HIGH PERFORMANCE RESEARCH COMPUTING

Introduction to LAMMPS

HPRC Short Course
1:30 pm
10/4/22

High Performance
Research Computing
DIVISION OF RESEARCH

Introduction to LAMMPS Outline

- FASTER Resources
- Modeling Overview
  - Molecular Dynamics
  - Atomic Force Fields
  - LAMMPS
  - Visualization
- Project 1 - Simulating a Lennard Jones Melt
- Project 2 - Melting temperature of Copper
- Project 3 - Building Copper Nanowire
- Project 4 - Uniaxial Tensile Test
- Project 5 - Calculation of Thermal Conductivity

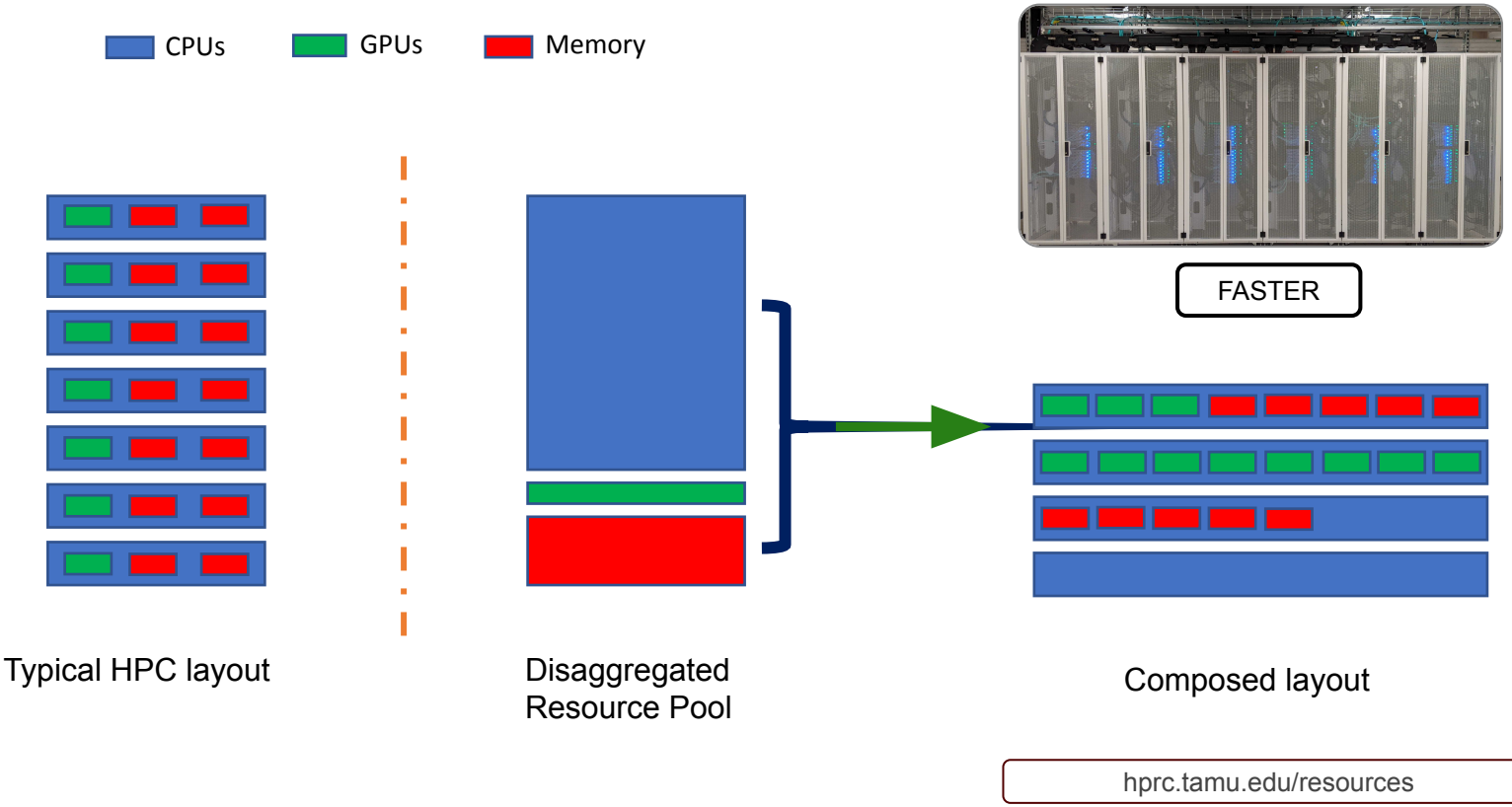
FASTER Cluster

hprc.tamu.edu/wiki/FASTER:Intro

Node Type	Quantity
64-core compute nodes (256GB RAM each)	180 (11,520 cores)
Composable GPUs	200 T4 16GB 40 A100 40GB 10 A10 24GB 4 A30 24GB 8 A40 48GB

FASTER (Fostering Accelerated Sciences Transformation Education and Research) is a 180-node Intel cluster.

Composability at the Hardware Level



hprc.tamu.edu/resources

Accessing FASTER via SSH

- SSH command is required for accessing FASTER:
  - On campus: `ssh userNetID@faster.hprc.tamu.edu`
  - Off campus:
    - Set up and start VPN (Virtual Private Network): u.tamu.edu/VPnetwork
    - Then: `ssh userNetID@faster.hprc.tamu.edu`
  - *Two-Factor Authentication* enabled for CAS, VPN, SSH
- SSH programs for Windows:
  - MobaXTerm (preferred, includes SSH and X11)
  - PuTTY SSH
  - Windows Subsystem for Linux
- https://portal-faster.hprc.tamu.edu/
  - Select the "Clusters" tab and then "_faster Shell Access"
- FASTER has 2 login nodes for TAMU users. Check the bash prompt.

  Login sessions that are idle for **60** minutes will be closed automatically

  Processes run longer than **60** minutes on login nodes will be killed automatically.

  Do not use more than 8 cores on the login nodes!

  Do not use the sudo command.

  hprc.tamu.edu/wiki/HPRC:Access

Accessing FASTER for ACCESS users

- The Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) is a virtual collaboration funded by the National Science Foundation that facilitates free, customized access to advanced digital resources, consulting, training, and mentorship.

- View the getting started documentation to create an ACCESS account.

