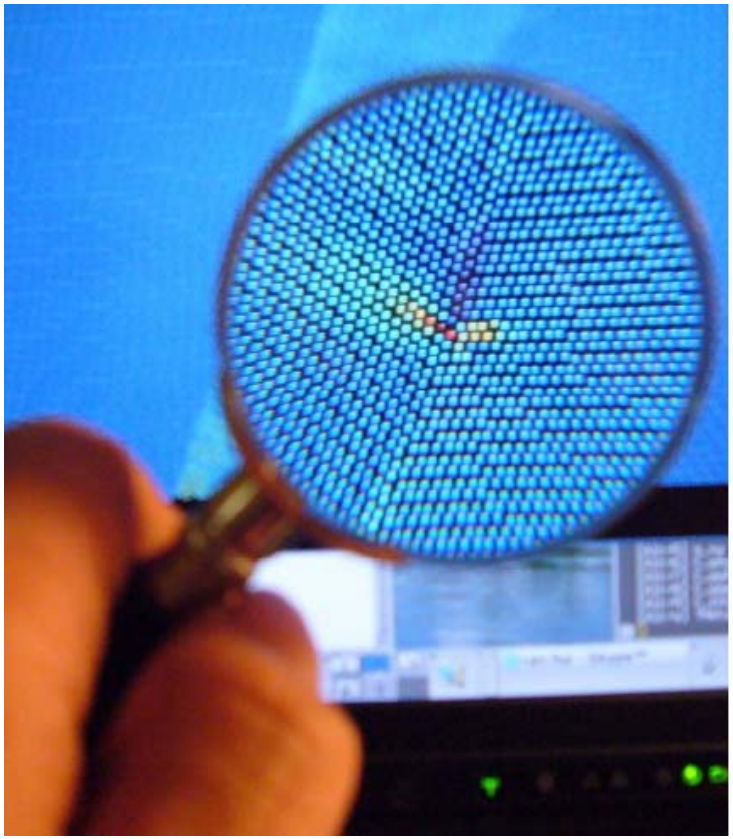
MD example: calculate the melting temperature

Dr. Xining Zang,
ME 138/ME238

Prof. Liwei Lin
Professor, Dept. of Mechanical Engineering Co-Director, Berkeley Sensor and Actuator Center
The University of California, Berkeley, CA94720
e-mail: lwlin@me.berkeley.edu
http://www.me.berkeley.edu/~lwlin

MD simulation: observing and measuring atoms in virtual lab



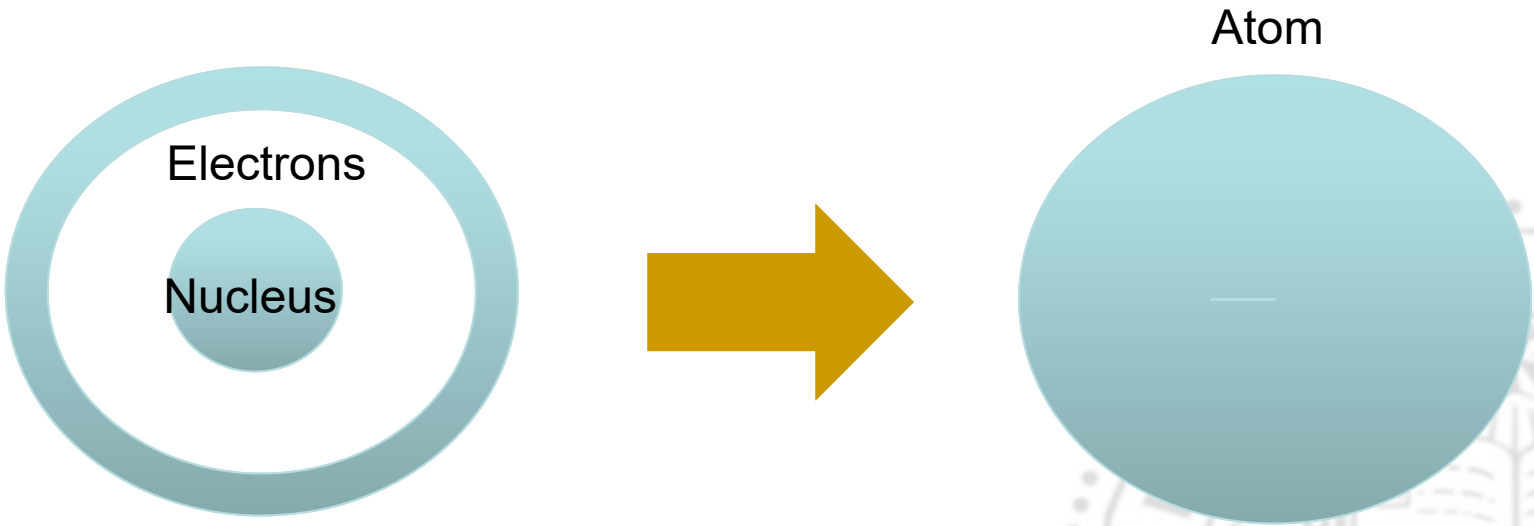
- Interatomic potential
- Initial condition
- Boundary condition
- Integrator (solver for New Second Law)
- Atomic/particle trajectories

Molecular Dynamics

- Using the information gained from ***r, v*** and application of the forces:
  - Initial positions are advanced towards a lower energy state.
    - Through some time step delta t
  - New positions are obtained.
  - New velocities are obtained
  - REPEAT.
- This is repeated as many times as is needed to obtained equilibrium.

Atomic model in MD

- Atoms are represented as spheres.
  - Point mass
- This means that the electron's role is neglected.
  - As is the electronic wave function.



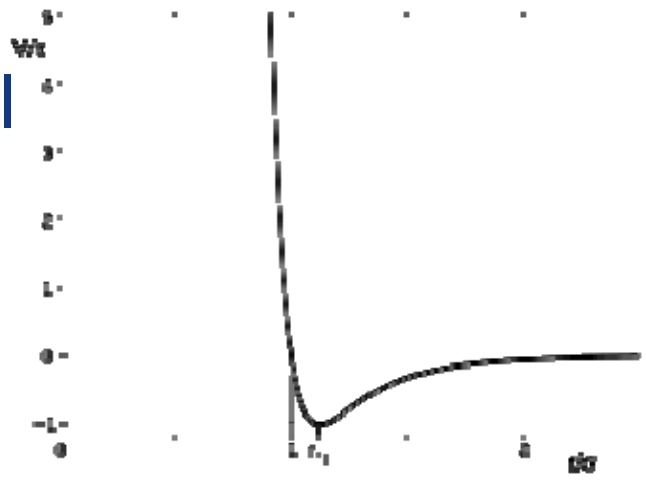
Thus the calculations become relatively simple.

