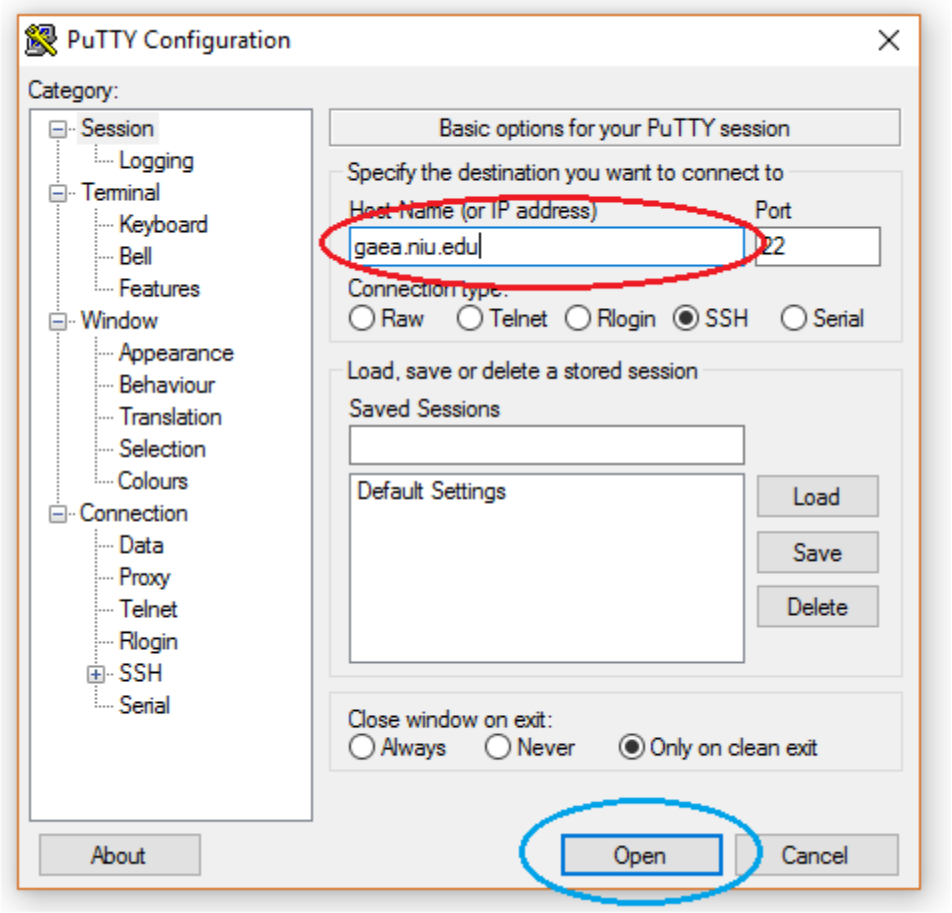
The transfer status of files can be viewed at the bottom of the FileZilla window:



With those four files now on the GAEA server it is time to run the simulation.

Running the Nano Indentation Code on GAEA:

Start by opening PuTTY. Enter ‘gaea.niu.edu’ for the host name and click open.



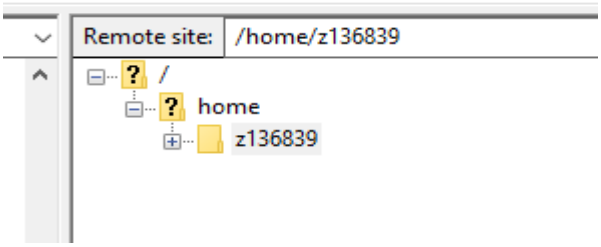
You should get a black login screen next.

Enter in you user name where it says login, and then type in your password when prompted. Note: your password will not display as you type it. Just type it in and hit enter.
Now that you are logged in, you will need to navigate to where you placed your files. Since GAEA runs on linux, you will need to use the linux commands to navigate the server.
cd – change directory
ls – list the files in the directory
ls –a – list all files in the directory

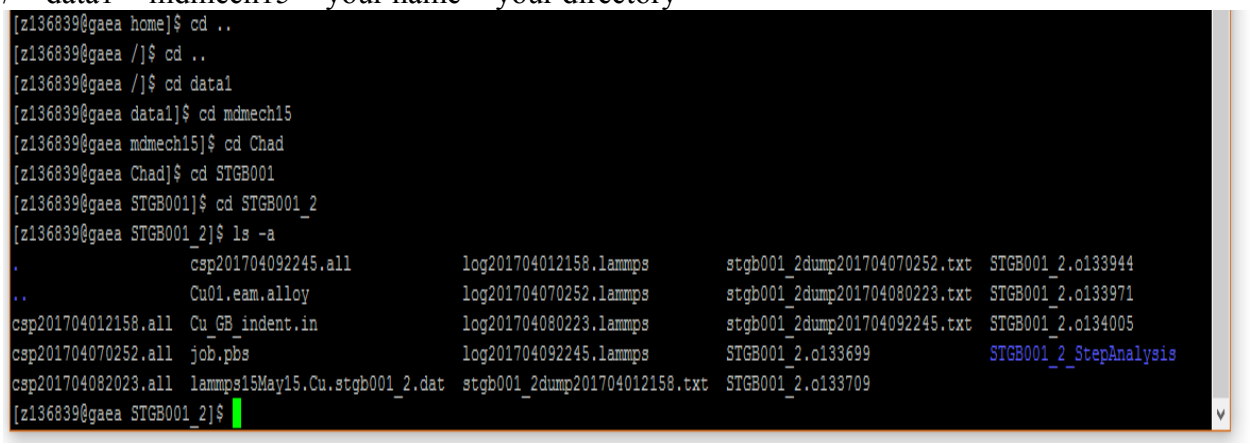
This site provides some good info on using PuTTY: http://www.mybloggingthing.com/ssh-putty-commands/

you can move up directories by typing 'cd ..' and hitting enter and then move into directories by typing 'cd filename' and hitting enter.

Note: you start in the same directory in PuTTY as you do in FileZilla:



So you need to move up two directories (do 'cd ..' twice). Then navigate to your directory:
/ > data1 > mdmech15 > your name > your directory



Note: White items are files and blue items are directories.

Now that you have found the directory with the files you uploaded, you are almost ready to run. But before you do so you must convert the job.pbs file to a unix format (since GAEA operates on a version of Linux, it has to read the commands in a unix format).

To do this simply type:
dos2unix job.pbs (or dos2unix filename.pbs)

```
csp201704082023.all  lammps15May15.Cu.stgb001_2.dat  st
[z136839@gaea STGB001_2]$ dos2unix job.pbs
dos2unix: converting file job.pbs to Unix format ...
[z136839@gaea STGB001_2]$
```

Now you can queue up your simulation in GAEA. Use these commands to manage the your simulations on the server through PuTTY:

qsub job.pbs (qsub filename.pbs) – start running your simulation

```
dos2unix: converting file job.pbs to Unix
[z136839@gaea STGB001_2]$ qsub job.pbs
135316.gaea-sn1.its.niu.edu
[z136839@gaea STGB001_2]$
```

Note: the six or seven digit number it outputs is the job number of your simulation

qstat – checks the status of the simulations currently running on GAEA

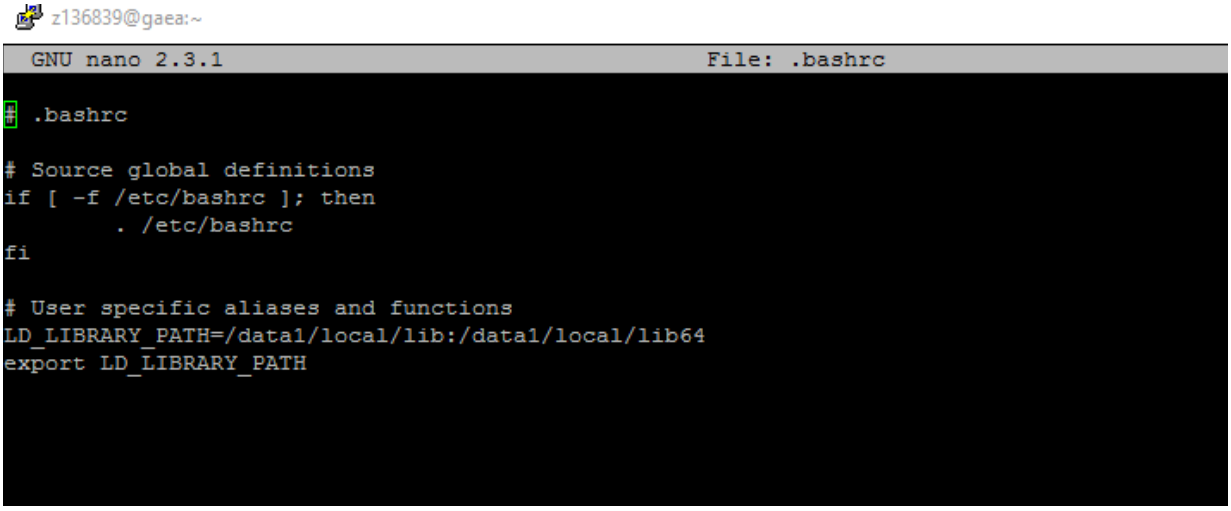
```
135316.gaea-sn1.its.niu.edu
[z136839@gaea STGB001_2]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
134756.gaea-sn1            ...NTrelax&kappa z1791832         1254:11: R special
135292.gaea-sn1            3WCNT_inner5     z1815572         287:25:4 R medium
135293.gaea-sn1            3WCNT_inner5     z1815572         287:10:5 R medium
135294.gaea-sn1            3WCNT_inner5     z1815572         287:02:4 R medium
135295.gaea-sn1            3WCNT_inner5     z1815572         286:55:3 R medium
135296.gaea-sn1            3WCNT_inner5     z1815572         286:41:5 R medium
135297.gaea-sn1            3WCNT_inner      z1815572         286:14:2 R medium
135298.gaea-sn1            3WCNT_Outer5     z1815572         286:14:1 R medium
135299.gaea-sn1            3WCNT_Outer5     z1815572         227:53:3 R medium
135300.gaea-sn1            3WCNT_Outer5     z1815572         228:06:1 R medium
135301.gaea-sn1            3WCNT_Outer5     z1815572         227:52:1 R medium
135302.gaea-sn1            HAP_ELASTIC      z1784827         76:23:11 R medium
135303.gaea-sn1            3WCNT_middle5    z1815572                0 Q medium
135304.gaea-sn1            3WCNT_middle5    z1815572                0 Q medium
135305.gaea-sn1            3WCNT_middle5    z1815572                0 Q medium
135306.gaea-sn1            3WCNT_middle5    z1815572                0 Q medium
135307.gaea-sn1            3WCNT_middle5    z1815572                0 Q medium
135316.gaea-sn1            STGB001_2        z136839                 0 Q medium
[z136839@gaea STGB001_2]$
```

The job number is listed first, then the job name (red), the user (blue), the wall run time (shows total computation time of all cores not the time it has spent on the wall), the status (green; Q-queued, R – running, C – complete), and priority (purple; this changes based on wall time requested and admin permissions).

Note: I had to modify my bashrc file because the simulations would not run for some reason. So navigate back to your starting folder:
/ > home > user id

Then type ‘nano .bashrc’ (you can find a tutorial on using nano here: https://wiki.gentoo.org/wiki/Nano/Basics_Guide)

Make sure it looks like this:



Now with the simulation queued, it's time to wait until it is complete. You will get an email from GAEA when it is done.

Analysing the Simulation Data:

Once you get the email that the simulation is complete it is time to gather the data. Once again open FileZilla, connect to GAEA and navigate to the folder the simulation was run in. The LAMMPS will create three files when it runs:

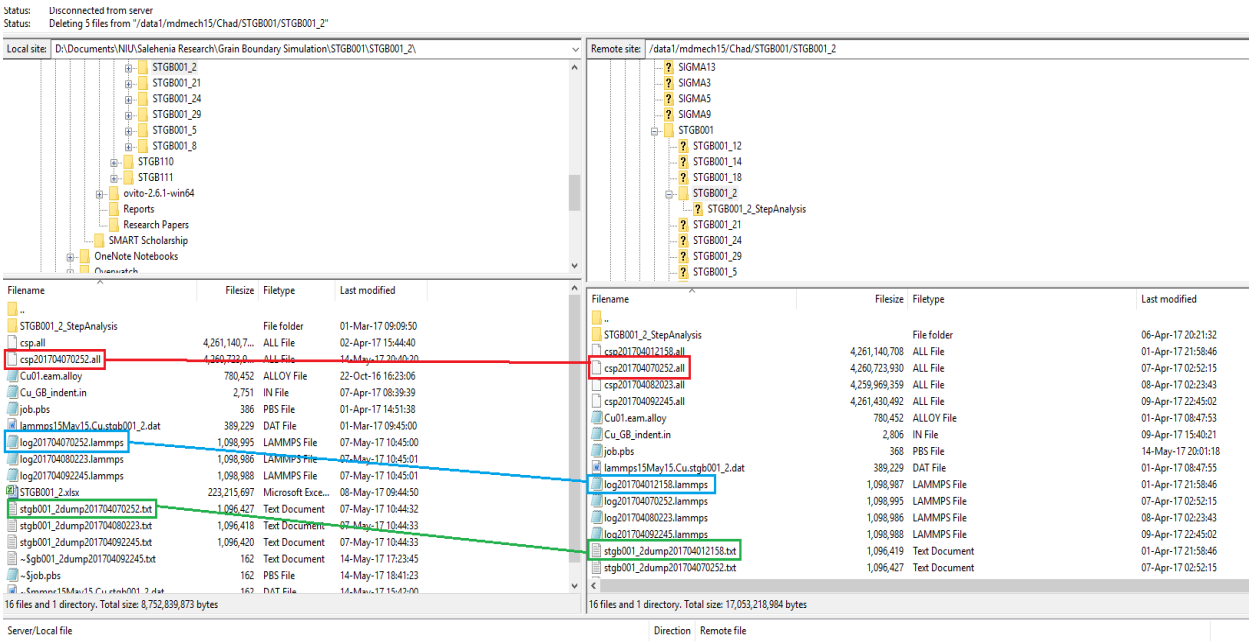
Csp.all (this is the visualization data specified in the Define Settings section of the indentation code)

Log.lammps (a log file created by LAMMPS)

'filename'dump.txt (this is the dump file where all the data from the minimization, relaxation, and indentation loops during the simulation are placed. The file name for this dump file is changed in job.pbs.)

