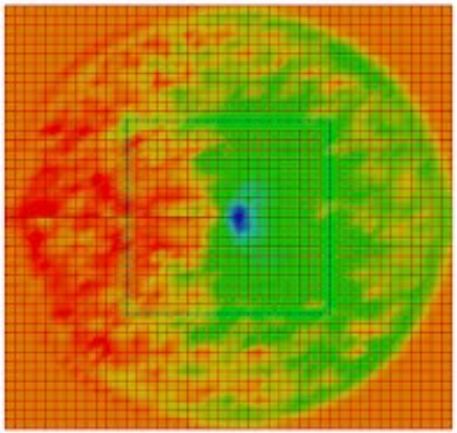
SAND2013-9651C

October 2013-January 2014

ESP900: Atomistic/Molecular Simulation:

Lecture 5: Introduction to LAMMPS

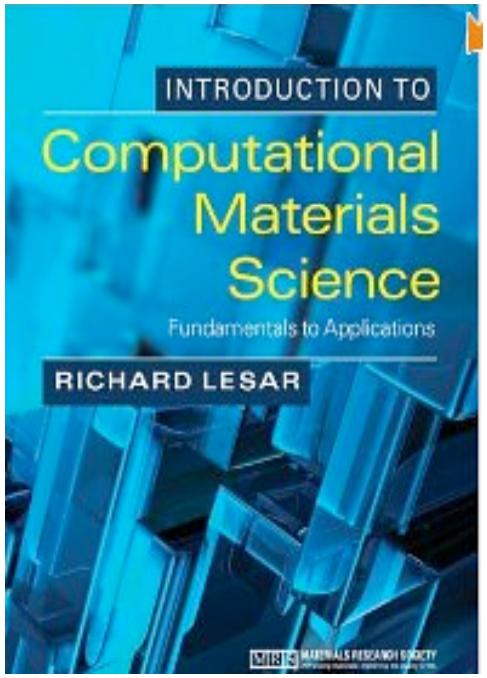


Instructor: Reese Jones

rjones@sandia.gov

(925) 294-4744

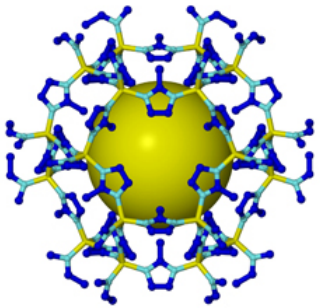
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

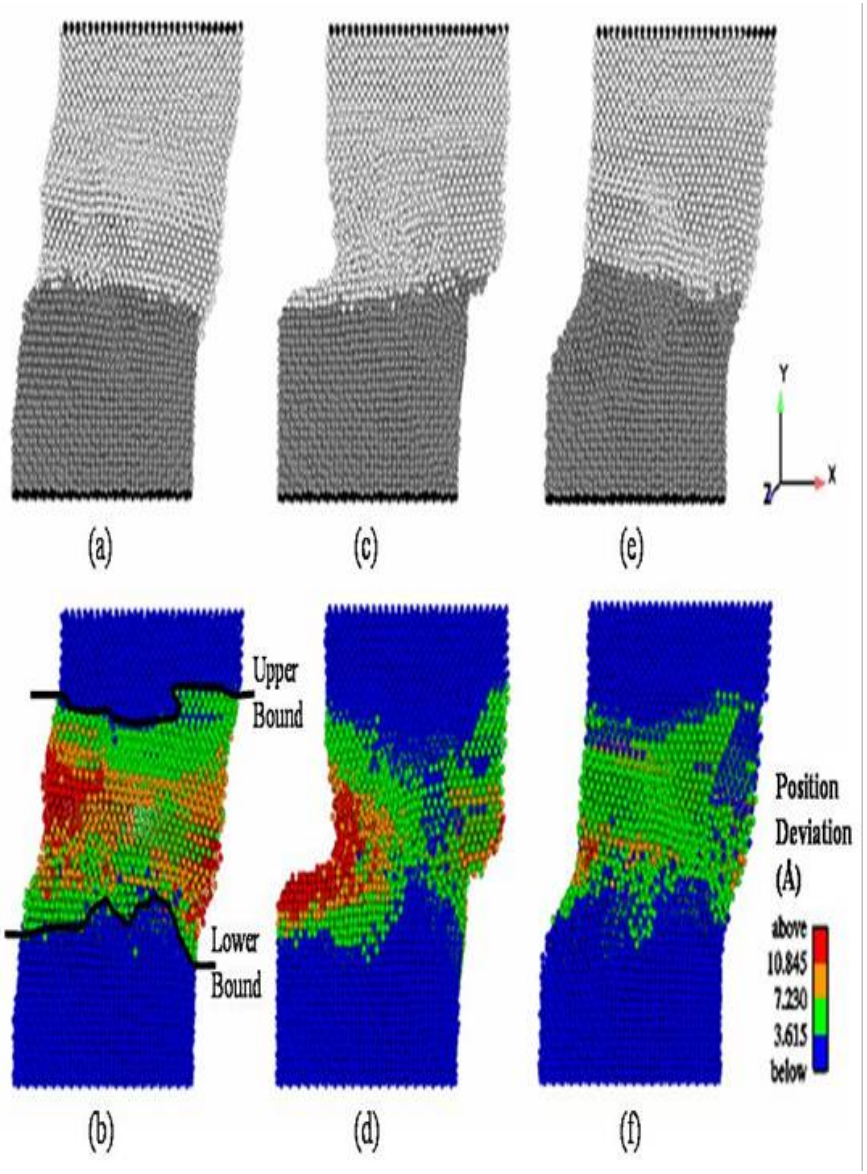


Why a whole class on LAMMPS?

(Large-scale Atomic/Molecular Massively Parallel Simulator)

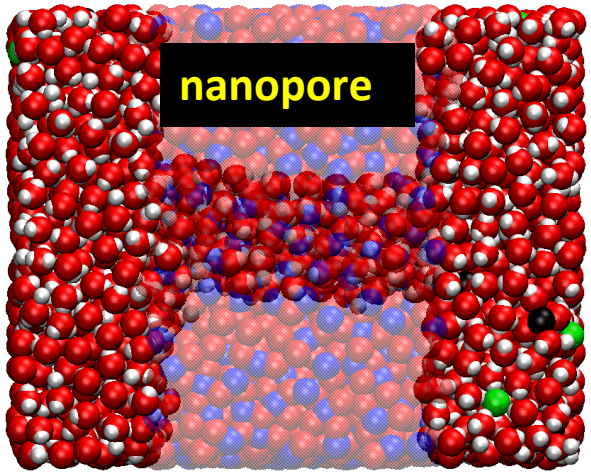
- Most complex and varied syntax of the codes used in the class
- Most of the class is on molecular dynamics
- LAMMPS is widely used both at Sandia and in the wider research community used for many reasons

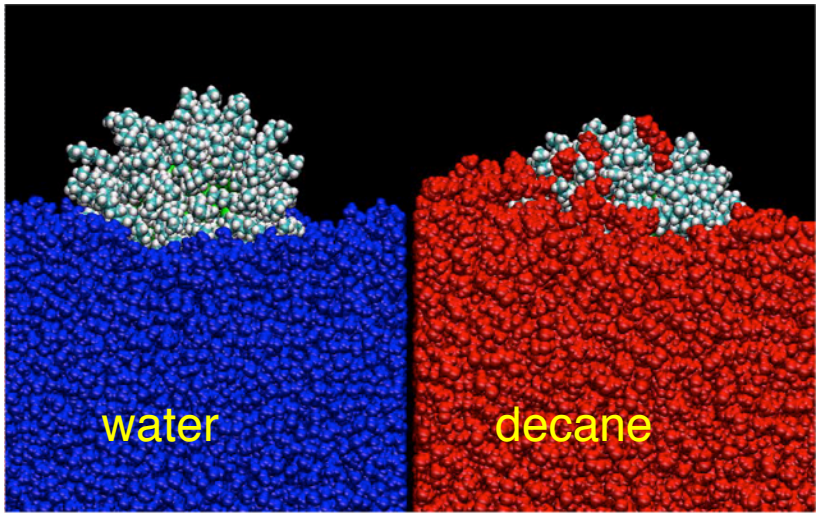


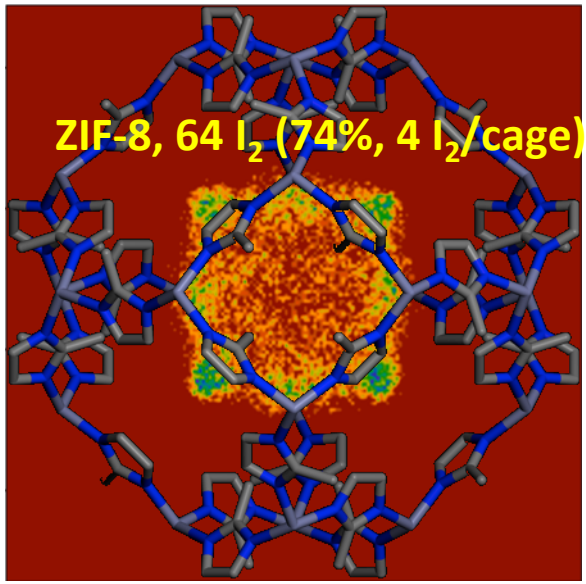


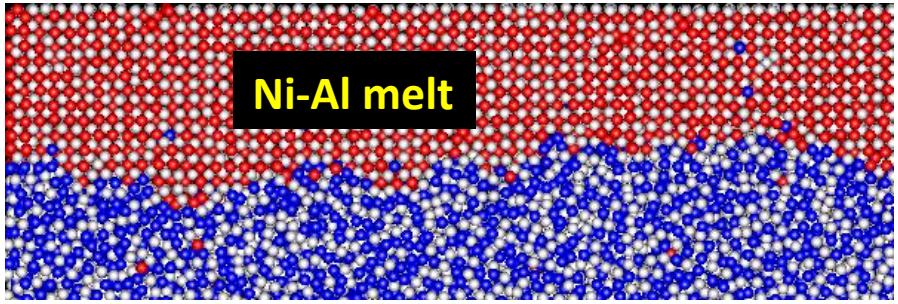
Outline

- Sales pitch for LAMMPS
- Big picture: scale, structure, expense
- Syntax
- Homework