Potential in MD

- Empirical Potentials
- Lennard-Jones (LJ) potential

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

$$V(r) = \frac{A}{r^{12}} - \frac{B}{r^{6}}$$

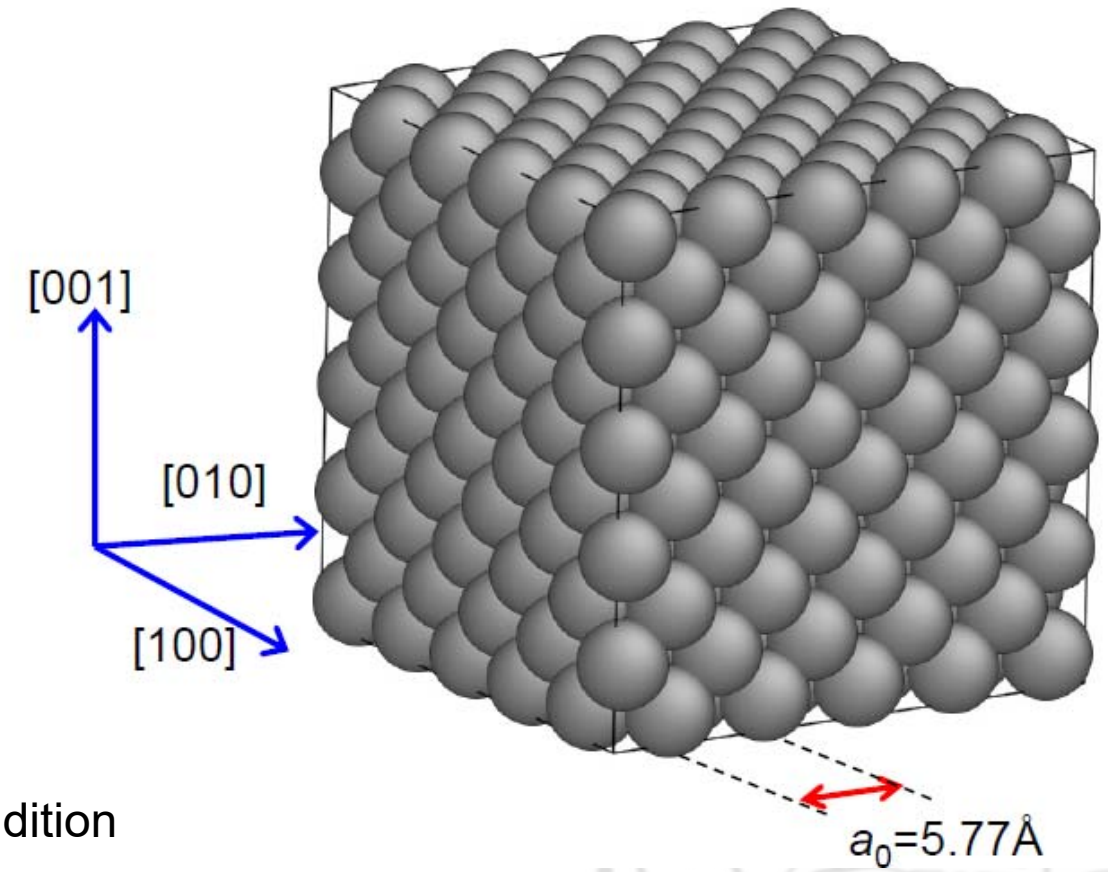


- Many-body potentials
- Semi-empirical potentials
- Polarizable potentials
- Potentials in ab initio methods
- Etc

Initial state: Example FCC (Argon)

N = 4x(5x5x5) = 500

T = 0 K

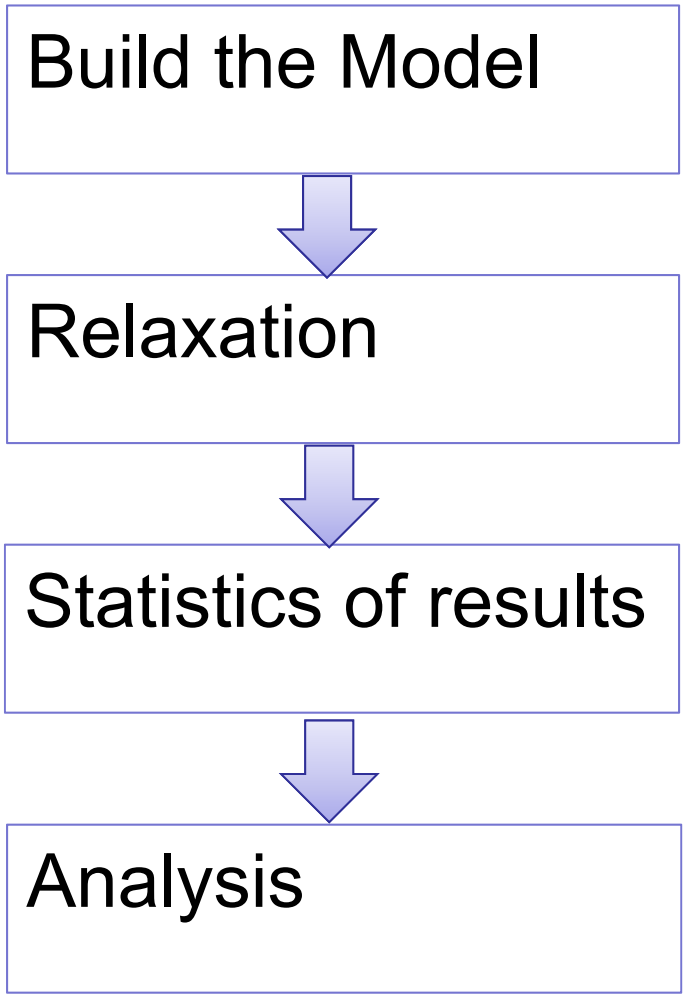


3D periodic boundary condition

MD simulation

Model system	Ar
Initial condition	N=500 fcc cubic box （supercell） T=10K， T=80K.
Boundary condition	Periodic Boundary Condition (PBCs)
Potential	L-J potential
Ensemble (integrator)	NVT: N (number), V (volume) , T(temperature)
Calculation	Reach equilibrium Observe the thermal fluctuation Study the solid to liquid transition

