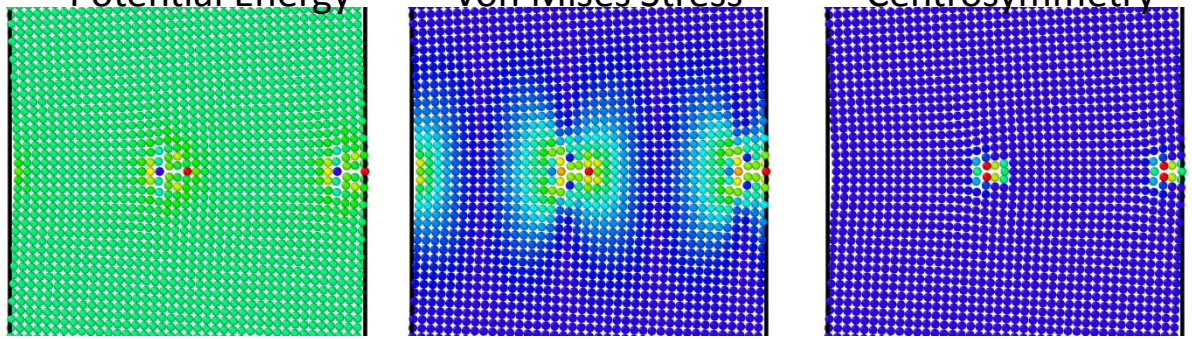
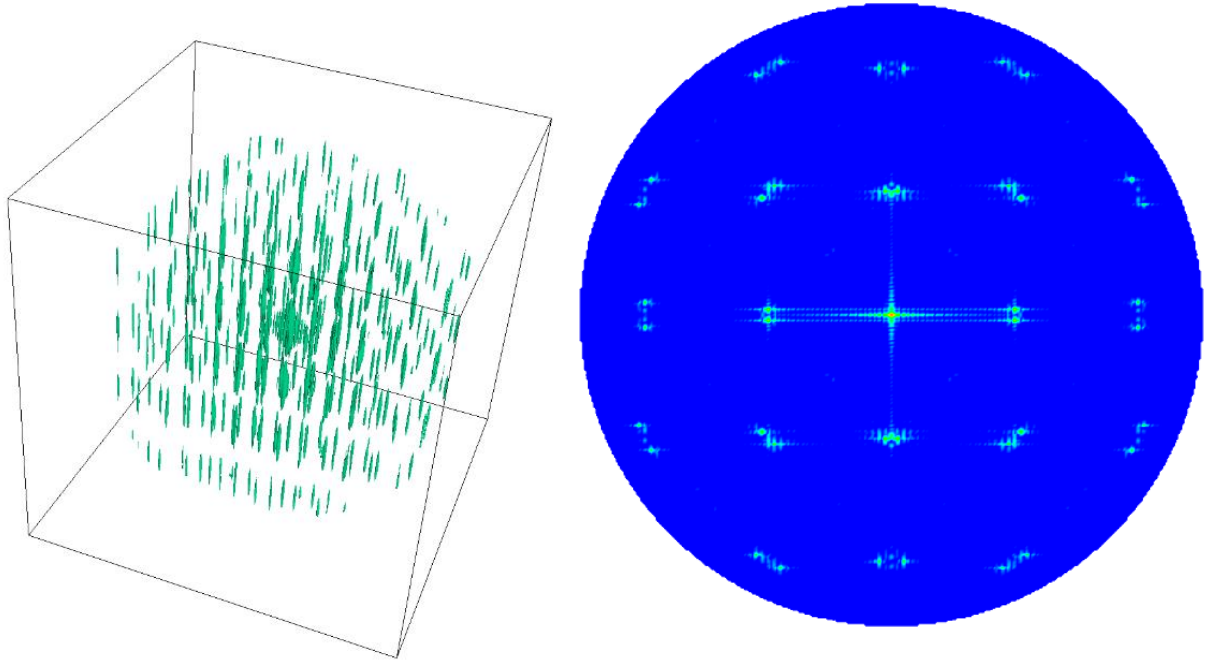
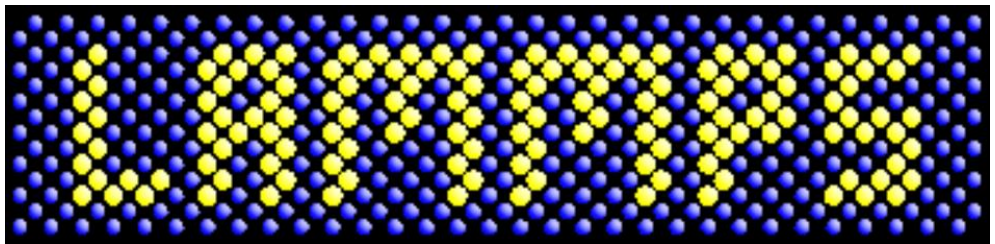






Tutorial- Virtual Diffraction

Installation

Commands / Syntax

Example Problem

- Input Script
- Helper Script
- Visualization

POC: Shawn Coleman
shawn.p.coleman8.ctr@mail.mil
(410) 306-0697

Installation:

A) Copy these files into the lammps /src directory

Computes*

- compute_saed.cpp
- compute_saed.h
- compute_saed_consts.h
- compute_xrd.cpp
- compute_xrd.h
- compute_xrd_consts.h

Fixes

- fix_ave_histo_weights.cpp
- fix_ave_histo_weights.h
- fix_saed_vtk.cpp
- fix_saed_vtk.h

There are special computes for MIC enabled architectures

B) Compile lammps with OpenMP enabled

Example Makefiles for Spirit and Excalibur located in tutorial documents

compute ID group-ID xrd lambda type1 type2 ... typeN keyword value ...