Note: I add in the time stamp of when the file was last modified on GAEA to help keep track of what simulation it was and when I conducted it. Also the timestamp will help automatically order the files from oldest to newest (YYYYMMDDhhmm). The spreadsheet with all of this tracked data is in the STGB001 Research Progress excel.

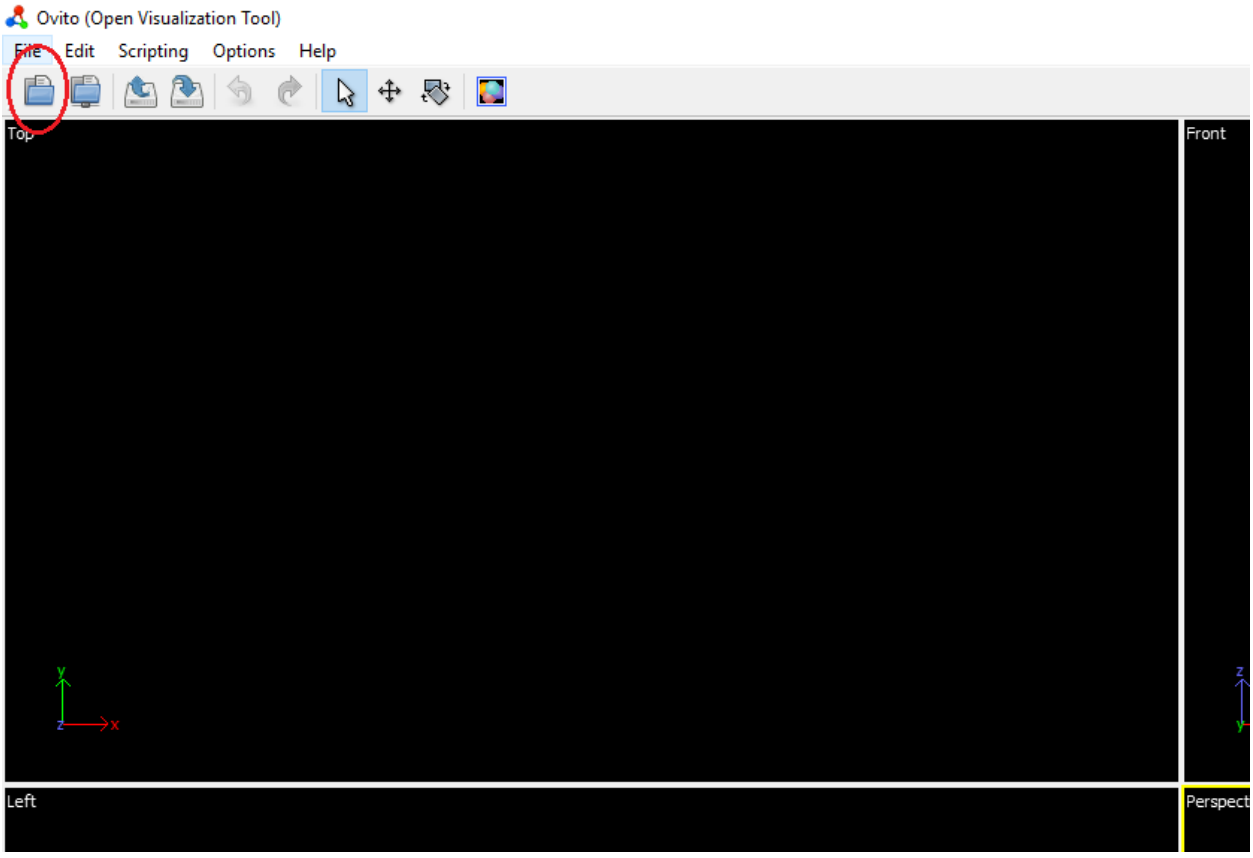
Double click on each of these files to transfer it from the server (right side) to your pc (left side). Beware of the file size for csp.all; it will be several gigabytes.



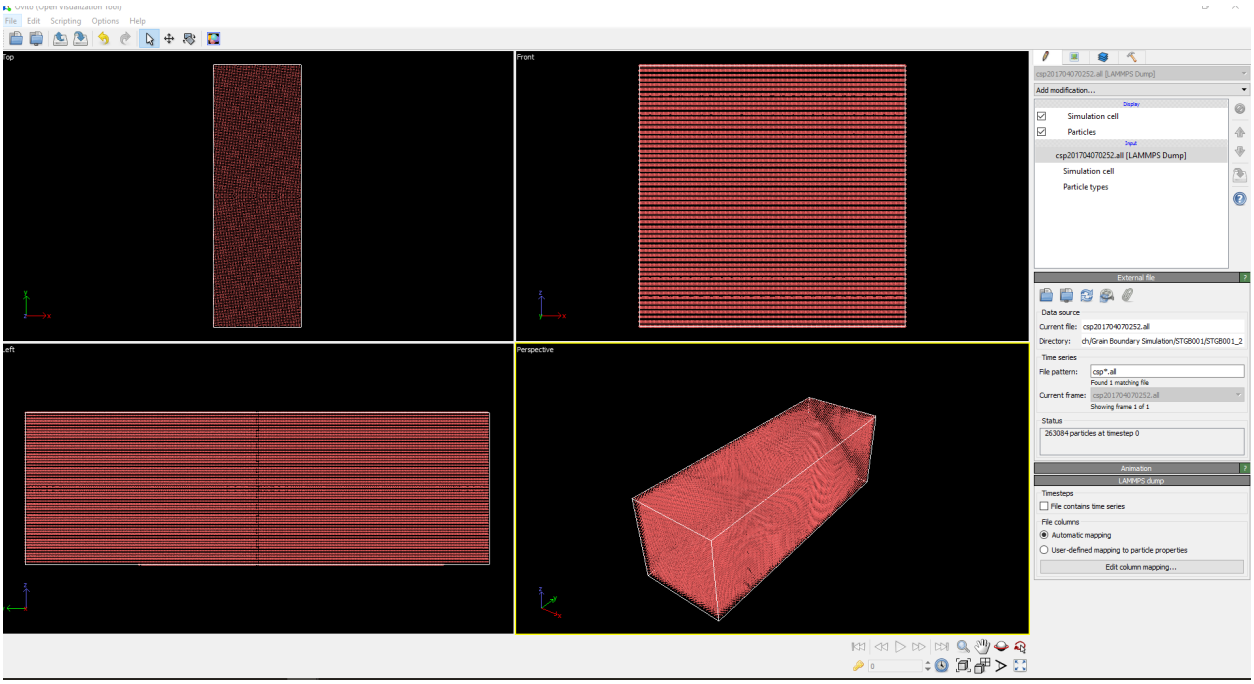
Visualization using OVITO:

Once the transfer of the files is complete open up OVITO.

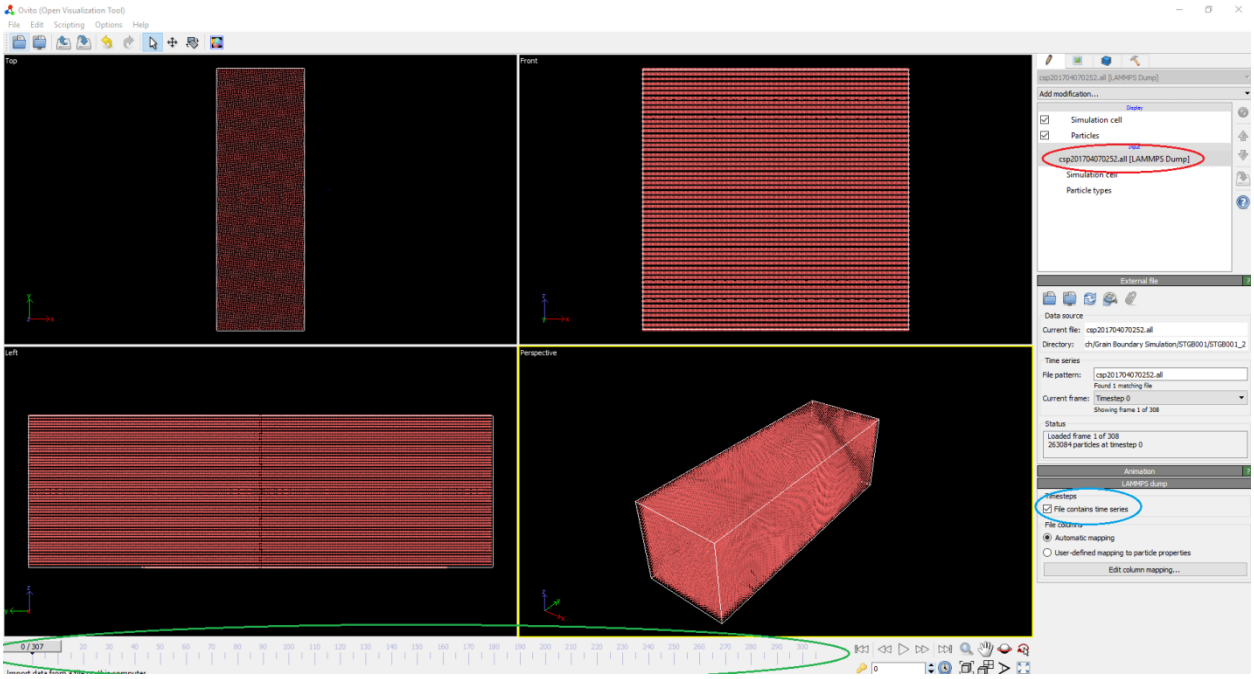
To load the cspYYYYMMDDhhmm.all file, either click file > load or click the circled button:



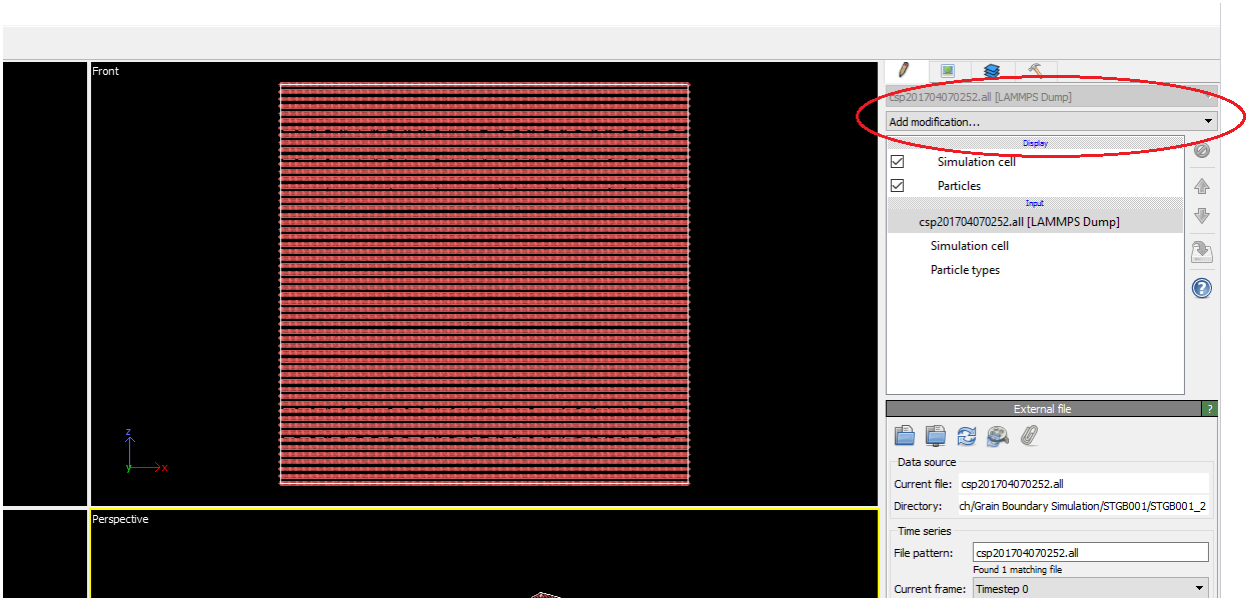
Navigate to the csp file, highlight it, and click open (or double click on it). OVITO should now look like this:

