SAND2016-7917C

Exceptional service in the national interest

Sandia National Laboratories

LAMMPS Tutorial, Tuesday AM

Stan Moore

Molecular Dynamics for Modern Materials with LAMMPS

Philadelphia, PA

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

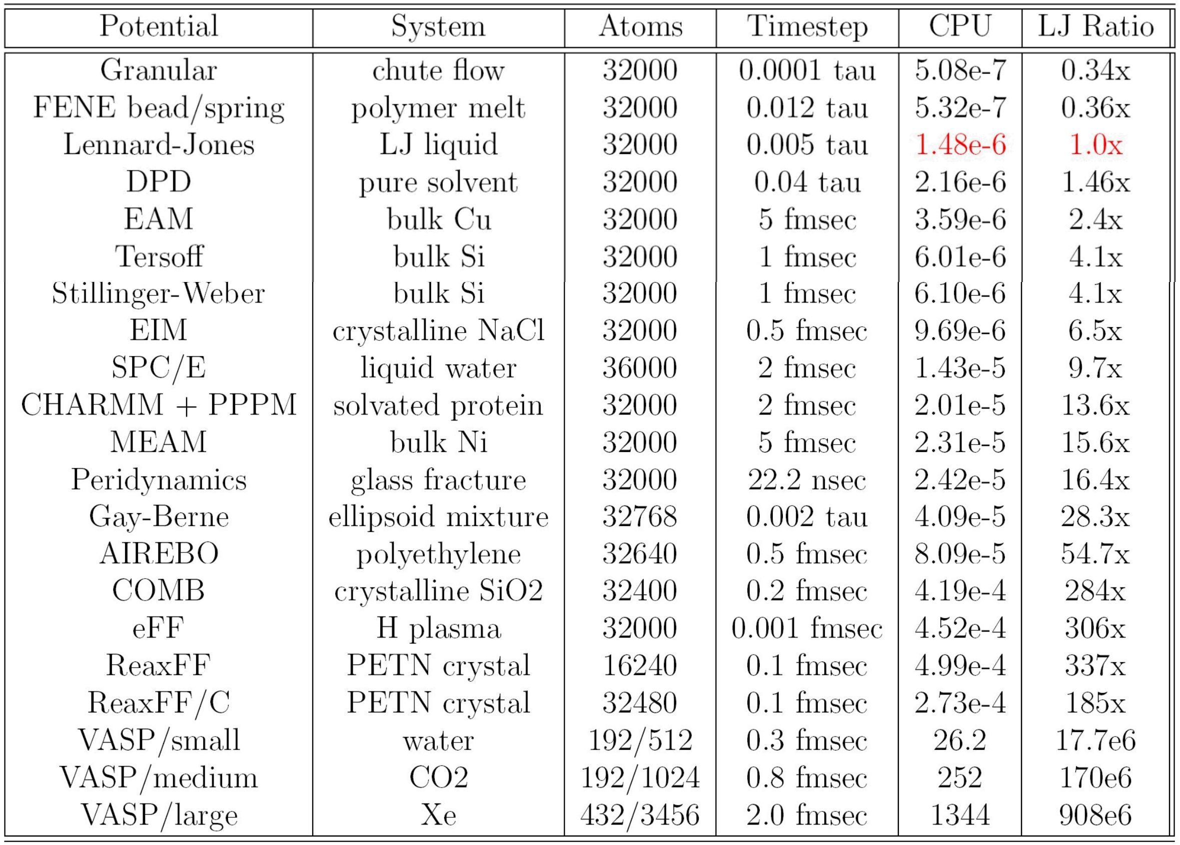
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Outline

- Beyond reactive force fields
- Hybrid force fields
- Syntax specifying advanced force field input
- Data file and molecule file formats
- Molecular builders