ID / group-ID are documented in compute command

xrd	Style name of this compute command
lambda	Wavelength of incident radiation (length units)
type1 type2 ... typeN	Chemical symbol of each atom type

Keywords

zero or more keyword/value pairs may be appended keyword = **2Theta** , **c** , **LP** , **manual** , or , **echo**

2Theta value	*Min2Theta Max2Theta* (Minimum and maximum 2 theta range to explore in either degrees or radians)
c values	c_1 c_2 c_3 (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
LP value	*1 or 0* (Switch to apply Lorentz-polarization factor)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Compute XRD

Examples:

compute 1 all xrd 1.541838 B O 2Theta 0.087 0.87 c 1 1 1 LP 1 echo

compute 2 all xrd 1.541838 C O 2Theta 10.0 100.0 c 0.05 0.05 0.05 LP 1 manual

Compute XRD

2Theta value	*Min2Theta Max2Theta* (Minimum and maximum 2 theta range to explore in either degrees or radians)
c values	c_1 c_2 c_3 (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
LP value	*1 or 0* (Switch to apply Lorentz-polarization factor)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Examples:

compute 1 all xrd 1.541838 B O 2Theta 0.087 0.87 c 1 1 1 LP 1 echo

compute 2 all xrd 1.541838 C O 2Theta 10.0 100.0 c 0.05 0.05 0.05 LP 1 manual

2Theta values can be inputted as radians or degrees. The output value will match the style of the input.

Compute XRD

2Theta value	*Min2Theta Max2Theta* (Minimum and maximum 2 theta range to explore in either degrees or radians)
c values	c_1 c_2 c_3 (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
LP value	*1 or 0* (Switch to apply Lorentz-polarization factor)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Examples:

compute 1 all xrd 1.541838 B O 2Theta 0.087 0.87 c 1 1 1 LP 1 echo

compute 2 all xrd 1.541838 C O 2Theta 10.0 100.0 c 0.05 0.05 0.05 LP 1 manual

2Theta values can be inputted as radians or degrees. The output value will match the style of the input.

LP 1 will turn on Lorentz-polarization factor and apply a scaling to the diffraction intensity.

Compute XRD

2Theta value	*Min2Theta Max2Theta* (Minimum and maximum 2 theta range to explore in either degrees or radians)
c values	c_1 c_2 c_3 (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
LP value	*1 or 0* (Switch to apply Lorentz-polarization factor)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

compute ID group-ID saed lambda type1 type2 ... typeN keyword value ...

ID / group-ID are documented in compute command

saed	Style name of this compute command
lambda	Wavelength of incident radiation (length units)
type1 type2 ... typeN	Chemical symbol of each atom type

Keywords
zero or more keyword/value pairs may be appended keyword = **Kmax** , **Zone** , **dR_Ewald** , **c** , **manual** , **echo**

Kmax value	Maximum distance explored from reciprocal space origin (inverse length units)
Zone values	z_1 z_2 z_3 (Zone axis of incident radiation) If z_1=z_2=z_3=0 all reciprocal space will be meshed up to **Kmax**
dR_Ewald value	Thickness of Ewald sphere slice intercepting reciprocal space (inverse length units)
c values	*c1 c2 c3* (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Compute SAED

Examples:

compute 1 all saed 0.0251 C H N O Kmax 1.70 Zone 0 0 1 dR_Ewald 0.01 c 0.5 0.5 0.5

compute 2 all saed 0.0251 Al O Kmax 1.70 Zone 0 0 0 c 0.05 0.05 0.05 manual echo

Compute SAED

Kmax value	Maximum distance explored from reciprocal space origin (inverse length units)
Zone values	z_1 z_2 z_3 (Zone axis of incident radiation) If z_1=z_2=z_3=0 all reciprocal space will be meshed up to **Kmax**
dR_Ewald value	Thickness of Ewald sphere slice intercepting reciprocal space (inverse length units)
c values	*c1 c2 c3* (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Examples:

compute 1 all saed 0.0251 C H N O Kmax 1.70 Zone 0 0 1 dR_Ewald 0.01 c 0.5 0.5 0.5

compute 2 all saed 0.0251 Al O Kmax 1.70 Zone 0 0 0 c 0.05 0.05 0.05 manual echo

Focusing the calculation on a particular slice of reciprocal space will reduce the computational costs.

Compute SAED

Kmax value	Maximum distance explored from reciprocal space origin (inverse length units)
Zone values	z_1 z_2 z_3 (Zone axis of incident radiation) If z_1=z_2=z_3=0 all reciprocal space will be meshed up to **Kmax**
dR_Ewald value	Thickness of Ewald sphere slice intercepting reciprocal space (inverse length units)
c values	*c1 c2 c3* (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

Examples:

compute 1 all saed 0.0251 C H N O Kmax 1.70 Zone 0 0 1 dR_Ewald 0.01 c 0.5 0.5 0.5

compute 2 all saed 0.0251 Al O Kmax 1.70 Zone 0 0 0 c 0.05 0.05 0.05 manual echo

Focusing the calculation on a particular slice of reciprocal space will reduce the computational costs.