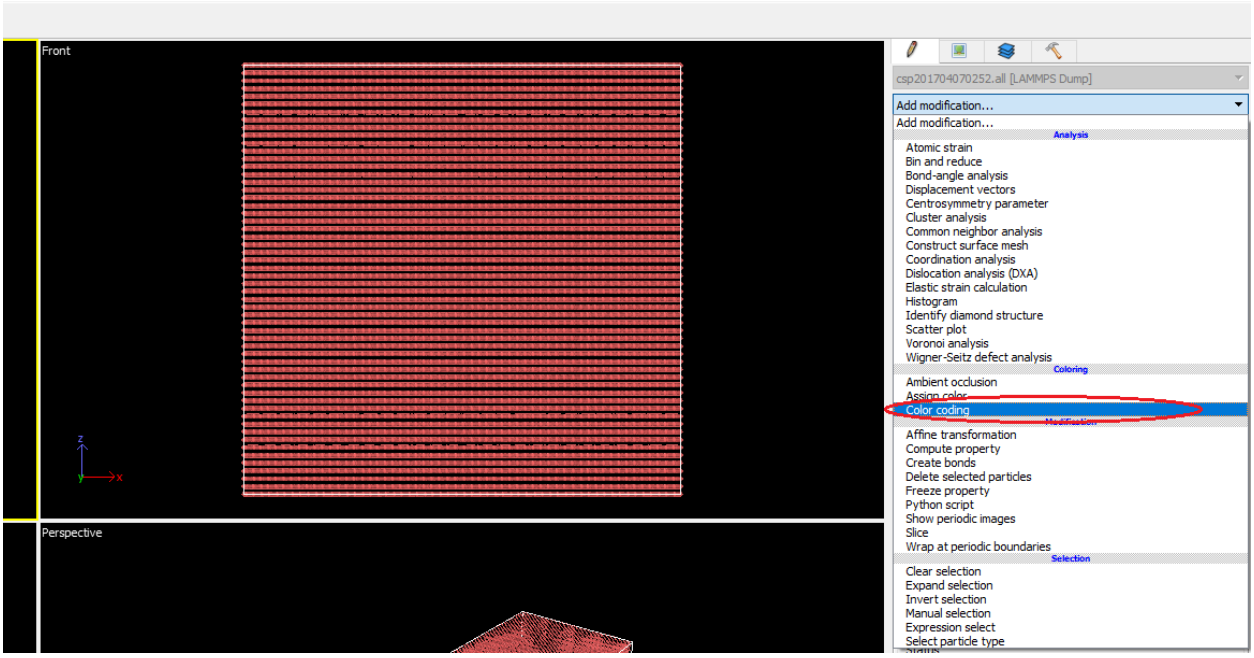


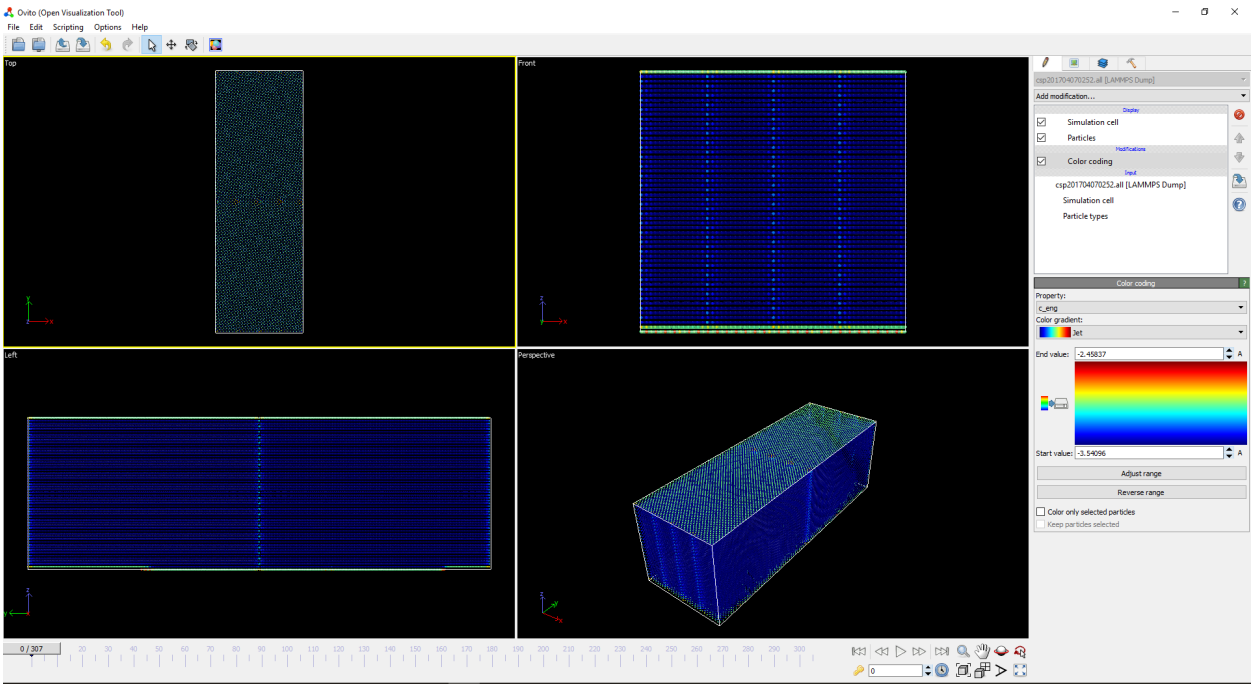
Next we want to load all of the step data. With the csp.all file highlighted (circled in red), check the ‘File contains time series’ box (circled in blue). When the data is loaded you should see bar with all of the scenes at the bottom (circled in green).



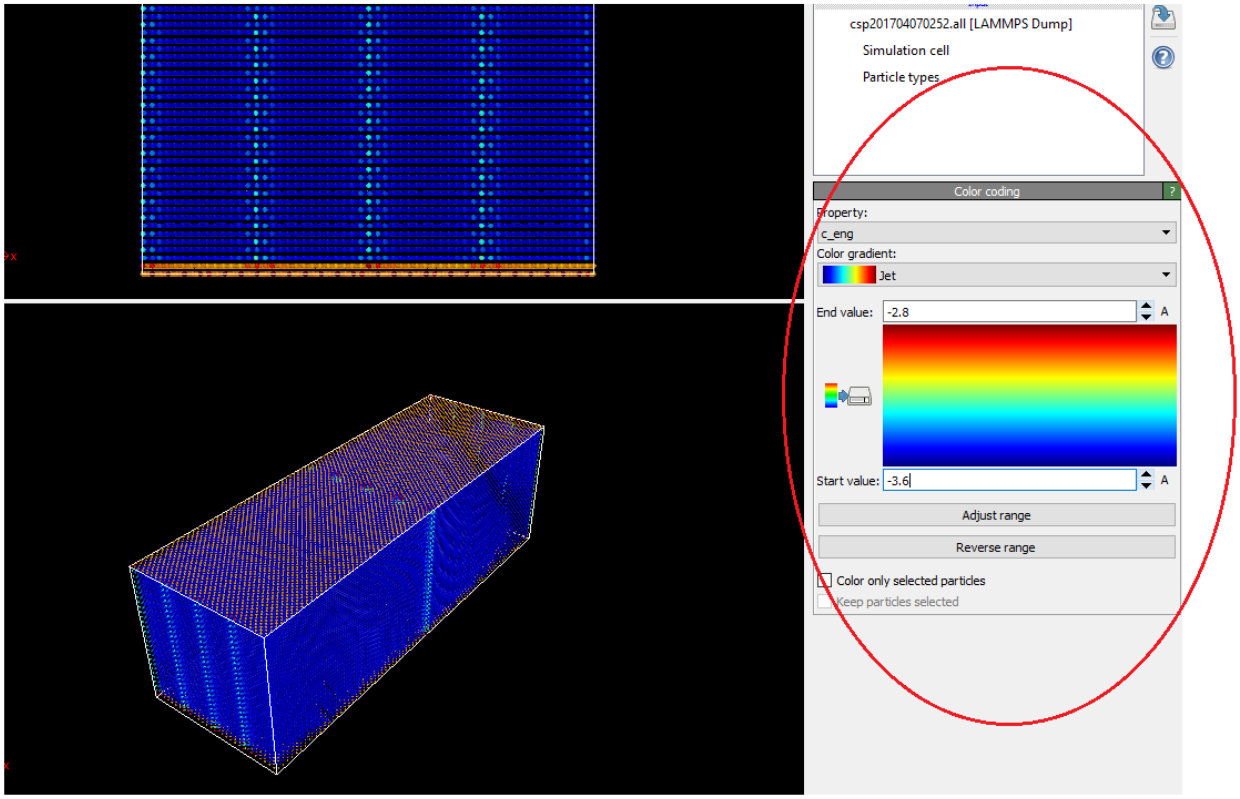
You can now hit the play button and watch the simulation progress. However with all the atoms displayed it can be difficult to see what is happening. So we will add some modifications. At the top click add modifications > color coding:



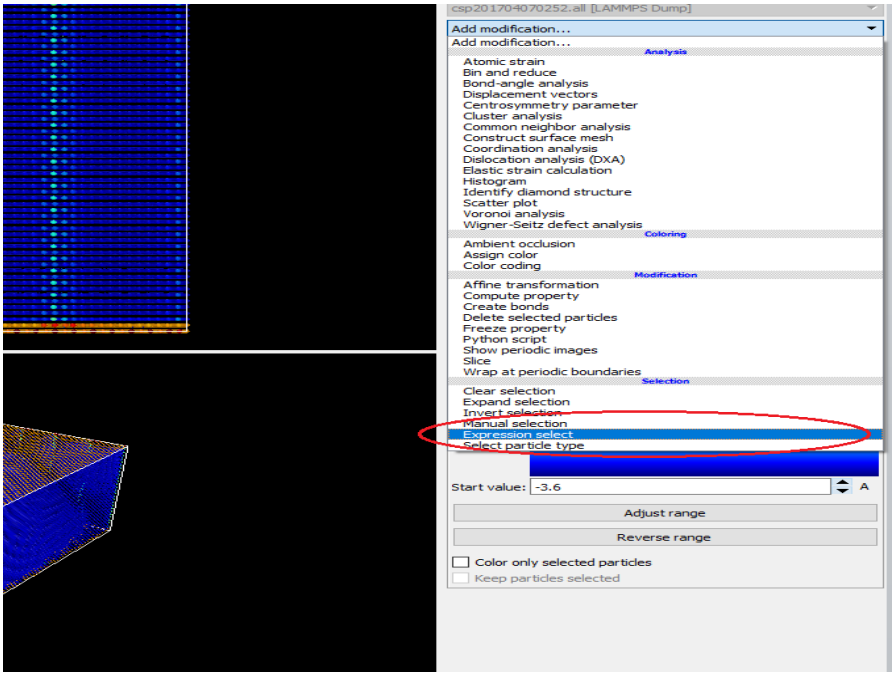




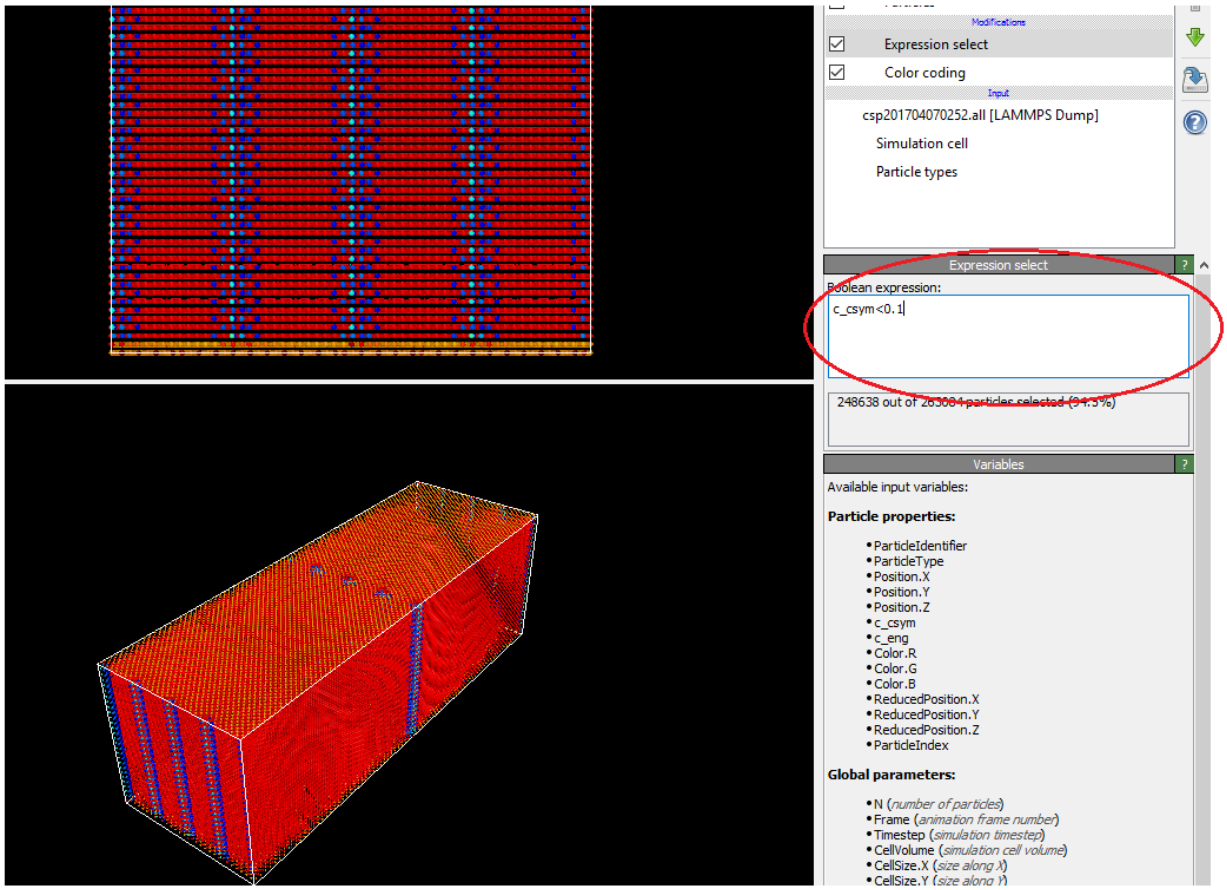
The property type (that get colored), the color gradient, and the values along the gradient can be modified in the side window:



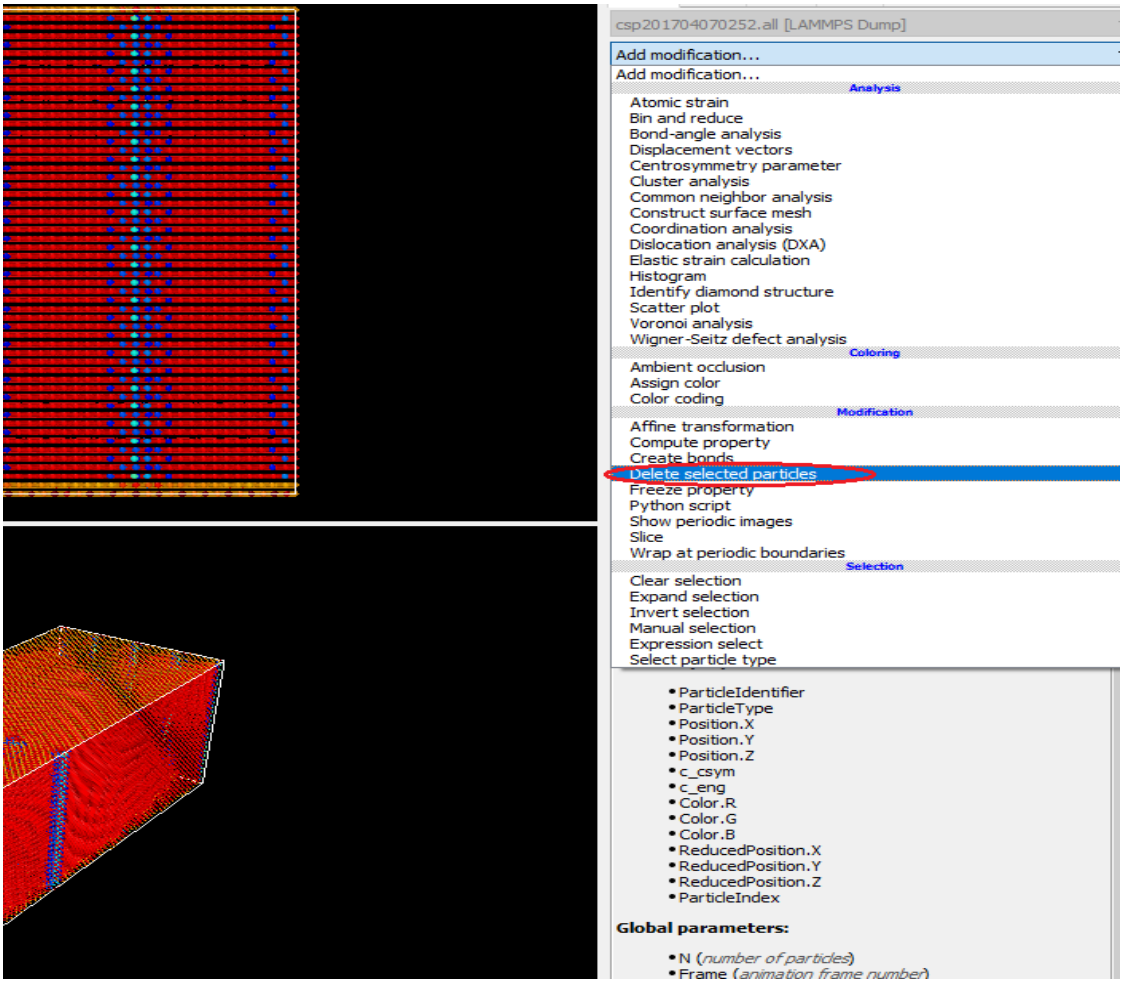
To remove atoms, we first have to select the atoms using the following:
Add modification > expression select

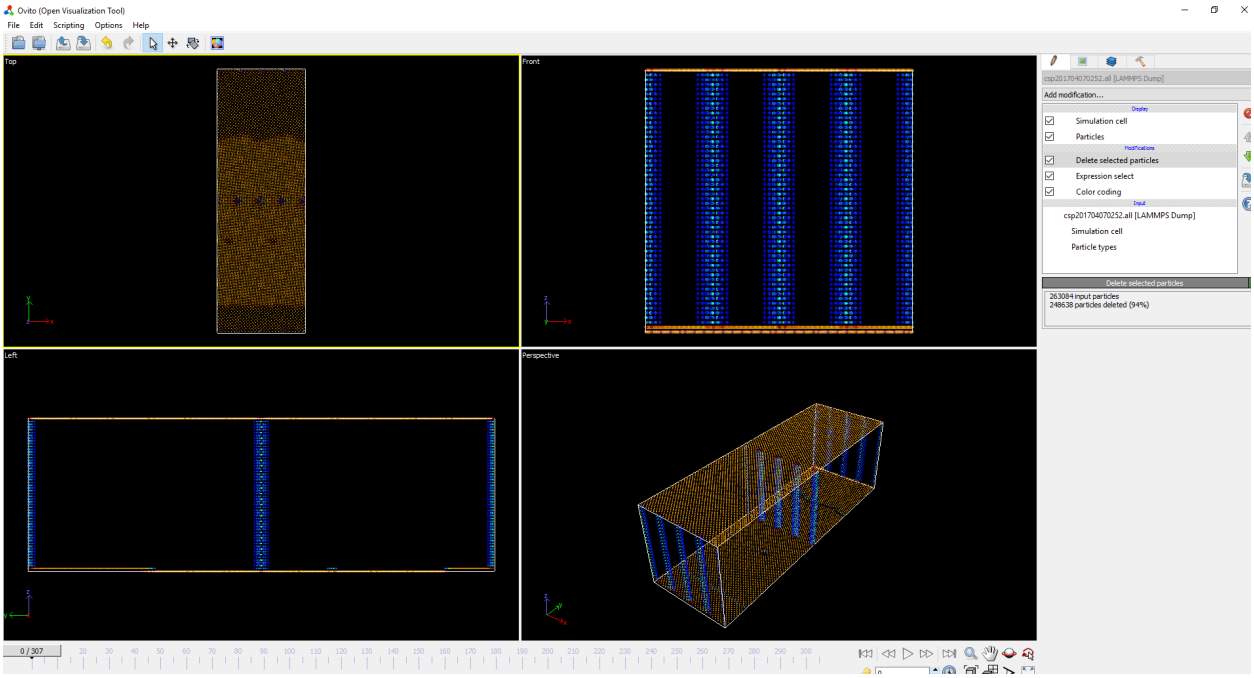


Particles can then be selected by entering in various inequalities in the Boolean expression window (in this case c_csym<0.1) and hitting enter.
Note: multiple expression can be used to target the particles you want.

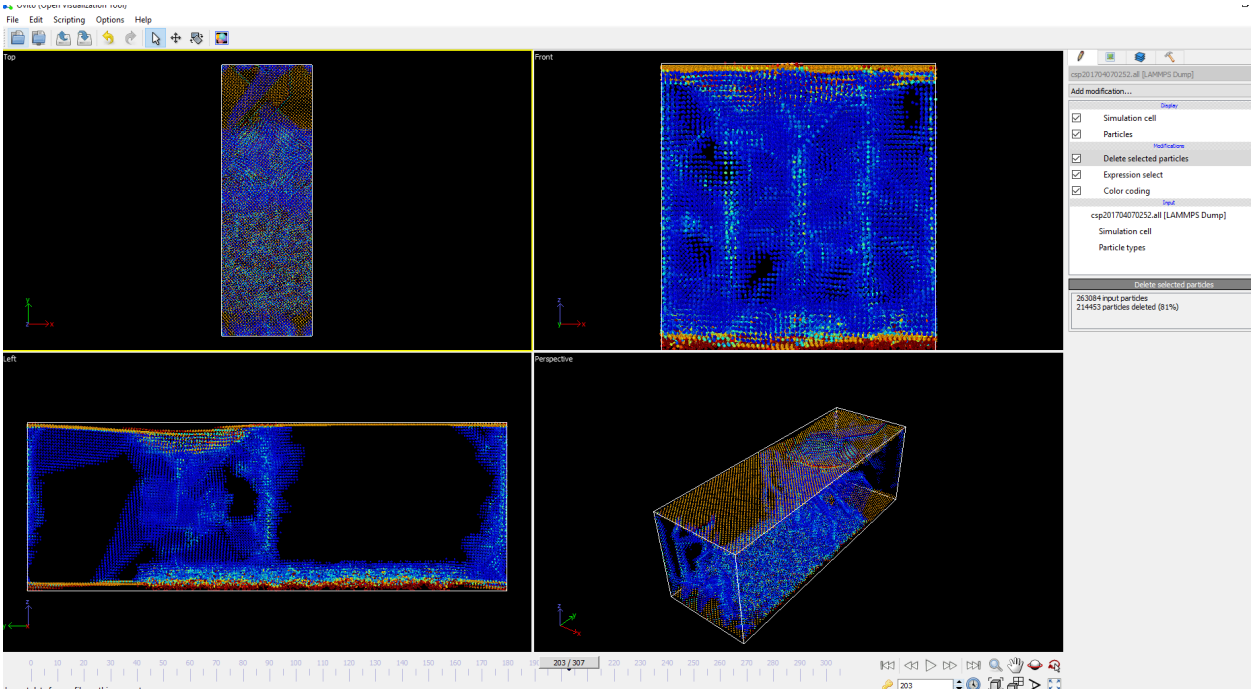


To remove the selected atoms:
Add modification > delete selected particles

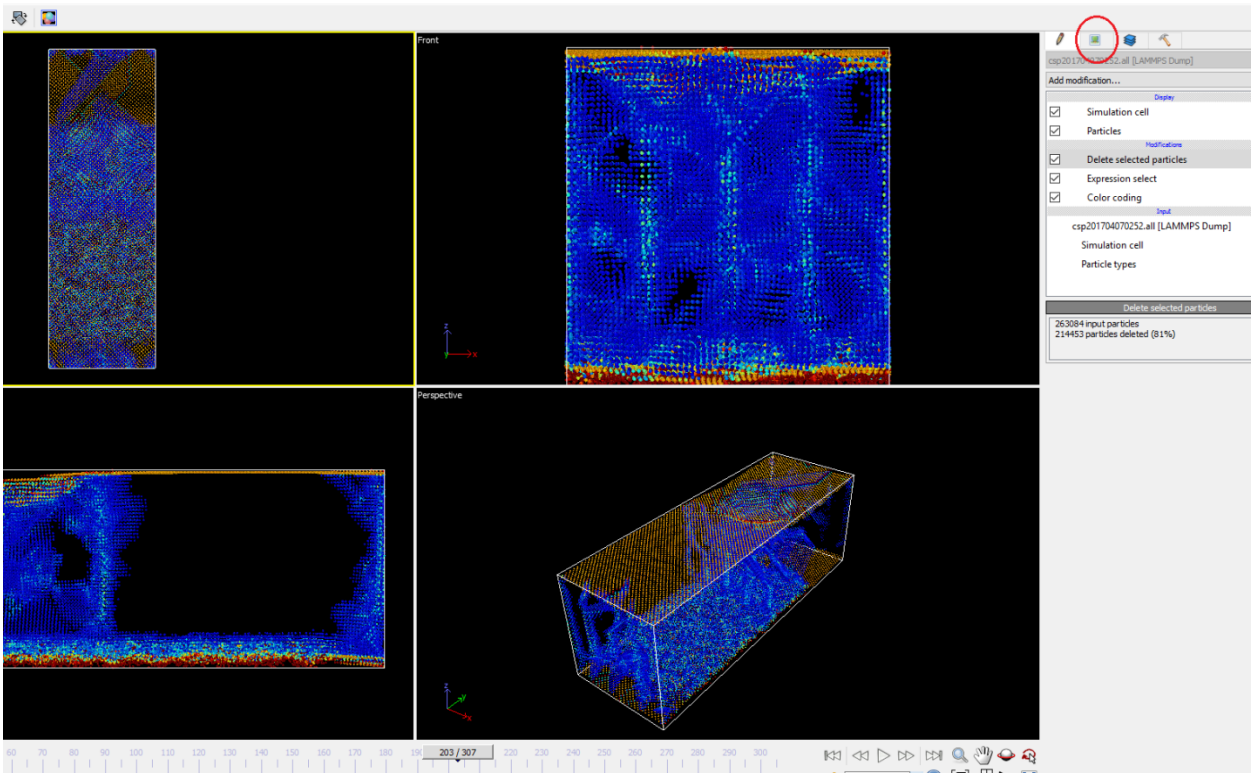

csp201704070252.all [LAMMPS Dump]
Add modification...
Add modification...
Analysis
Atomic strain
Bin and reduce
Bond-angle analysis
Displacement vectors
Centrosymmetry parameter
Cluster analysis
Common neighbor analysis
Construct surface mesh
Coordination analysis
Dislocation analysis (DXA)
Elastic strain calculation
Histogram
Identify diamond structure
Scatter plot
Voronoi analysis
Wigner-Seitz defect analysis
Coloring
Ambient occlusion
Assign color
Color coding
Modification
Affine transformation
Compute property
Create bonds
Delete selected particles
Freeze property
Python script
Show periodic images
Slice
Wrap at periodic boundaries
Selection
Clear selection
Expand selection
Invert selection
Manual selection
Expression select
Select particle type
ParticleIdentifier
ParticleType
Position.X
Position.Y
Position.Z
c_csym
c_eng
Color.R
Color.G
Color.B
ReducedPosition.X
ReducedPosition.Y
ReducedPosition.Z
ParticleIndex
Global parameters:
N (number of particles)
Frame (animation frame number)


Ovito (Open Visualization Tool)
File Edit Scripting Options Help
Top
Front
Left
Perspective
Simulation cell
Particles
Delete selected particles
Expression select
Color coding
csp201704070252.all [LAMMPS Dump]
Simulation cell
Particle types
263084 input particles
248638 particles deleted (94%)

Now when you hit play at the bottom of the window, you will be able to see the indentation more clearly.