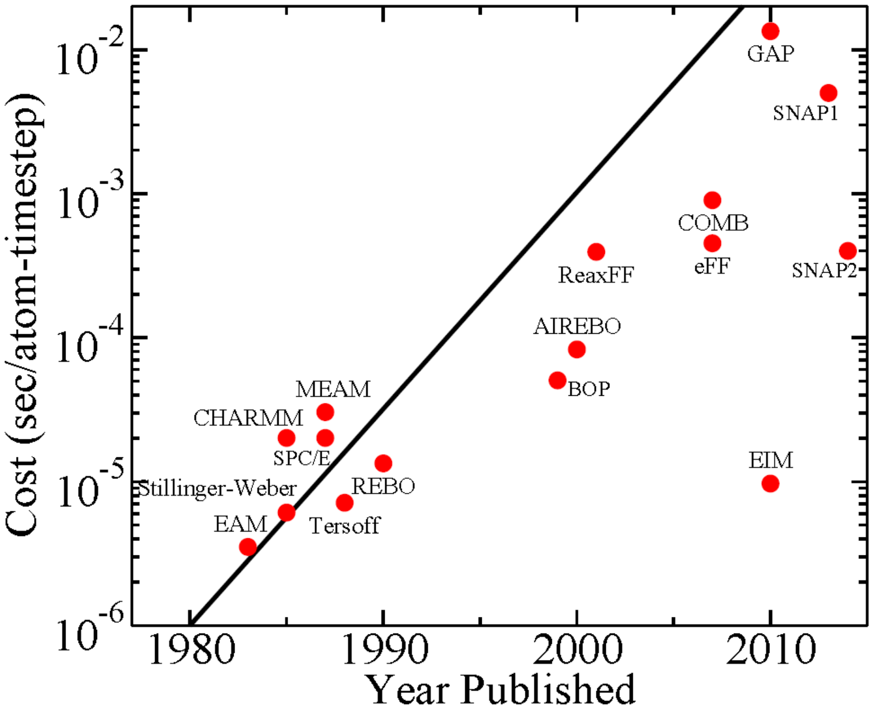
LAMMPS Potentials

- See lammps.sandia.gov/bench.html#potentials



Potential	System	Atoms	Timestep	CPU	LJ Ratio
Granular	chute flow	32000	0.0001 tau	5.08e-7	0.34x
FENE bead/spring	polymer melt	32000	0.012 tau	5.32e-7	0.36x
Lennard-Jones	LJ liquid	32000	0.005 tau	1.48e-6	1.0x
DPD	pure solvent	32000	0.04 tau	2.16e-6	1.46x
EAM	bulk Cu	32000	5 fmsec	3.59e-6	2.4x
Tersoff	bulk Si	32000	1 fmsec	6.01e-6	4.1x
Stillinger-Weber	bulk Si	32000	1 fmsec	6.10e-6	4.1x
EIM	crystalline NaCl	32000	0.5 fmsec	9.69e-6	6.5x
SPC/E	liquid water	36000	2 fmsec	1.43e-5	9.7x
CHARMM + PPPM	solvated protein	32000	2 fmsec	2.01e-5	13.6x
MEAM	bulk Ni	32000	5 fmsec	2.31e-5	15.6x
Peridynamics	glass fracture	32000	22.2 nsec	2.42e-5	16.4x
Gay-Berne	ellipsoid mixture	32768	0.002 tau	4.09e-5	28.3x
AIREBO	polyethylene	32640	0.5 fmsec	8.09e-5	54.7x
COMB	crystalline SiO2	32400	0.2 fmsec	4.19e-4	284x
eFF	H plasma	32000	0.001 fmsec	4.52e-4	306x
ReaxFF	PETN crystal	16240	0.1 fmsec	4.99e-4	337x
ReaxFF/C	PETN crystal	32480	0.1 fmsec	2.73e-4	185x
VASP/small	water	192/512	0.3 fmsec	26.2	17.7e6
VASP/medium	CO2	192/1024	0.8 fmsec	252	170e6
VASP/large	Xe	432/3456	2.0 fmsec	1344	908e6

Towards Quantum Accurate Force Fields



Moore’s Law for Interatomic Potentials

Plimpton and Thompson, MRS Bulletin (2012).

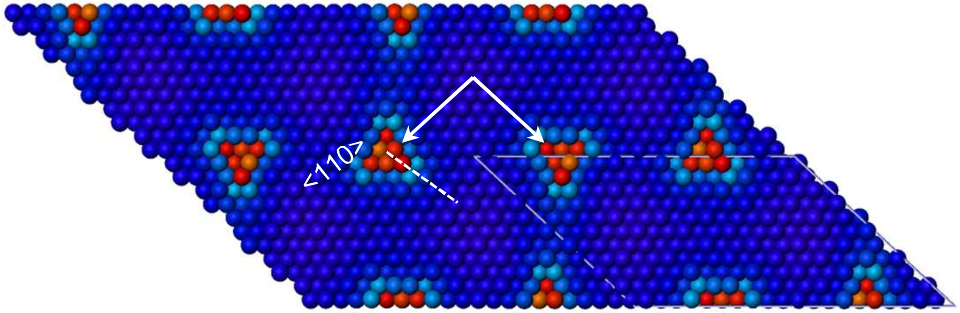
Towards Quantum Accurate Force Fields

- GAP = Gaussian approximation potentials
  - Gabor Csanyi, Albert Bartok-Partay (U Cambridge)
- SNAP = spectral neighbor analysis potentials
  - Aidan Thompson and collaborators (Sandia)
- Aim for quantum-level accuracy in some cases:
  - interpolate to ab initio potential energy surface
  - trained on set of configurations via quantum calculations
  - SNAP cost still linear in N = number of atoms
- Our interest: semiconductors & metals like InP, Ta, SiO_2