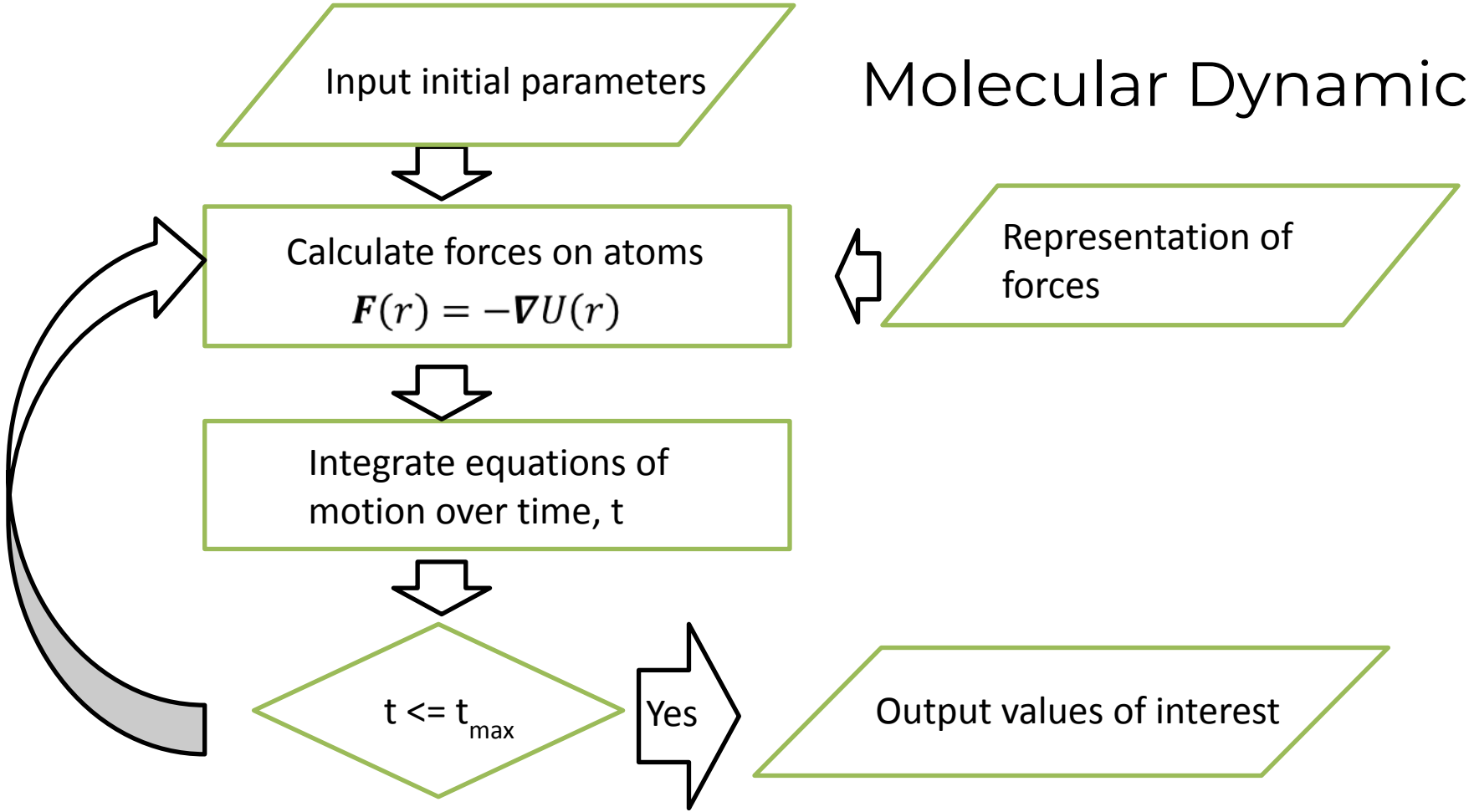
- FASTER has 1 login node for ACCESS users

- SSH via Jump Host:
  - `ssh -J fasterusername@faster-jump.hprc.tamu.edu:8822 fasterusername@login.faster.hprc.tamu.edu`

Hands On Activity

- Please login to FASTER now
- **`cp -r /scratch/training/Intro-LAMMPS/ $SCRATCH/Intro-LAMMPS`**
- **`cd $SCRATCH/Intro-LAMMPS`**

Molecular Dynamics



Atomic Force Fields

- Represents the forces between atoms.
- An accurate force field is needed for accurate results!
- Force fields are usually fitted to material properties around a certain set of conditions.
  - They may not be accurate in regions outside where they were fitted!
- Some types of force fields usable in LAMMPS:
  - LJ, EAM, Bond Order, CHARMM, AMBER, COMPASS, AIREBO, REAXFF

Embedded Atom Method Force Field
Copper: Cu_u3

$$E_i = F_\alpha\left(\sum_{j\neq i}\rho_\beta(r_{ij})\right) + \frac{1}{2}\sum_{j\neq i}\phi_{\alpha\beta}(r_{ij})$$