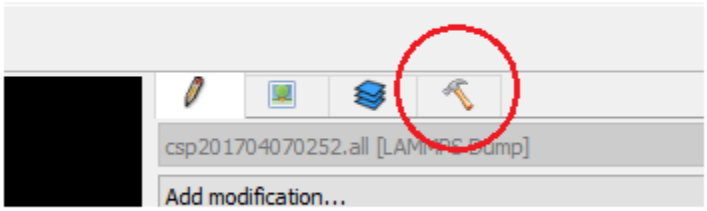
You can take pictures and video of these visualizations in the highlighted tab:

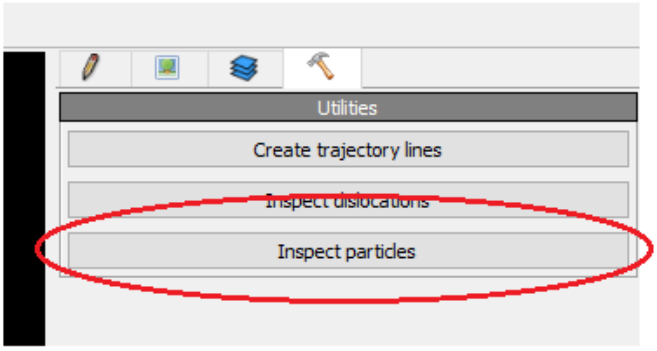


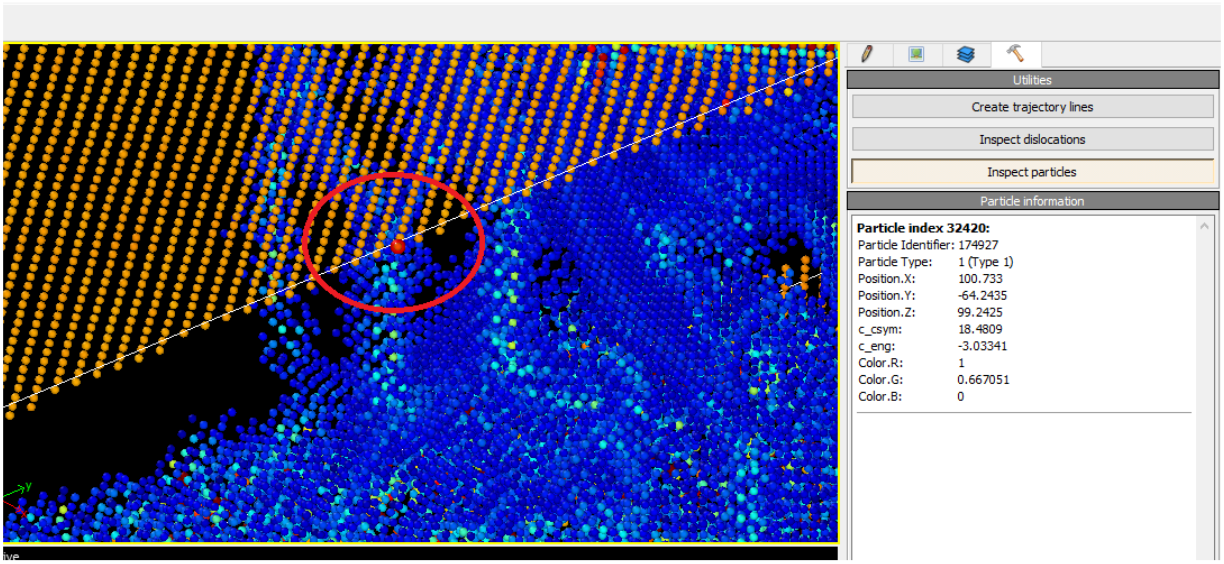
Details on how to do so can be found here:

https://www.ovito.org/manual/usage.rendering.html

Note: a useful tool can be found in the last tab. This is the inspect particles tool. It allows you to click on a particle in the model and it will output the properties of that particle. This is useful when adjustments need to be made to the indentation code.



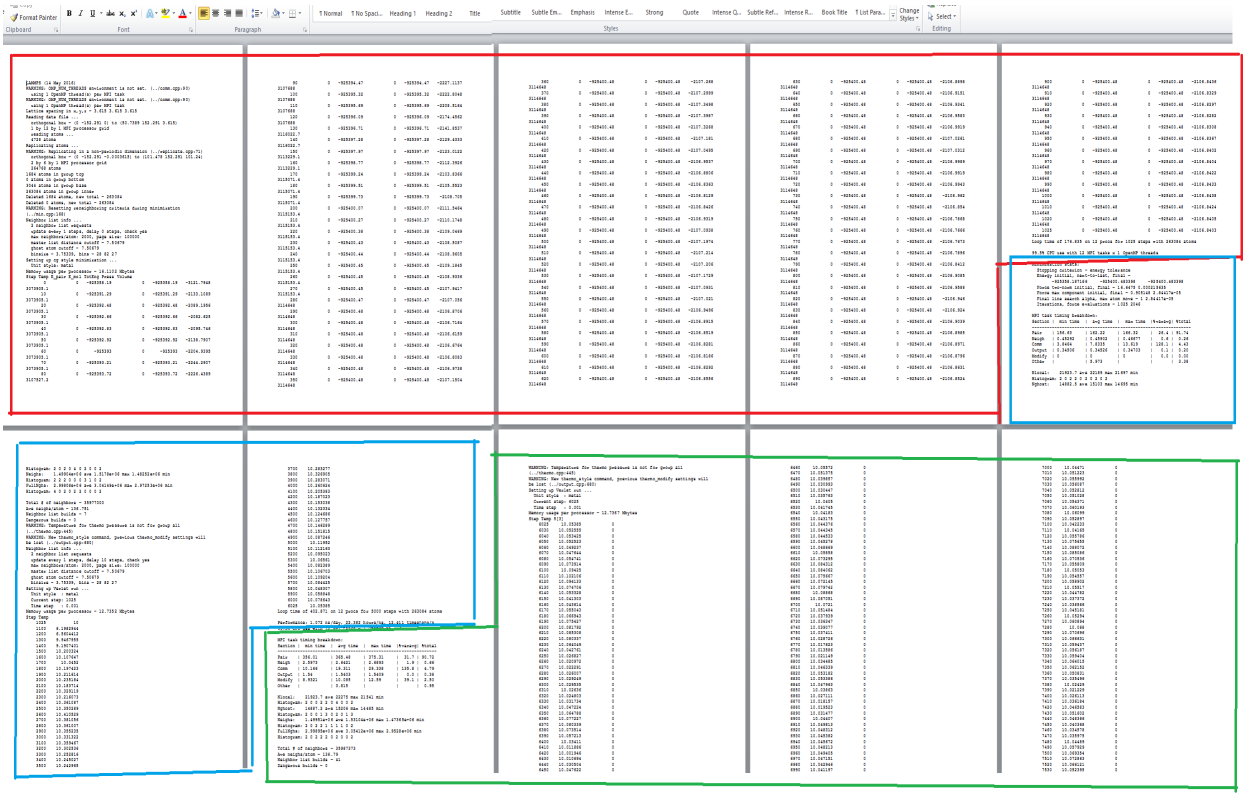




Another useful OVITO tool is the Dislocation Analysis (DXA) modification. Information on this can be found here: http://ovito.org/manual/particles.modifiers.dislocation_analysis.html

Creating Load Displacement Curves:

Now it is time to take the raw data from the dump.txt file and create usable graphs from it. Go to the data.txt file and open it in Microsoft Word. You can see in the document that we have the dumps from the minimization (red), the relaxation (blue), and the indentation (green).

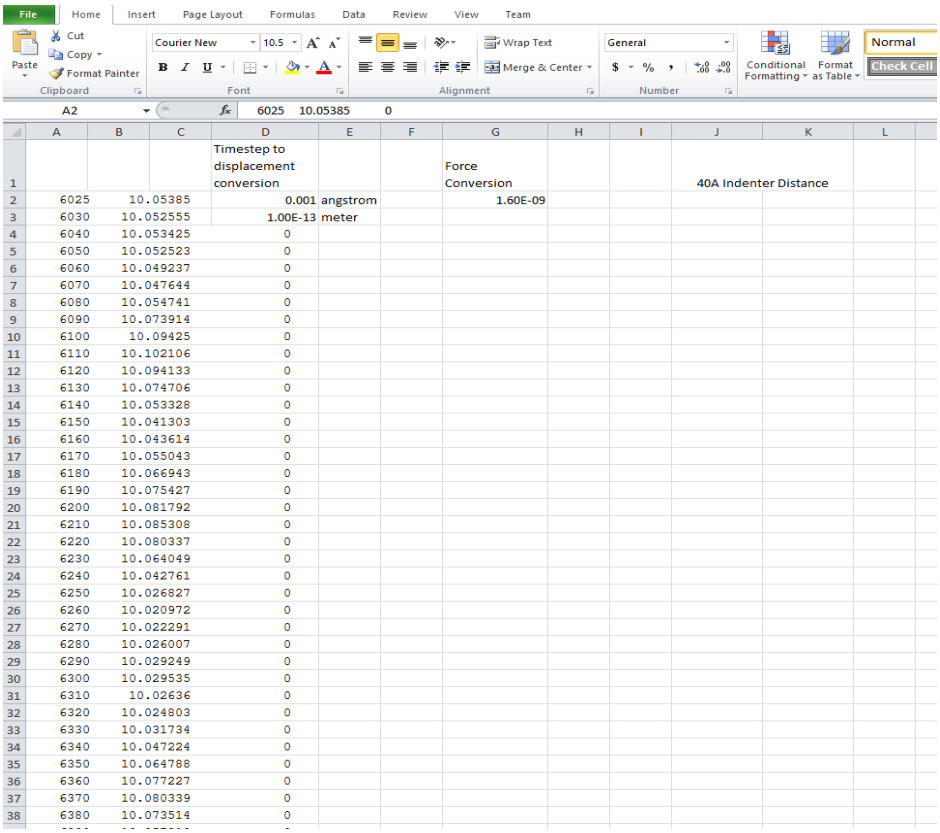


We only need the data from indentation so delete everything above the purple line, and everything below the blue line.

```
WARNING: Temperature for thermo pressure is not for group all
(../thermo.cpp:445)
WARNING: New thermo_style command, previous thermo_modify settings will
be lost (../output.cpp:680)
Setting up Verlet run ...
  Unit style    : metal
  Current step: 6025
  Time step     : 0.001
Memory usage per processor = 12.7367 Mbytes
Step Temp 5[3]
      6025     10.05385            0
      6030    10.052555            0
      6040    10.053425            0
      6050    10.052523            0
      6060    10.049237            0
      6070    10.047644            0
      6080    10.054741            0
      6090    10.073914            0
      6100     10.09425            0
      6110    10.102106            0
      6120    10.094133            0
      6130    10.074706            0
      6140    10.053328            0
      6150    10.041303            0
      6160    10.043614            0
      6170    10.055043            0
      6180    10.066943            0
      6190    10.075427            0
      6200    10.081792            0
      6210    10.085308            0
      6220    10.080337            0
      6230    10.064049            0
      6240    10.042761            0
      6250    10.026027            0

305710    10.025584    285.76456
305720     10.02211    286.53732
305730    10.014867    286.82643
305740    10.011127    286.62476
305750    10.014797    286.16666
305760    10.022558    285.81321
305770    10.029376    285.82262
305780    10.033396     286.1581
305790    10.034384    286.50044
305800     10.03327    286.42228
305810     10.03105    285.61666
305820    10.026922     284.2496
305830    10.020985    282.89481
305840    10.016789    282.13765
305850    10.017454    282.22474
305860    10.021064    283.05382
305870    10.023073    284.37967
305880    10.021501     285.9683
305890    10.018227    287.51754
305900    10.017092    288.57631
305910    10.020015    288.75315
305920    10.024927    288.05038
305930    10.028063    286.85256
305940    10.026985    285.65179
305950    10.022054    284.79212
305960    10.016513    284.36818
305970    10.013911    284.32889
305980    10.015075    284.58716
305990    10.018391    284.93277
306000     10.02218    285.08903
306010    10.025523    284.90927
306020    10.027374     284.4737
306025    10.027302    284.26328
Loop time of 24908 on 12 procs for 300000 steps with 263084 atoms

Performance: 1.041 ns/day, 23.063 hours/ns, 12.044 timesteps/s
99.5% CPU use with 12 MPI tasks x 1 OpenMP threads

MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
---------------------------------------------------------------
Pair    | 22030      | 22458      | 22873      | 192.6 | 90.16
Neigh   | 87.739     | 90.754     | 92.876     |  19.4 |  0.36
Comm    | 878.54     | 1296.5     | 1788.3     | 931.8 |  5.21
Output  | 95.271     | 95.511     | 95.798     |   1.5 |  0.38
```

Note: You should have 300,000 lines total since that is the number of loops that were specified with 'run' in the indenter code. Also you can jump the bottom of the text by click and dragging the slider at the side of the word document to the bottom.