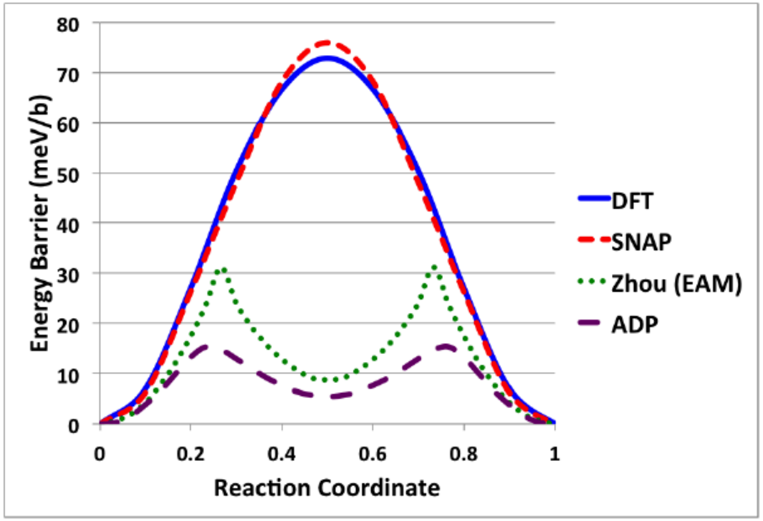
Towards Quantum Accurate Force Fields

- SNAP Ta: captures correct screw dislocation behavior

Screw Dislocation Motion in BCC Tantalum



Weinberger, Tucker, and Foiles, PRB (2013)



- SNAP SiO_2: Good agreement for lattice constants and liquid radial distribution function with DFT

Electron Force Field (eFF)

USER-EFF package, Jaramillo-Botero et al. (Caltech)

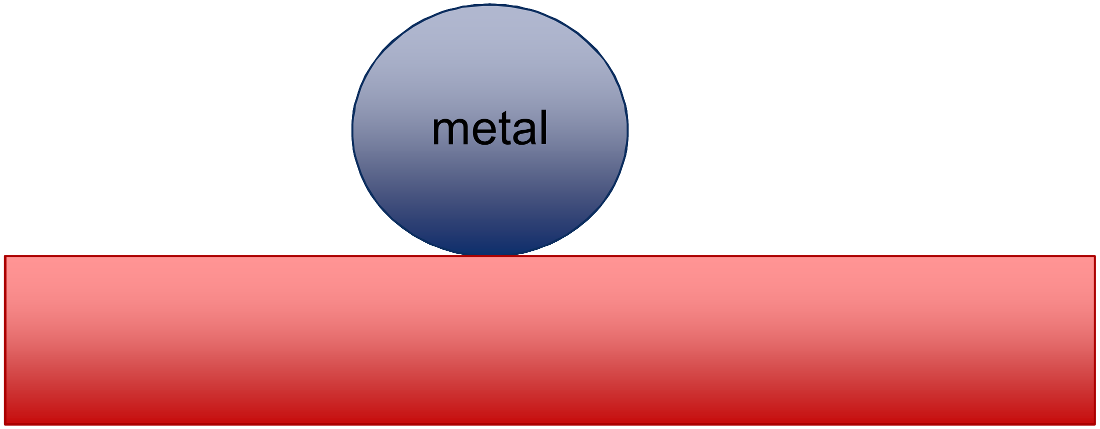
- eFF is approximation to QM wave packet dynamics and Fermionic molecular dynamics
- Combines the ability of electronic structure methods to describe atomic structure, bonding, and chemistry in materials, and of plasma methods to describe nonequilibrium dynamics of large systems with a large number of highly excited electrons
- Relies on a simplification of the electronic wavefunction in which electrons are described as floating Gaussian wave packets
- Excels at computing the properties of materials in extreme conditions and tracing the system dynamics over multi-picosecond timescales