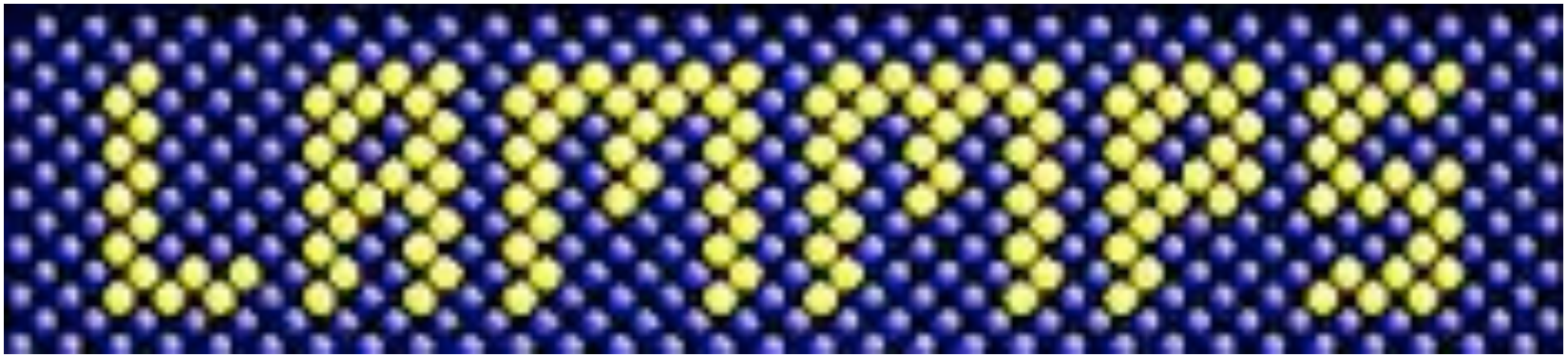




Why use LAMMPS?

Instead of NAMD, AMBER, GROMACS etc

- Open source, portable C++ http://lammps.sandia.gov/download.html
- Well documented http://lammps.sandia.gov/doc/Manual.html
- Easy to modify or extend with new features and functionality.
- Active, monitored user's e-mail list http://lammps.sandia.gov/mail.html
- Users' workshops
- Good parallel scaling and speed.
- Built-in data reduction
- Can be used for atomistic, mesoscale, and coarse-grain simulations.
- Variety of potentials, e.g. many-body and coarse-grained, for solids, fluids, molecules.
- Variety of dynamics/ensembles, boundary conditions, constraints, etc.

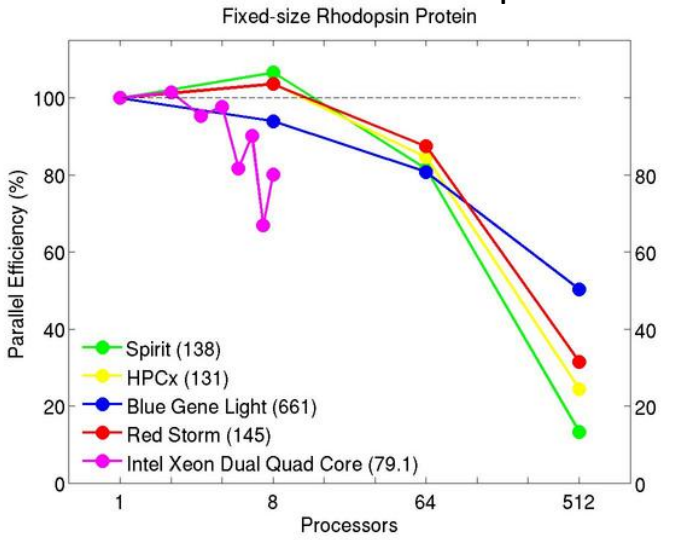


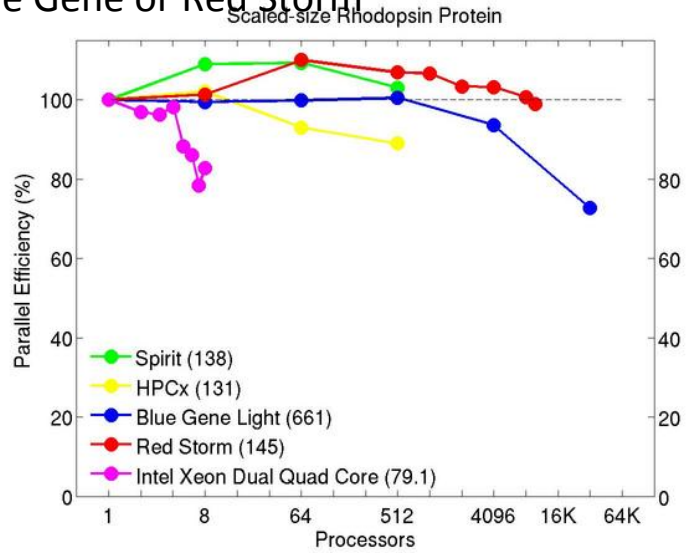
Why Use LAMMPS? performance

Highly scalable – MD is only interesting in *large* systems

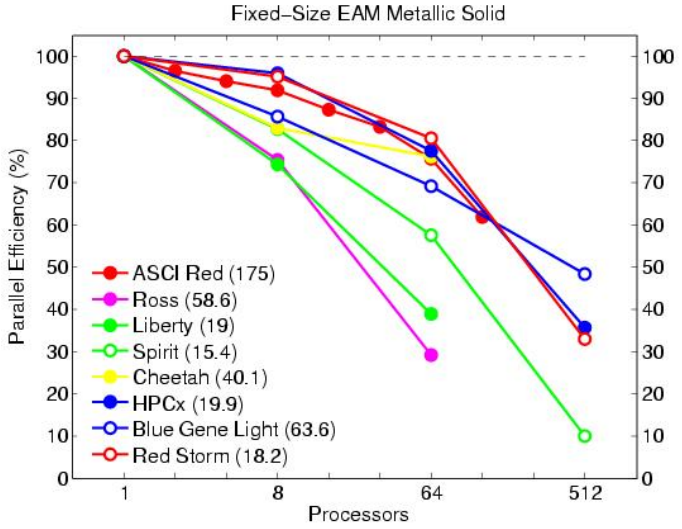
Typical speed: 5E-5 core-sec/atom-step (LJ 1E-6, ReaxFF 1E-3)

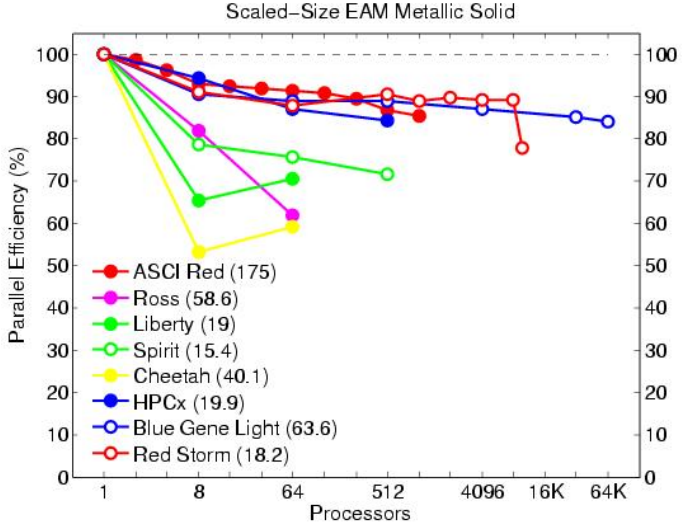
- Fixed-size (32K atoms) & scaled-size (32K/proc) parallel efficiencies
- Billions of atoms on 64K procs of Blue Gene or Red Storm





Protein (rhodopsin) in solvated lipid bilayer



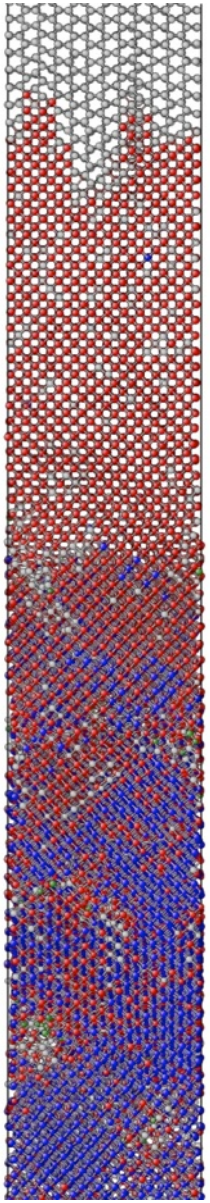


Metallic solid with EAM potential

Why Use LAMMPS? potentials

Has most of the commonly used potentials built in

- **Pairwise** : Lennard-Jones, Buckingham, ...
- **Charged pairwise**: Coulombic, point-dipole
- **Manybody**: EAM, Finnis/Sinclair, modified EAM (MEAM), embedded ion method (EIM), Stillinger-Weber, Tersoff, AI-REBO, ReaxFF, COMB
- **Long-range Coulomb** & dispersion: Ewald, PPPM
- **Water**: TIP3P, TIP4P, SPC
- **Implicit solvent** : hydrodynamic lubrication, Debye
- **Bond, angle & dihedral**: harmonic, FENE, CHARMM..
- **Polymer**: all-atom, united-atom, bead-spring, breakable
- compatibility w/CHARMM, AMBER, OPLS, GROMACS
- electron force field (eFF)
- coarse-grained potentials: DPD, GayBerne, ...
- mesoscopic potentials: granular, peridynamics
- improper potentials: harmonic, cvff, class 2 (COMPASS)



Biomolecules:
CHARMM, AMBER, OPLS, COMPASS (class 2), long-range Coulombics via PPPM, point dipoles, ...