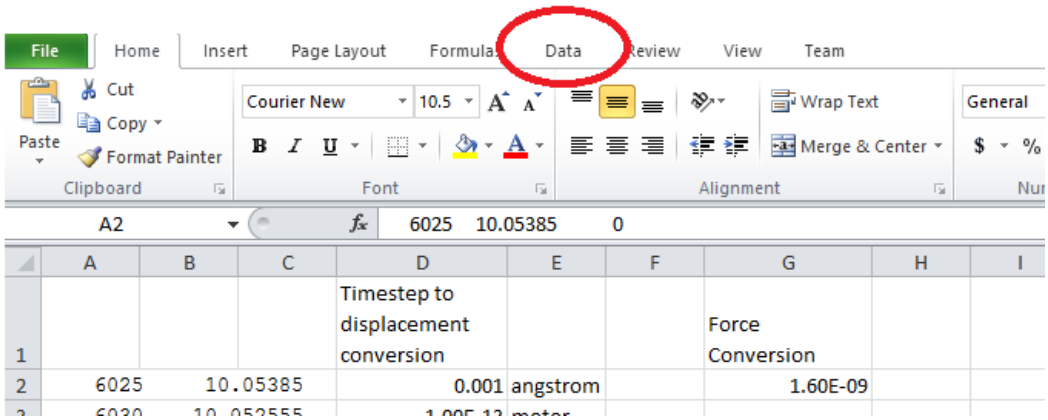
Now ctrl + a, to select all of the data, and ctrl + c to copy it; then with Microsoft Excel open paste it. (You can close out the word document but don't save it otherwise you lose the data that was deleted)

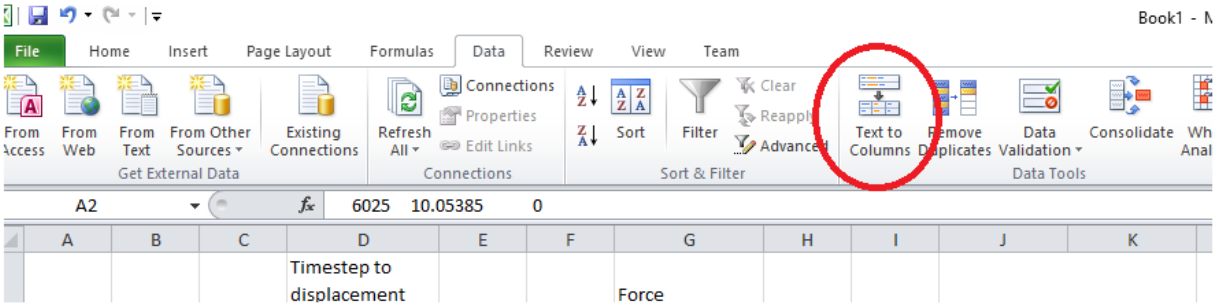


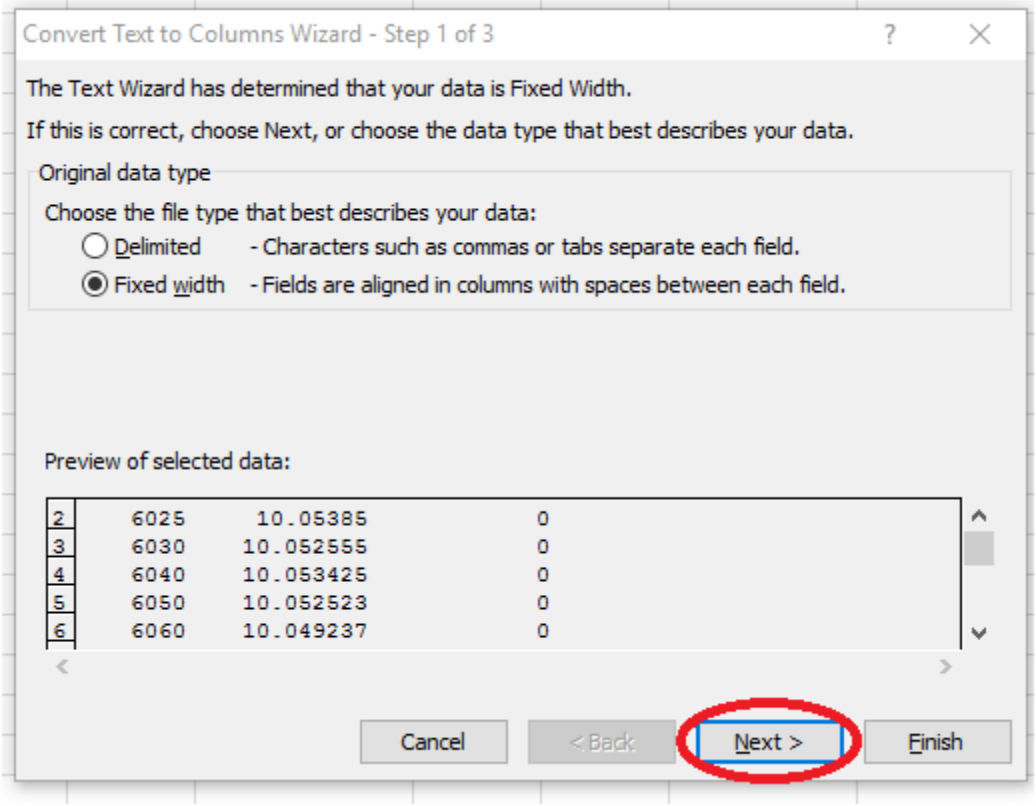
	A	B	C	D	E	F	G	H	I	J	K	L
1				Timestep to displacement conversion			Force Conversion			40A Indenter Distance		
2	6025	10.05385		0.001	angstrom		1.60E-09					
3	6030	10.052555		1.00E-13	meter							
4	6040	10.053425		0								
5	6050	10.052523		0								
6	6060	10.049237		0								
7	6070	10.047644		0								
8	6080	10.054741		0								
9	6090	10.073914		0								
10	6100	10.09425		0								
11	6110	10.102106		0								
12	6120	10.094133		0								
13	6130	10.074706		0								
14	6140	10.053328		0								
15	6150	10.041303		0								
16	6160	10.043614		0								
17	6170	10.055043		0								
18	6180	10.066943		0								
19	6190	10.075427		0								
20	6200	10.081792		0								
21	6210	10.085308		0								
22	6220	10.080337		0								
23	6230	10.064049		0								
24	6240	10.042761		0								
25	6250	10.026827		0								
26	6260	10.020972		0								
27	6270	10.022291		0								
28	6280	10.026007		0								
29	6290	10.029249		0								
30	6300	10.029535		0								
31	6310	10.02636		0								
32	6320	10.024803		0								
33	6330	10.031734		0								
34	6340	10.047224		0								
35	6350	10.064788		0								
36	6360	10.077227		0								
37	6370	10.080339		0								
38	6380	10.073514		0								

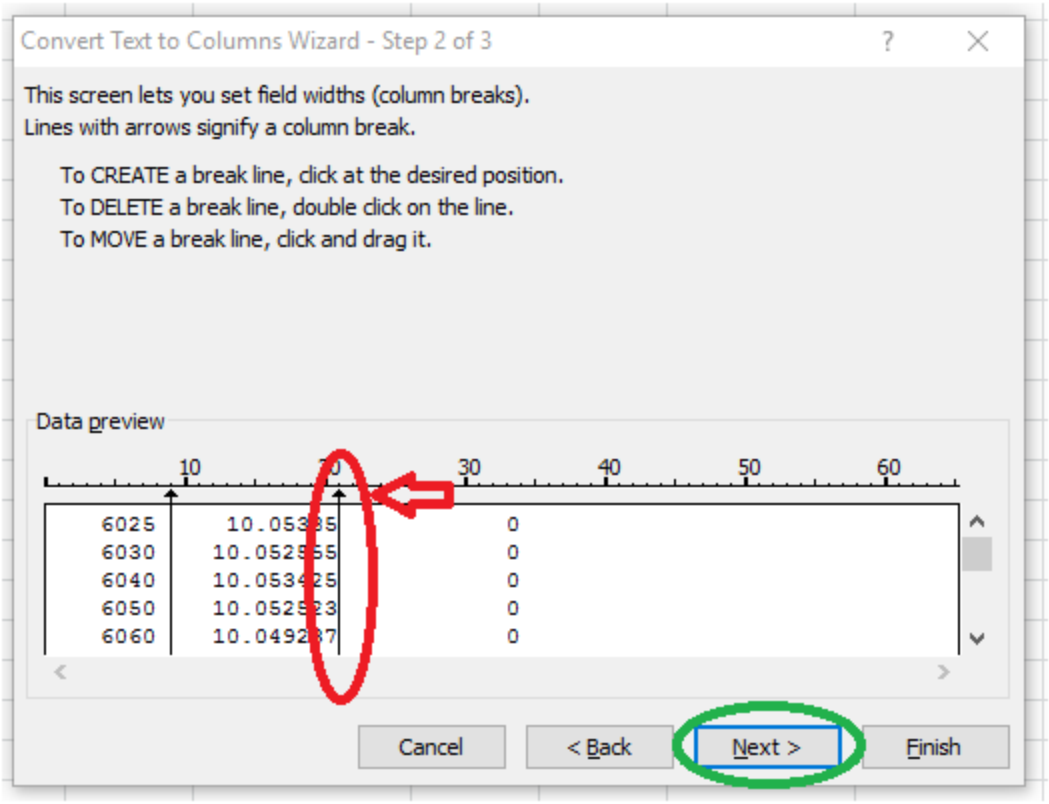
Since it was imported into excel as a single set of data, it now needs to be separated. With the column highlighted click:
Data > Text to columns

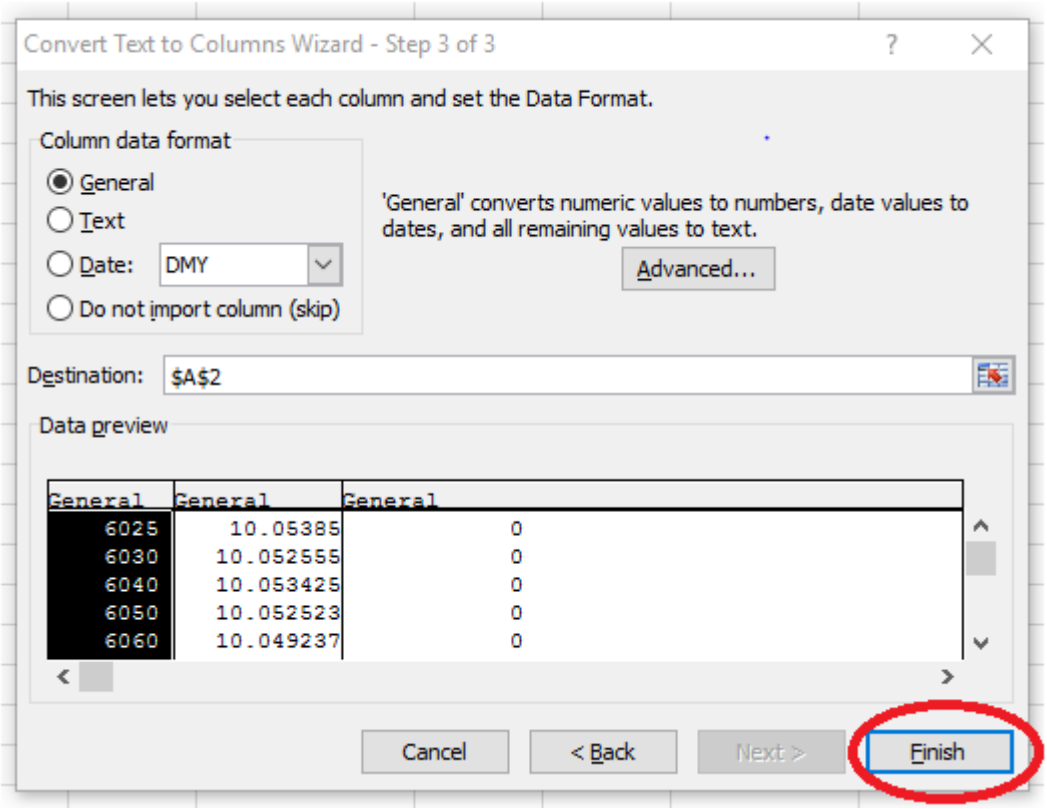
When the window pops up click:
Next > (move the slider as close to the second column as possible) then hit next > Finish


File
Home
Insert
Page Layout
Formulas
Data
Review
View
Team
Cut
Copy
Format Painter
Paste
Clipboard
Courier New
10.5
Font
Wrap Text
Merge & Center
Alignment
General
A2
6025 10.05385 0
Timestep to displacement conversion
Force Conversion
6025
10.05385
0.001
angstrom
1.60E-09