When using the manual flag, the **c** values define the spacing used in reciprocal space. This allows precise control of the mesh.

Compute SAED

Kmax value	Maximum distance explored from reciprocal space origin (inverse length units)
Zone values	z_1 z_2 z_3 (Zone axis of incident radiation) If z_1=z_2=z_3=0 all reciprocal space will be meshed up to **Kmax**
dR_Ewald value	Thickness of Ewald sphere slice intercepting reciprocal space (inverse length units)
c values	*c1 c2 c3* (Parameters to adjust the spacing of the reciprocal lattice nodes in the h, k, and l directions respectively)
manual	flag to use manual spacing of reciprocal lattice points based on the values of the **c** parameters
echo	flag to provide extra output for debugging purposes

fix ID group-ID ave/histo/weights Nevery Nrepeat Nfreq lo hi Nbin value1 value2 ... keyword args ..

ID / group-ID are documented in compute command

ave/histo/weights	Style name of this fix command
Nevery	Use input values every this many timesteps
Nrepeat	# of times to use input values for calculating averages
Nfreq	calculate averages every this many timesteps
lo,hi	low/high bounds within which to histogram
Nbin	# of histogram bins
Value1	parameter over which the histogram is created

Keywords

Weights *value2*	parameter (same size as Value1) to weight histogram

Fix ave/histo/weights

Note, this has the same functionality as fix ave/histo but has the special keyword weights to compute a weighted histogram.

fix ID group-ID saed/vtk Nevery Nrepeat Nfreq c_ID ... keyword args ...

ID / group-ID are documented in compute command

saed/vtk	Style name of this fix command
Nevery	Use input values every this many timesteps
Nrepeat	# of times to use input values for calculating averages
Nfreq	calculate averages every this many timesteps
c_ID	saed compute ID

Keywords

zero or more keyword/value pairs may be appended keyword = **file**, **ave**, **start** , **file**, or **overwrite**

Note, this has the same functionality as fix ave/time but is specially modified for compute saed to output into the 3rd generation vtk image data format for use in parallelized visualization software (i.e., Paraview or Visit).

Fix saed/vtk

Memory usage:

These can be a very memory intensive computations. Here are some helpful hints I've found:

1) Run diffraction computations with MPI/OpenMP.
   - I typically output data/dump files from LAMMPS and run the diffraction computations separate from minimizations and dynamics (MPI based).

2) Be mindful of the number of reciprocal lattice nodes being generated.
   - Several parameters will help control the number of reciprocal lattice nodes. I typically start with a courser grid and fine tune as needed.

   - Using the "echo" flag will report the number of reciprocal lattice nodes and progress of the calculation to the std_out.

```
compute 2 all saed 0.0251 Al O Kmax 1.70 Zone 0 0 0 c 0.05 0.05 0.05 manual echo
```

Example Problem: Al Grain Boundary (dump file)

Input Files:

- Al.stgb001_1.dump - dump file with type x y z (and more) specified
- DiffFromDump.in - general input script to create both XRD and SAED data
- Loop_Dump_Diffraction_(Excalibur/Spirit) - helper scripts that creates PBS script that will run separate LAMMPS diffraction jobs for all *dump files in the directory