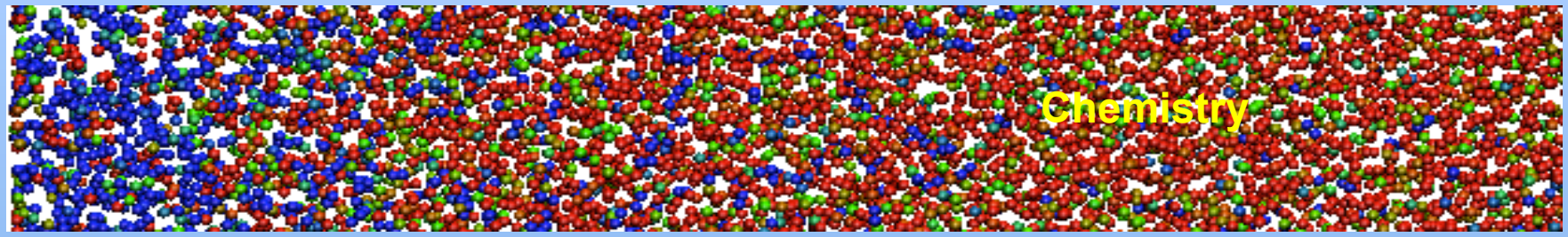
Polymers:
all-atom, united-atom, coarse-grain (bead-spring FENE), bond-breaking, …

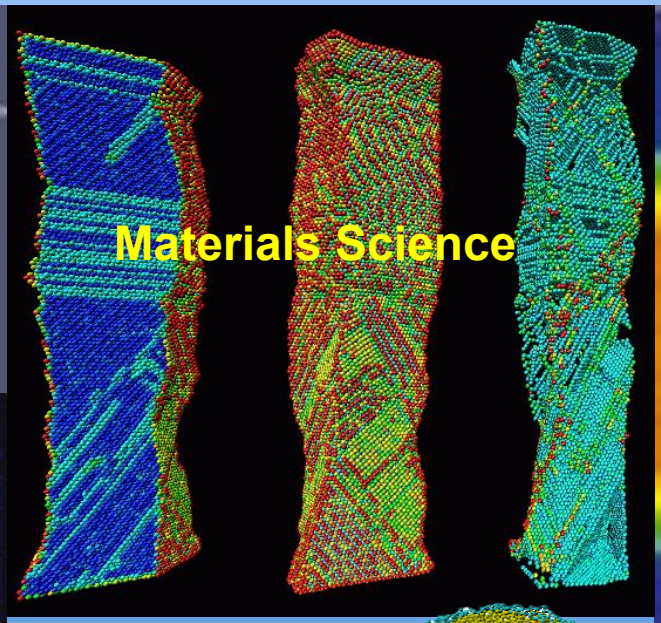
Materials:
EAM and MEAM for metals, Buckingham, Morse, Yukawa, Stillinger-Weber, Tersoff, AI-REBO, ReaxFF, COMB, eFF...

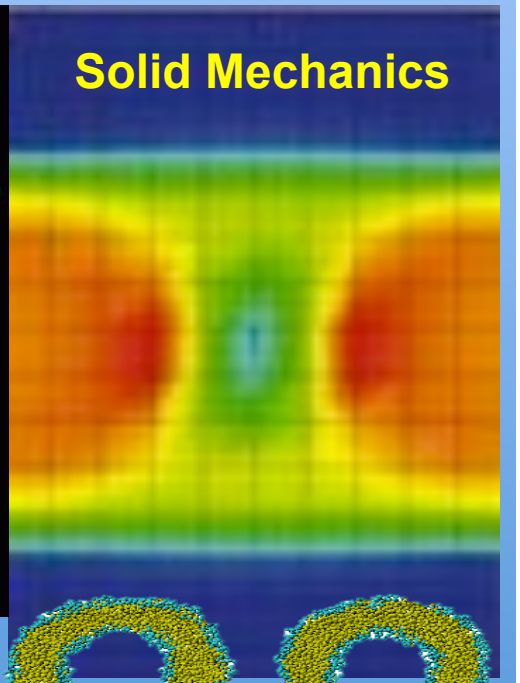
Mesoscale:
granular, DPD, Gay-Berne, colloidal, peri-dynamics, DSMC...

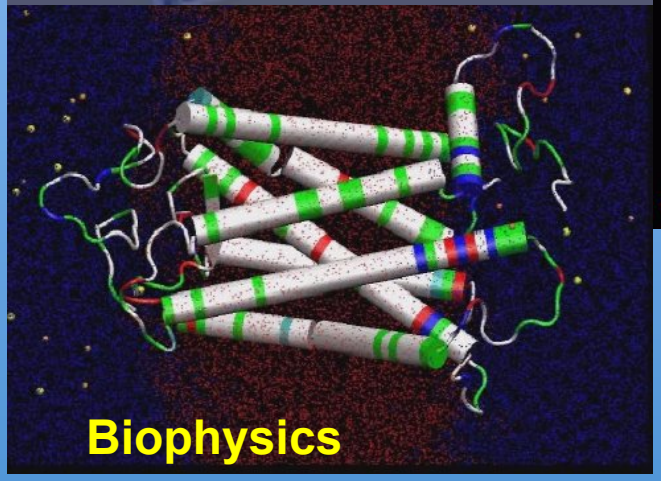
Hybrid:
can use combinations of potentials for hybrid systems: water on metal, polymers/ semiconductor interface,
colloids in solution, …

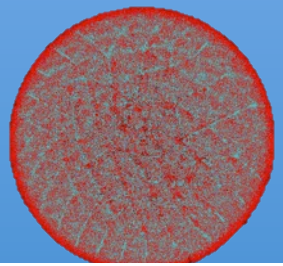
Why Use LAMMPS? versatility

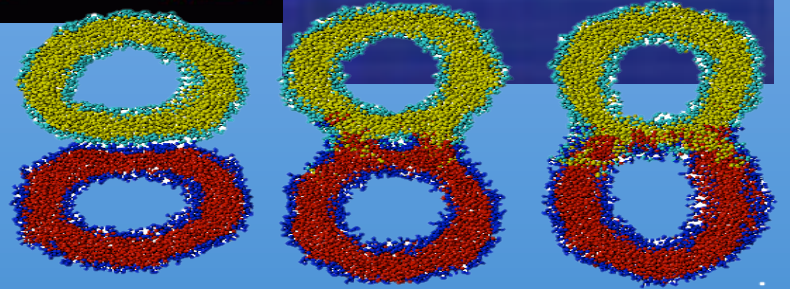












Why Use LAMMPS? modularity

Objects/constructs

atom *styles:* atom, charge, colloid, ellipsoid, point dipole

pair *styles:* LJ, Coulomb, Tersoff, ReaxFF, AI-REBO, COMB, MEAM, EAM, Stillinger-Weber, eg. `pair_reax.h`

fix *styles:* NVE dynamics, Nose-Hoover, Berendsen, Langevin, SLLOD, Indentation, eg. `fix_nve.h`

compute *styles:* temperatures, pressures, per-atom energy, pair correlation function, mean square displacemnts, spatial and time averages e.g. `compute_reduce.h`

All computes works with all fixes work with all pair styles work with all atom styles

Why Use LAMMPS? extensible

80% of code is extensions via ***styles*** (only 35K of 175K lines is core of LAMMPS)

- Easy to add new features via 14 ***styles***
  - –particle types = atom style
  - –force fields = pair, bond, angle, dihedral, improper styles
  - –long range forces = kspace style
  - –energy minimizers = min style
  - –geometric region = region style
  - –output = dump style
  - –time integrator = integrate style
  - –computation/data reduction = compute style (global, per-atom, local)
  - –"fix" = fix style = BC, constraint, time integration, ...
  - –input command = command style, e.g. read_data, velocity, run, …
- Enabled by C++ **inheritance**
  - –virtual parent class for all styles, e.g. pair potentials
  - –defines interface the feature must provide e.g.: compute(), init(),
  - – to add feature: add 2 lines to header file, add files to src dir, re-compile

The feature won't exist if not used, won't conflict with rest of code

BIG PICTURE

F=ma + lots of particles → "statistical mechanics by computation"