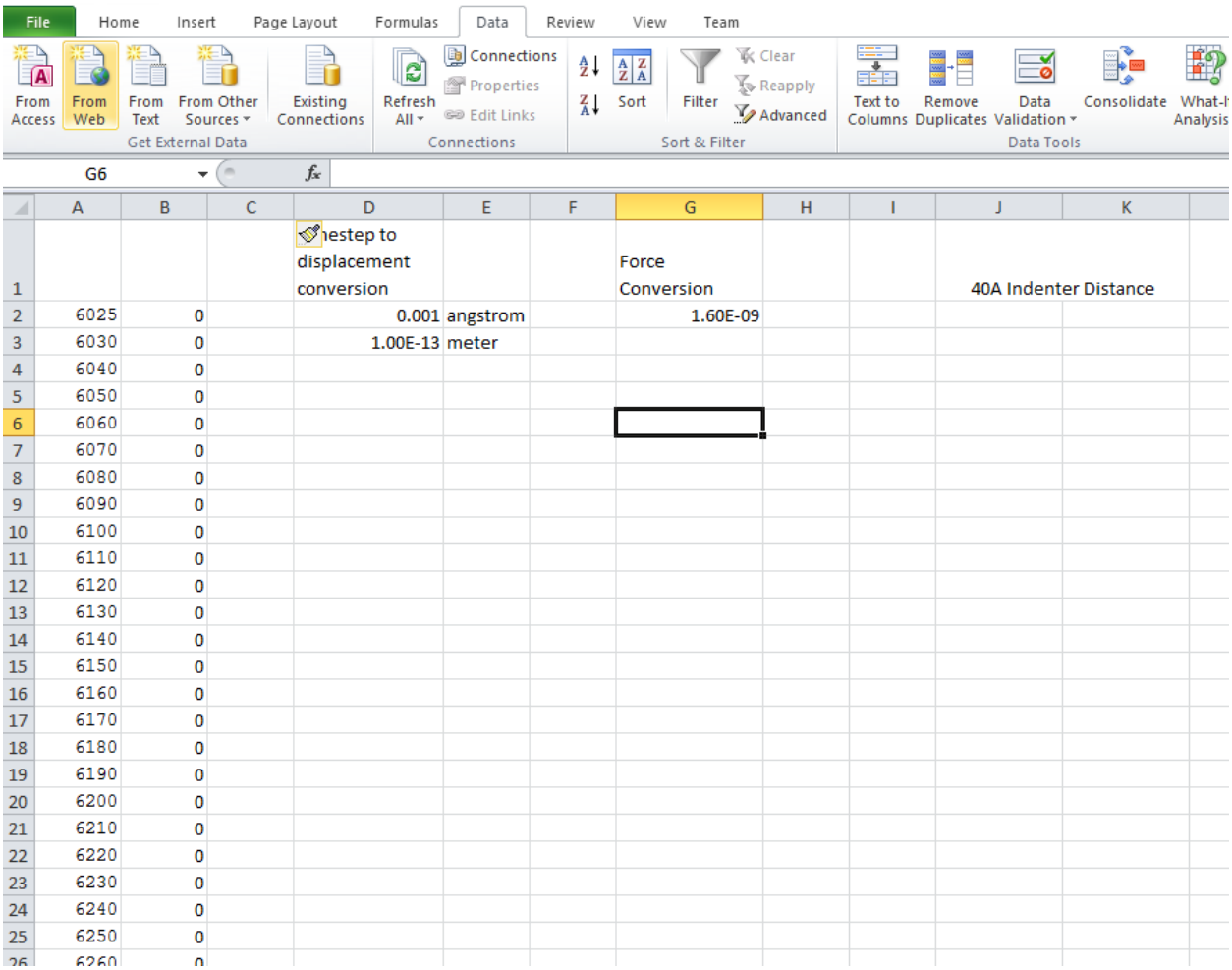

Book1 - M
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Team
From Access
From Web
From Text
From Other Sources
Existing Connections
Get External Data
Refresh All
Connections
Properties
Edit Links
Connections
Sort
Filter
Clear
Reappl
Advanced
Sort & Filter
Text to Columns
Remove Duplicates
Data Validation
Consolidate
Data Tools
A2
6025 10.05385 0
Timestep to displacement
Force


Convert Text to Columns Wizard - Step 1 of 3
The Text Wizard has determined that your data is Fixed Width.
If this is correct, choose Next, or choose the data type that best describes your data.
Original data type
Choose the file type that best describes your data:
Delimited - Characters such as commas or tabs separate each field.
Fixed width - Fields are aligned in columns with spaces between each field.
Preview of selected data:
2 6025 10.05385 0
3 6030 10.052555 0
4 6040 10.053425 0
5 6050 10.052523 0
6 6060 10.049237 0
Cancel
< Back
Next >
Finish


Convert Text to Columns Wizard - Step 2 of 3
This screen lets you set field widths (column breaks).
Lines with arrows signify a column break.
To CREATE a break line, click at the desired position.
To DELETE a break line, double click on the line.
To MOVE a break line, click and drag it.
Data preview
10
30
40
50
60
6025 10.05385 0
6030 10.052555 0
6040 10.053425 0
6050 10.052523 0
6060 10.049237 0
Cancel
< Back
Next >
Finish


Convert Text to Columns Wizard - Step 3 of 3
This screen lets you select each column and set the Data Format.
Column data format
General
Text
Date: DMY
Do not import column (skip)
'General' converts numeric values to numbers, date values to dates, and all remaining values to text.
Advanced...
Destination: A2
Data preview
General General General
6025 10.05385 0
6030 10.052555 0
6040 10.053425 0
6050 10.052523 0
6060 10.049237 0
Cancel
< Back
Next >
Finish

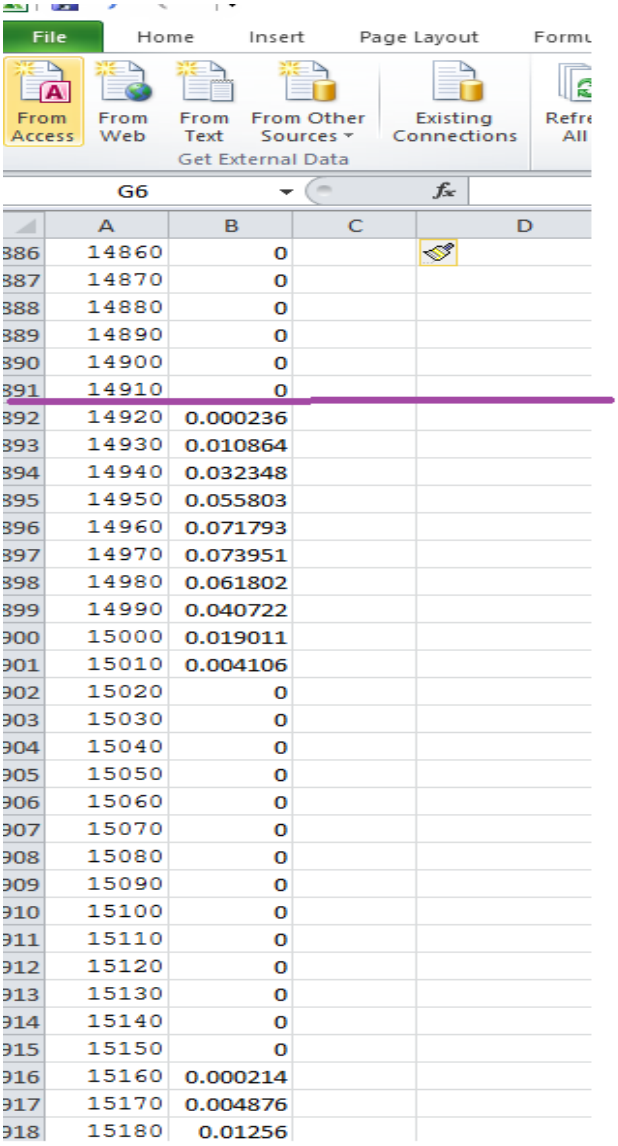
The data should now be separated into three separate columns. We do not need the middle column (lattice temperature) so that can be deleted.



	A	B	C	D	E	F	G	H	I	J	K
1				nestep to displacement conversion			Force Conversion			40A Indenter Distance	
2	6025	0		0.001	angstrom		1.60E-09				
3	6030	0		1.00E-13	meter						
4	6040	0									
5	6050	0									
6	6060	0									
7	6070	0									
8	6080	0									
9	6090	0									
10	6100	0									
11	6110	0									
12	6120	0									
13	6130	0									
14	6140	0									
15	6150	0									
16	6160	0									
17	6170	0									
18	6180	0									
19	6190	0									
20	6200	0									
21	6210	0									
22	6220	0									
23	6230	0									
24	6240	0									
25	6250	0									
26	6260	0									

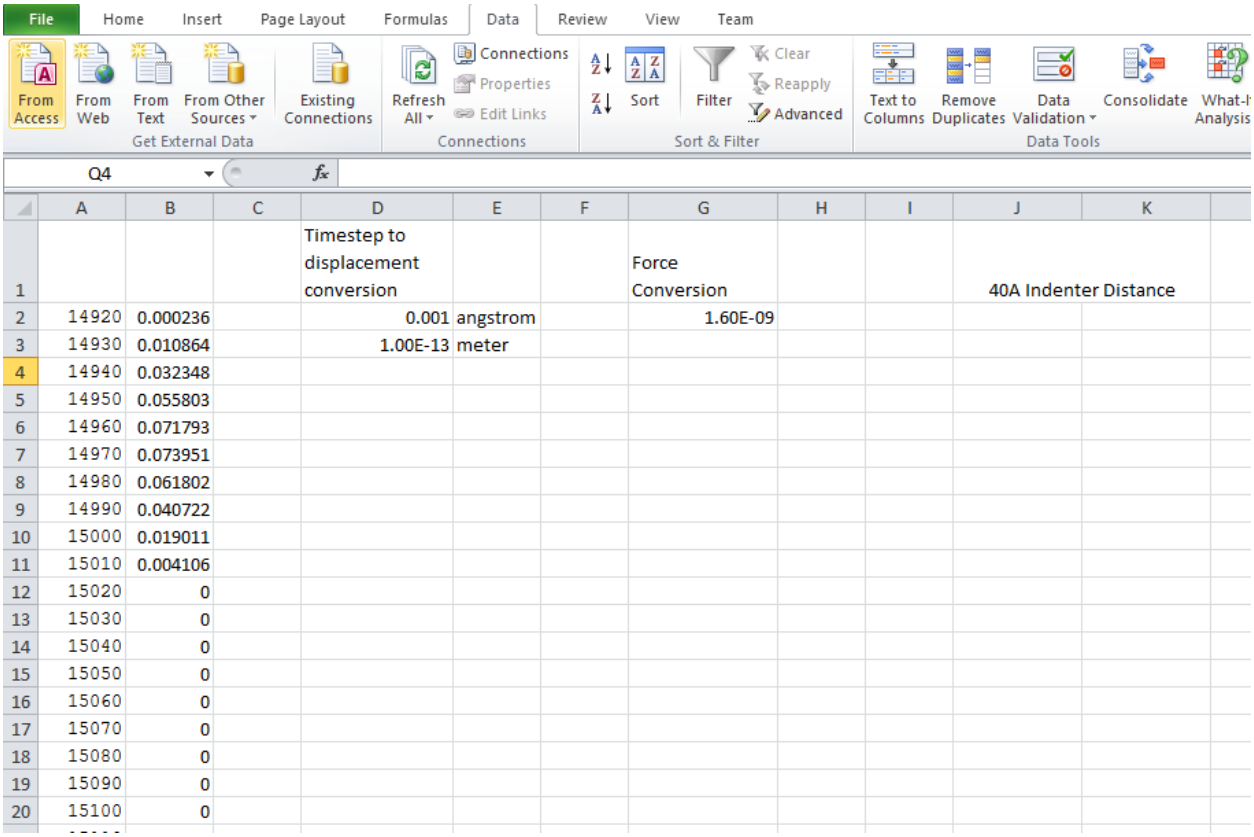
Now we need to get rid of useless data (or the data from before the indenter comes into contact with the grain). So scroll down the rows until the force column starts to show values other than zero.

Note: The force column starts off with non zero numbers in it. That means the indenter was generated inside of the grain. The indenter will need to be moved up (in the indenter code) and the simulation re-run.



	A	B	C	D
886	14860	0		
887	14870	0		
888	14880	0		
889	14890	0		
890	14900	0		
891	14910	0		
892	14920	0.000236		
893	14930	0.010864		
894	14940	0.032348		
895	14950	0.055803		
896	14960	0.071793		
897	14970	0.073951		
898	14980	0.061802		
899	14990	0.040722		
900	15000	0.019011		
901	15010	0.004106		
902	15020	0		
903	15030	0		
904	15040	0		
905	15050	0		
906	15060	0		
907	15070	0		
908	15080	0		
909	15090	0		
910	15100	0		
911	15110	0		
912	15120	0		
913	15130	0		
914	15140	0		
915	15150	0		
916	15160	0.000214		
917	15170	0.004876		
918	15180	0.01256		

Delete the values above this line. This is our new zero, or indenter contact point.