Embedding energy: Function of atomic electron density

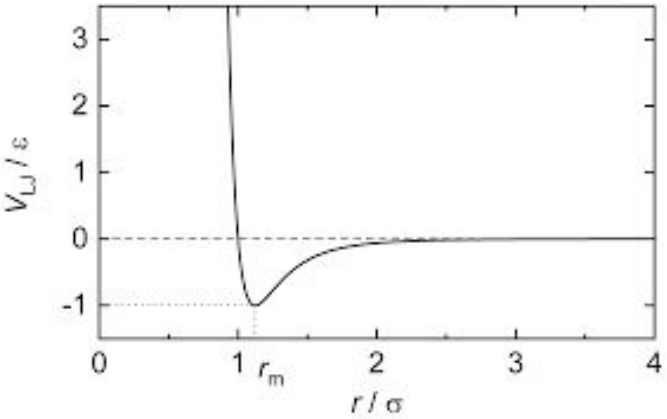
Pairwise interaction

- https://docs.lammps.org/pair_eam.html
- Foiles et al, Phys Rev B, 33, 7983 (1986)

Lennard Jones 12-6 Force Field

$$E = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad r < r_c$$

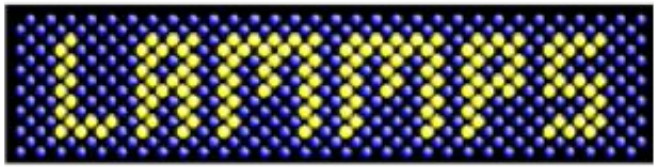
- ϵ (energy units)
- σ (distance units)
- LJ cutoff (distance units)



https://docs.lammps.org/pair_lj.html

Molecular Dynamics Software - LAMMPS

- Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)
- Open Source MD Software
- Optimized to run on large numbers of CPUs and GPUs.
- Great Documentation:
  - docs.lammps.org



https://www.lammps.org

Visualization

- Ovito
  - Free visualization software
  - Reads lammps files
  - https://www.ovito.org/

Running LAMMPS

- The files ending in *.job are slurm job submission files for a parallel cpu lammps job.
- LAMMPS Jobs are too intensive to be run on login nodes!

Running LAMMPS

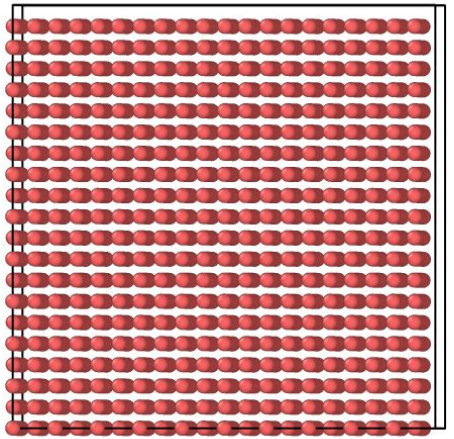
- Load the LAMMPS module
  - **`ml GCC/10.2.0 CUDA/11.1.1 OpenMPI/4.0.5`**
  - **`ml LAMMPS/29Sep2021-kokkos-sm80-noMP`**
- Run lammps using an <inputFile>
  - **`lmp -k on g 1 -sf kk -in `*`inputFile`***
  - The input file will contain all the information needed to setup and execute the simulation

5 Minute Break

Project #1 - LJ

- Learn the basics of a LAMMPS input file by running a short simulation of a Lennard Jones melt.
- Use 12-6 LJ forces between atoms.

$$E = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \qquad r < r_c$$



Project #1 - LJ

- `cd LJ`
- `sbatch lj_gpu.job`

Basics of an Input Script - in.lj - 1

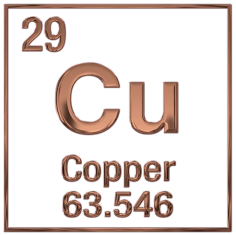
```
# 3d Lennard-Jones melt      # - comments out lines

units           lj           # units - Describes what units will be used
atom_style      atomic       # atom_style - Describes what information is reported for atoms

lattice         fcc 0.8442   # lattice - creates a lattice using a lattice parameter
region          box block 0 10 0 10 0 10    # region - Defines a region of our simulation
create_box      1 box        # create_box - Defines a simulation box made up of our region
create_atoms    1 box        # create_atoms - Will fill our simulation box with atoms

mass            1 1.0        # mass - inputs the mass of the atoms

velocity    all create 1.44 87287 loop geom    # velocity - assigns initial velocities to atoms
```