- scale
- limitations
- expense

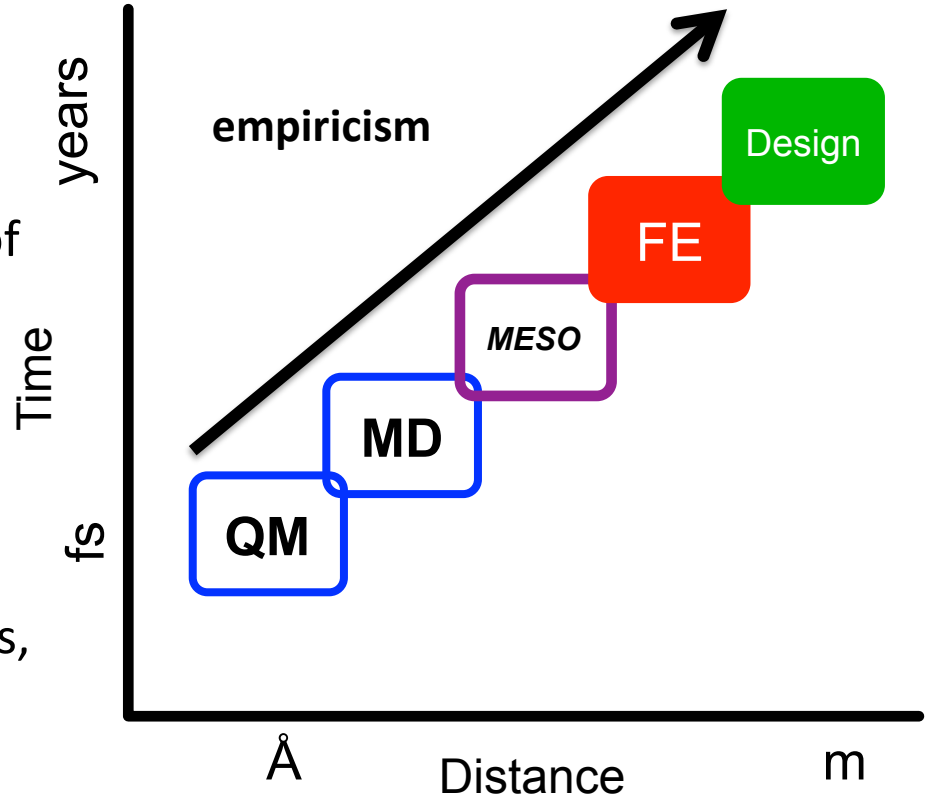
MD is in the middle

Quantum mechanics (QM)

- electronic degrees of freedom,
- chemical reactions
- Schrodinger equation, wave functions
- sub-femtosecond timestep, thousands of atoms, $O(n^3)$

Atomistic models

- molecular statics/dynamics (MD)
- particles, empirical forces, Newton's equations
- femtosecond timestep, millions of atoms, O(N)



Mesoscale

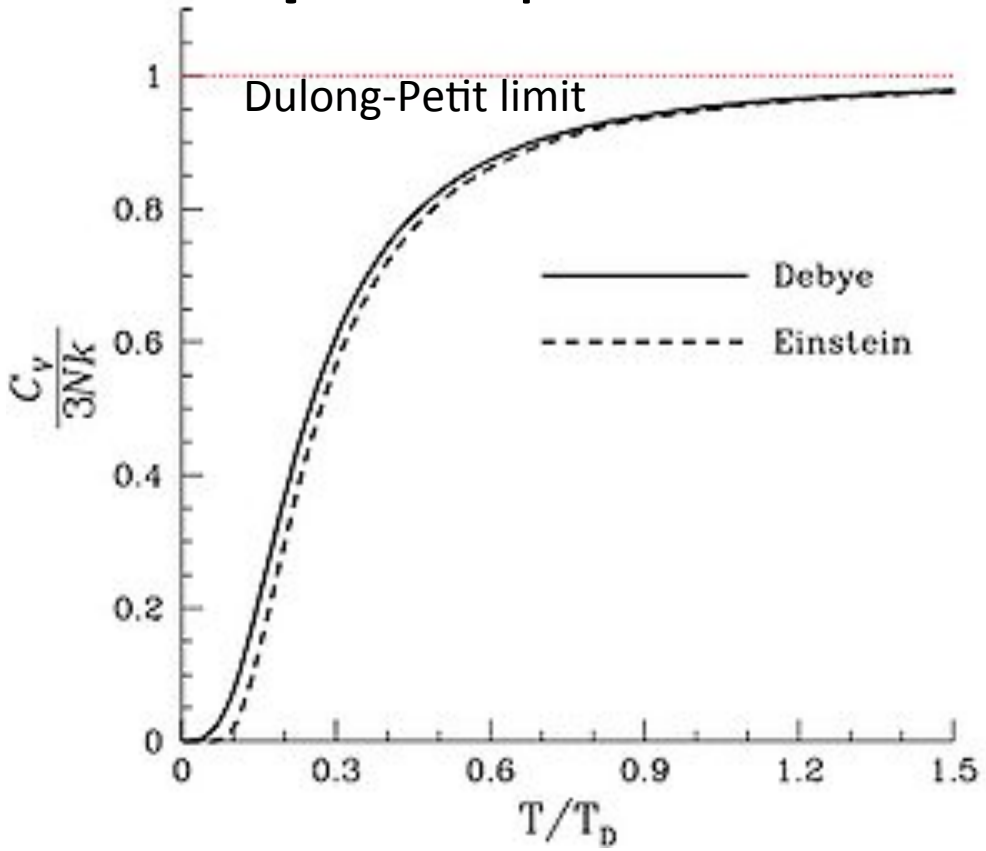
- Monte Carlo (MC)
- Coarse-grained particles: DPD, PeriDynamics, …

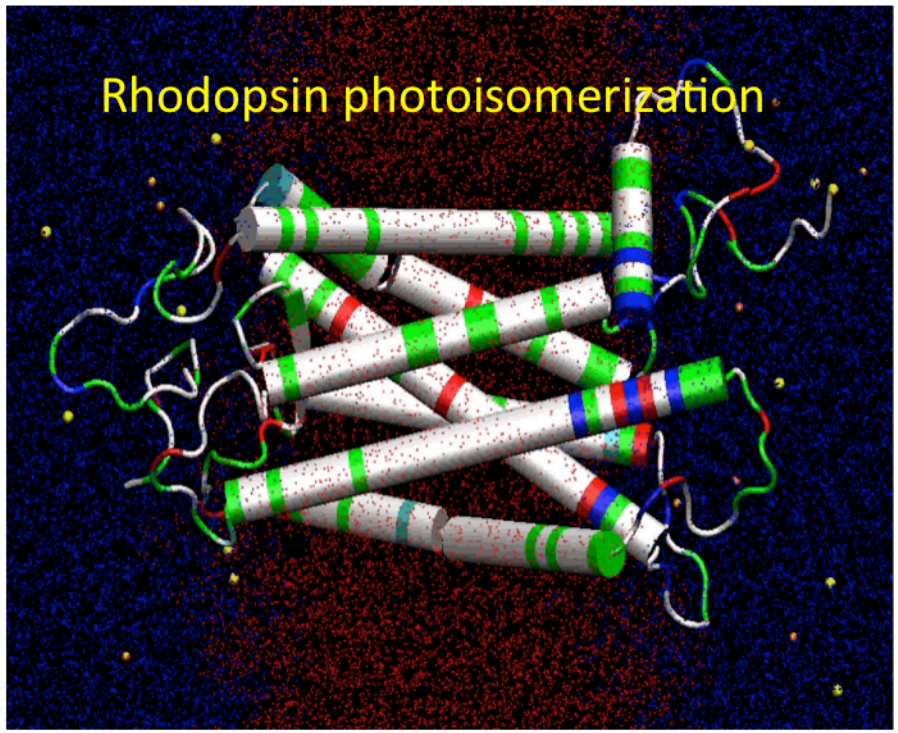
Continuum

- finite elements (FE) or finite difference on grids, SPH/RKPM
- PDEs e.g. elasticity, Navier-Stokes with scales µs to s, µm to m, $O(M^{3/2})$

What do we lose with classical dynamics

- **Quantum effects** e.g. low temperature heat capacity
- **Empirical** potentials aren't always predictive





Time scale

The limits on the maximum timescale of molecular dynamics is its most serious drawback

- Timestep **size limited by atomic vibrations oscillations**

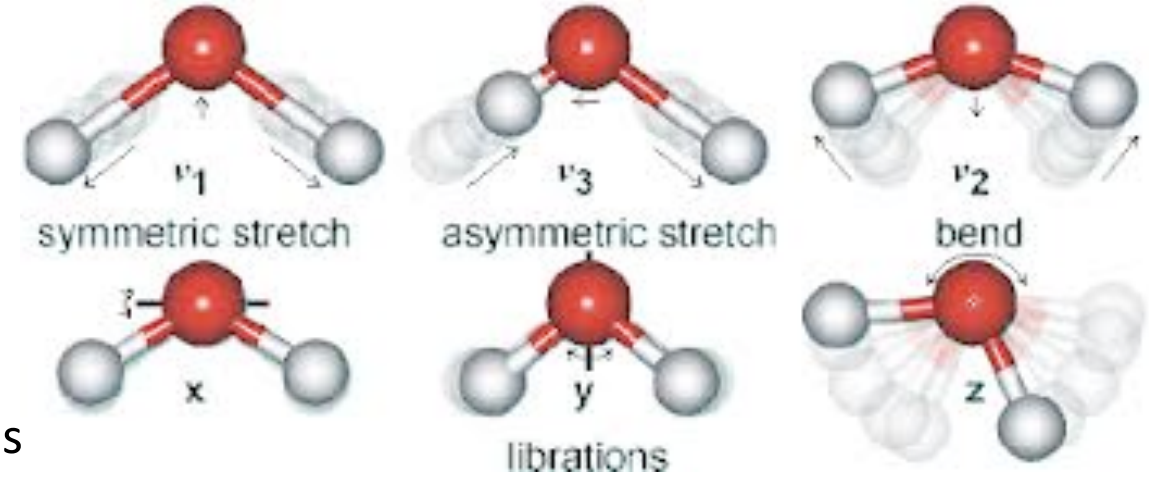
can't skip over it without losing all the interesting physics (so explicit time integration is appropriate)

  - C-H bond = 10 fs → ½ to 1 fs timestep
  - **Debye** (max) **frequency** ~ 10 THz → 2 fs timestep

