Technion - Israel Institute of Technology
Faculty of Mechanical Engineering
The Nanomechanics Simulations Laboratory

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת מכונות
המעבדה לסימולציות בננומכניקה

LAMMPS

(Large−scale Atomic/Molecular Massively Parallel Simulator)

From A User's Point of View

Dan Mordehai

Department of Mechanical Engineering, Technion, 32000 Haifa, Israel.

PRACE

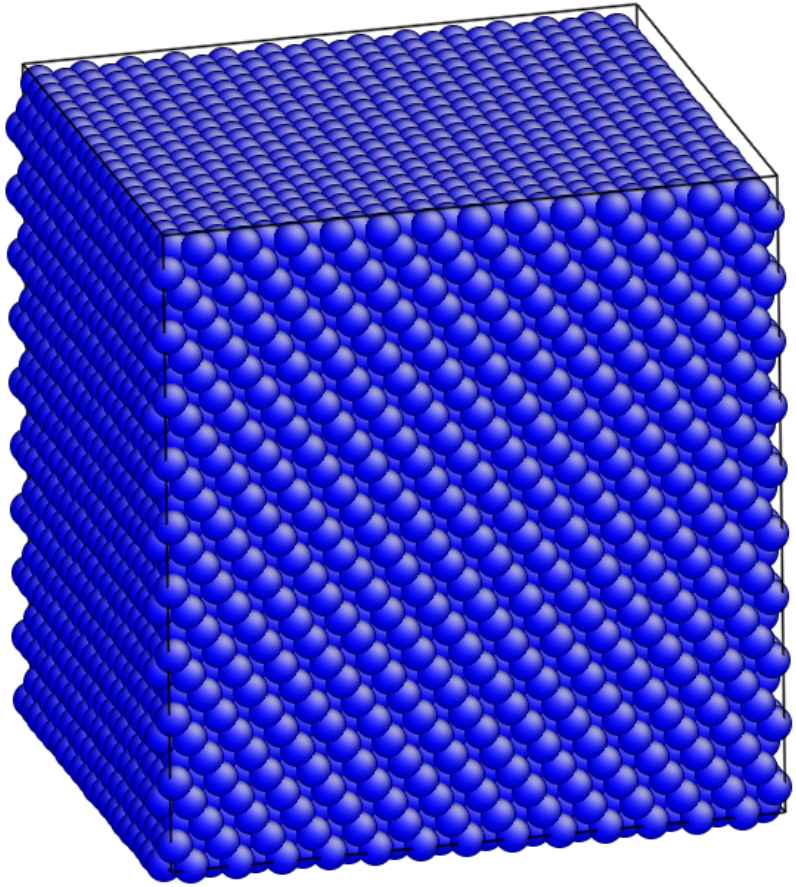
PRACE 2017 Winter School-Tel Aviv

Overview

- Molecular Dynamics (MD) Simulations – A brief Overview
- How to construct basic MD simulations in LAMMPS
- Example – running and analyzing MD simulations
- Hybrid MPI/OpenMP simulations.
- Visualizing the simulation results.

Molecular Dynamics (MD) Simulations

Molecular Dynamics simulation (MD) is a computational method to determine the trajectories of atoms in phase space according to Newton's equations of motions. The atoms are interacting via an interatomic potential that provides a description of atomic interactions.

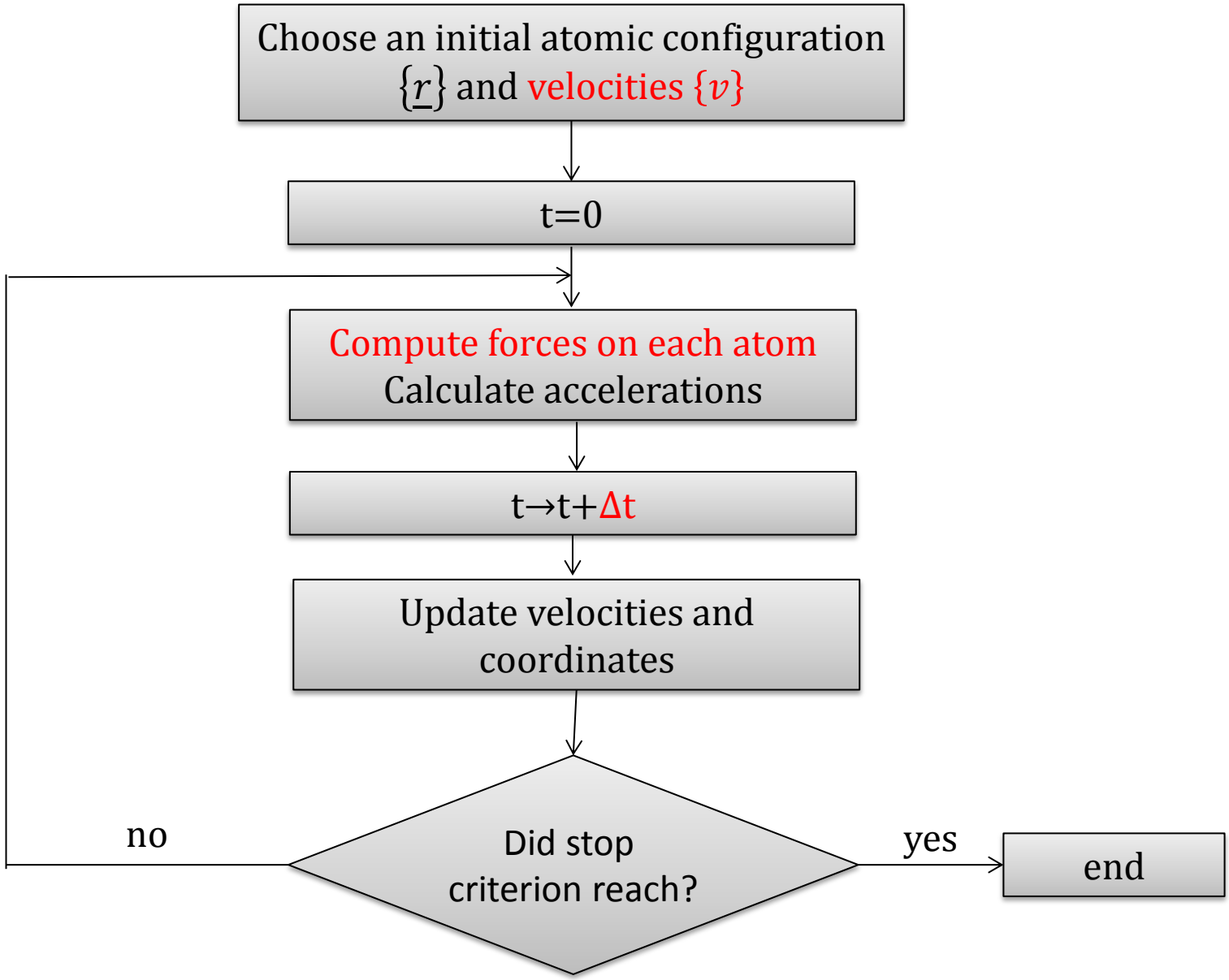


If the position of the *i*-th atom is $\vec{r}_i$ then the equation of motion for each atom is

$$m_i \ddot{\vec{r}}_i = \vec{F}_i + \ldots$$

$\vec{F}_i$ is the force acting on the *i*-th atom due to its interaction with its neighboring atoms.

Molecular Dynamics – Schematic Structure

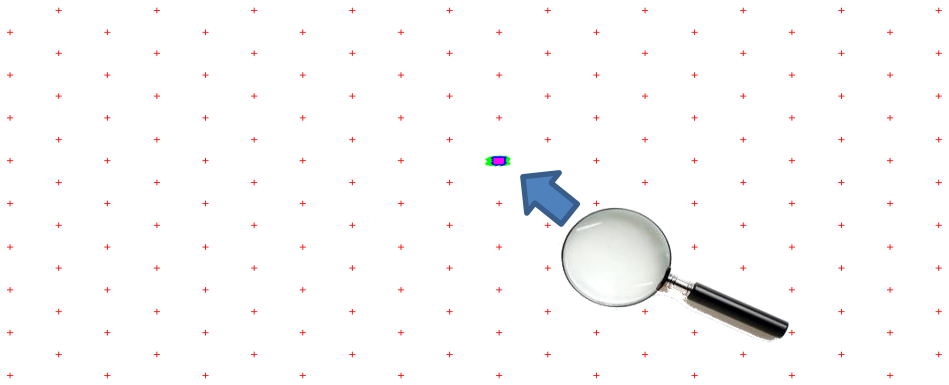


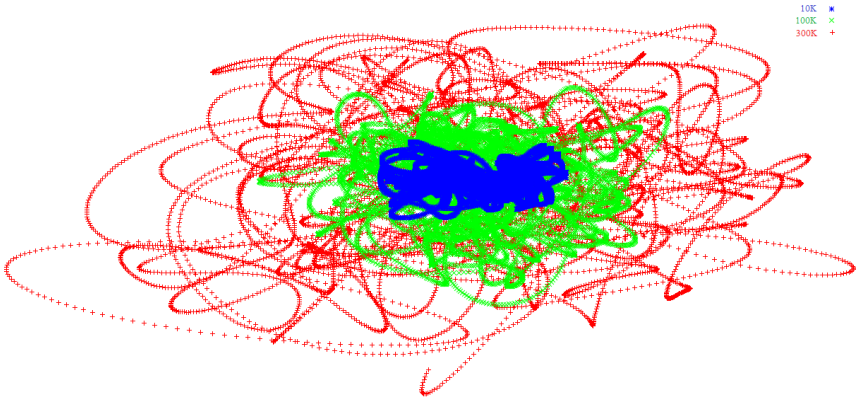
Initialization

Atom initial positions are set according to the desired problem.