Al.stgb001_1.dump

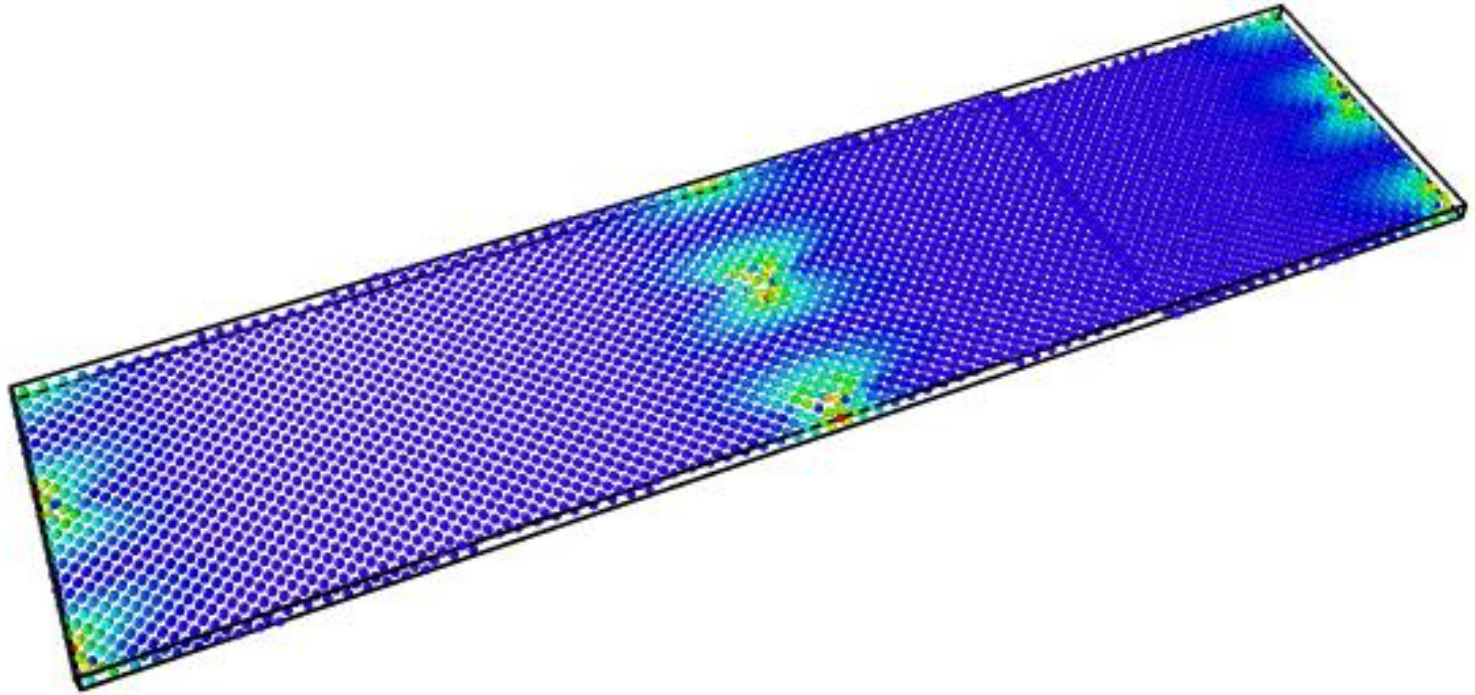
```
ITEM: TIMESTEP
1234
ITEM: NUMBER OF ATOMS
6408
ITEM: BOX BOUNDS pp pp pp
-0.0163336 81.1175
-162.221 162.221
0.000734495 4.04927
ITEM: ATOMS id type x y z c_PEatom c_Centro c_Voronoi[1] c_Voronoi[2] c_Voronoi[3] c_StrAtom[1] c_StrAtom[2] c_StrAtom[3] c_StrAtom[4] c_StrAtom[5] c_StrAtom[6] v_VonMises
664 1 2.26004 -160.1 0.000752652 -3.34756 0.0102832 17.0578 14 35.4252 111487 569825 211080 -38970.5 -0.401599 -0.86682 1.14865e+06
667 1 0.273626 -160.062 2.02502 -3.32373 0.0473431 17.1706 14 35.5702 65578.9 741402 197822 -82040.7 -1.07232 -0.766046 1.47112e+06
627 1 4.25886 -160.131 2.02502 -3.35472 0.00657964 16.9697 14 35.3018 127323 455503 181849 -2696.99 -0.147733 -0.912994 929016
624 1 6.26988 -160.15 0.000752568 -3.3569 0.0044887 16.8861 14 35.1846 113773 346869 150968 15448 -0.0583065 -0.931529 717780
587 1 8.28655 -160.163 2.02502 -3.35845 0.00337383 16.8129 14 35.0825 91048.7 256032 115073 27140.1 -0.032342 -0.937643 533933
547 1 12.3266 -160.181 2.02502 -3.35959 0.00238687 16.6855 14 34.9043 26450.3 108202 42834.5 37701.9 -0.0312845 -0.946344 219860
```

Example Problem: Al Grain Boundary (dump file)

Input Files:

- Al.stgb001_1.dump - dump file with type x y z (and more) specified
- DiffFromDump.in - general input script to create both XRD and SAED data
- Loop_Dump_Diffraction_(Excalibur/Spirit) - helper scripts that creates PBS script that will run separate LAMMPS diffraction jobs for all *dump files in the directory

```
ITEM: TIMESTEP
1234
ITEM: NUMBER OF ATOMS
6408
ITEM: BOX BOUNDS pp pp pp
-0.0163336 81.1175
-162.221 162.221
0.000734495 4.04927
ITEM: ATOMS id type x y z c_PEatom c_Centro c_Voronoi[1] c
664 1 2.26004 -160.1 0.000752652 -3.34756 0.0102832 17.0578 14 35
667 1 0.273626 -160.062 2.02502 -3.32373 0.0473431 17.1706 14 35.
627 1 4.25886 -160.131 2.02502 -3.35472 0.00657964 16.9697 14 35.
624 1 6.26988 -160.15 0.000752568 -3.3569 0.0044887 16.8861 14 35
587 1 8.28655 -160.163 2.02502 -3.35845 0.00337383 16.8129 14 35.
547 1 12.3266 -160.181 2.02502 -3.35959 0.00238687 16.6855 14 34.
```