Basics of an Input Script - in.lj - 2

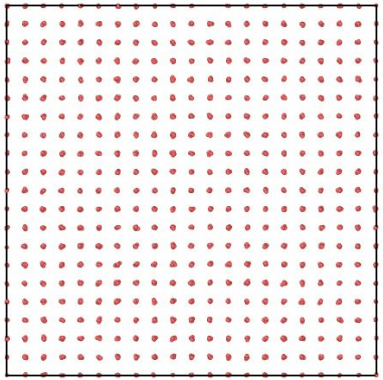
$$E = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \qquad r < r_c$$

```
pair_style  lj/cut 2.5          # pair_style - defines a pairwise force between atoms.
pair_coeff 1 1 1.0 1.0 2.5      # pair_coeff - provides coefficients to be used in the pairstyle

neighbor   0.3 bin
neigh_modify    delay 0 every 20 check no      # Change how neighbor lists are made

dump trj all custom 1000 traj.lj.dump id type x y z     # Create a trajectory file for visualization

fix             1 all nve       # Apply constraints to your simulation

run             10000           # Run the simulation
```

Project #2.a - Cu Cohesive Energy



- We will simulate a bulk metal (cu) using periodic boundary conditions and an embedded atom method (eam) force field.
- We will calculate the cohesive energy of copper
- This calculation has been setup in Cu_Melt



Project #2.a - Cu Cohesive Energy

- `cd ../Cu_Melt`
- `sbatch energy_gpu.job`

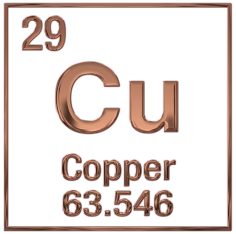
in.energy

```
units         metal
atom_style    atomic
boundary      p p p              # Sets up periodic boundary conditions - default is p

lattice           fcc 3.615
region            box block 0 10 0 10 0 10
create_box        1 box
create_atoms      1 box

pair_style    eam
pair_coeff    1 1 Cu_u3.eam      # embedded atom method type potential
thermo 1
run 0                            # Tells lammps just to perform the energy calculation
```

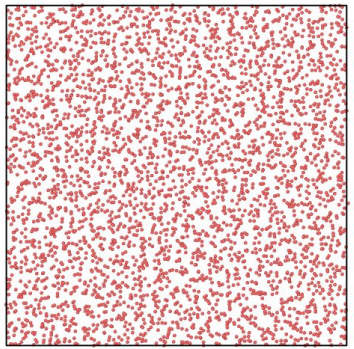
Analysis of Results

- Identify the cohesive energy of the system.
- Cohesive Energy:
  - The energy required to move an atom from the bulk to the vacuum.
  - Potential Energy / Number of Atoms
- Potential energy and number of atoms can be found in the log.lammps file.

Project #2.b - Cu_Melt



- We will simulate a bulk metal (cu) using periodic boundary conditions and an embedded atom method (eam) force field.
- We will calculate the melting temperature of cu in this simulation by raising the temperature until the system melts.
- This simulation has been setup in Cu_Melt