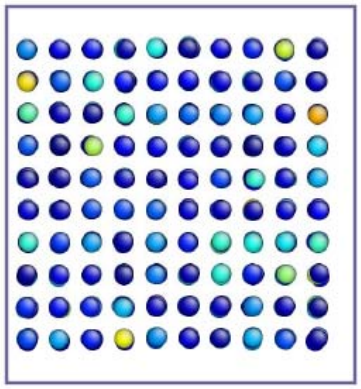
How MD works for melting



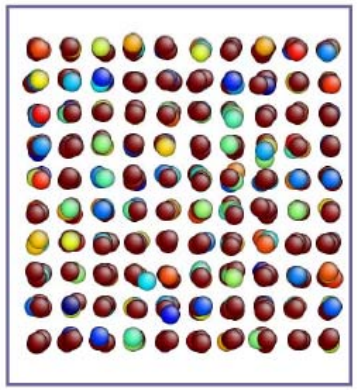
```
# 2d Lennard-Jones melt

units           lj
atom_style      atomic

boundary        p p p

lattice         fcc 1.073
region          box block 0 8 0 8 0 5
create_box      1 box
create_atoms    1 box

mass            1 1.0
velocity        all create 0.01 872877
timestep        0.01

dump            1 all xyz 1000     melt.xyz

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5

neighbor        0.3 bin
neigh_modify    every 10 delay 0 check yes

thermo          1000

fix             1 all npt 0.01 0.01 1.00 xyz 0.0 0.0 1.0 drag 0.2
run             50000
unfix           1

thermo          1000

fix             1 all npt 0.01 0.85 2.0 xyz 0.0 0.0 1.0 drag 0.2

run             1000000
```

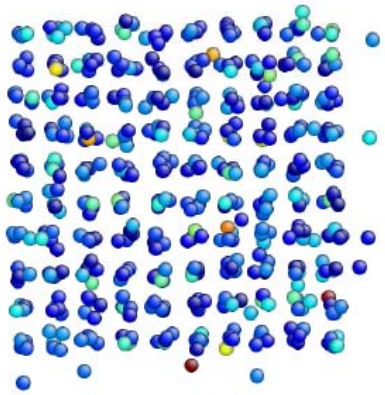
Visualization of melting process

T = 10 K



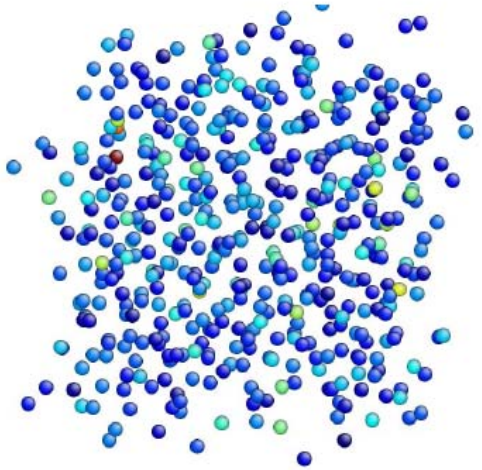
T = 80 K





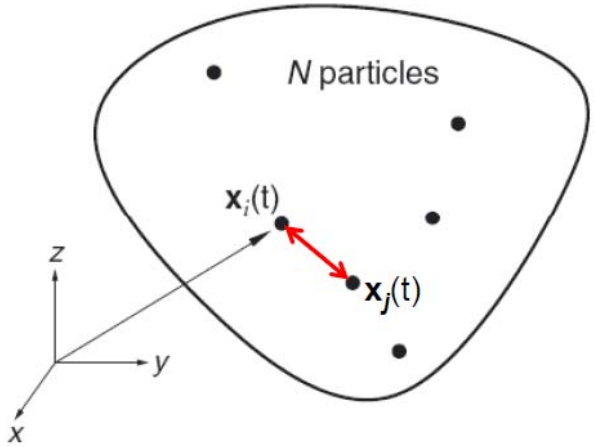
Hot crystal

Melt



Liquid

Calculate the force



$$r_{ij} = \left|\mathbf{x}_{ij}(t)\right|$$

$$\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$$

$$\hat{\mathbf{x}}_{ij} \equiv \frac{\mathbf{x}_{ij}}{r_{ij}}$$

unite vector

Force on atom i $$\mathbf{f}_i = -\sum_{j\neq i} \frac{\partial V(r_{ij})}{\partial \mathbf{x}_i} = \sum_{j\neq i} \left(-\frac{\partial V(r)}{\partial r}\bigg|_{r=r_{ij}} \right) \hat{\mathbf{x}}_{ij}$$

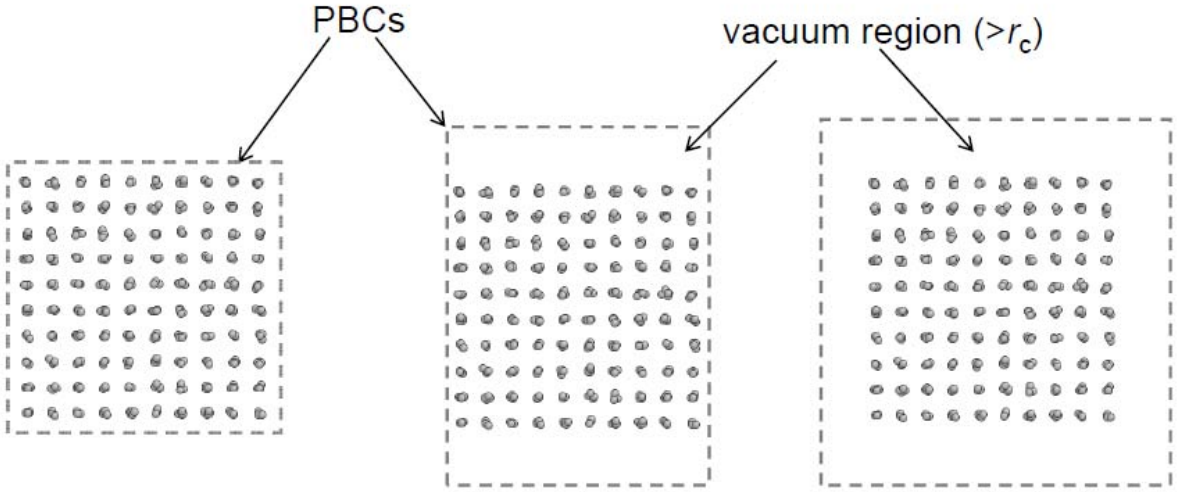
$$= \sum_{j\neq i} \left(-\frac{1}{r}\frac{\partial V(r)}{\partial r}\bigg|_{r=r_{ij}} \right) \mathbf{x}_{ij}$$