Ideally , velocities should be also initialized in equilibrium conditions. The conditions satisfy $k_B T = \left\langle \frac{{p_i}^2}{m} \right\rangle$, and the momentum distribution satisfies the Maxwell-Boltzmann distribution $P(p) = \left(\frac{1}{2\pi m k_B T}\right)^{3/2} e^{-\frac{p^2}{2mk_BT}}$.

Practically, velocities can be initialized in a different distribution and they will evolve towards the equilibrium distribution.





Interatomic Potentials

$$\vec{F}(\vec{r}) = -\vec{\nabla}U(\vec{r})$$

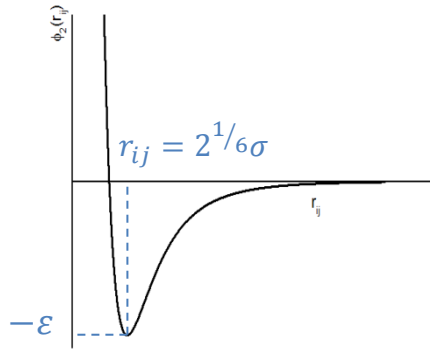
U is the interatomic potential

Pair Potentials $U = \frac{1}{2}\sum_i \sum_j \phi_2(r_{ij})$

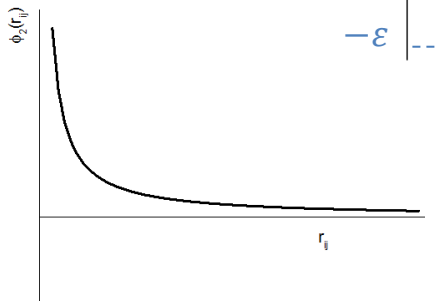
Lennard-Jones potential $\phi_2(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$

Repulsive term

Attractive VdW term (first order)



Coulomb interaction $\phi_2(r_{ij}) = \frac{Kq_iq_j}{r_{ij}}$



Buckingham Potential $\phi_2(r_{ij}) = Ae^{-r_{ij}/R_{BM}} - \frac{B}{r_{ij}^6}$

Morse Potential $\phi_2(r_{ij}) = \epsilon\left(e^{-2\alpha(r_{ij}-r_0)} - 2e^{-\alpha(r_{ij}-r_0)}\right)$

Interatomic Potentials

$$\vec{F}(\vec{r}) = -\vec{\nabla}U(\vec{r})$$

U is the interatomic potential

Many-body Potentials

Embedded-Atom Method (EAM)

$$U = \frac{1}{2}\sum_i \sum_j \phi_2(r_{ij}) + \sum_i F\left[\sum_j \rho(r_{ij})\right]$$

Stillinger-Weber, Tersoff

$$U = \frac{1}{2}\sum_i \sum_j \phi_2(r_{ij}) + \sum_i \sum_j \sum_j \phi_3(r_{ij}, r_{ik}, r_{jk})$$

ReaxFF

$$U = U_{bonds} + U_{angle} + U_{torsion} + U_{VdW} + E_{Coulomb} + \ldots$$

Interatomic Potentials in LAMMPS

Force fields (taken from LAMMPS's website 8/2/17)

- pairwise potentials: Lennard-Jones, Buckingham, Morse, Born-Mayer-Huggins, Yukawa, soft, class 2 (COMPASS), hydrogen bond, tabulated
- charged pairwise potentials: Coulombic, point-dipole
- manybody potentials: EAM, Finnis/Sinclair EAM, modified EAM (MEAM), embedded ion method (EIM), EDIP, ADP, Stillinger-Weber, Tersoff, REBO, AIREBO, ReaxFF, COMB, SNAP, Streitz-Mintmire, 3-body polymorphic
- long-range interactions for charge, point-dipoles, and LJ dispersion: Ewald, Wolf, PPPM (similar to particle-mesh Ewald)
- polarization models: QEq, core/shell model, Drude dipole model
- charge equilibration (QEq via dynamic, point, shielded, Slater methods)
- coarse-grained potentials: DPD, GayBerne, REsquared, colloidal, DLVO
- mesoscopic potentials: granular, Peridynamics, SPH
- electron force field (eFF, AWPMD)
- bond potentials: harmonic, FENE, Morse, nonlinear, class 2, quartic (breakable)
- angle potentials: harmonic, CHARMM, cosine, cosine/squared, cosine/periodic, class 2 (COMPASS)
- dihedral potentials: harmonic, CHARMM, multi-harmonic, helix, class 2 (COMPASS), OPLS
- improper potentials: harmonic, cvff, umbrella, class 2 (COMPASS)
- polymer potentials: all-atom, united-atom, bead-spring, breakable
- water potentials: TIP3P, TIP4P, SPC
- implicit solvent potentials: hydrodynamic lubrication, Debye
- force-field compatibility with common CHARMM, AMBER, DREIDING, OPLS, GROMACS, COMPASS options
- access to KIM archive of potentials via pair kim
- hybrid potentials: multiple pair, bond, angle, dihedral, improper potentials can be used in one simulation
- overlaid potentials: superposition of multiple pair potentials

The Velocity-Verlet Algorithm

For simplicity, we shall consider the 1-dimensional numerical solution of an ODE of the form

$$\ddot{r} = a(\underline{r}) = F(\underline{r})/m$$

When we propagate time "forwards":

$$r(t+\Delta t) = r(t) + v(t)(\Delta t) + \frac{1}{2}a(t)(\Delta t)^2 + \mathcal{O}\left[(\Delta t)^3\right]$$

$$v(t+\Delta t) = v(t) + a(t)(\Delta t) + \mathcal{O}\left[(\Delta t)^2\right]$$

Taking time "backwards"

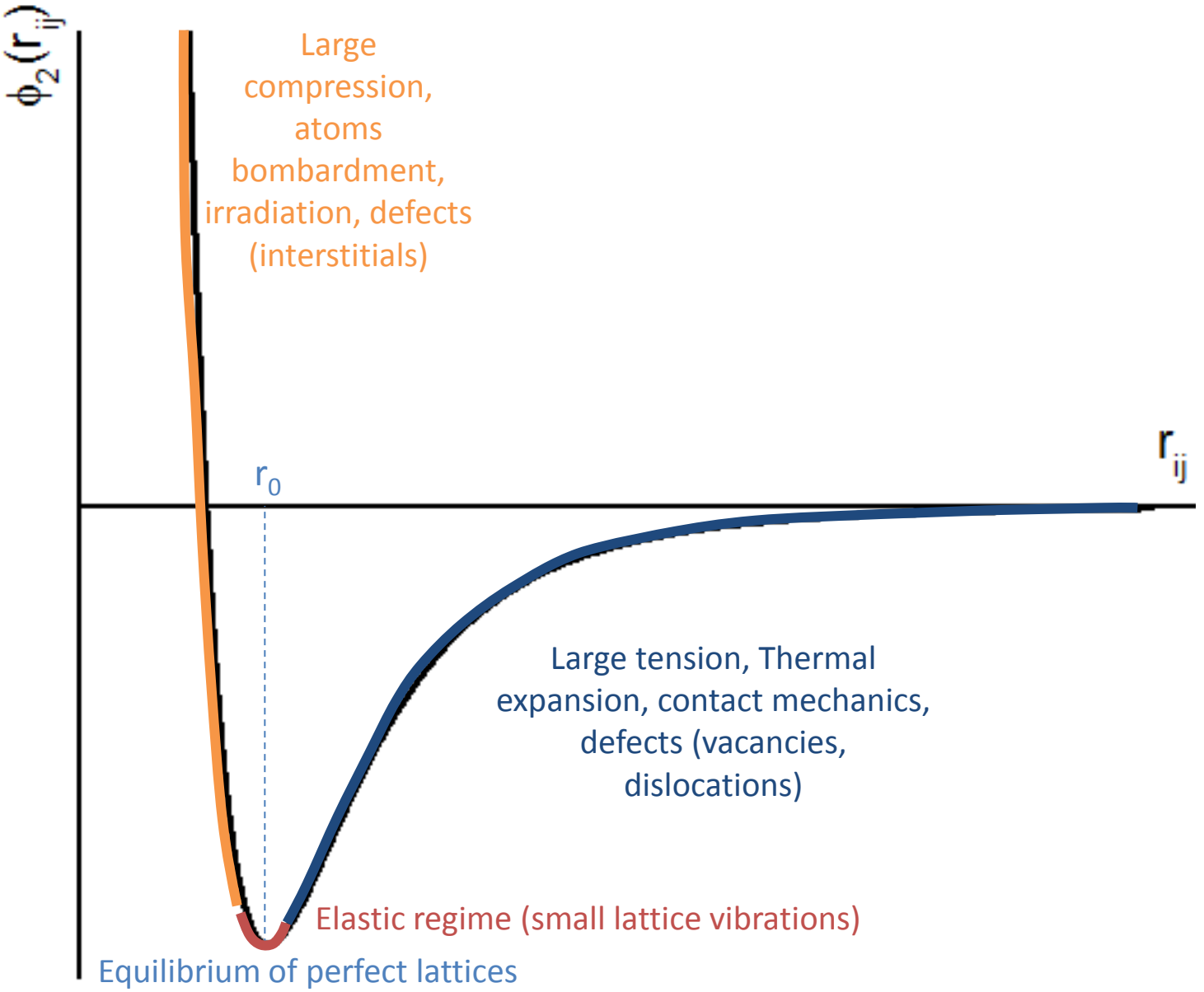
$$r(t) = r\left[(t+\Delta t) - \Delta t\right] = r(t+\Delta t) - v(t+\Delta t)(\Delta t) + \frac{1}{2}a(t+\Delta t)(\Delta t)^2 + \mathcal{O}\left[(\Delta t)^3\right]$$