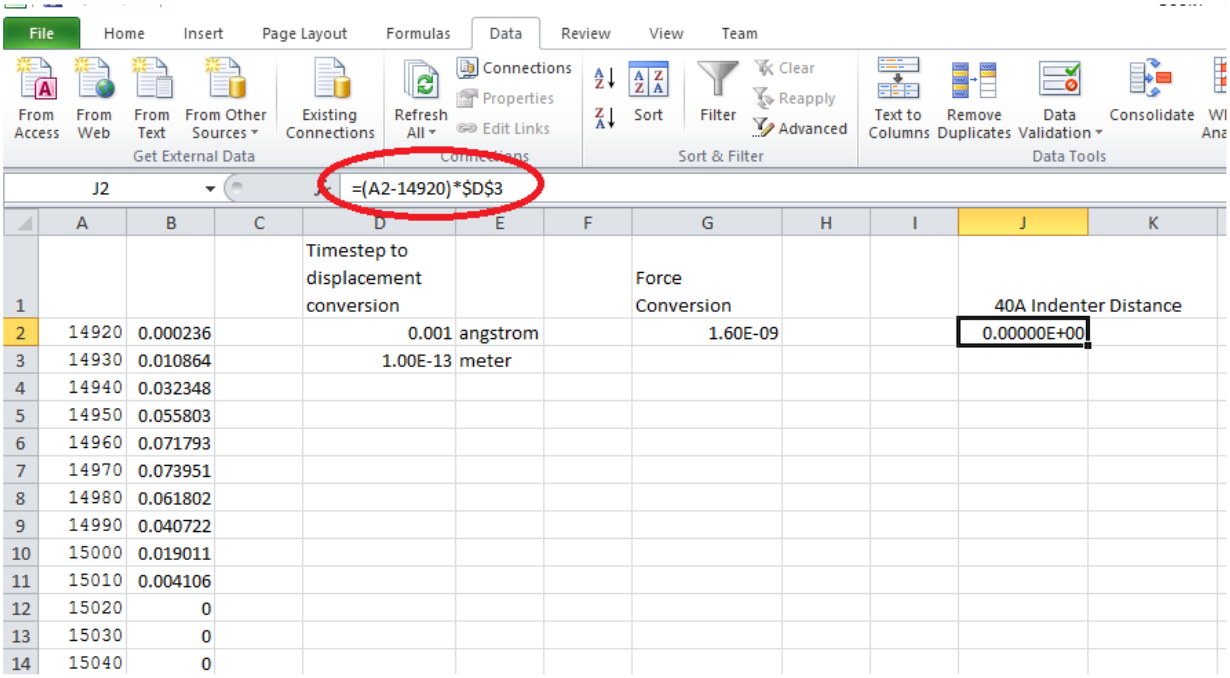
	A	B	C	D	E	F	G	H	I	J	K
1				Timestep to displacement conversion			Force Conversion			40A Indenter Distance	
2	14920	0.000236		0.001	angstrom		1.60E-09				
3	14930	0.010864		1.00E-13	meter						
4	14940	0.032348									
5	14950	0.055803									
6	14960	0.071793									
7	14970	0.073951									
8	14980	0.061802									
9	14990	0.040722									
10	15000	0.019011									
11	15010	0.004106									
12	15020	0									
13	15030	0									
14	15040	0									
15	15050	0									
16	15060	0									
17	15070	0									
18	15080	0									
19	15090	0									
20	15100	0									

The first column now tells us the time step the simulation is on, and the second column tells us the force on the indenter in the z direction (the units is ev/angstrom). We need to convert this into a depth into the material (or displacement). This can be done by subtracting the starting timestep from all the values in column A and multiplying them by a conversion factor to get the value in meters. We can find the conversion from the indenter code:

```
# ---------- indentation -----------------------------------------

variable z equal "162.5-0.1*step*dt"
fix             5 all indent 100.0 sphere 50.7 40.0 v_z 60 units box

# ---------- Run with indenter -----------------------------------

timestep        0.001
thermo          10
thermo_modify   temp new
thermo_style    custom step temp f_5[3]

run 300000
```

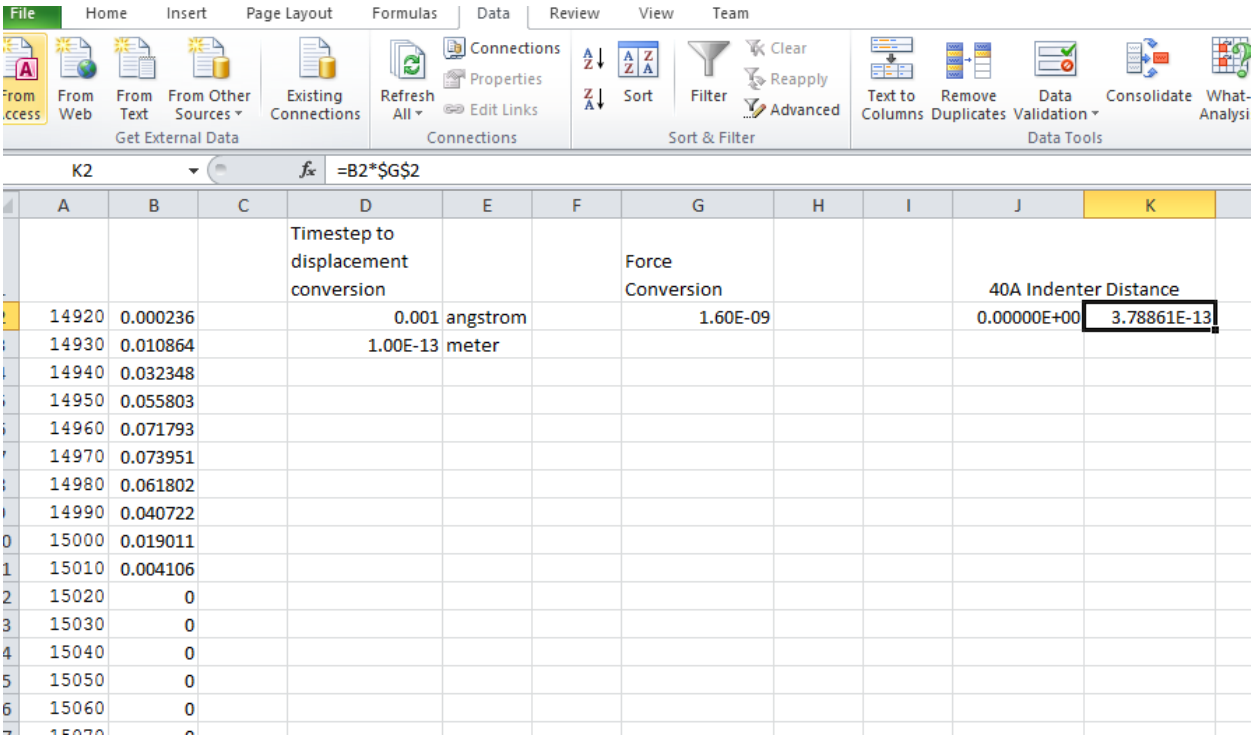
Our conversion is derived from 0.1*step*dt
step = 10 (thermo 10)
dt = 0.001 (timestep 0.001)

This gives us a conversion of 0.001. Since we are using metal units, that is 0.001 angstrom; which is 1E-13m. Our excel displacement conversion will look like this:



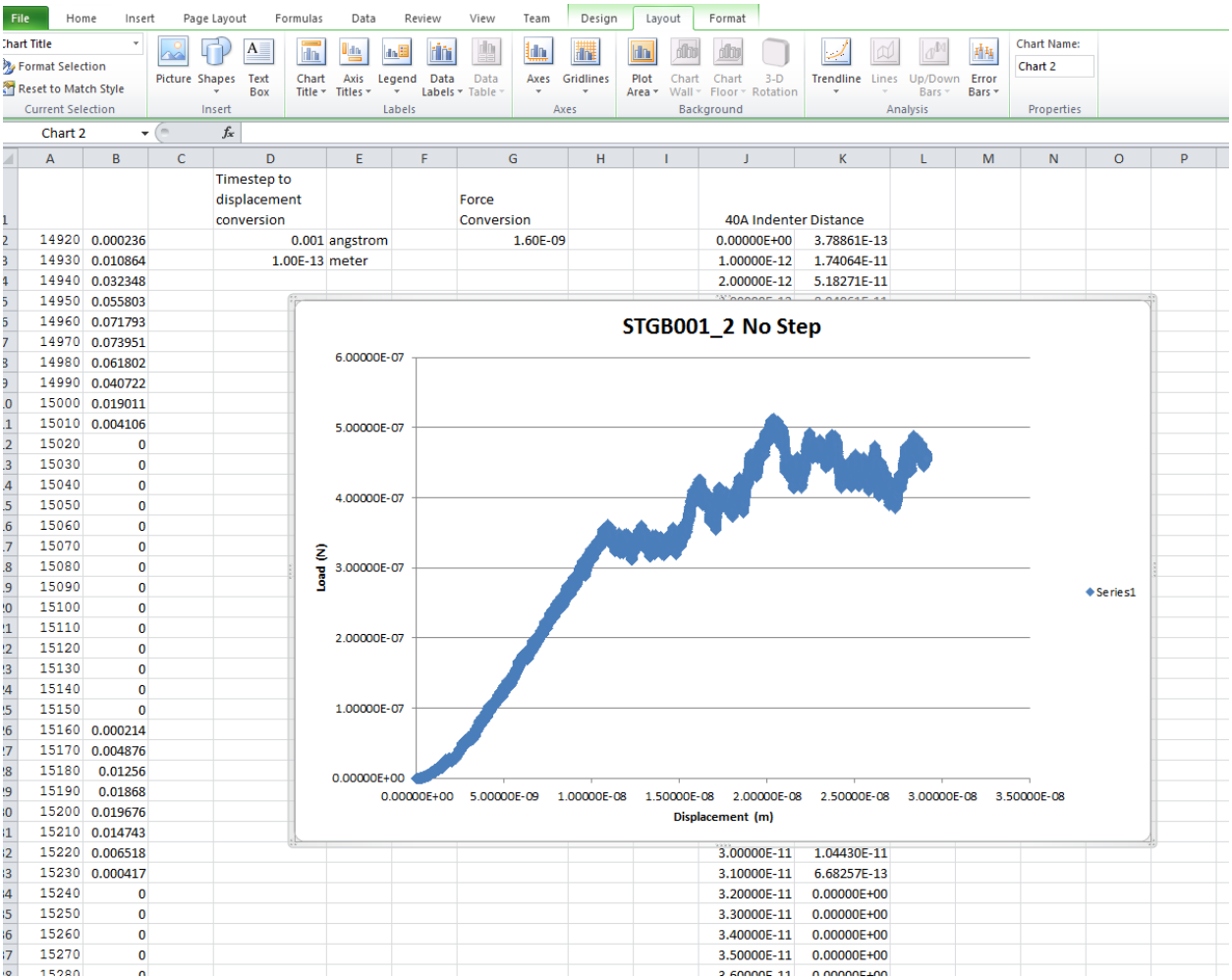
	A	B	C	D	E	F	G	H	I	J	K
1				Timestep to displacement conversion			Force Conversion			40A Indenter Distance	
2	14920	0.000236		0.001	angstrom		1.60E-09			0.00000E+00	
3	14930	0.010864		1.00E-13	meter						
4	14940	0.032348									
5	14950	0.055803									
6	14960	0.071793									
7	14970	0.073951									
8	14980	0.061802									
9	14990	0.040722									
10	15000	0.019011									
11	15010	0.004106									
12	15020	0									
13	15030	0									
14	15040	0									

Converting the force from ev/angstrom to N is a straight forward multiplication of 1.60217662E-9:



	A	B	C	D	E	F	G	H	I	J	K
				Timestep to displacement conversion			Force Conversion			40A Indenter Distance	
	14920	0.000236		0.001	angstrom		1.60E-09			0.00000E+00	3.78861E-13
	14930	0.010864		1.00E-13	meter						
	14940	0.032348									
	14950	0.055803									
	14960	0.071793									
	14970	0.073951									
	14980	0.061802									
	14990	0.040722									
0	15000	0.019011									
1	15010	0.004106									
2	15020	0									
3	15030	0									
4	15040	0									
5	15050	0									
6	15060	0									

Then expand the formulas for the entire set of data and plot:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1				Timestep to displacement conversion			Force Conversion			40A Indenter Distance						
2	14920	0.000236		0.001	angstrom		1.60E-09			0.00000E+00	3.78861E-13					
3	14930	0.010864		1.00E-13	meter					1.00000E-12	1.74064E-11					
4	14940	0.032348								2.00000E-12	5.18271E-11					
5	14950	0.055803														
6	14960	0.071793														
7	14970	0.073951														
8	14980	0.061802														
9	14990	0.040722														
10	15000	0.019011														
11	15010	0.004106														
12	15020	0														
13	15030	0														
14	15040	0														
15	15050	0														
16	15060	0														
17	15070	0														
18	15080	0														
19	15090	0														
20	15100	0														
21	15110	0														
22	15120	0														
23	15130	0														
24	15140	0														
25	15150	0														
26	15160	0.000214														
27	15170	0.004876														
28	15180	0.01256														
29	15190	0.01868														
30	15200	0.019676														
31	15210	0.014743														
32	15220	0.006518								3.00000E-11	1.04430E-11					
33	15230	0.000417								3.10000E-11	6.68257E-13					
34	15240	0								3.20000E-11	0.00000E+00					
35	15250	0								3.30000E-11	0.00000E+00					
36	15260	0								3.40000E-11	0.00000E+00					
37	15270	0								3.50000E-11	0.00000E+00					

This concludes the tutorial on conducting nano indentation simulations using LAMMPS.