Force between atom i and j $$\mathbf{f}_{ij} \equiv \left(-\frac{1}{r}\frac{\partial V(r)}{\partial r}\bigg|_{r=r_{ij}} \right) \mathbf{x}_{ij}$$

Since $\mathbf{f}_i = \sum_{j\neq i} \mathbf{f}_{ij}$ so $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$

PBC

Supercell: one can always use 3D PBCs in MD simulations



infinite system

$$\mathbf{x}_i + l\mathbf{h}_1 + m\mathbf{h}_2 + n\mathbf{h}_3$$
$$(l, m, n = -\infty, \infty)$$

infinite film/slab/wire

Equivalent to 1D or 2D PBCs, e.g.

$$\mathbf{x}_i + l\mathbf{h}_1 + m\mathbf{h}_2$$
$$(l, m = -\infty, \infty)$$

finite system

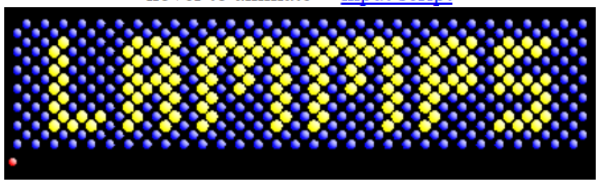
(no PBC)

LAMMPS

LAMMPS Molecular Dynamics Simulator

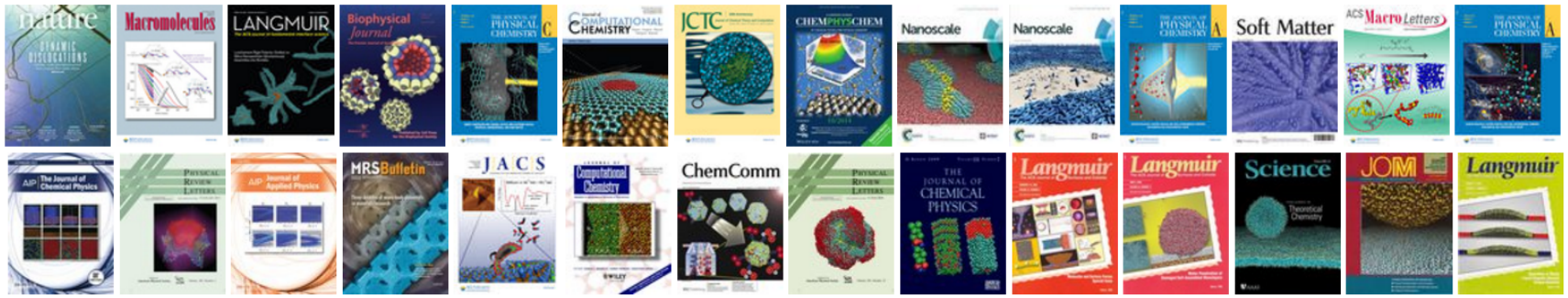
***lamp**: a device that generates light, heat, or therapeutic radiation; something that illumines the mind or soul* -- www.dictionary.com

hover to animate -- input script



physical analog (start at 3:25) & explanation

Big Picture	Code	Documentation	Results	Related Tools	Context	User Support
Features	Download	Manual	Publications	Pre/Post processing	Authors	Mail list
Non-features	GitHub	Developer guide	Pictures	Pizza.py Toolkit	History	Workshops
Packages	SourceForge	Tutorials	Movies	Offsite LAMMPS packages & tools	Funding	User scripts and HowTos
FAQ	Latest features & bug fixes	MD to LAMMPS glossary	Benchmarks	Visualization	Open source	Contribute to LAMMPS
Wish list	Report bugs & request features	Commands	Citing LAMMPS	Related modeling codes	.	.



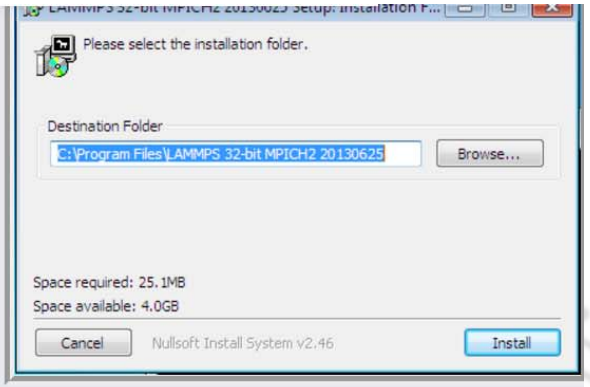
http://lammps.sandia.gov/

LAMMPS installation

http://rpm.lammps.org/windows.html

http://rpm.lammps.org/windows/64bit/

- Latest versions
- 32-bit Windows download area with all available installer versions
- 64-bit Windows download area with all available installer versions