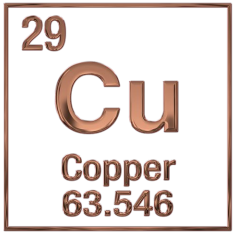


$$E_i = F_\alpha\left(\sum_{j\neq i}\rho_\beta(r_{ij})\right) + \frac{1}{2}\sum_{j\neq i}\phi_{\alpha\beta}(r_{ij})$$

embedding energy

pairwise interaction

Project #2.b - Cu_Melt



- **`sbatch cu_melt_gpu.job`**

in.melt - 1

```
# bulk Cu lattice

units         metal
atom_style    atomic
boundary      p p p              # Sets up periodic boundary conditions

lattice           fcc 3.615
region            box block 0 10 0 10 0 10
create_box        1 box
create_atoms      1 box

pair_style    eam                # embedded atom method type potential
pair_coeff    1 1 Cu_u3.eam
```

in.melt - 2

```
velocity     all create 300.0 376847 loop geom

neighbor    1.0 bin
neigh_modify    every 1 delay 5 check yes

fix          1 all npt temp 300.0 300.0 100.0 iso 0.0 0.0 1000.0

timestep    0.005       # Specifies a timestep length
thermo      500         # Prints thermodynamic information

run          50000
reset_timestep 0        # Resets timestep
```

#Keeps the simulation at a specified temperature and pressure

in.melt - 3

```
dump trj all custom 10000 traj.melt.dump id type x y z

variable stp equal "step"
variable den equal "density"    # Defines variables which we will print later
variable pe equal "pe"
variable temp equal "temp"
fix st2 all print 10000 "${stp} ${temp} ${den} ${pe}" file melt.dat    # Prints variables to an output file

unfix 1
fix             2 all npt temp 300.0 2000.0 100.0 iso 0.0 0.0 1000.0
run             1000000
```

Analysis of Results

- Identify the melting temperature from the density or pe data outputs.
- If you are using mobaxterm (windows) or xquartz (mac):
  - **`gnuplot`**
  - **`plot 'melt.dat' u 2:3`**
  - **`plot 'melt.dat' u 2:4`**

5 Minute Break

Project #3 - Wire

- We will adjust the parameters in our input script for project 1 to create and equilibrate a nanowire.

- Think how you would edit the in.melt lammps script to create a nanowire.
  - Extend the simulation box to a rectangular prism.
  - Leave about 10 angs of space between the wire and the edge of the box.

in.wire - 1

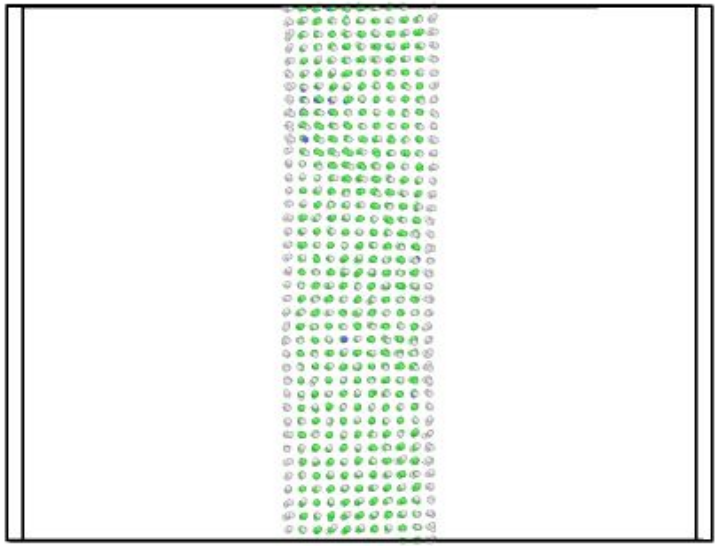
```
units           metal
atom_style      atomic
boundary        p p p

lattice         fcc 3.615
region          box1 block 10 15 10 15 0 20
region          box2 block 0 25 0 25 0 20

create_box      1 box2
create_atoms    1 region box1

pair_style eam
pair_coeff 1 1 Cu_u3.eam
```

in.wire - 2

```
velocity      all create 300.0 376847 loop geom
velocity      all zero linear
velocity      all zero angular

neighbor      1.0 bin
neigh_modify    every 1 delay 5 check yes

dump trj all custom 20000 traj.wire.dump id type x y z

fix           1 all npt temp 300.0 300.0 100.0 aniso 0.0 0.0 1000.0
timestep      0.005
thermo        10000
run           500000
write_data    wire_output.lammps        # Write output file in lammps format
```