Demanding that "forward" and "backward" motion will yield the same *r(t+Δt) yields a correction to the velocity intergration*

$$r_{n+1} = r_n + v_n(\Delta t) + \frac{1}{2}a_n(\Delta t)^2$$

$$v_{n+1} = v_n + \frac{a_{n+1} + a_n}{2}\Delta t$$

Properties of Pair Potentials



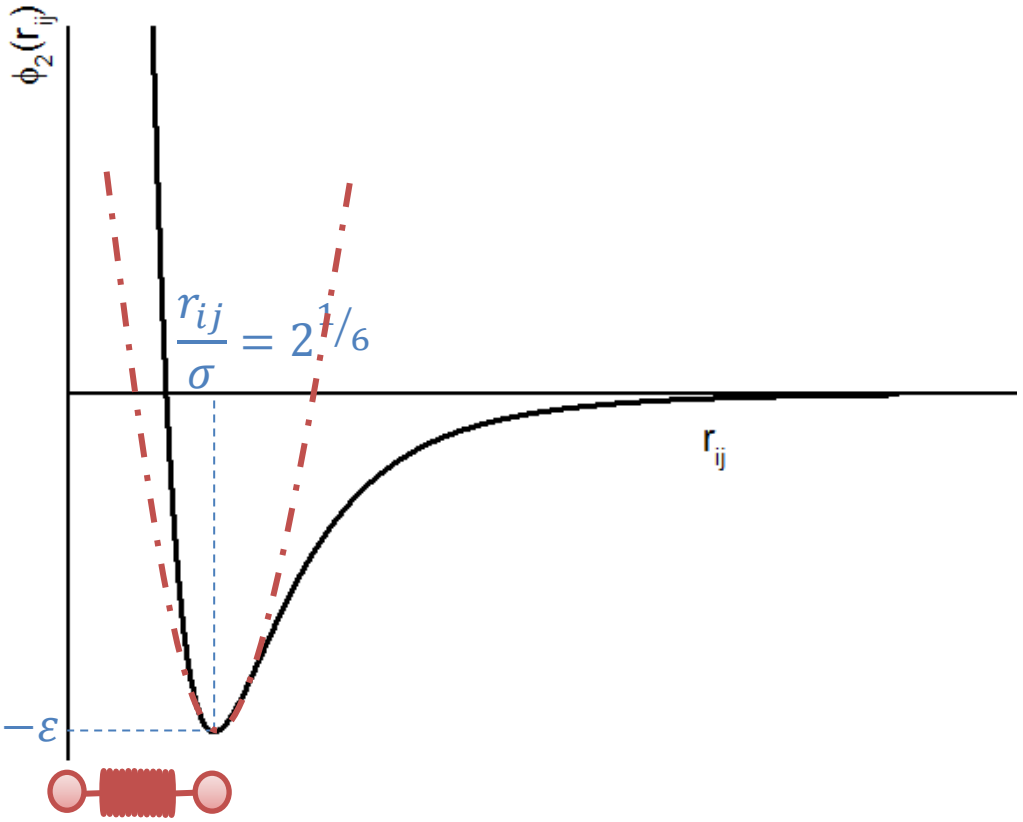
Time step in MD simulation

Let us consider a Lennard-Jones potential and approximate the potential to an harmonic oscillator near equilibrium.

$$\phi_2(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

$$K = \left.\frac{d\phi_2^2(r_{ij})}{dr_{ij}^2}\right|_{r_0=2^{1/6}\sigma} = 36\cdot 2^{\frac{2}{3}}\frac{\epsilon}{\sigma^2}$$

$$T = 2\pi\sqrt{\frac{m}{K}} = 2\pi\sqrt{\frac{1}{36\cdot 2^{\frac{2}{3}}}\frac{m\sigma^2}{\epsilon}}$$



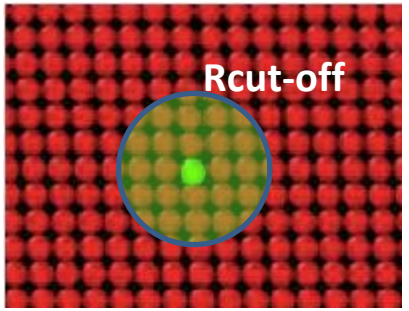
Typical values for metals:

- $\varepsilon \sim 1\ eV$
- $\sigma \sim 2\ Å$
- $m \sim 1\cdot 10^{-25}\ kg$

Which leads to periods of $T \sim 1\cdot 10^{-13}\ sec$ → ***$\Delta t \sim 1\cdot 10^{-15}\ sec = 1\ fsec$***

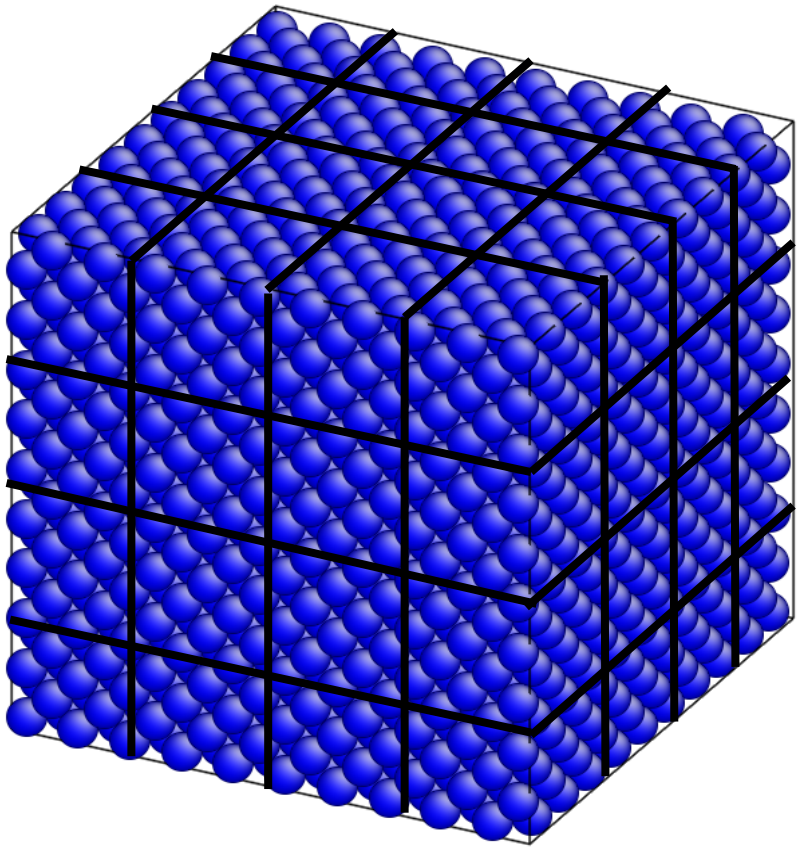
Finding neighbors – Cut-off distance

The most time consuming part of the MD simulation is finding neighbors. To save time, only atoms within a distance smaller than the cut-off distance are accounted in the non-bonded interaction energy.



But we still need to do it efficiently…

- Sort the atoms into bins.
- For each atom, seek for its neighbors from the atoms inside its bin and the surrounding bins.
- The bin size is taken slightly larger than $R_{cut\text{-}off}$ so that the bins are not updated every time step.



Statistical Ensembles

The equations of motions $m_i \ddot{\vec{r}}_i = \vec{F}_i$ corresponds to a system that conserves energy (potential + kinetic). In statistical mechanics, the ensemble of states that satisfy these conditions are called a **microcanonical ensemble** or the **NVE ensemble**.

Number of atoms — Volume (shape) — Energy

NVT ensemble (canonical ensemble) – Conserves number of atoms, shape and temperature.

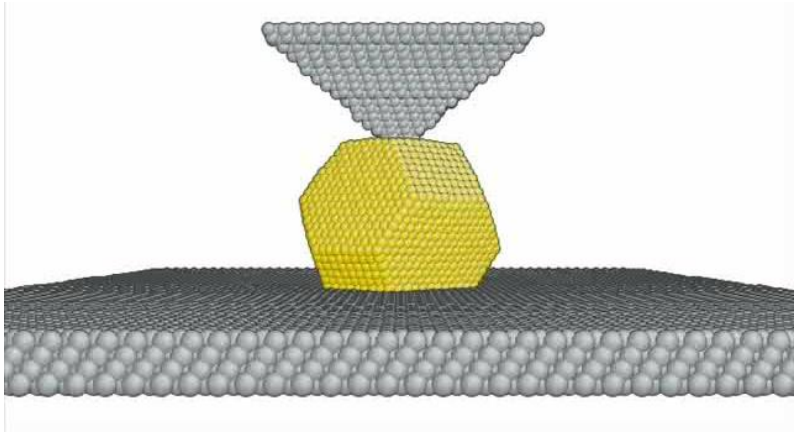
e.g. Nosé-Hoover Thermostat

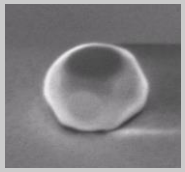
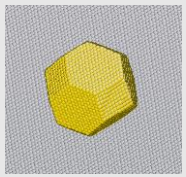
$$\ddot{\vec{r}}_i = \frac{\vec{F}_i}{m_i} - \xi \dot{\vec{r}}_i \qquad \dot{\xi} = \frac{3NK_B}{Q}(T_{inst} - T)$$

NPH ensemble (canonical ensemble) – Conserves number of atoms, pressure (stress) and enthalpy.

NPT ensemble (canonical ensemble) – Conserves number of atoms, pressure (stress) and temperature.