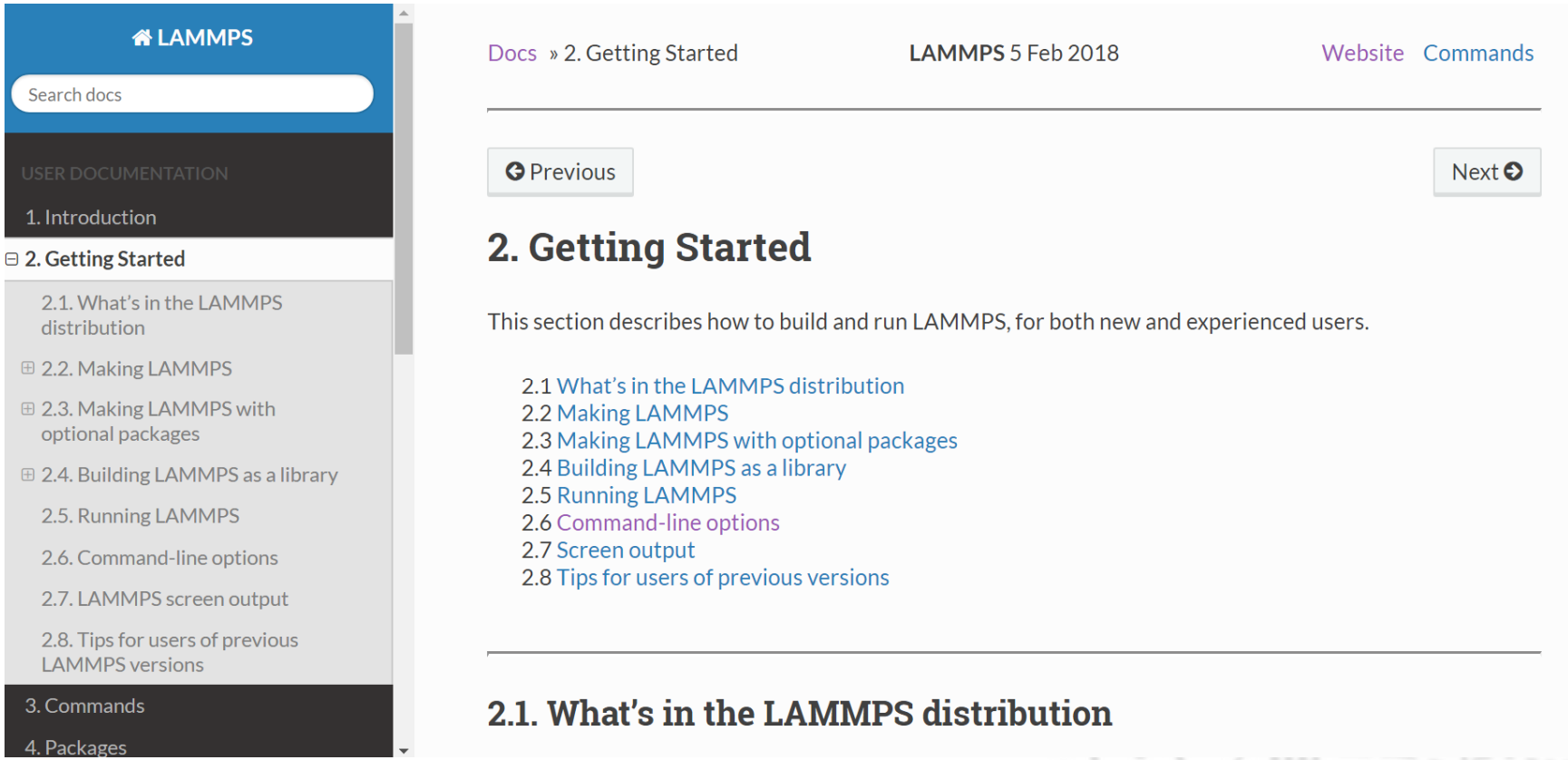


The respective download directory will contain installer packages that are labeled with the date of the LAMMPS version and packages labeled as *latest*. It is usually recommended to download and install the latest package. The other packages are provided in case there is a problem with it. Download the installer executable suitable for your machine, execute it, and follow the instructions in the dialogs. Each version will install into a different directory, so it is possible to have multiple versions installed at the same time (however it is not recommended). Both kinds of packages contain:

- Either: a regular multi-threaded LAMMPS executable called `lmp_serial`. This should always work.
- Or: a multi-threaded LAMMPS executable that also supports parallel execution via MPI message passing. This executable is called `lmp_mpi` and requires installation of a suitable MPICH2 package to work.
- the LAMMPS manual in PDF format
- the LAMMPS developer guide in PDF format
- the colvars reference manual in PDF format
- several additional PDF format guides for specific packages or styles
- the potential files bundled with the LAMMPS source code
- most of the example inputs, reference outputs and related files

LAMMPS Manuel

http://lammps.sandia.gov/doc/Section_start.html#start_6



LAMMPS

Search docs

USER DOCUMENTATION

1. Introduction
2. Getting Started
   - 2.1. What's in the LAMMPS distribution
   - 2.2. Making LAMMPS
   - 2.3. Making LAMMPS with optional packages
   - 2.4. Building LAMMPS as a library
   - 2.5. Running LAMMPS
   - 2.6. Command-line options
   - 2.7. LAMMPS screen output
   - 2.8. Tips for users of previous LAMMPS versions
3. Commands
4. Packages

Docs » 2. Getting Started **LAMMPS** 5 Feb 2018 Website Commands

Previous Next

2. Getting Started

This section describes how to build and run LAMMPS, for both new and experienced users.

2.1 What's in the LAMMPS distribution
2.2 Making LAMMPS
2.3 Making LAMMPS with optional packages
2.4 Building LAMMPS as a library
2.5 Running LAMMPS
2.6 Command-line options
2.7 Screen output
2.8 Tips for users of previous versions

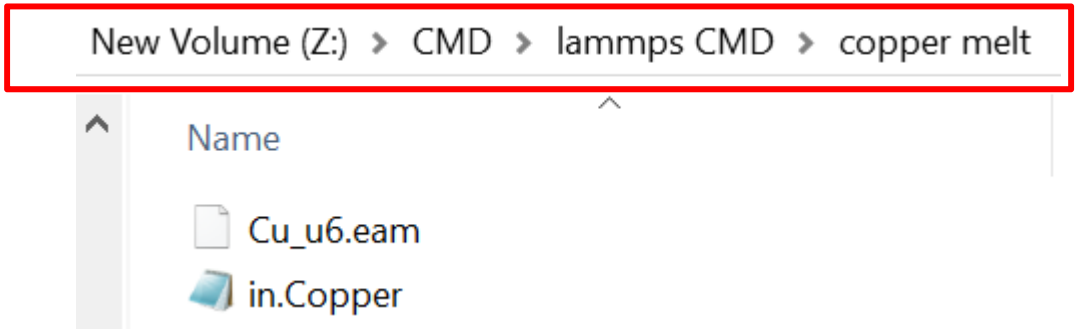
2.1. What's in the LAMMPS distribution

Command Prompt in windows

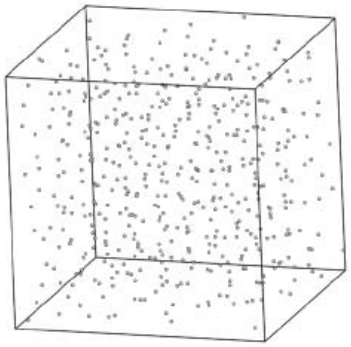
```
Z:\>cd CMD

Z:\CMD>cd "lammps CMD"

Z:\CMD\lammps CMD>cd "copper melt"

Z:\CMD\lammps CMD\copper melt>lmp_serial < in.Copper
```

Run

To the directory



Control T and P, calculate V



1、Temperature（T）

$$< E_k >_{NVT} = \sum_{i=1}^{N} \frac{m_i v_i^2}{2} = \frac{k_B T}{2} N_f$$

k_B — Boltzmann constant

N_f —number of degree of freedom, N_f~3N if N>>1

2、Pressure（P）

$$PV = Nk_B T + \frac{1}{D}\left\langle \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i \right\rangle$$