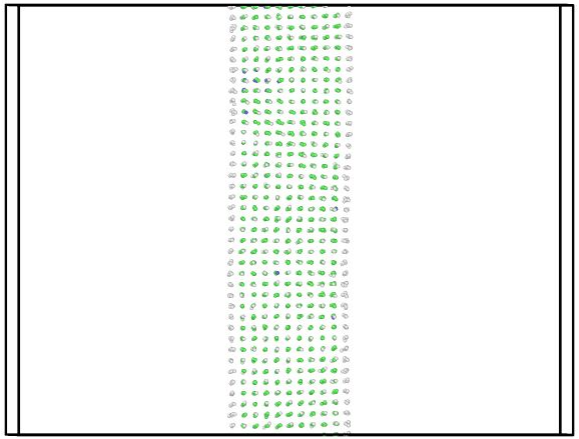
Project #3 - Wire

- `cd ../Wire`
- `sbatch wire_gpu.job`



Project #4 - Tensile Test

- We will run a tensile test on our previously created nanowire.
- We will read the output file of our in.wire job.



Project #4 - Tensile Test

- `cd ../Tensile`
- `sbatch tensile_gpu.job`

in.tensile - 1

```
units          metal
atom_style     atomic
boundary       p p p

read_data      wire_output.lammps
pair_style     eam
pair_coeff     1 1 Cu_u3.eam

neighbor       1.0 bin
neigh_modify   every 1 delay 5 check yes

dump trj all custom 10000 traj.dump id type x y z
```

in.tensile - 2

```
fix             1 all nvt temp 300.0 300.0 10.0

timestep   0.005
thermo     10000

fix    2 all deform 200 z erate 0.0002
reset_timestep 0
variable stp equal "step"
variable lz equal "lz"
variable pe equal "pe"
variable pzz equal "pzz"
fix st2 all print 10000 "${stp} ${lz} ${pe} ${pzz}" file test.dat
run             500000
```

Analysis of Results

- Approximately what strain does failure occur?

- What is the mechanism of failure?

Project #5 - Thermal Conductivity

- We will calculate the thermal conductivity of an Ar gas using the Green - Kubo method.
- The calculations will be handled by LAMMPS and the result will be output to the log file.