- A state-of-the-art, massively parallel "long" simulation is nanoseconds to a microsecond of real time

- Reality is usually on a much longer timescale:
  - protein folding (ms to seconds)
  - polymer entanglement (ms and up)
  - glass relaxation (seconds to decades)

Extending time



- **SHAKE** = bond-angle constraints to freeze fast DOF
  - up to 2-3 fs timestep
  - E.g. rigid water, C-H bonds
  - Matrix equations to enforce constraints
- **rRESPA** = hierarchical time stepping, sub-cycle on fast DOF
  - inner loop on bonds (0.5 fs)
  - next loop on angle, torsions (3-4 body forces)
  - next loop on short-range LJ and Coulombic
  - outer loop on long-range Coulombic forces (~ 4 fs)
- **Rigid body time integration** via quaternions
  - treat groups of atom as rigid bodies (portions of polymer or protein)
  - 3N DOF → 6 DOF
  - save computation of internal forces, longer stable timestep
- Temperature accelerated dynamics (TAD), parallel replica dynamics (PRD) and more exotic methods by A Voter, G Henkleman, R Elber

Length-scale

Limited length scale is also serious but there are more standard means of ameliorating it

$$v+ = \Delta t / 2 \, f$$

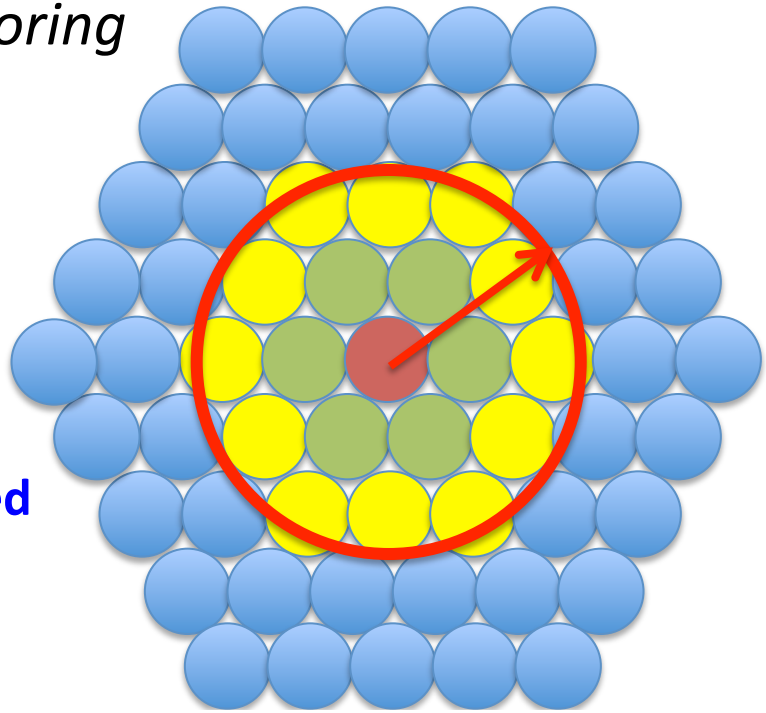
$$x+ = \Delta t \, f$$

$$f = f(x)$$

$$v+ = \Delta t / 2 \, f$$

What is expensive?

- *Not* time integration (simple explicit Verlet scheme)
- Force evaluations which entail determining which atoms are in close proximity aka *neighboring*
- Long-range forces : Coulomb
- Energy minimization since it entails solving non-linear equations



For short-range, non-bonded interactions neighbors are determined using a cutoff radius

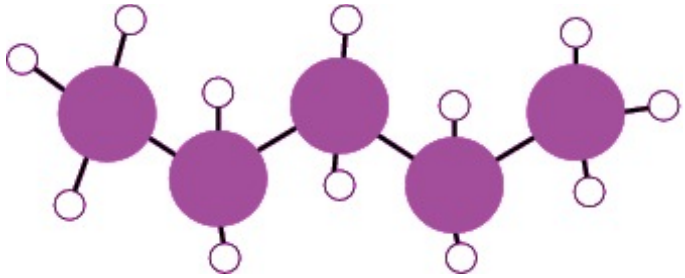
Extending the length-scale via coarse-graining

- **All-atom**:

  Δt = 0.5-1.0 fmsec for C-H

  C-C distance = 1.5 Angs

  cutoff = 10 Angs



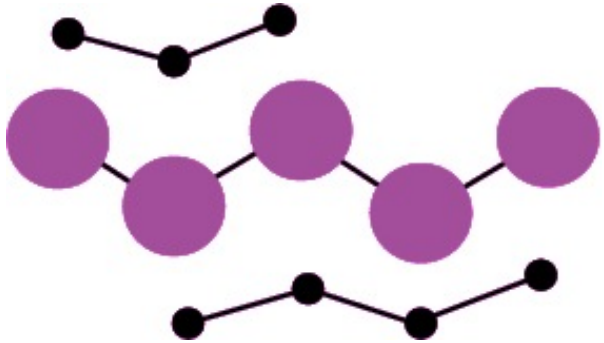
- **United-atom:**

  ~ 9x less interactions

  Δt = 1.0-2.0 fs for C-C

  cutoff = 10 A

  20-30x savings over all-atom

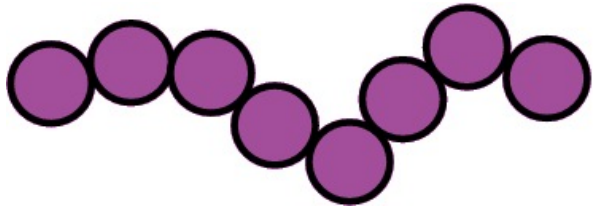


- **Bead-Spring:**

  2-3 C per bead

  $\Delta t \leftrightarrow$ fs mapping is T-dependent

  $2^{1/6}\,\sigma$ cutoff → ~ 8x interactions

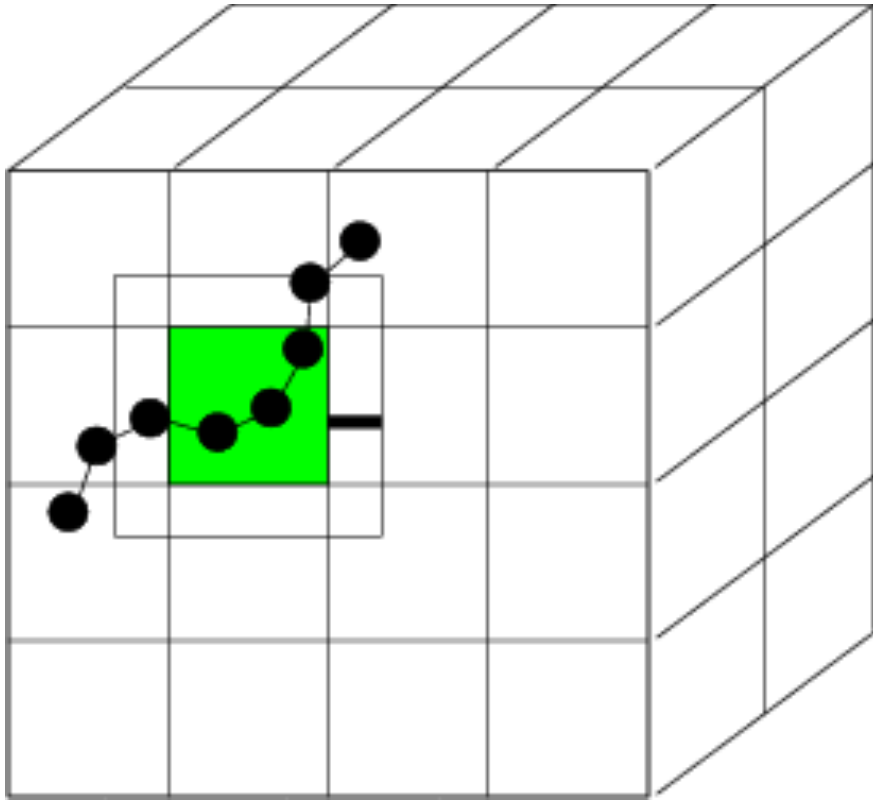


Extending the length-scale
via parallelism

- MD (*with short-range interactions*) is **inherently parallel**
  - forces on each atom are local and can be computed simultaneously
  - positions and velocities can be updated simultaneously
- Most MD codes use a **distributed-memory** message-passing paradigm (MPI)
  - Computation scales as **N**, *the number of atoms*
    - ideally would scale as N/P in parallel, where **P** *is the number of processors*
  - Distributed:
    - atoms communication = scales as N
    - forces communication = scales as N/sqrt(P)
    - space communication = scales as N/P or $(N/P)^{2/3}$

Spatial-decomposition

- Physical domain divided into boxes, one per processor
- Each processor computes forces on atoms in its box using communicated information from nearby processor
- Atoms "carry along" molecular topology as they migrate to new processors
- Communication via nearest-neighbor 6-way stencil
- Optimal scaling N/P *if load-balanced*