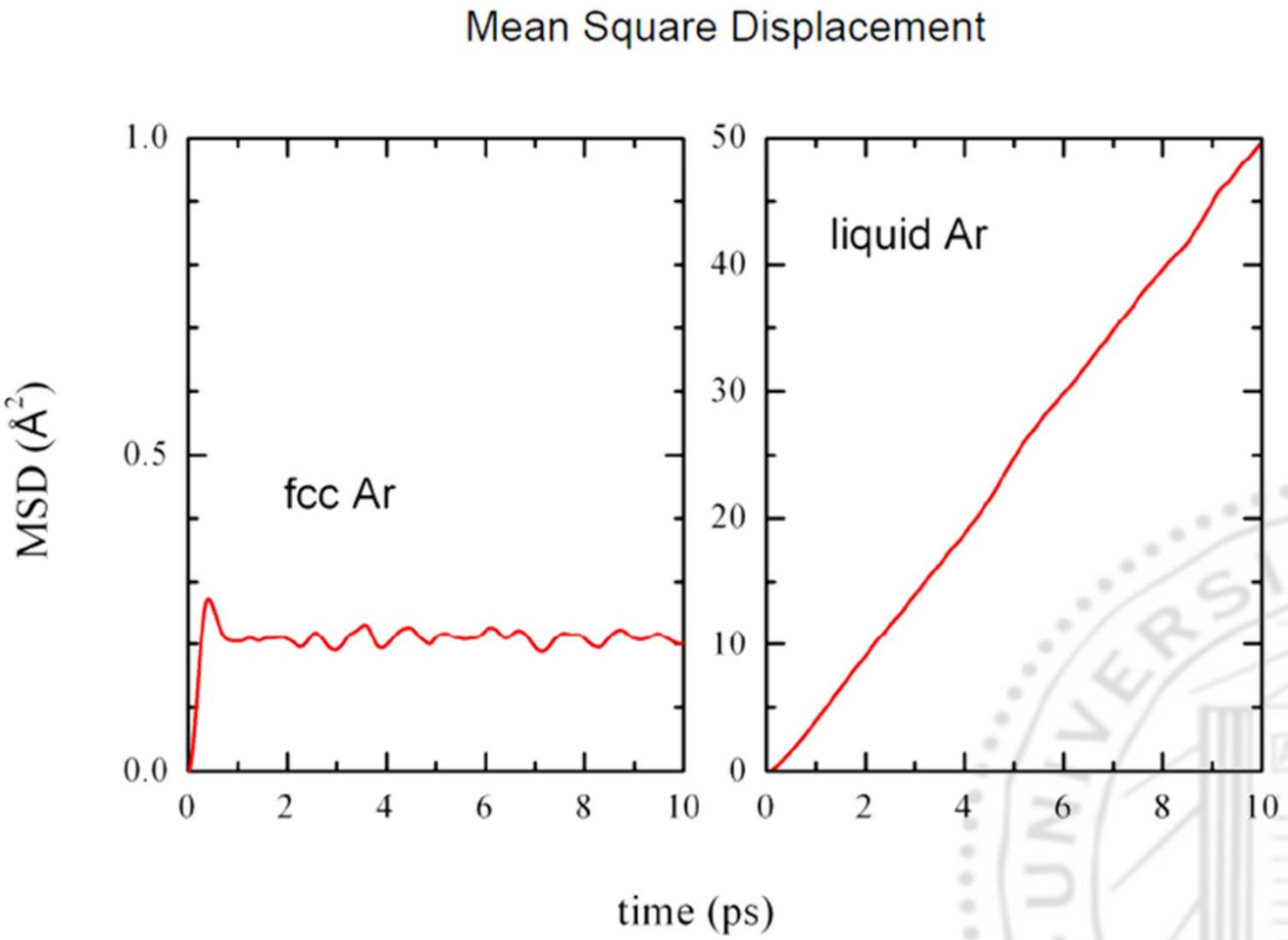
— the *virial* equation

D — system dimensions

Control T and P :

NPT ensemble

MSD for solid and liquid

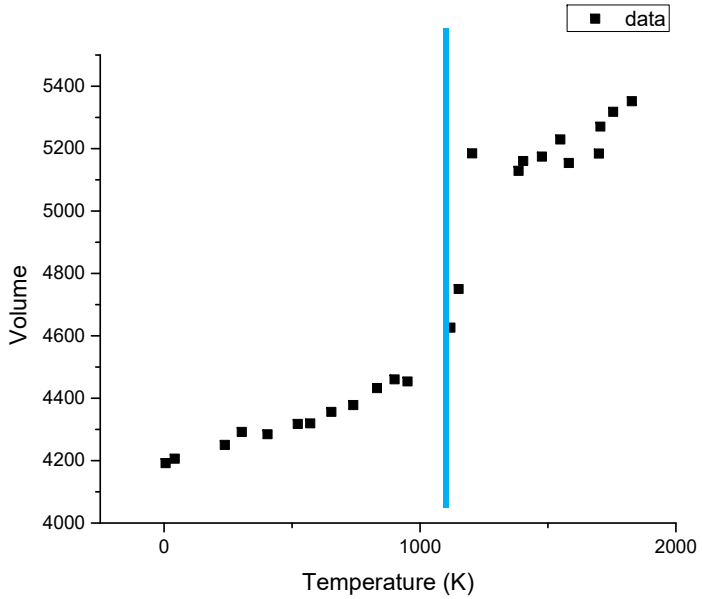


Melt Cu

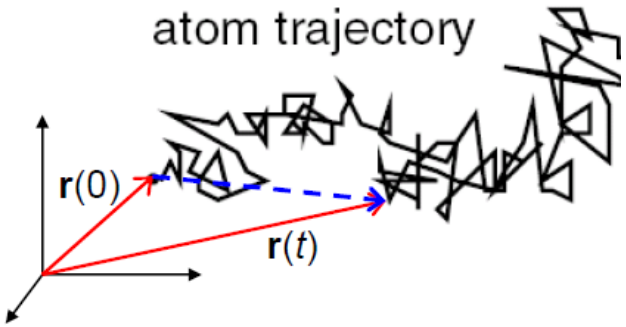
```
change_box all triclinic

#----------Output Configuration----------------------
compute eperat all pe/atom

#------------Output x,y,z of atoms LAMMPS standard format--------------------

#Custom output of atom properties
dump 1 all image 5000 a*.jpg type type
dump 3 all xyz 5000 a*.xyz

#id type x y z c_eperat

#------------Get the radial distribution function
compute rdf all rdf 100
fix rdf1 all ave/time 100 10 1000 c_rdf[1] c_rdf[2] c_rdf[3] &
ave running file nanogold.rdf mode vector

#-------------Energy & Force tolerence , max iterations .
minimize 1.0e-8 1.0e-8 1000 100000

#-------------Minimize using conjugate gradient method
min_style cg
timestep 0.01

#-------------Frequency of Ensemble data output to s c r e e n
thermo 1000
thermo_style custom step pe ke etotal temp vol press

#-------------Thermalize to 298 K
#-------------Create velocity distribution
velocity all create 298 39849 mom yes rot yes dist gaussian

#-------------NVT ensemble ramp temperature to melt

fix 2 all npt temp $x 2400 4.00 tri 0.0 0.0 6.0 drag 0.2
#fix 2 all nvt temp
#print"1"
#fix 3 all msd 1 msd_Cu_$x.dat
#compute 1 all msd Cu_$x.dat

run 200000
```