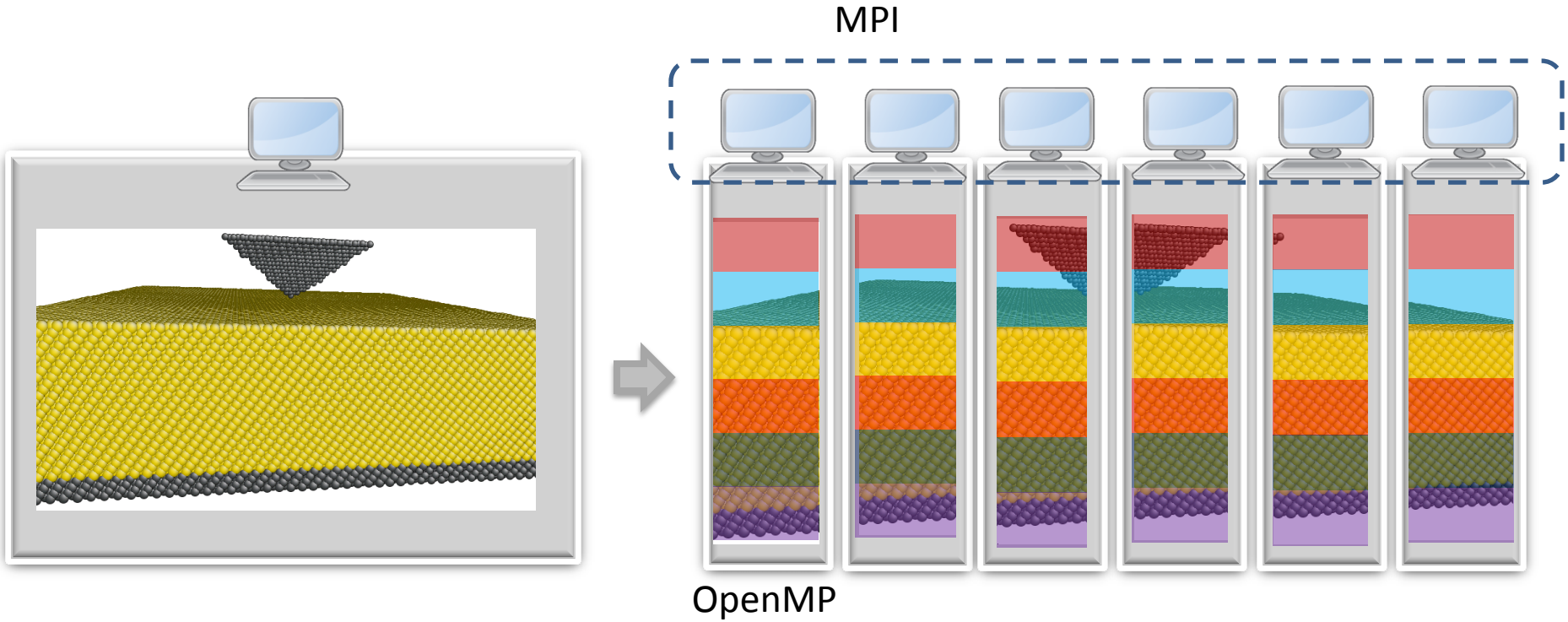
So can we start performing MD simulations?



	No. of Atoms	"Real" Time	Time to perform
	~8000 Millions	30 sec	1 minute
	~8000 Millions	30 sec	450 Milliard Years!

High Performance Computing (HPC)

The MD simulations are performed using the open source Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)



Constructing a basic MD simulation in LAMMPS

LAMMPS (Large−scale Atomic/Molecular Massively Parallel Simulator) is an open−source code classical molecular dynamics, developed at Sandia National Laboratories, designed to run efficiently on parallel computers. LAMMPS is distributed freely under the terms of the GNU Public License (GPL)

General features (taken from LAMMPS's website 8/2/17)

- runs on a single processor or in parallel
- distributed-memory message-passing parallelism (MPI)
- spatial-decomposition of simulation domain for parallelism
- open-source distribution
- highly portable C++
- optional libraries used: MPI and single-processor FFT
- GPU (CUDA and OpenCL), Intel Xeon Phi, and OpenMP support for many code features
- easy to extend with new features and functionality
- runs from an input script
- syntax for defining and using variables and formulas
- syntax for looping over runs and breaking out of loops
- run one or multiple simulations simultaneously (in parallel) from one script
- build as library, invoke LAMMPS thru library interface or provided Python wrapper
- couple with other codes: LAMMPS calls other code, other code calls LAMMPS, umbrella code calls both

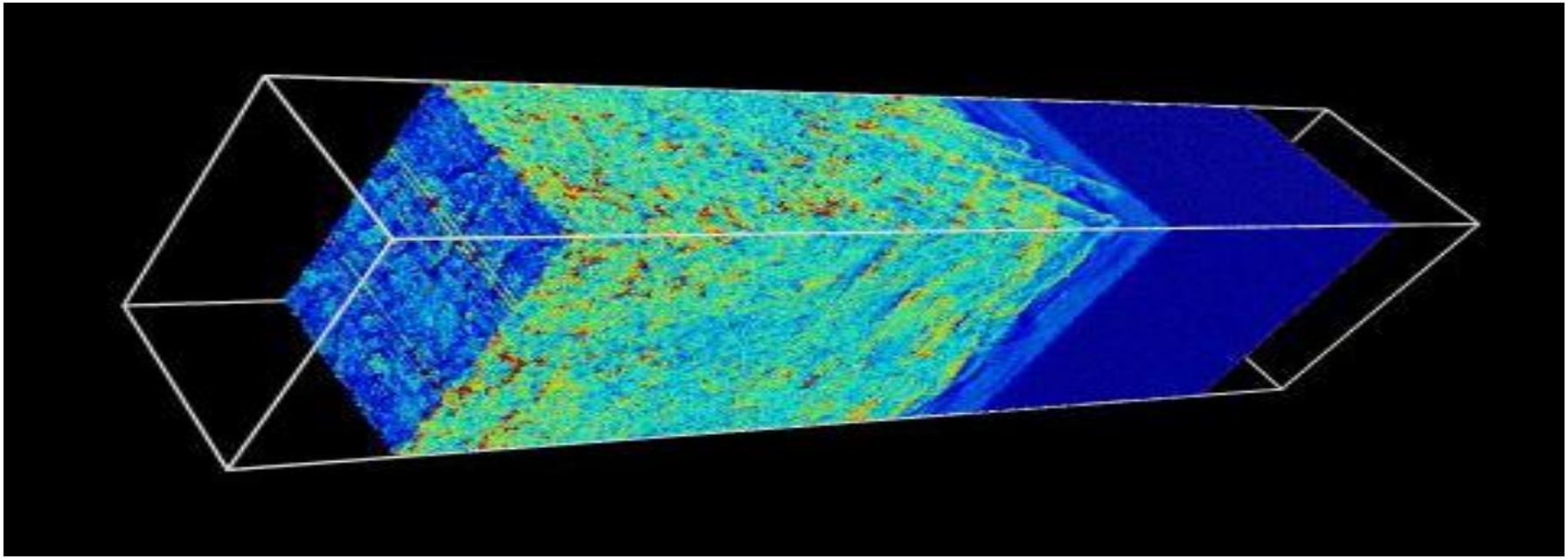
Constructing a basic MD simulation in LAMMPS

What do we need?

- **Input script** (**obligatory**)
- Initial atoms positions / restart (optional)
- Tabulated interatomic potential (optional)

What can we find?

- Global thermodynamic quantities on the **whole** computational cell (*lammps.log*)
- Global thermodynamic quantities on **groups** of atoms
- Positions/velocities of atoms.



Input script - example

```
units           metal
atom_style      atomic

boundary        p p p
newton          off off
processors      1 1 1

lattice         fcc 3.50 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1  origin 0.05 0.05 0.05
region          sim_region block 0 10 0 10 0 10 units lattice
create_box      2 sim_region
create_atoms    1 box

pair_style      eam/alloy
pair_coeff      * *  Mishin-Ni-Al-2009.eam.alloy Ni Al

neighbor        2.0 bin
neigh_modify    delay 3

timestep        0.001
thermo_style    custom step pe  etotal pxx pyy pzz lx ly lz
thermo          10
restart         50000 NICYCLESTAT.STA

dump            drelax all atom 1000 dump_relax1.atom
fix             frelax all nve
run             5000
unfix           frelax
undump          drelax
```

Input Script – Initialization

```
units           metal
atom_style      atomic

boundary        p p p
newton          off off
processors      1 1 1
```

units (metal):

mass = grams/mole, distance = Å, time = picoseconds energy = eV, temperature = ^{o}K, pressure = bars

atom_style: defines the identity of the particles in the simulations (may be atoms, ellipsoid grains, bonds etc.).

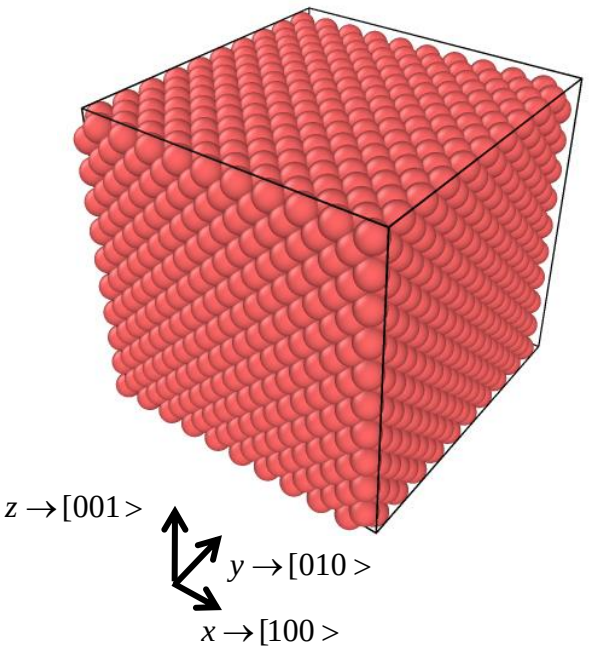
boundary: boundary conditions in each direction

newton: determines if to consider Newton's 3rd law for pairwise and bonded interactions

processors: division of the computational cell into sub-domains, for each processor (in the x,y and z direction).
Important: Make sure the atoms are as equally distributed as possible between the processors and that the surface-to-volume ratio of each sub-domain is minimal. LAMMPS can do it for you by putting wildcard asteriks *.