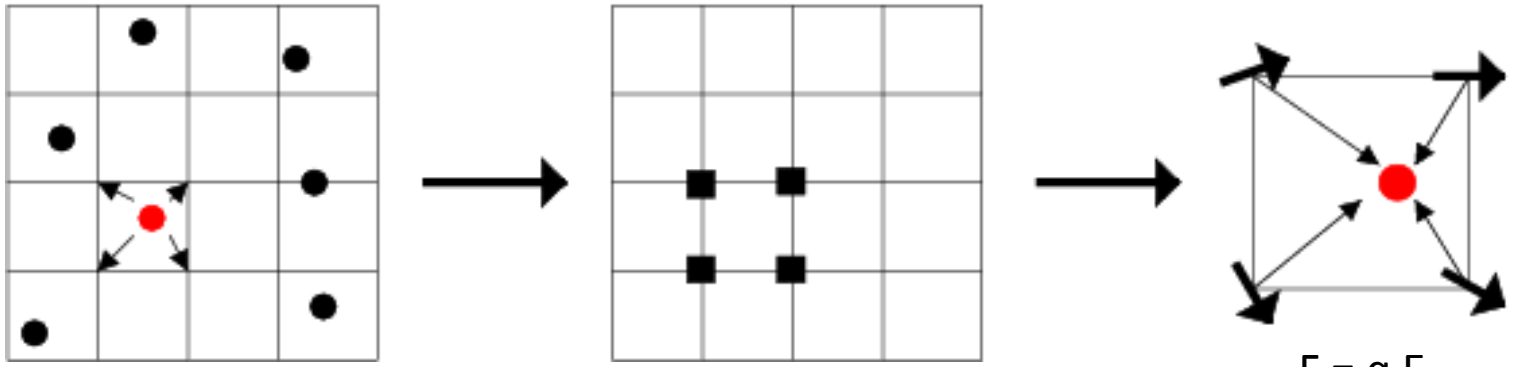


➢ **Computation** ~ N/P
➢ **Communicatio**n ~$(N/P)^{2/3}$
➢ **Memory** ~ N/P

Particle-mesh methods

for Coulomb interactions

- Coulomb forces decay as 1/r so can't be truncated
- Particle-mesh methods:
  - **decompose** into **short-range** (within the cutoff) and **long-range** (outside the cutoff) contributions
  - compute short-range via direct sums of pairwise forces
  - long-range:
    - interpolate atomic charge **density ρ** to 3d mesh
    - solve Poisson's equation $\nabla^2\phi = \rho$ on mesh (4 FFTs)
    - Interpolate electric field $E = -\nabla\phi$ back to atoms to effect forces $f = qE$
- FFTs scale as N logN if cutoff is held fixed



Energy minimization

Unlike dynamics $m\ddot{x} = f(x)$

Energy minimization requires solving nonlinear equations

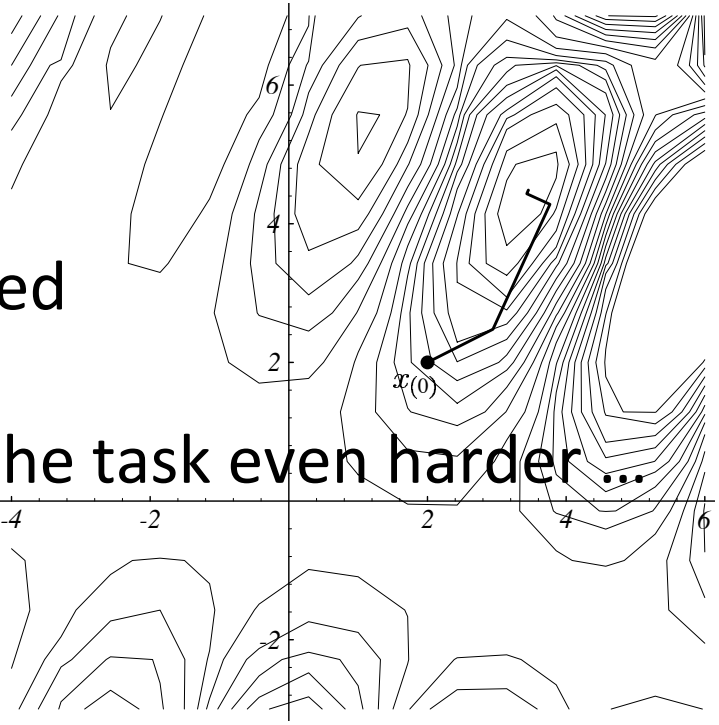
$$0 = f(x) \Leftrightarrow -\partial_x \Phi = 0 \Leftrightarrow \min_x \Phi$$

For small systems Newton's

Method has quadratic convergence

But for large systems methods like (nonlinear) conjugate gradient that have linear convergence are employed

Multiple neighboring minima make the task even harder …



SYNTAX

```
echo both
units    metal
dimension  3
boundary  p p p
atom_style     atomic
read_data propane.data
pair_style     airebo 1.5 1 0
pair_coeff     *  *  CH.airebo C H
mass           1 12.0107
mass           2 1.00794
velocity all create 300.0 4928459 dist gaussian
fix NVE all nve
timestep 0.001
thermo  10
dump IMAGE all image 10 propane_*.ppm type type
run  1000
```

Top six LAMMPS commands!

1. **Fix**: "*everything is a fix*" (operations that change the state of the atoms e.g. integrators, thermostats)
2. **Compute**: "*everything that isn't a fix is a compute*" (data reduction e.g. center-of-mass, average)
3. **Pair_style**: "pair" is a euphemism for potential
4. **Variable**: user variables for parameterizing the dynamics
5. **Thermo_style**: text output of observable: temperature, etc
6. **Dump**: output of positions, velocities, charges

fix command

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

A "fix" is any operation that is applied to/operates on the system during time-stepping or minimization.

Examples:

- Time integration to updateof atom positions and velocities
- Temperature control
- Enforcing constraints and boundary conditions

Syntax: fix ID group-ID style args

- ID = user-assigned name for the fix
- group-ID = ID of the group of atoms to apply the fix to
- style = one of a long list of possible style names
- args = arguments used by a particular style

Fix NVT all nvt temp 300.0 300.0 0.01

compute command

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

A “compute" is a diagnostic calculated for a group of atoms to extract quantitative information from a simulation while it is running.

Examples:

- Temperature
- Center of mass

three styles of quantities: global, per-atom, or local to processor

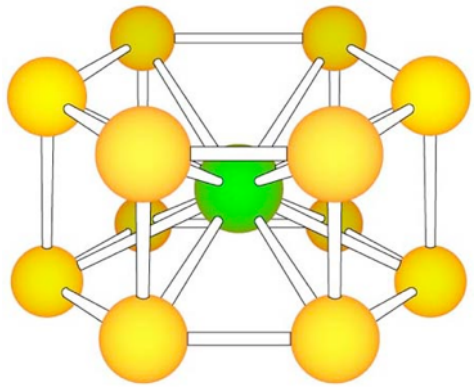
Syntax: compute ID group-ID style args

- ID = user-assigned name for the computation
- group-ID = ID of the group of atoms to perform the computation on
- style = one of a list of possible style names (see below)
- args = arguments used by a particular style

***compute AVE IONS ave/time* 100 1 100 *c_T vx* # stample every step for 100 steps**

pair_style command

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



A “pair_style" is a type of empirical potential used in “non-bonded” interactions based on proximity

A cutoff value is usually required

Syntax: pair_style style args

- style = one of the styles from the list below
- args = arguments used by a particular style

Examples:

- *pair_style lj/cut **2.5***
- *pair_style eam/alloy*
- *pair_style hybrid lj/charmm/coul/long **10.0** eam*
- *pair_style table linear 1000*
- *pair_style none*

variable command

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

Assigns a name to a value or values to be used elsewhere in the input script for parameterization or output

- For variable styles **loop & index** that store multiple strings, the next command can be used to increment which string is assigned to the variable.
- Variables of style ***equal*** store a formula which when evaluated produces a single numeric value which can be output either directly (e.q. print command) or as part of thermodynamic output (via the thermo_style command), or used as input to an averaging fix (e.q.fix ave/time command).
- Variables of style ***atom*** store a formula which when evaluated produces one numeric value per atom which can be output to a dump file (via the dump custom command) or used as input to an averaging fix (e.g. the fix ave/spatial and fix ave/atom commands).

Syntax: variable name style args ...

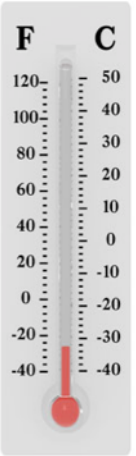
- name = name of variable to define
- style = *delete* or *index* or *loop* or *world* or *universe* or *uloop* or *string* or *equal* or *atom*

Examples:

- ***variable x index run1 run2 run3 run4 run5 run6 run7 run8***
- ***variable i loop $n***
- ***variable T equal c_myTemp***
- ***variable beta equal 1.0/${kB}/$T***

thermo_style command

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



Set the style and content for printing data to the screen & log file.

Syntax: thermo_style style args

- style = *one* or *multi* or *custom*
- args = list of arguments for a particular style

Examples:

- *thermo_style multi*
- *thermo_style custom* ***step temp pe etotal press vol***
- *thermo_style custom step temp etotal c_myTemp v_abc*

dump command

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

Output a snapshot of various atom quantities to one or more files every N timesteps

Syntax:

- dump ID group-ID style N file args ID = user-assigned name for the dump
- group-ID = ID of the group of atoms to be dumped
- style = *atom* or *cfg* or *dcd* or *xtc* or *xyz* or *local* or *custom*
- N = dump every this many timesteps
- file = name of file to write dump info to
- args = list of arguments for a particular style *atom*

Examples:

- ***dump ATOMS all atom 100 dump.atm***
- ***dump PERSTEP all custom 100 dump.myforce.* id type x y vx fx***
- ***dump PERPROC fluid custom 100 dump.%.myforce id type c_myF[3] v_ke***

How do I **fix a group of atoms so they don't move**?

- Leave them out of the integrator's group

fix NVE MOBILE_GROUP nve

- OR set the forces

fix F0 FIXED_GROUP setforce NULL 0.0 0.0

And velocities to zero

velocity FIXED_GROUP set NULL 0.0 0.0

How do I **set up a finite temperature system**?