DiffFromDump.in

```
variable  A string "PREFIX"
log             $A.log

processors * * *

#########################################################
###################### Variables ########################
#########################################################

variable        Restart       string "DUMPFILE"
variable        Species       string "Al "
variable        NTypes        equal   1

variable        eLambda       equal 0.0251
variable        Kmax          equal 1.25
variable        Zone0         equal 0
variable        Zone1         equal 0
variable        Zone2         equal 0
variable        eRes0         equal 0.0075
variable        eRes1         equal 0.0075
variable        eRes2         equal 0.0075
variable        dR_Ewald      equal 0.0015

variable        xLambda       equal 1.541838
variable        ThetaMin      equal 30.0
variable        ThetaMax      equal 110.0
variable        xRes0         equal 0.0075
variable        xRes1         equal 0.0075
variable        xRes2         equal 0.0075
variable        LP            equal 1
variable        Nbins         equal 3000


#########################################################
```

```
lattice         none 1.0
region          chamber block 0 1 0 1 0 1
create_box      ${NTypes} chamber

read_dump       ${Restart} XXXXX x y z box yes add yes
mass             * 26.9820

atom_modify     sort 0 0.0
comm_style      tiled
balance         0.9 rcb

pair_style       none

#########################################################
###################### Computes #########################
#########################################################

compute          SAED all saed ${eLambda} ${Species} Kmax ${Kmax} &
                 Zone ${Zone0} ${Zone1} ${Zone2} c ${eRes0} ${eRes1} ${eRes2} &
                 dR_Ewald ${dR_Ewald} echo manual

compute          XRD all xrd ${xLambda} ${Species} 2Theta ${ThetaMin} &
                 ${ThetaMax} c ${xRes0} ${xRes1} ${xRes2} LP ${LP} echo manual

fix              1 all saed/vtk 1 1 1 c_SAED file $A_saed

fix              2 all ave/histo/weights 1 1 1  ${ThetaMin} ${ThetaMax} &
                 ${Nbins} c_XRD[1] weights c_XRD[2] mode vector file $A.xrd

run              0
unfix            1
unfix            2
uncompute        SAED
uncompute        XRD
```

DiffFromDump.in

```
variable  A string "PREFIX"
log             $A.log

processors * * *

#####################################
##################### Variables ####
#####################################################

variable        Restart         string "DUMPFILE"
variable        Species         string "Al "
variable        NTypes          equal   1

variable        eLambda         equal 0.0251
variable        Kmax            equal 1.25
variable        Zone0           equal 0
variable        Zone1           equal 0
variable        Zone2           equal 0
variable        eRes0           equal 0.0075
variable        eRes1           equal 0.0075
variable        eRes2           equal 0.0075
variable        dR_Ewald        equal 0.0015

variable        xLambda         equal 1.541838
variable        ThetaMin        equal 30.0
variable        ThetaMax        equal 110.0
variable        xRes0           equal 0.0075
variable        xRes1           equal 0.0075
variable        xRes2           equal 0.0075
variable        LP              equal 1
variable        Nbins           equal 3000

#####################################################
```

Species name for each atom type, up to NTypes

Keeping all diffraction compute options up front as variables.

```
lattice         none 1.0
region          chamber block 0 1 0 1 0 1
create_box      ${NTypes} chamber

read_dump       ${Restart} XXXXX x y z box yes add yes
mass             * 26.9820

atom_modify     sort 0 0.0
comm_style      tiled
balance         0.9 rcb

pair_style       none

#####################################################
##################### Computes ######################
#####################################################

compute          SAED all saed ${eLambda} ${Species} Kmax ${Kmax} &
                 Zone ${Zone0} ${Zone1} ${Zone2} c ${eRes0} ${eRes1} ${eRes2} &
                 dR_Ewald ${dR_Ewald} echo manual

compute          XRD all xrd ${xLambda} ${Species} 2Theta ${ThetaMin} &
                 ${ThetaMax} c ${xRes0} ${xRes1} ${xRes2} LP ${LP} echo manual

fix              1 all saed/vtk 1 1 1 c_SAED file $A_saed

fix              2 all ave/histo/weights 1 1 1  ${ThetaMin} ${ThetaMax} &
                 ${Nbins} c_XRD[1] weights c_XRD[2] mode vector file $A.xrd

run              0
unfix            1
unfix            2
uncompute        SAED
uncompute        XRD
```

DiffFromDump.in

```
variable  A string "PREFIX"
log             $A.log

processors * * *

########################################################
###################### Variables ######################
########################################################

variable        Restart       string "DUMPFILE"
variable        Species       string "Al "
variable        NTypes        equal   1

variable        eLambda       equal 0.0251
variable        Kmax          equal 1.25
variable        Zone0         equal 0
variable        Zone1         equal 0
variable        Zone2         equal 0
variable        eRes0         equal 0.0075
variable        eRes1         equal 0.0075
variable        eRes2         equal 0.0075
variable        dR_Ewald      equal 0.0015

variable        xLambda       equal 1.541838
variable        ThetaMin      equal 30.0
variable        ThetaMax      equal 110.0
variable        xRes0         equal 0.0075
variable        xRes1         equal 0.0075
variable        xRes2         equal 0.0075
variable        LP            equal 1
variable        Nbins         equal 3000
```

The helper script automatically fills in these values using each dump file in the directory. (or change these manually)

```
lattice         none 1.0
region          chamber block 0 1 0 1 0 1
create_box      ${NTypes} chamber

read_dump       ${Restart} XXXXX x y z box yes add yes
mass             * 26.9820

atom_modify     sort 0 0.0
comm_style      tiled
balance         0.9 rcb

pair_style       none

########################################################
###################### Computes ########################
########################################################

compute          SAED all saed ${eLambda} ${Species} Kmax ${Kmax} &
                 Zone ${Zone0} ${Zone1} ${Zone2} c ${eRes0} ${eRes1} ${eRes2} &
                 dR_Ewald ${dR_Ewald} echo manual

compute          XRD all xrd ${xLambda} ${Species} 2Theta ${ThetaMin} &
                 ${ThetaMax} c ${xRes0} ${xRes1} ${xRes2} LP ${LP} echo manual

fix              1 all saed/vtk 1 1 1 c_SAED file $A_saed

fix              2 all ave/histo/weights 1 1 1  ${ThetaMin} ${ThetaMax} &
                 ${Nbins} c_XRD[1] weights c_XRD[2] mode vector file $A.xrd

run              0
unfix            1
unfix            2
uncompute        SAED
uncompute        XRD
```

No effect on the computation of diffraction patterns!