Input Script – Simulation Box Definition (1)

```
lattice       fcc 3.52 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 origin 0.05 0.05 0.05
region        sim_region block 0 10 0 10 0 10 units lattice
create_box    2 sim_region
create_atoms  1 box
```



lattice: used to define the lattice structure in the cell (can be replaced with an *external* positions file). Notice that the origin is slightly shifted, not to have an atom at (0,0,0)

region : defines a geometric region in space (**does not** include the atoms in this space)
In the example, the region, which is named *sim_region,* is of a box shape (*block)*. The box boundaries are 0 and 10 in both x, y and z directions in the units of the lattice (each period of the lattice is unity in lattice units)

create_box: creates the simulation box (in which the atoms will be defined). In the example, the box is created in the region called *sim_region* and 2 types of atoms will be employed in the cell. Although we use only one type of atoms in this example, we need to prepare the simulation for 2 atoms since the potential definition file includes two atoms.

create_atoms: fills the simulation box region with atoms, according to the lattice definition. In this case, only one type of atom is used.

Input Script – Simulation Box Definition (2)

```
pair_style      eam/alloy
pair_coeff      * *  Mishin-Ni-Al-2009.eam.alloy Ni Al
```

In these simulations we employ and EAM potential for alloys (describe atomic interaction between atoms of more than one type), which is defined in a table file.

pair_style : defines the type of interatomic potential (in this case eam/alloy)

pair_coeff: the coefficient of the potential type chosen in pair_style. In the case of eam/alloy we need to define the file with the interaction functions and give names to each type of atom.
In the example, the pair potential, the density function and the embedding function are defined as tables in *Mishin-Ni-Al-2009.eam.alloy*. The file includes information for alloy with 2 types of atoms. The first type of atom is called *Ni* and the second type is *Al*, which are defined in the potential files.
You can choose the consider only part of the information from the potential file.For instance, to consider only the Ni-Ni and Al-Al interactions from the file, but to define differently Ni-Al interaction. In our case, we wish to use all the information, and we do that by using the wildcards asterisk (* * means to consider the interaction defined in the file between all types of atoms with all types of atoms)

Input Script – Simulation Box Definition (3)

```
neighbor        2.0 bin
neigh_modify    delay 3
```

Defines the way the neighbors of each atoms is calculated

neighbor: defines the size of the bin for the neighbors list, beyond the interatomic potential cutoff.

neigh_modify: defines parameters on how and when the neighbors list is updated. In the example, the number of time steps between refreshing the neighbors list will not fall below 3 timesteps (even if needed).

Settings

```
timestep        0.001
thermo_style    custom step pe  etotal pxx pyy pzz lx ly lz
thermo          10
restart         50000 NICYCLESTAT.STA
```

timestep: in time units.

thermo_style: defines the **global** parameters that our being plotted (see list in link).

thermo: the frequency at which the global parameters (defined at thermo_style) are being computed and plotted.

restart: write a restart file with all the information needed to restart a calculation from the middle. In the example, a file with the prefix NICYCLESTAT.STA is written every 50000 timesteps (the suffix is the timestep number, for instance NICYCLESTAT.STA.100000 is the restart file written at the 100000-th timestep).

Calculation

```
dump        drelax all atom 1000 dump_relax1.atom
fix         frelax all nve
run         5000
unfix       frelax
undump      drelax
```

dump/undump: sends information on **atoms** to a file. Each dump is given an id-name (*derelax* in this case). In the example, a list of atoms every 1000 timesteps is written to *dump_relax1.atom* file, with their type, id number and reduced coordinates (style=*atom*). *all* is actually a group with all atoms. We will later better define the concept of "groups".

fix/unfix: all operations on a groups of atoms during calculation are defined via *fix*. There are many type of fixes. The fix in the example defines the calculation ensemble (nve). You can define different statistical ensembles on different groups of atoms at the same timestep.

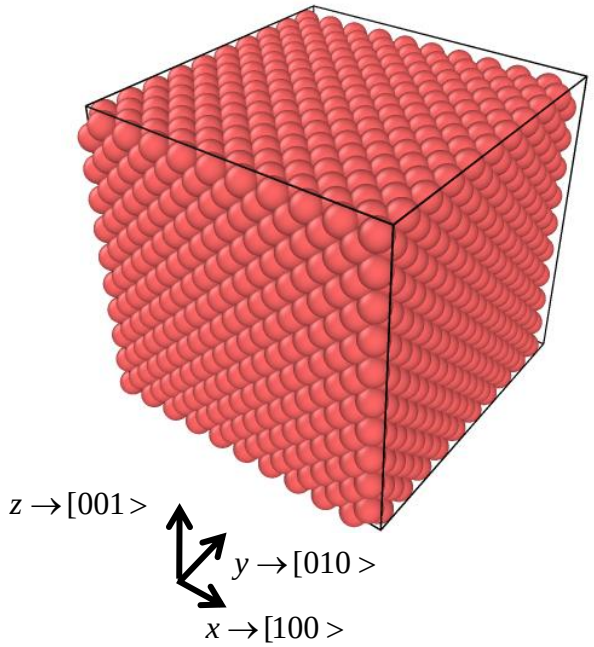
run: propagates the systems in the number of timesteps specified after run, during which the operations defined earlier are being performed.

Summary of script

```
units           metal
atom_style      atomic

boundary        p p p
newton          off off
processors      1 1 1

lattice         fcc 3.50 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1  origin 0.05 0.05 0.05
region          sim_region block 0 10 0 10 0 10 units lattice
create_box      2 sim_region
create_atoms    1 box

pair_style      eam/alloy
pair_coeff      * *  Mishin-Ni-Al-2009.eam.alloy Ni Al

neighbor        2.0 bin
neigh_modify    delay 3

timestep        0.001
thermo_style    custom step pe  etotal pxx pyy pzz lx ly lz
thermo          10
restart         50000 NICYCLESTAT.STA

dump            drelax all atom 1000 dump_relax1.atom
fix             frelax all nve
run             5000
unfix           frelax
undump          drelax
```

The script build an **infinite <u>Ni</u>** fcc lattice with a lattice parameter of 3.50Å.

Then the simulation cell is relaxed for 5000 timesteps, conserving the number of atoms, shape and energy.



Running the simulation on Rocks-Frontend

- Enter with account on Rocks-Frontend (128.139.196.20)
- Create a directory

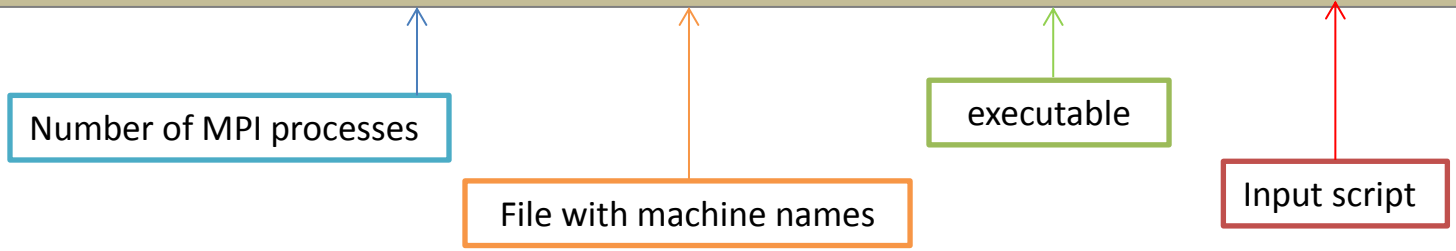
```
mkdir example
```

- Copy the input script and the interatomic potential file

```
cp /share/apps/lammps_example/in.fccnve example/
cp /share/apps/lammps_example/ Mishin-Ni-Al-2009.eam.alloy example/
```

- and enter the directory (of course…)
- Choose two machines on which you will run the simulation (machines.txt)
- Run the simulation

```
/opt/openmpi/bin/mpirun -nolocal -np 64 -machinefile machines.txt /share/apps/lammps_example/lmp_omp -in in.fccnve
```



Analyze the simulation – log file

>> less log.lammps

```
LAMMPS (17 Nov 2016)
  using 1 OpenMP thread(s) per MPI task
package omp 0
using multi-threaded neighbor list subroutines
units           metal
atom_style      atomic

boundary        p p p
newton          off off
processors      4 2 2

lattice         fcc 3.50 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 origin 0.05 0.05 0.05
Lattice spacing in x,y,z = 3.5 3.5 3.5
region          sim_region block 0 10 0 10 0 10 units lattice
create_box      2 sim_region
Created orthogonal box = (0 0 0) to (35 35 35)
  4 by 2 by 2 MPI processor grid
create_atoms    1 box
Created 4000 atoms

pair_style      eam/alloy
pair_coeff      * *  Mishin-Ni-Al-2009.eam.alloy Ni Al

neighbor        2.0 bin
neigh_modify    delay 3

timestep        0.001
thermo_style    custom step pe  etotal pxx pyy pzz lx ly lz
thermo          10
restart         50000 NICYCLESTAT.STA

dump            drelax all atom 1000 dump_relax1.atom
fix             frelax all nve
run             5000
Last active /omp style is pair_style eam/alloy/omp
Neighbor list info ...
  1 neighbor list requests
  update every 1 steps, delay 3 steps, check yes
  max neighbors/atom: 2000, page size: 100000
  master list distance cutoff = 8.28721
  ghost atom cutoff = 8.28721
  binsize = 4.1436 -> bins = 9 9 9
Memory usage per processor = 3.82291 Mbytes
Step PotEng TotEng Pxx Pyy Pzz Lx Ly Lz
       0   -17792.698   -17792.698    32146.969    32146.969    32146.969           35           35           35
      10   -17792.698   -17792.698    32146.969    32146.969    32146.969           35           35           35
```

Changing the lattice parameter

Create a variable (alat) which is the lattice parameter

```
variable    alat equal 3.50
lattice     fcc ${alat} orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 origin 0.05 0.05 0.05
```

The variable command can be replaced with a switch in the command line

```
/opt/openmpi/bin/mpirun -nolocal -np 64 -machinefile machines.txt /share/apps/lammps_example/lmp_omp  -v alat 3.50 -in in.fcc
```

Calculate the total energy with two different lattice parameters: 3.49 and 3.50

Why are they different?