1. Assign velocities to atoms

velocity all create 600.0 1234 dist gaussian

2. Allow the system to come to equilibrium

fix RESCALE all temp/rescale 1 300. 300. 0.02 1.0

run 1000

3. Apply a thermostat and further relax

fix NVT all nvt temp 300.0 300.0 100.0

run 1000

Oddities

- **Units are not self consistent**, the m, L, t-scales do not determine the energy scale, e.g.
  - mass = grams/mole
  - distance = Angstroms
  - time = picoseconds
  - energy = eV

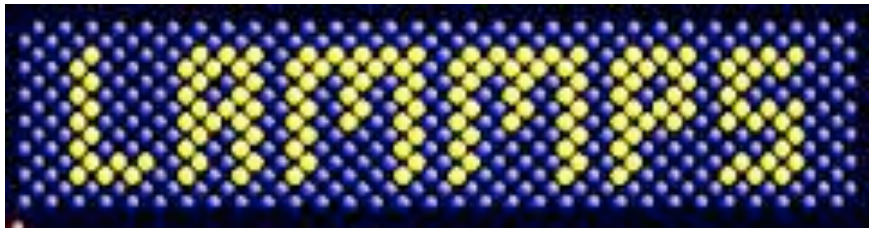
http://lammps.sandia.gov/doc/units.html

- **Error messages need interpreting**

http://lammps.sandia.gov/doc/Section_errors.html#err_3

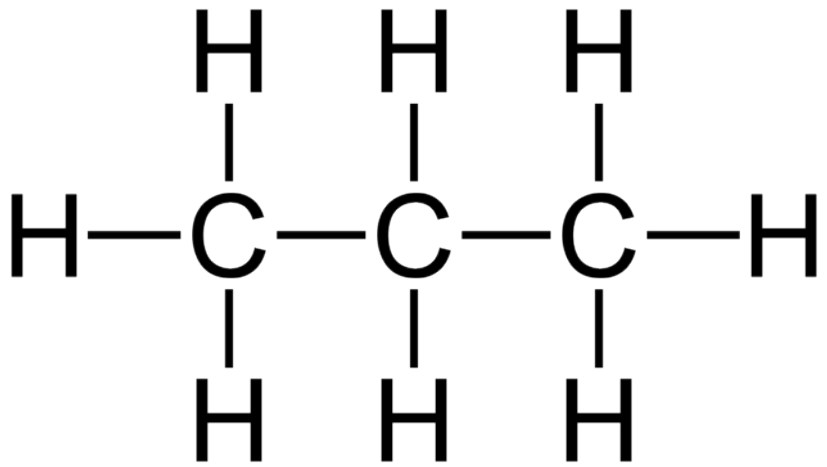
Angle atom missing in delete_bonds

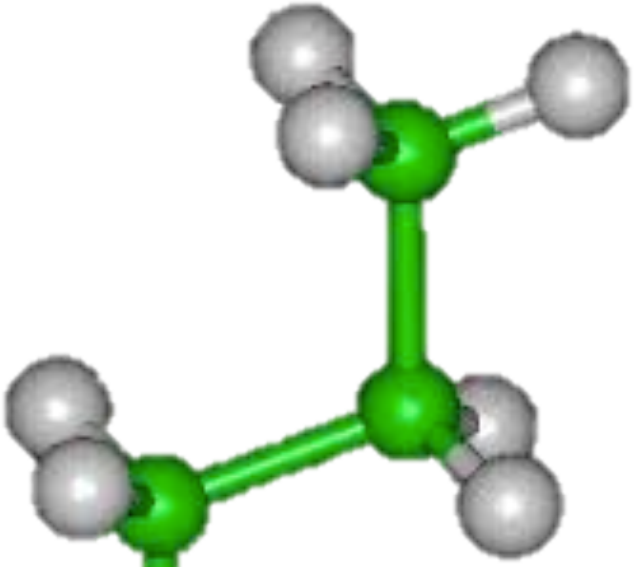
➔ The delete_bonds command cannot find one or more atoms in a particular angle on a particular processor. The pairwise cutoff is too short or the atoms are too far apart to make a valid angle.

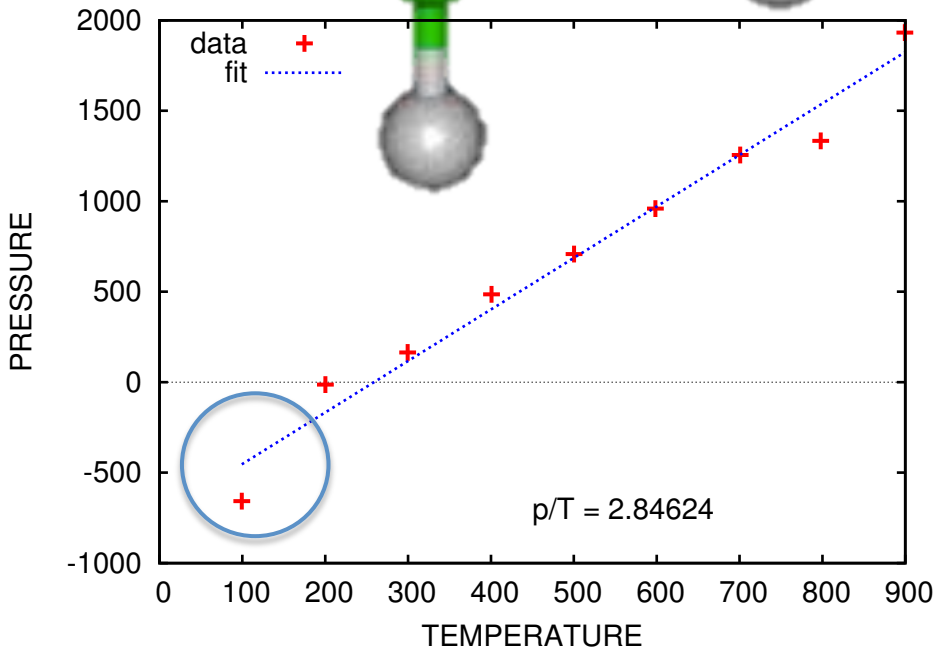


"Homework"

- Create a molecule, e.g. propane (3 carbons and 8 hydrogens
- Select potentials/bonds
- Run and visualize dynamics



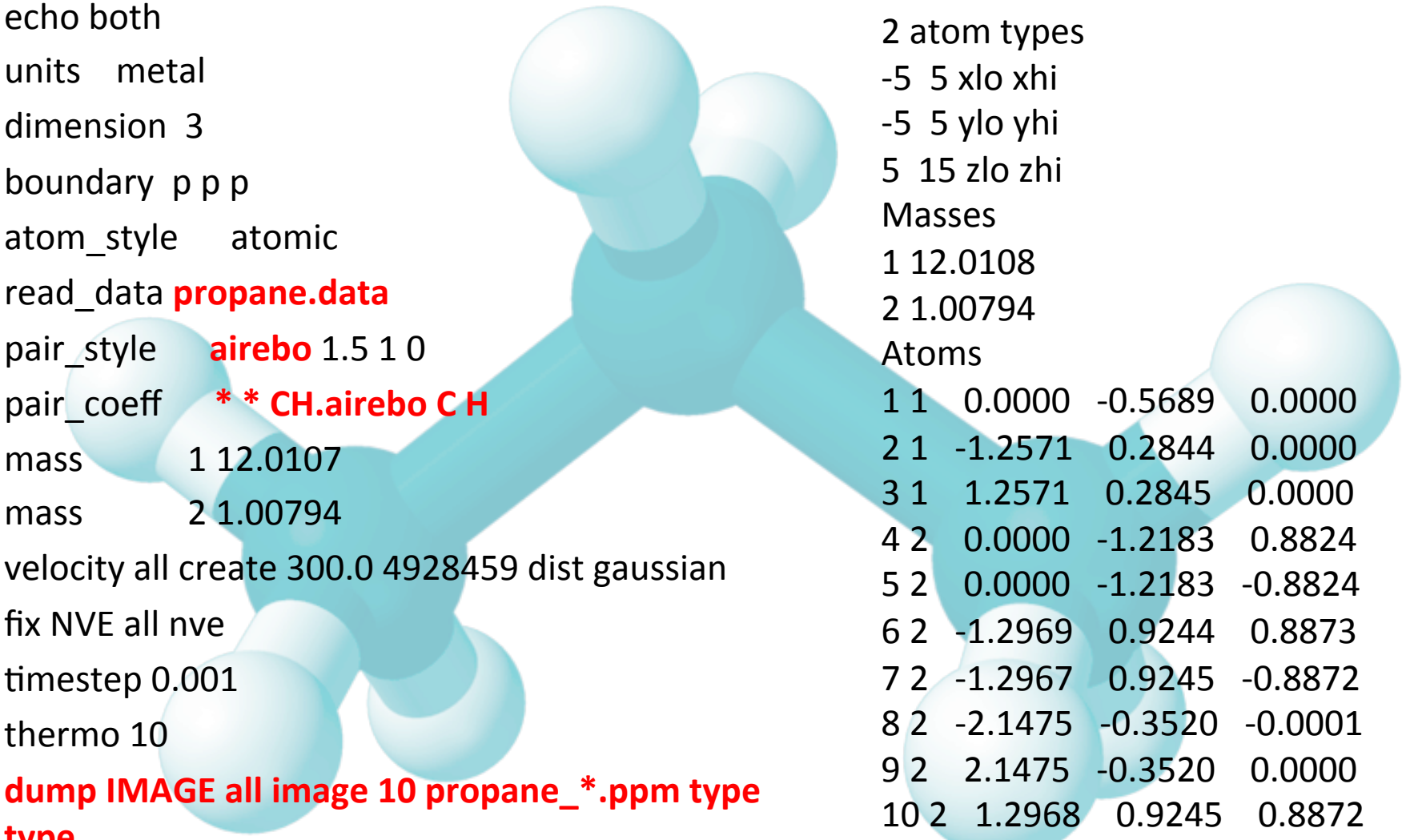




INPUT

Input file

```
echo both
units    metal
dimension  3
boundary  p p p
atom_style      atomic
read_data propane.data
pair_style      airebo 1.5 1 0
pair_coeff      * * CH.airebo C H
mass            1 12.0107
mass            2 1.00794
velocity all create 300.0 4928459 dist gaussian
fix NVE all nve
timestep 0.001
thermo 10
dump IMAGE all image 10 propane_*.ppm type type
run 1000
```