Create/Submit Jobs with Helper Script

*Modify the following defaults values in Loop_Dump_Diffraction_(Excalibur/Spirit)

ACCOUNT=YOUR_ACCOUNT_HERE

EMAIL=YOUR_EMAIL_HERE

LAMMPS=YOUR_PATH_TO_LAMMPS_HERE

```
excalibur:> ls
Al.stgb001_1.dump  DiffFromDump.in  Loop_Dump_Diffraction_Excalibur

excalibur:> Loop_Dump_Diffraction_Excalibur -q d -n 1 -t 30 -D DiffFromDump.in Example
```

Create/Submit Jobs with Helper Script

*Modify the following defaults values in Loop_Dump_Diffraction_(Excalibur/Spirit)

ACCOUNT=YOUR_ACCOUNT_HERE

EMAIL=YOUR_EMAIL_HERE

LAMMPS=YOUR_PATH_TO_LAMMPS_HERE

```
excalibur:> ls
Al.stgb001_1.dump  DiffFromDump.in  Loop_Dump_Diffraction_Excalibur

excalibur:> Loop_Dump_Diffraction_Excalibur -q d -n 1 -t 30 -D DiffFromDump.in Example
```

Dump file and DiffFromDump.in must be located in the current directory

Create/Submit Jobs with Helper Script

*Modify the following defaults values in Loop_Dump_Diffraction_(Excalibur/Spirit)

ACCOUNT=YOUR_ACCOUNT_HERE

EMAIL=YOUR_EMAIL_HERE

LAMMPS=YOUR_PATH_TO_LAMMPS_HERE

```
excalibur:> ls
Al.stgb001_1.dump  DiffFromDump.in  Loop_Dump_Diffraction_Excalibur

excalibur:> Loop_Dump_Diffraction_Excalibur -q d -n 1 -t 30 -D DiffFromDump.in Example
```

Last input value = Run name

queue
d=debug
s=standard

# of nodes

time = dd:hh:mm

Basename for
diffraction calculation

15 min 30 seconds 32 processors on Excalibur

Outputs

- (100Kb) Al.stgb001_1.xrd
  - 2θ values in 2nd column
  - Intensity values in 3rd column
  - Plot these values

4 line header

```
# Histogrammed data for fix 2
# TimeStep Number-of-bins Total-counts Missing-counts Min-value Max-value
# Bin Coord Count Count/Total
1234 3000 4.08972e+10 0 30.0016 109.992
1 30.0133 9.63929e+06 0.000235696
2 30.04 6.17153e+06 0.000150904
3 30.0667 3.50073e+06 8.55983e-05
4 30.0933 1.47572e+06 3.60837e-05
5 30.12 1.24187e+07 0.000303656
…
```

- (246Mb) Al.stgb001_1_saed.0.vtk
  - This vtk format is for a regularly spaced data filling a cubic volume, but our data has spherical limits.
  - Ghost data (value=-1) fills in the volume outside the sphere to make this data structure cubic.

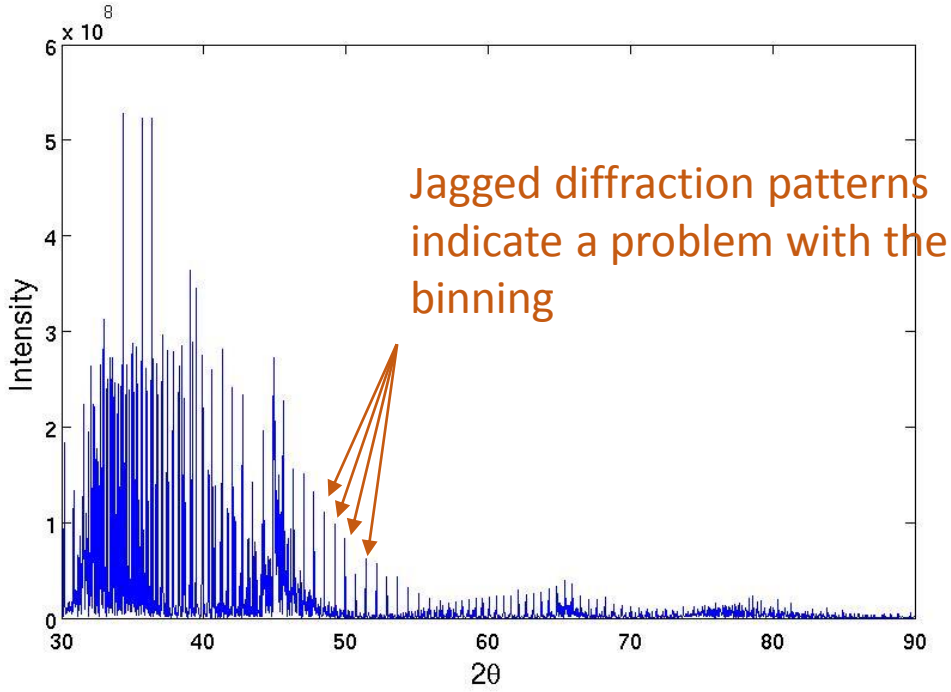
10 line header

```
# vtk DataFile Version 3.0 c_SAED
Image data set
ASCII
DATASET STRUCTURED_POINTS
DIMENSIONS 335 335 335
ASPECT_RATIO 0.0075 0.0075 0.0075
ORIGIN -1.2525 -1.2525 -1.2525
POINT_DATA 37595375
SCALARS intensity float
LOOKUP_TABLE default
-1
-1
…
```