Melt Al

```
# LAMMPS Melt_Al
units           metal
boundary        p p p
atom_style      atomic
variable        x equal 2.5
lattice         fcc 4.05
region          box block 0 4 0 4 0 4
create_box      1 box
create_atoms    1 box
pair_style      eam/fs
pair_coeff      * * Al_FM.eam.fs Al
timestep        0.01
thermo          1000
neighbor        0.6 bin
neigh_modify    every 5 delay 0 check yes
variable        N equal step
variable        pote equal pe
variable        Etotal equal etotal
variable        T equal temp
variable        Press equal press
variable        V equal vol
compute         3 all pe/atom
compute         4 all ke/atom
velocity        all create $x 825577 dist gaussian
fix             extra all print 1000 "${N} ${T} ${V} ${pote} ${Etotal} ${Press}" file data
#dump            1 all cfg 10000 a$x.*.cfg id type xs ys zs c_3 c_4 c_5
dump 1 all image 10000 a*.jpg type type
dump 3 all xyz 10000 a*.xyz
dump_modify     1 element Al
fix             1 all nvt temp $x $x 2 drag 0.2
run             10000
unfix           1
fix             1 all npt temp $x 2000 10 iso 0 0 10
run             120000
```



Mean square displacement



The mean square displacement of atoms in a simulation can be easily computed by its definition

$$\mathrm{MSD} = \langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle$$

where $\langle \ldots \rangle$ denotes here averaging over all the atoms (or all the atoms in a given subclass). Care must be taken to *avoid* considering the "jumps" of particles to refold them into the box when using periodic boundary conditions as contributing to diffusion.

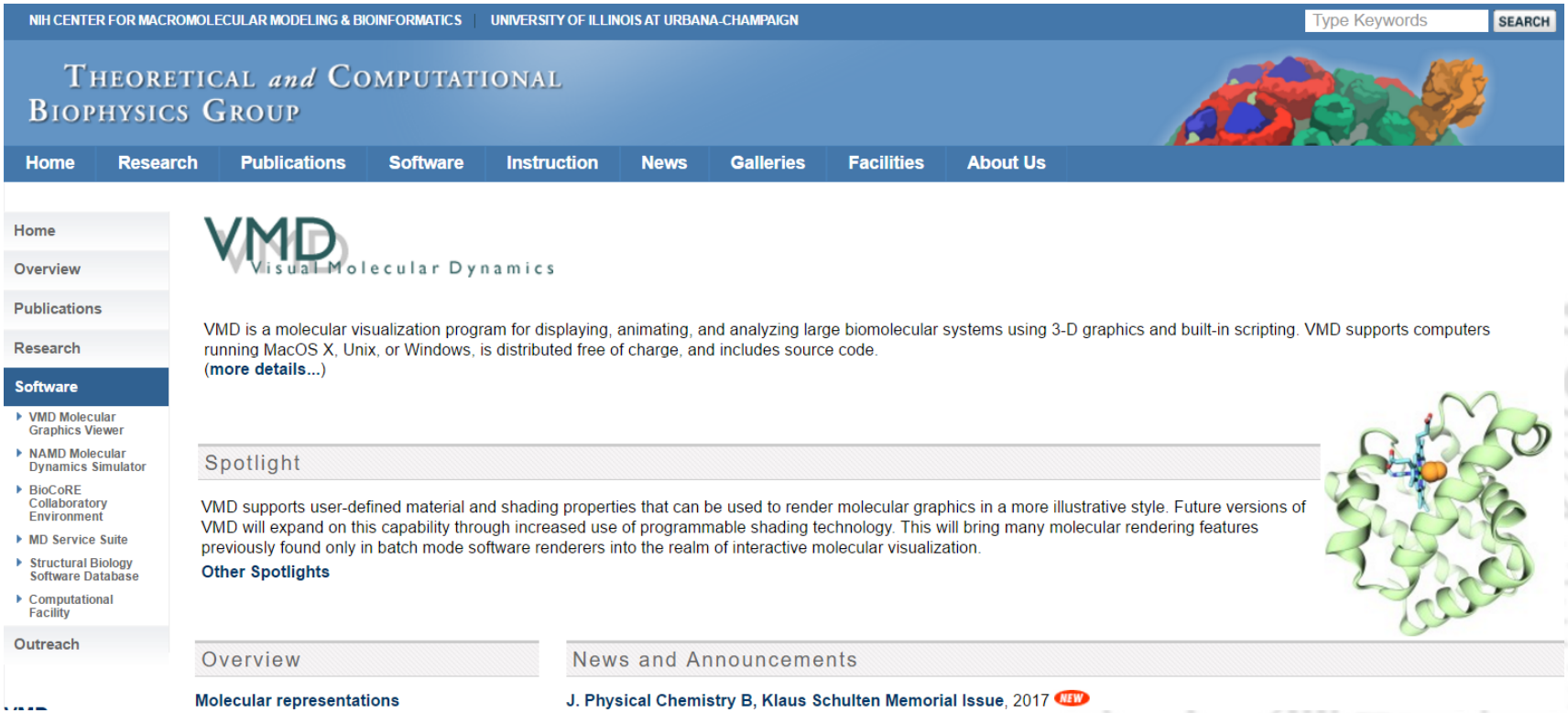
The MSD contains information on the atomic diffusivity. If the system is solid, MSD saturates to a finite value, while if the system is liquid, MSD grows linearly with time. In this case it is useful to characterize the system behavior in terms of the slope, which is the *diffusion coefficient* D:

$$D = \lim_{t\to\infty} \frac{1}{6t} \langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle$$