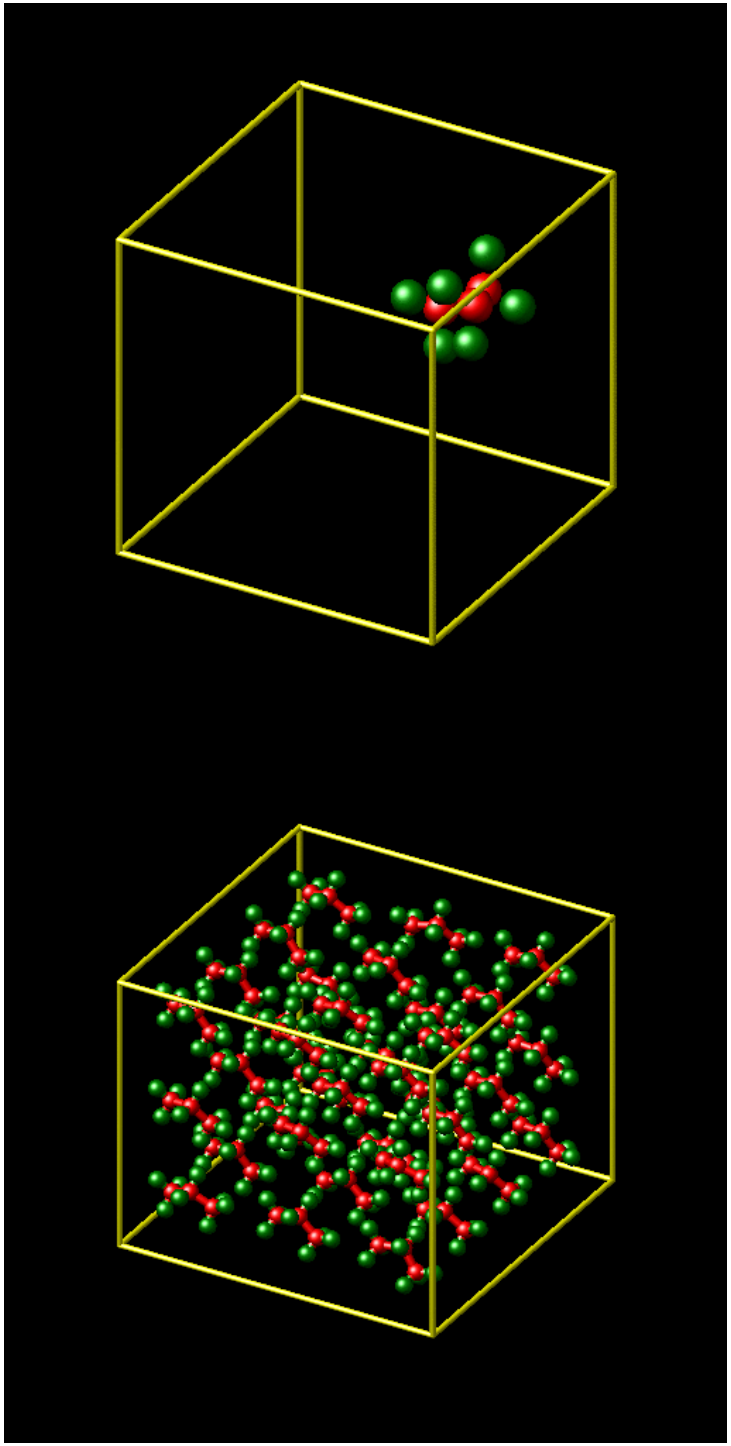


data file

```
11  atoms
2 atom types
-5  5 xlo xhi
-5  5 ylo yhi
5  15 zlo zhi
Masses
1 12.0108
2 1.00794
Atoms
1 1    0.0000   -0.5689    0.0000
2 1   -1.2571    0.2844    0.0000
3 1    1.2571    0.2845    0.0000
4 2    0.0000   -1.2183    0.8824
5 2    0.0000   -1.2183   -0.8824
6 2   -1.2969    0.9244    0.8873
7 2   -1.2967    0.9245   -0.8872
8 2   -2.1475   -0.3520   -0.0001
9 2    2.1475   -0.3520    0.0000
10 2   1.2968    0.9245    0.8872
11 2   1.2968    0.9245   -0.8872
```

BONUS

- Create a *bonded* molecule, do the dynamics visibly differ?
- Create a small ensemble of molecules and see if the thermodynamic properties are different from the single molecule system e.g. density as a function of temperature or pressure?
- Does this model of propane act like an ideal gas? P = (nR/V) T
- Compare the thermodynamic properties of an ensemble of bonded vs non-bonded molecules.



Lecture 6

Week 5 : **LAMMPS tutorial**

- Overview of LAMMPS as a molecular simulation tool
- Input file elements: variables, fixes, computes, etc.
- Units
- Output options
- Tools, potentials, etc.

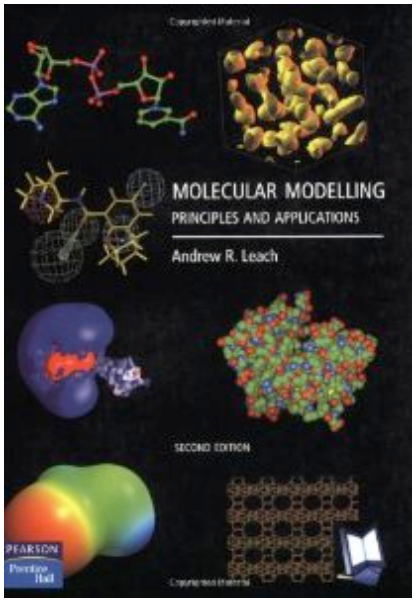
Week 6: **Molecular Statics**

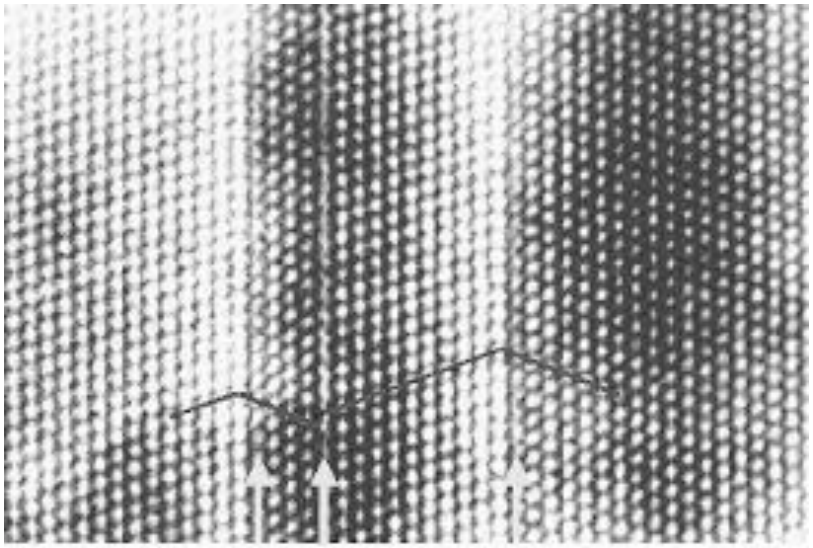
- Boundary conditions
- Energy minimization, e.g. the conjugate gradient algorithm
- Energy surface exploration, e.g. Nudged Elastic Band, transition states

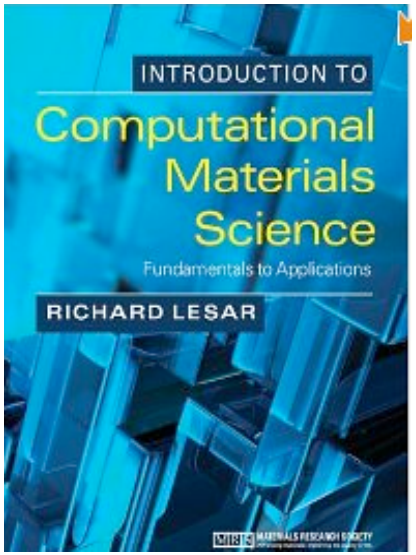
Homework : Minimization of a point defect.

Reading Suggestions for Lec. 6

- Chapter 5 & 6 of LeSar
- Chapter 5 of Leach
- http://en.wikipedia.org/wiki/Molecular_dynamics
- http://lammps.sandia.gov/







How to get help with LAMMPS

1. User's **Manual**:
http://lammps.sandia.gov/doc/Manual.html
http://lammps.sandia.gov/doc/Section_commands.html#3_5

2. **Tutorials**:
https://icme.hpc.msstate.edu/mediawiki/index.php/LAMMPS_tutorials

3. **Search**: include "lammps-users" as keyword

4. Send e-mail to the **user's list**:
http://lammps.sandia.gov/mail.html

5. **Contact** LAMMPS developers:
http://lammps.sandia.gov/authors.html
Steve Plimpton, sjplimp@sandia.gov
Aidan Thompson, athomps@sandia.gov
Paul Crozier, pscrozi@sandia.gov