Project #5 - Thermal Conductivity

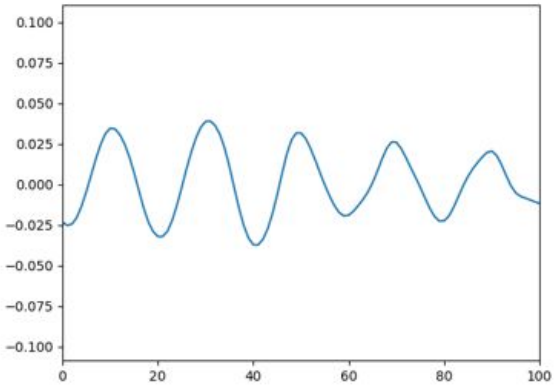
- `cd ../GK`
- `sbatch GK_gpu.job`

https://docs.lammps.org/compute_heat_flux.html

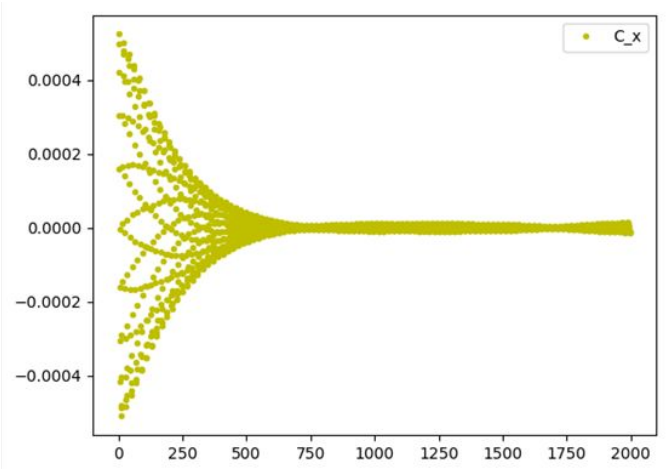
Green Kubo Method

$$k = \frac{V}{k_B T^2} \int_0^\infty \langle J_x(0)\, J_x(t) \rangle dt$$

Heat Flux



Autocorrelation function



Kubo, Ryogo (1957-06-15). "Statistical-Mechanical Theory of Irreversible Processes. I. General Theory and Simple Applications to Magnetic and Conduction Problems". *Journal of the Physical Society of Japan*. **12** (6): 570–586. 10.1143/jpsj.12.570

in.GK - 1

```
# Sample LAMMPS input script for thermal conductivity of solid Ar

units       real
variable    T equal 70
variable    V equal vol
variable    dt equal 4.0
variable    p equal 200         # correlation length
variable    s equal 10          # sample interval
variable    d equal $p*$s       # dump interval

# convert from LAMMPS real units to SI

variable    kB equal 1.3806504e-23    # [J/K] Boltzmann
variable    kCal2J equal 4186.0/6.02214e23
variable    A2m equal 1.0e-10
variable    fs2s equal 1.0e-15
variable    convert equal ${kCal2J}*${kCal2J}/${fs2s}/${A2m}
```

in.GK - 2

```
# setup problem
dimension    3
boundary     p p p
lattice      fcc 5.376 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1
region       box block 0 4 0 4 0 4
create_box   1 box
create_atoms 1 box
mass         1 39.948
pair_style   lj/cut 13.0
pair_coeff   * * 0.2381 3.405
timestep     ${dt}
thermo       $d

# equilibration and thermalization

velocity     all create $T 102486 mom yes rot yes dist gaussian
fix          NVT all nvt temp $T $T 10 drag 0.2
run          8000
```

in.GK - 3

```
reset_timestep 0
compute      myKE all ke/atom
compute      myPE all pe/atom
compute      myStress all stress/atom NULL virial
compute      flux all heat/flux myKE myPE myStress
variable     Jx equal c_flux[1]/vol
variable     Jy equal c_flux[2]/vol
variable     Jz equal c_flux[3]/vol
fix          JJ all ave/correlate $s $p $d &
        c_flux[1] c_flux[2] c_flux[3] type auto file J0Jt.dat ave running
```

in.GK - 4

```
variable    scale equal ${convert}/${kB}/$T/$T/$V*$s*${dt}
variable    k11 equal trap(f_JJ[3])*${scale}
variable    k22 equal trap(f_JJ[4])*${scale}
variable    k33 equal trap(f_JJ[5])*${scale}
thermo_style custom step temp v_Jx v_Jy v_Jz v_k11 v_k22 v_k33
run         100000
variable    k equal (v_k11+v_k22+v_k33)/3.0
variable    ndens equal count(all)/vol
print       "average conductivity: $k[W/mK] @ $T K, ${ndens} /A\^3"
```

Analysis of Results

- Check the end of the log.lammps file. The thermal conductivity should be reported.

CHALLENGE:

- Edit the in.GK script to calculate the thermal conductivity of bulk copper (project #2).
    - Why might this result be inaccurate?

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

https://hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu
Phone: 979-845-0219

Please Help us help you. Please include details in your request for support, such as, Cluster (Faster, Grace, Terra, ViDaL), NetID (UserID), Job information (Job ID(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.

Acknowledgements

This course is supported by NSF 2112356: ACES - Accelerating Computing for Emerging Sciences