Visualizations: xrd

XRD plots are sensitive to the spacing of the reciprocal lattice points, the bin size of the histogram, and relrods from the finite simulation volume.

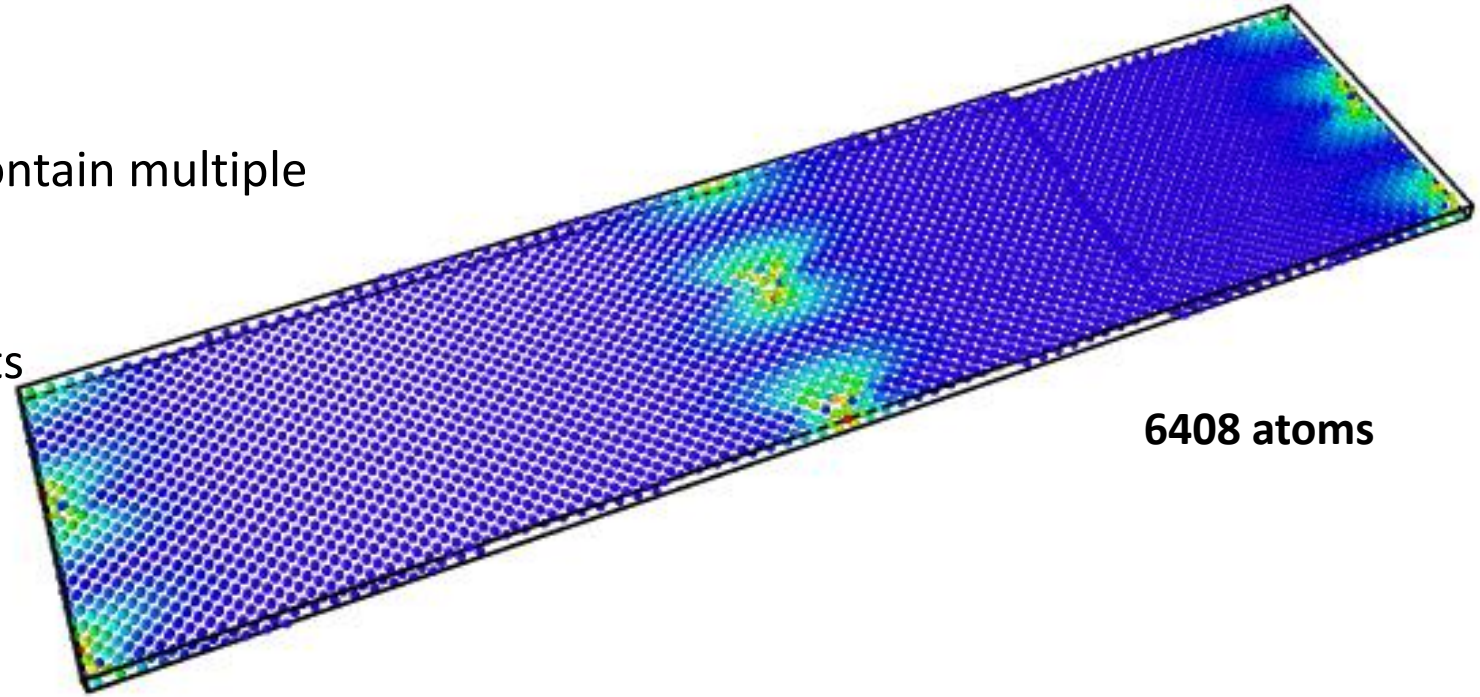
- The example showcases all these problems!



Hints:

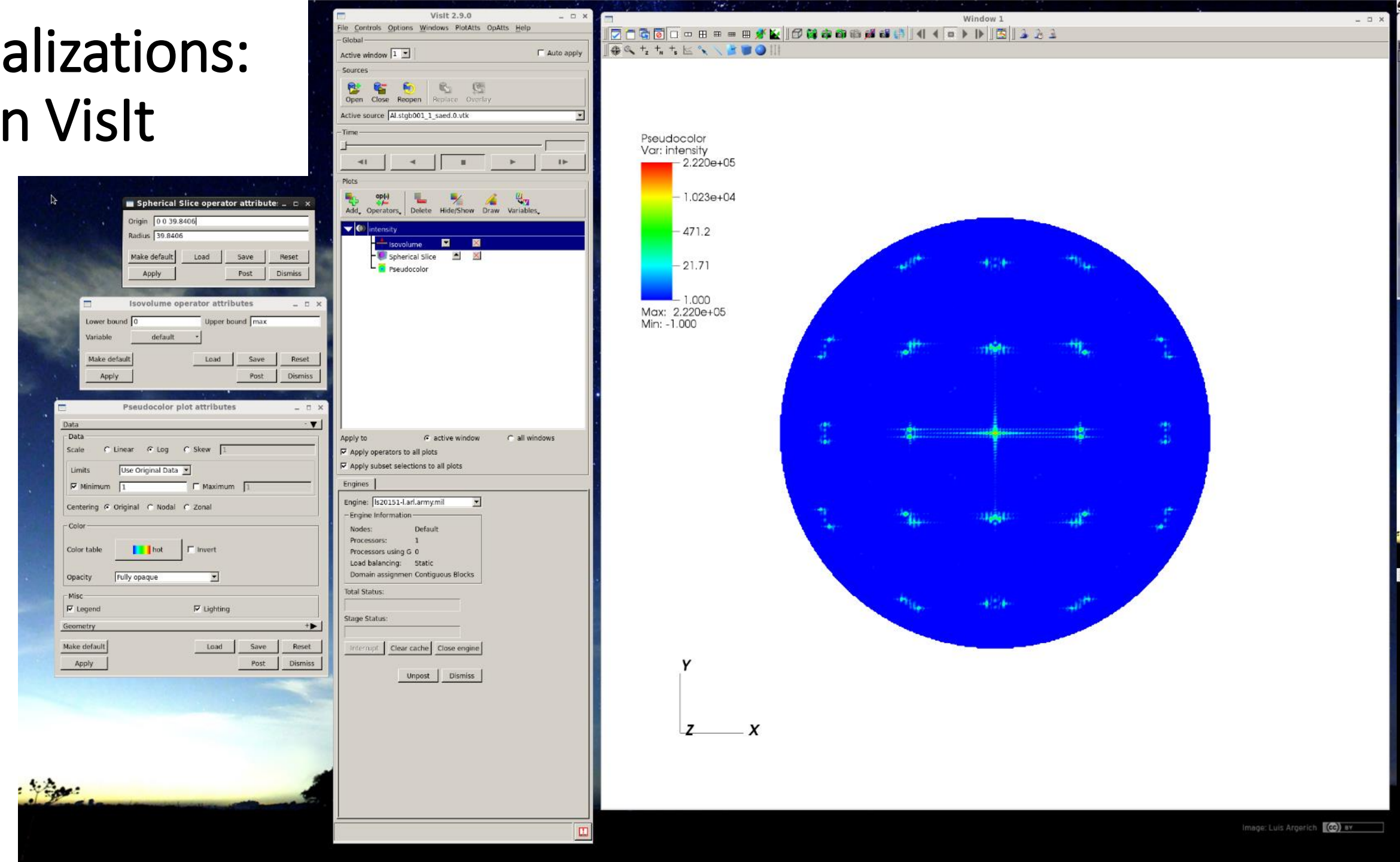
2θ binning should be large enough to contain multiple reciprocal lattice points.

Larger simulations will reduce the effects of relrods. Periodic boundaries are not accounted for in the diffraction calculations.



6408 atoms

Visualizations: vtk in VisIt



Visualizations: vtk in VisIt

a) Open the diffraction .vtk files in VisIt

b) Click add->pseudocolor->intensity
- Green name means its ready to be drawn

c) Click Draw
- You will see the full rectangular mesh grid which is not good. We need to remove ghost data

d) Select intensity plot;
Click Operators->Selection->Isovolume;
Double click isovolume (under intensity plot);
Change lower bound to 0
Click Apply
Click Dismiss
Click Draw

- Now you will see only where there was valid intensity data (if you don't use Zone it will be a sphere). But most intensity are low so I typically color the intensity by log scale.

e) Select intensity plot
Double click Pseudocolor;
Change Scale to Log
Check Mimimum Limit
Input non-zero minimum limit (i.e. 1)
Click Apply
Click Dismiss

- Clicking apply will change the scale automatically. But unless you have used Zone and a thin dR_Ewald you likely have lots of extra data visualized that does not intersect the Ewald sphere. Next we want to take a spherical slice to mimic the intersection of Ewald's sphere with reciprocal space.

f) Select intensity plot
Click Operators->Slicing>Spherical Slice
Double click Spherical Slice
Change the origin/radius to represent the center of your Ewald sphere
  - (i.e. 39.84063 0 0) and radius (i.e. 39.84063)

Click Apply
Click Dismiss
Click Draw

- This is the saed pattern, but we are not guaranteed to be aligned correctly with the zone axis.

g) On the menu bar... select Controls -> View
Change View Normal to the zone axis of choice (i.e. -1 0 0)
Adjust Up Vector if desired (i.e. 0 1 0)
Turn of Perspective view
Click Apply
Click Dismiss

- Note that for the primary x, y, and z zone axis you can use the drop down Align to axis. However, it is likely that there is a bunch of extra annotations on the image that don't look appealing.

h) On the menu bar... select Control-> Annotation

- Here you can change what is visualized along side the data and how you display axis information. I typically turn most all the display information off.

i) Save the image

Click File-> Set save options
Click Save

Visualizations: vtk in VisIt