Change the lattice parameter until reaching the minimum energy

– interpret the result.

NPH ensemble

Change the thermodynamic conditions to the NPH ensemble

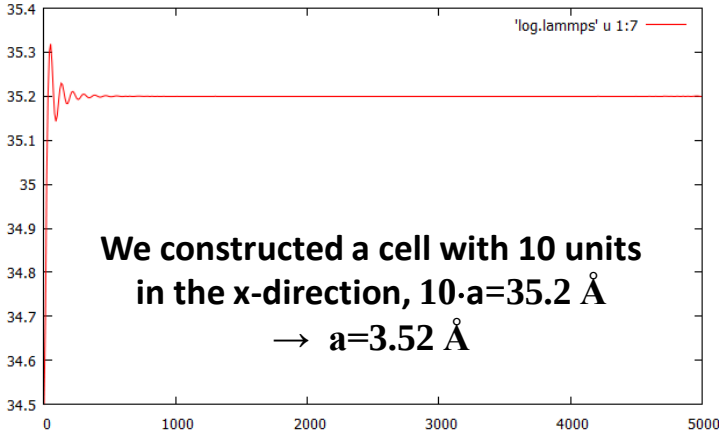
```
fix             frelax all nph x 0.0 0.0 0.5 y 0.0 0.0 0.5 z 0.0 0.0 0.5
```

Launch the simulation with an *initial* lattice parameter of 3.45Å

```
/opt/openmpi/bin/mpirun -nolocal -np 64 -machinefile machines.txt /share/apps/lammps_example/lmp_omp  -v alat 3.45 -in in.fcc
```

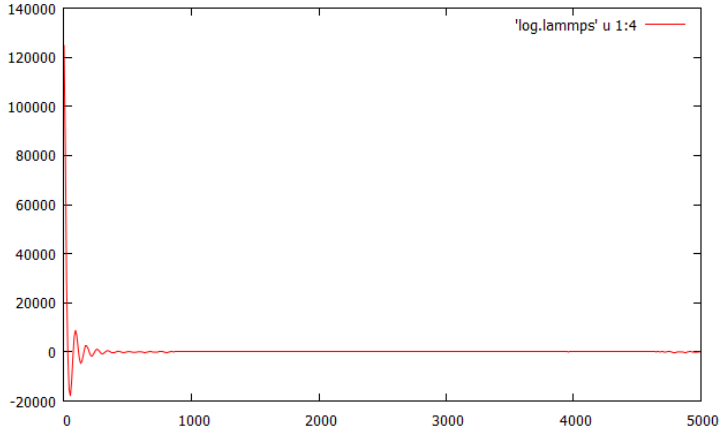
Check in the log file how the system is equilibrated.

```
/share/apps/gnuplot-5.0.5/bin/gnuplot

gnuplot> plot 'log.lammps' u 1:4
gnuplot> plot 'log.lammps' u 1:7
```



We constructed a cell with 10 units in the x-direction, 10·a=35.2 Å

→ a=3.52 Å

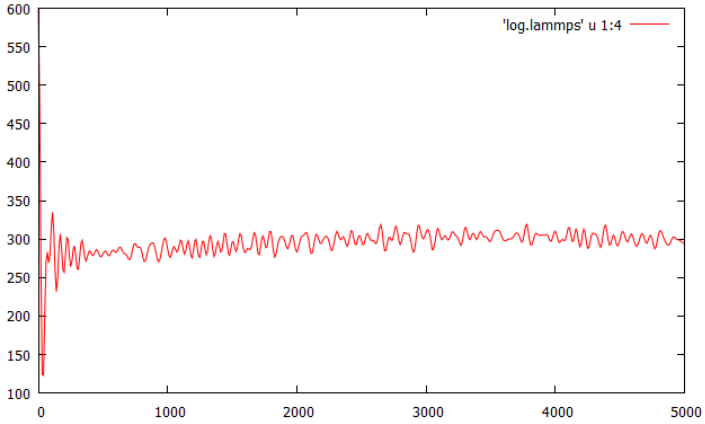


NPT ensemble

In order to control the temperature, let us make the following modifications

```
units           metal
atom_style      atomic

boundary        p p p
newton          off off
processors      4 2 2

lattice         fcc 3.45 orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 origin 0.05 0.05 0.05
region          sim_region block 0 10 0 10 0 10 units lattice
create_box      2 sim_region
create_atoms    1 box

pair_style      eam/alloy
pair_coeff      * *  Mishin-Ni-Al-2009.eam.alloy Ni Al

neighbor        2.0 bin
neigh_modify    delay 3

timestep        0.001
thermo_style    custom step pe etotal temp pxx pyy pzz lx ly lz
thermo          10
restart         50000 NICYCLESTAT.STA

velocity        all create 600.0 34593 dist gaussian
dump            drelax all atom 1000 dump_relax1.atom
fix             frelax all npt temp 300.0 300 0.7 x 0.0 0.0 0.5 y 0.0 0.0 0.5 z 0.0 0.0 0.5
run             5000
unfix           frelax
undump          drelax
```

```
/share/apps/gnuplot-5.0.5/bin/gnuplot

gnuplot> plot 'log.lammps' u 1:4
```



Calculating the thermal expansion coefficient

The thermal expansion coefficient describes the volume change of a material when its temperature is varying at zero pressure.

$$\alpha_T = \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_{P=0} = \left(\frac{\partial \ln(\frac{V}{V_0})}{\partial T}\right)_{P=0}$$

where V_0 is volume at zero temperate and pressure.

Use the NPT ensemble to calculate the thermal expansion coefficient:

- Calculate the steady state volume at different temperatures between 100°K and 500°K (V_0 is known from previous simulations).
  Tip: change the output to plot the volume of the computational cell.
- Calculate $\ln(\frac{V}{V_0})$ and plot as function of T.
- Calculate α_T from the slope.

Running the hybrid simulations

For Hybrid MPI/OpenMP simulations we need to define provide the number of MPI processes and how many OpenMP threads for each MPI process

```
pair_style      eam/alloy
```

```
pair_style      eam/alloy/omp
```

```
/opt/openmpi/bin/mpirun -nolocal -x OMP_NUM_THREADS=8 -np 2 -machinefile machines.txt /share/apps/lammps_example/lmp_omp  -sf omp -in in.fccnve
```

Number of threads

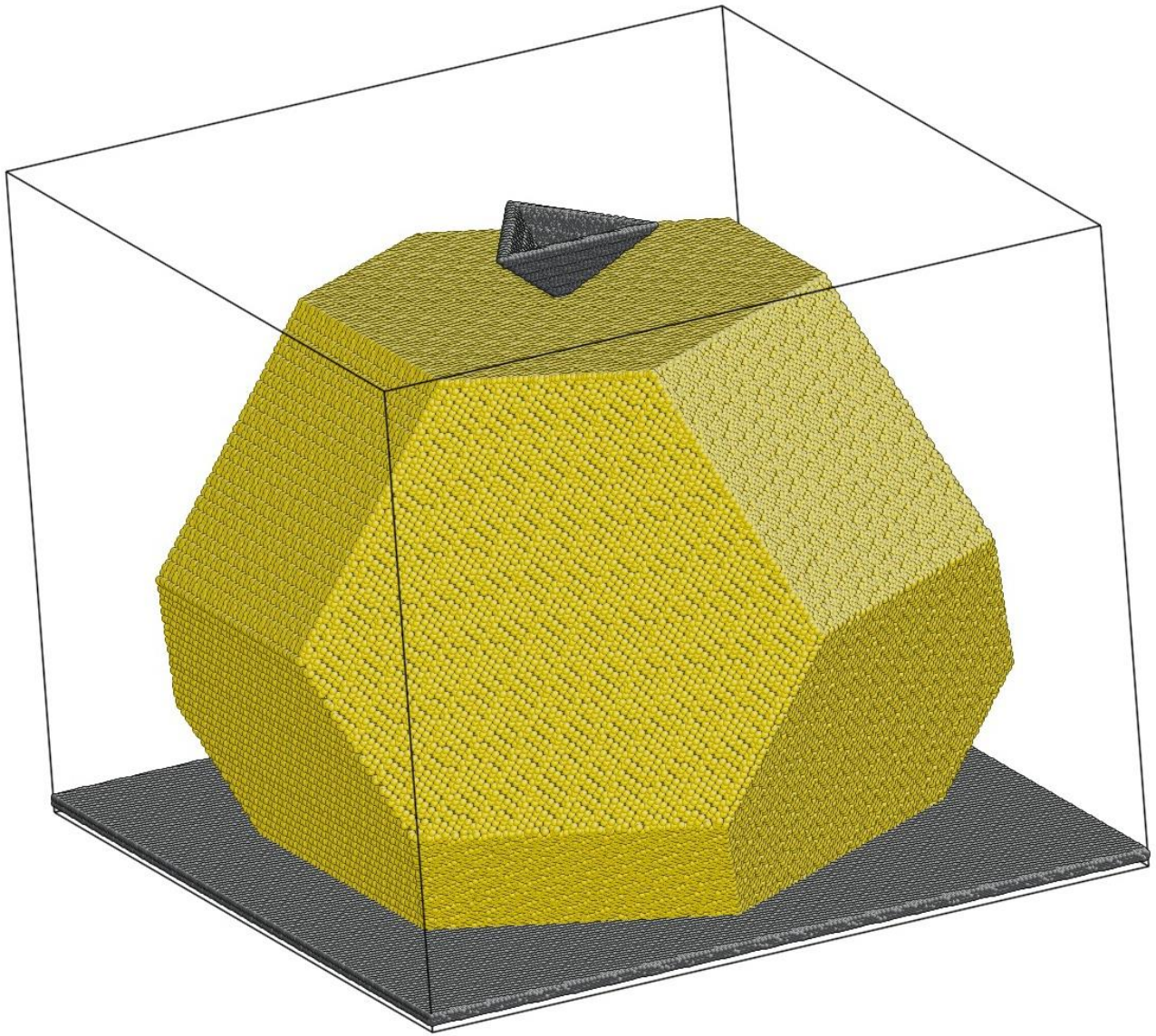
Number of MPI processes

2 MPI × 8 OpenMP threads = 16 processors

Extend to the simulation cell size to 20 × 20 × 20 units and try to partition the simulation differently (do no exceed 16 processors). Check the time spent in the MD run in each case

```
4990      -142400      -142400 -1.3877922e-08 -1.3864354e-08 -1.2037358e-08    70.400001    70.400001    70.400001
5000      -142400      -142400 -1.0673261e-05 -1.0673281e-05 -1.0671441e-05    70.400001    70.400001    70.400001
Loop time of 106.658 on 16 procs for 5000 steps with 32000 atoms
```