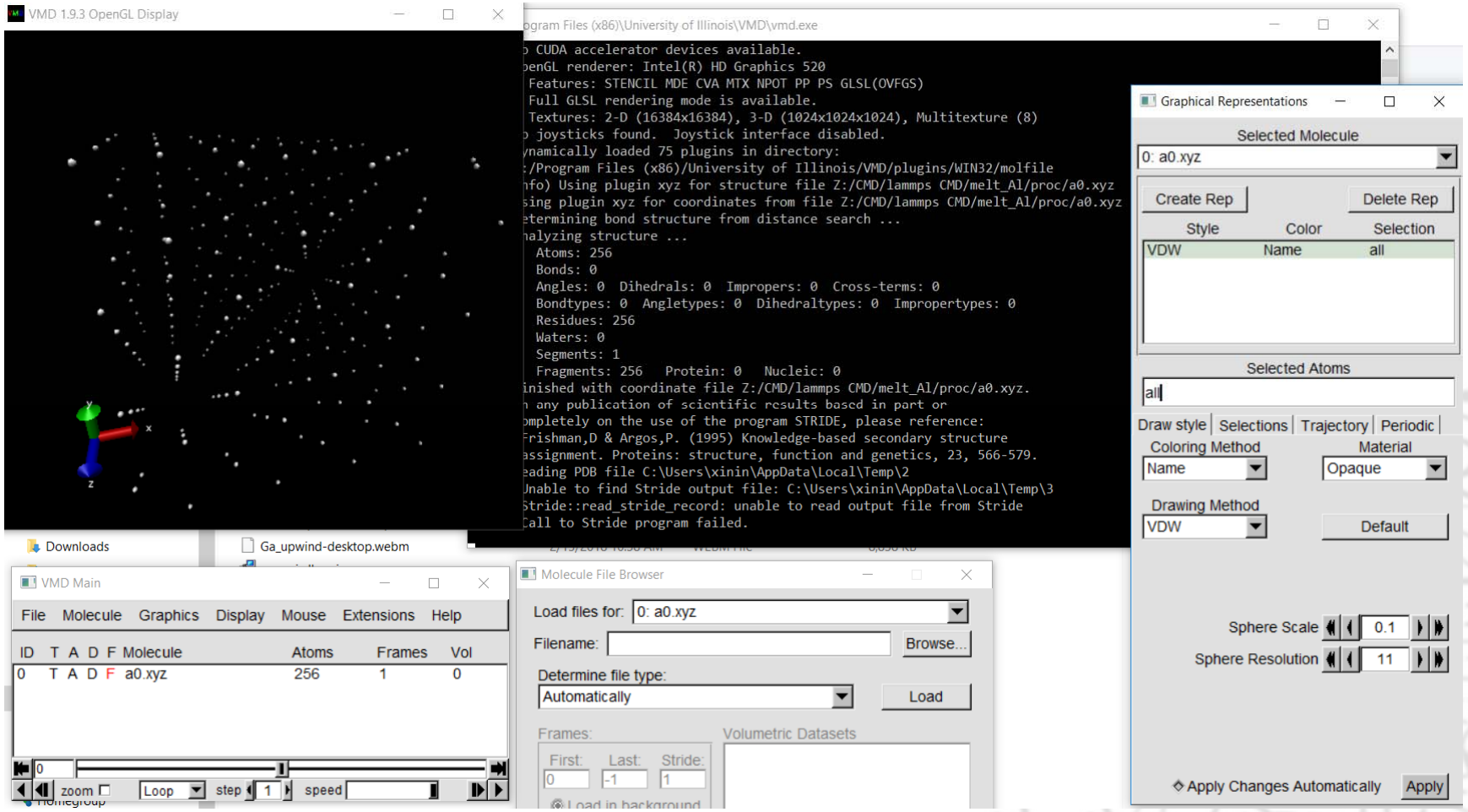
The 6 in the above formula must be replaced with 4 in two-dimensional systems.

VMD Visualization of Atoms

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



VMD


VMD 1.9.3 OpenGL Display
Program Files (x86)\University of Illinois\VMD\vmd.exe
CUDA accelerator devices available.
OpenGL renderer: Intel(R) HD Graphics 520
Features: STENCIL MDE CVA MTX NPOT PP PS GLSL(OVFGS)
Full GLSL rendering mode is available.
Textures: 2-D (16384x16384), 3-D (1024x1024x1024), Multitexture (8)
joysticks found. Joystick interface disabled.
Dynamically loaded 75 plugins in directory:
Program Files (x86)/University of Illinois/VMD/plugins/WIN32/molfile
Info) Using plugin xyz for structure file Z:/CMD/lammps CMD/melt_Al/proc/a0.xyz
Using plugin xyz for coordinates from file Z:/CMD/lammps CMD/melt_Al/proc/a0.xyz
Determining bond structure from distance search ...
Analyzing structure ...
Atoms: 256
Bonds: 0
Angles: 0 Dihedrals: 0 Impropers: 0 Cross-terms: 0
Bondtypes: 0 Angletypes: 0 Dihedraltypes: 0 Impropertypes: 0
Residues: 256
Waters: 0
Segments: 1
Fragments: 256 Protein: 0 Nucleic: 0
Finished with coordinate file Z:/CMD/lammps CMD/melt_Al/proc/a0.xyz.
If any publication of scientific results based in part or
completely on the use of the program STRIDE, please reference:
Frishman,D & Argos,P. (1995) Knowledge-based secondary structure
assignment. Proteins: structure, function and genetics, 23, 566-579.
Reading PDB file C:\Users\xinin\AppData\Local\Temp\2
Unable to find Stride output file: C:\Users\xinin\AppData\Local\Temp\3
Stride::read_stride_record: unable to read output file from Stride
Call to Stride program failed.
Graphical Representations
Selected Molecule
0: a0.xyz
Create Rep
Delete Rep
Style
Color
Selection
VDW
Name
all
Selected Atoms
all
Draw style
Selections
Trajectory
Periodic
Coloring Method
Name
Material
Opaque
Drawing Method
VDW
Default
Sphere Scale
0.1
Sphere Resolution
11
Apply Changes Automatically
Apply
Downloads
Ga_upwind-desktop.webm
VMD Main
File
Molecule
Graphics
Display
Mouse
Extensions
Help
ID T A D F Molecule
Atoms
Frames
Vol
0 T A D F a0.xyz
256
1
0
zoom
Loop
step
1
speed
Molecule File Browser
Load files for:
0: a0.xyz
Filename:
Browse...
Determine file type:
Automatically
Load
Frames:
First:
Last:
Stride:
0
-1
1
Volumetric Datasets