Scaling on PRACE infrastructure

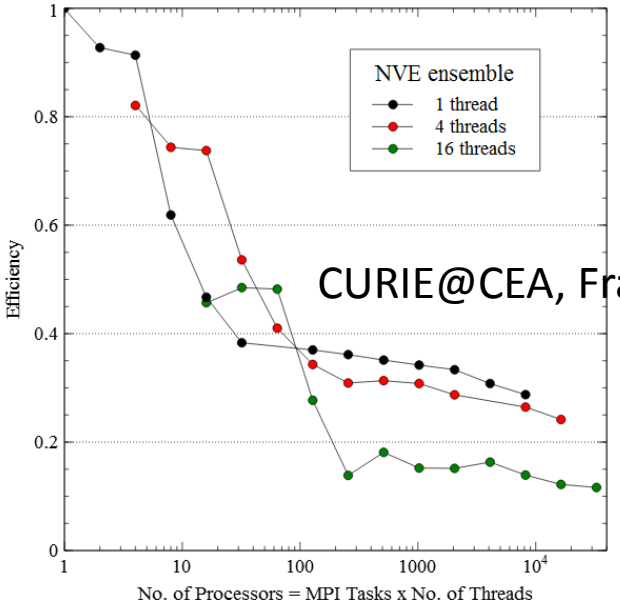


~16M atoms

=

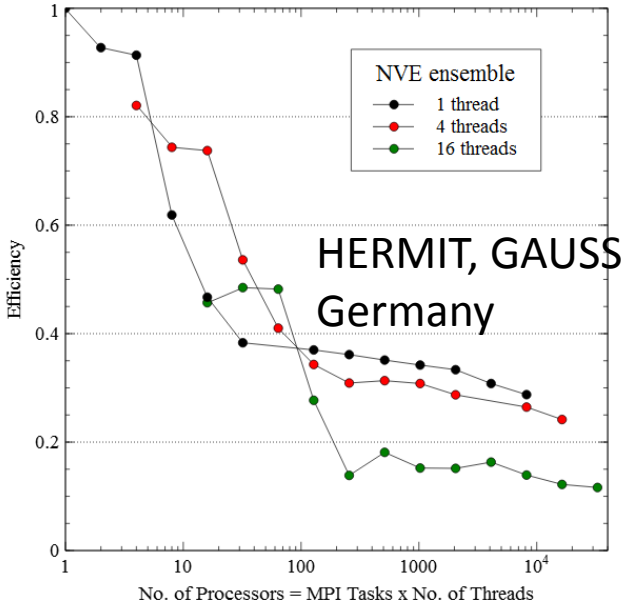
~32 nm high

Scaling on PRACE infrastructure

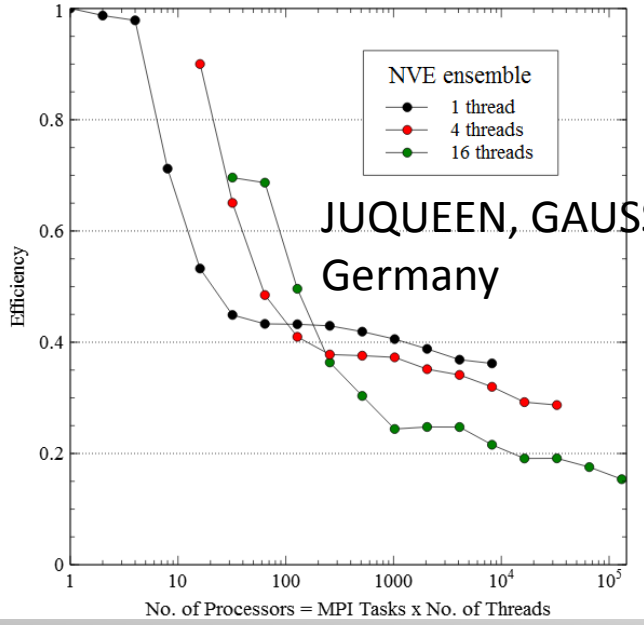


CURIE@CEA, France

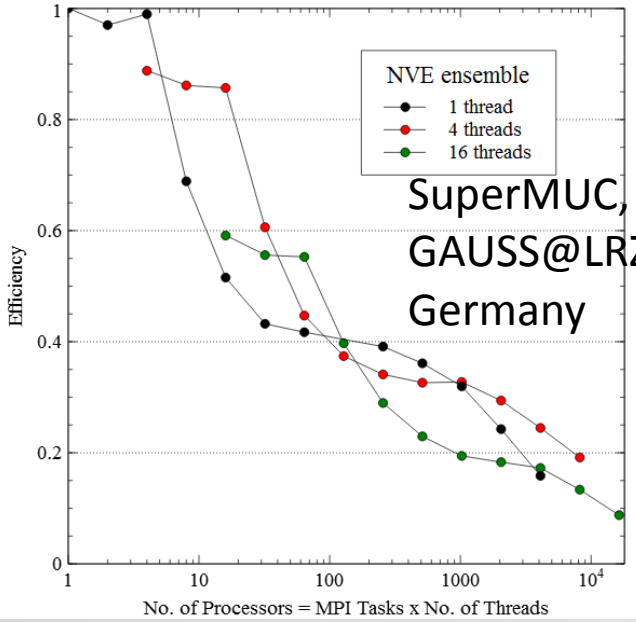
$$efficiency = \frac{T_{serial} / T_{parallel}}{\# \ processors}$$



HERMIT, GAUSS@HLRS, Germany



JUQUEEN, GAUSS@FZJ, Germany



SuperMUC, GAUSS@LRZ, Germany

Running a simulation with an external file

In order to tell LAMMPS to read the lattice information from a file, you should replace the lattice definitions with the command read_data from the external file Ni_lattice.xyz

```
lattice       fcc 3.52 orient x -1 1 0 orient y -1 -1 2 orient z 1 1 1 origin 0.0 0.0 0.0
region        sim_region block 0 10 0 10 0 10 units lattice
create_box    2 sim_region
create_atoms  1 box
```

```
read_data     Ni_lattice.xyz
```

The file Ni_lattice.xyz includes all information needed to construct the simulation cell:

Header

Total Number of atoms

Number of atom types

Computational cell shape

List of atoms: Number, type, X, Y, Z

```
fully periodic Ni lattice (z in the [111] direction)
# created by Dan Mordehai 11/04/2008

6000 atoms

2 atom types

    0.00000   43.11102 xlo xhi
    0.00000   24.89016 ylo yhi
    0.00000   60.96819 zlo zhi

Atoms

1 1    3.95184    1.86676    5.08068
2 1    1.07778    1.86676    3.04841
....
```

Reminder: although you have only a single type, your potential definitions are for two types of atoms.

Pre-processing tools

There are different pre-processing tools available (some appears on LAMMPS's website)

Materials Design Inc. - http://www.materialsdesign.com/

Scienomics - http://www.scienomics.com/

Scifes Inc. - http://www.scifes.com/

XenoView - http://xenoview.mat.rpi.edu/xenoview.html

Atomman - https://github.com/usnistgov/atomman

nanoSCULPT - http://www.gmp.ww.uni-erlangen.de/nanoSCULPT.php

AtomSK - http://atomsk.univ-lille1.fr/tutorials.php

"Seeing" the atoms

Atom properties at different time steps are plotted in dump files

```
dump        drelax all atom 1000 dump_relax1.atom.*
dump        drelax all atom 1000 dump_relax1.atom
```

```
ITEM: TIMESTEP
0
ITEM: NUMBER OF ATOMS
4000
ITEM: BOX BOUNDS pp pp pp
0.0000000000000000e+00 3.4500000000000000e+01
0.0000000000000000e+00 3.4500000000000000e+01
0.0000000000000000e+00 3.4500000000000000e+01
ITEM: ATOMS id type xs ys zs
1 1 0.005 0.005 0.005
2 1 0.055 0.055 0.005
3 1 0.055 0.005 0.055
4 1 0.005 0.055 0.055
5 1 0.105 0.005 0.005
8 1 0.105 0.055 0.055
11 1 0.005 0.105 0.005
13 1 0.055 0.105 0.055
15 1 0.105 0.105 0.005
51 1 0.005 0.005 0.105
```

Time step

Number of Atoms

Cell dimensions and boundary conditions

List of atoms

"Seeing" the atoms

There are different tools to visualize the atoms (only a few examples):

AtomEye - http://li.mit.edu/Archive/Graphics/A/

Aviz - http://phony1.technion.ac.il/~aviz/

VMD - http://www.ks.uiuc.edu/Research/vmd/

RasMol - http://www.openrasmol.org/

Pizza.Py - http://pizza.sandia.gov/ (from the same house of LAMMPS)

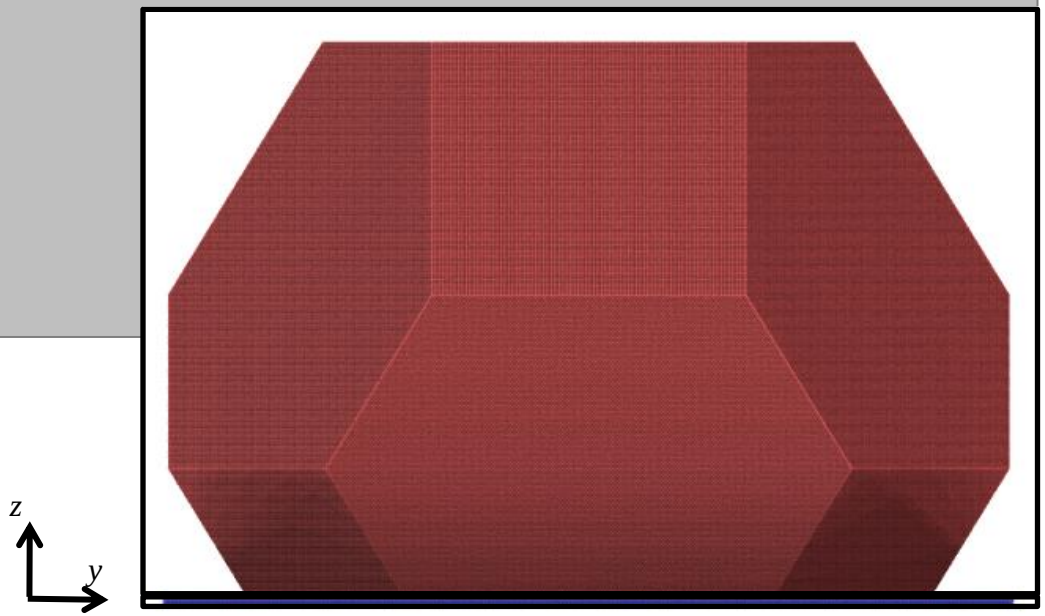
OVITO - http://www.ovito.org/

```
/share/apps/ovito-2.8.2-x86_64/bin/ovito
```

Example – Nanoindentation (1)

```
units           metal
atom_style      atomic

boundary        p p f
newton          off off
processors      * * *

pair_style      eam/alloy
pair_coeff      * * Au_GRS5017_LAMMPS.Gaussian_new.0p25 1 2

#set neighbor list update coefficients
neighbor        2.0 bin
neigh_modify delay 3

region          subfix block 0.0 1000.0 0.0 1000.00 0.0 18.80000 units box
region          part block 0.0 1000.0 0.0 1000.00 18.80100 626.28 units box

group           gpart region part
group           gsubfix region subfix

timestep        0.005

thermo          10
restart         200000 NICYCLESTAT.STA
```

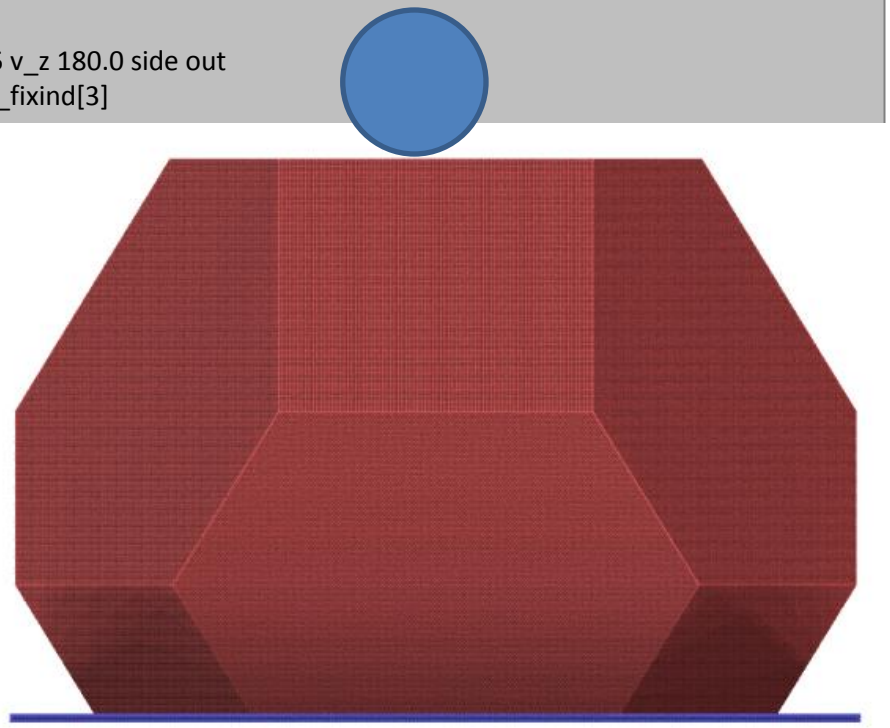
part, gpart



subfix, gsubfix

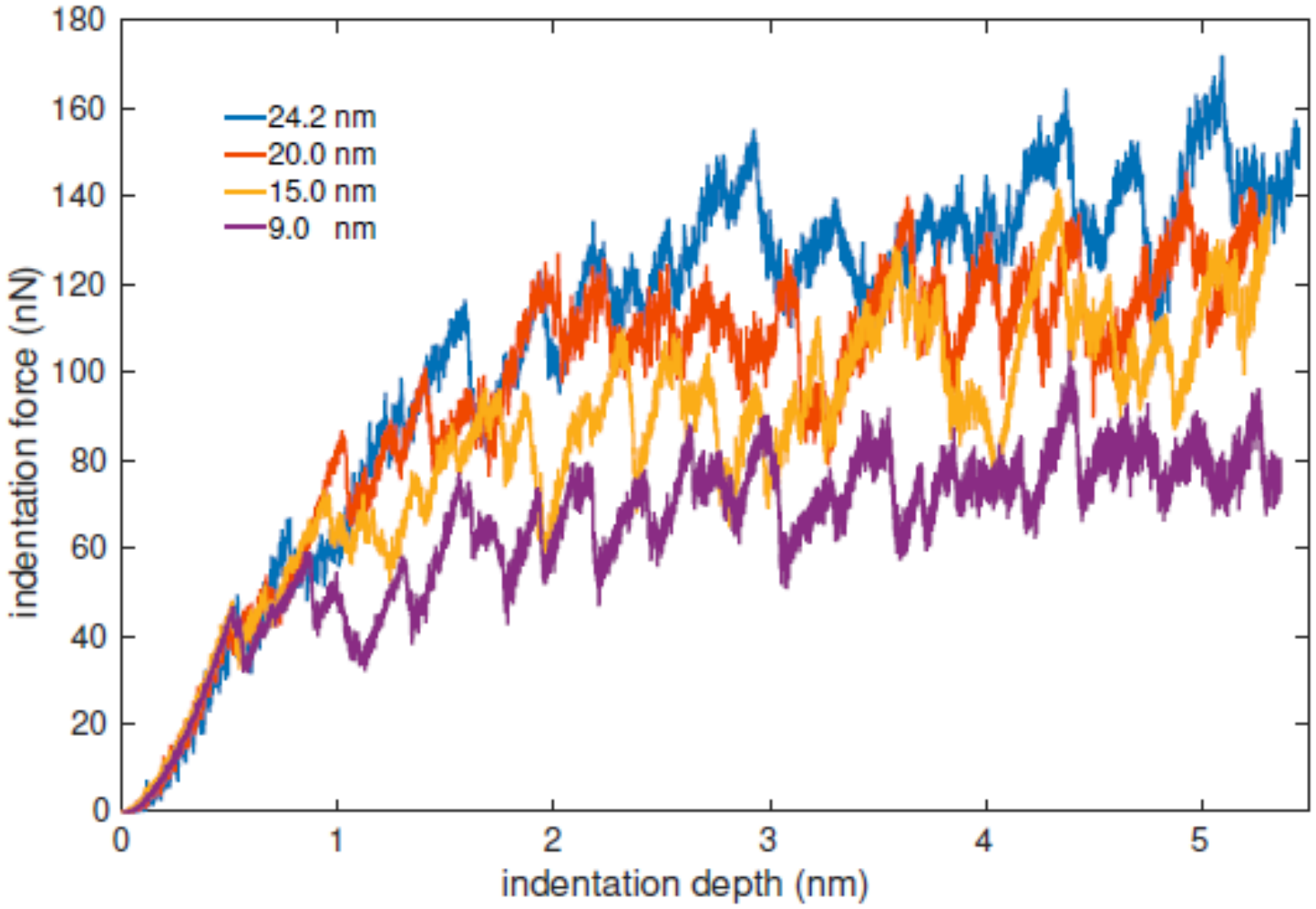
Example – Nanoindentation (1)

```
fix             nvesub gsubfix nve
fix             fixsub gsubfix setforce 0.0 0.0 0.0
variable        z equal 806.28
fix             fixind gpart indent 10.0 sphere 439.7324 461.599305 v_z 180.0 side out
thermo_style    custom step temp etotal epair f_fixind[1] f_fixind[2] f_fixind[3]

fix             heatpart0 gpart nvt temp 10.0 10.0 0.113
run             10000
unfix           heatpart0
unfix           fixind

variable        z equal vdisplace(806.28,-1.0)
fix             fixind gpart indent 10.0 sphere 439.7324 461.599305 v_z 180.0 side out
thermo_style    custom step temp etotal epair f_fixind[1] f_fixind[2] f_fixind[3]

fix             heatpart1 gpart nve
dump            dindent1 gpart atom 10000 dump_relaxA.atom.*
run             30000
undump          dindent1
unfix           heatpart1

unfix           fixind
```



Substrate atoms are “frozen”

Example – Nanoindentation



Example – Nanoindentation