Hybrid force fields

- Pair hybrid: exactly one pair style is assigned to each pair of atom types (can be different pair styles for different atom types)
- Pair hybrid/overlay: one or more pair styles can be assigned to each pair of atom types
- The assignment of pair styles to type pairs is made via the pair_coeff command

Hybrid force fields—Example

- Metal droplet on an LJ surface
  - metal → metal atoms interact with *eam* potential
  - surface → surface atoms interact with *lj/cut* potential
  - metal/surface interaction is also computed via a *lj/cut* potential

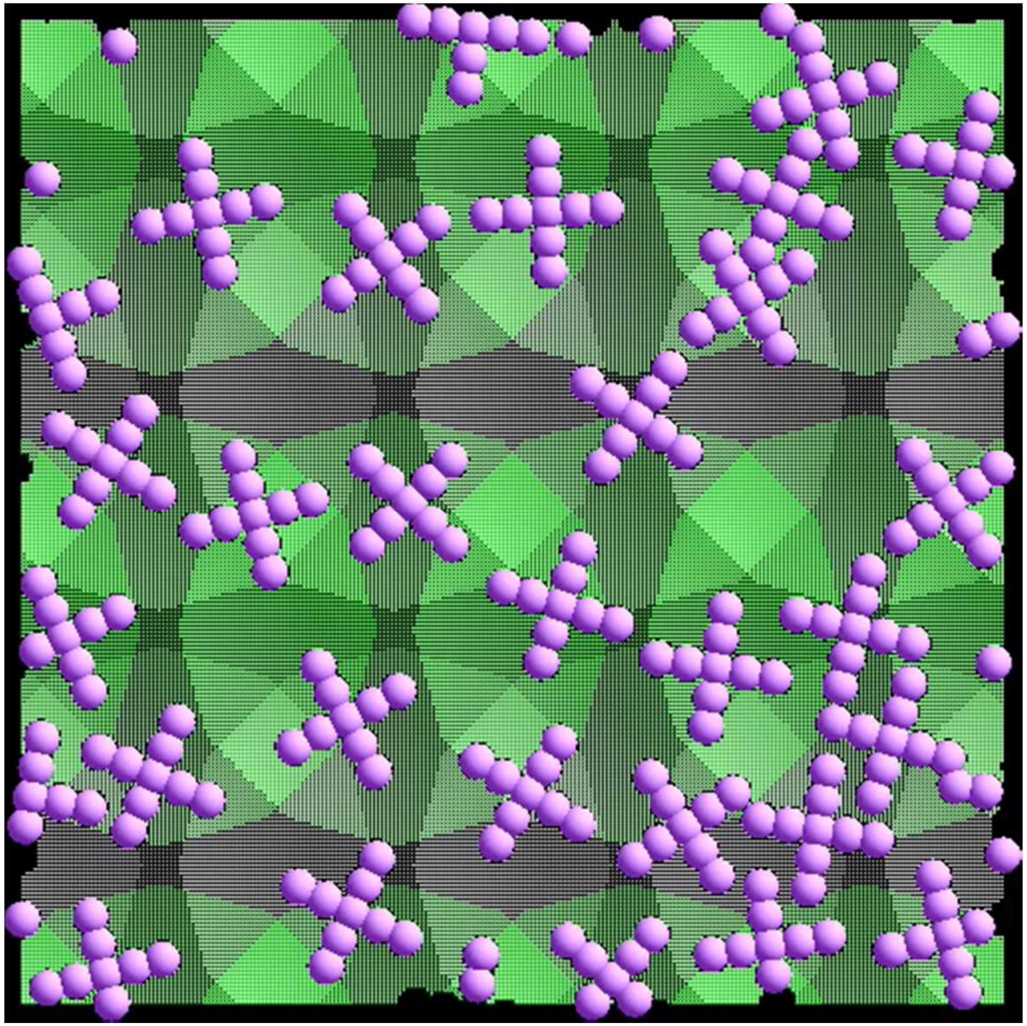


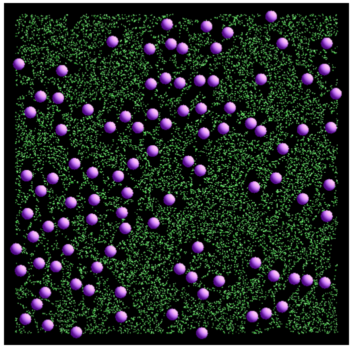
Coarse-grained models

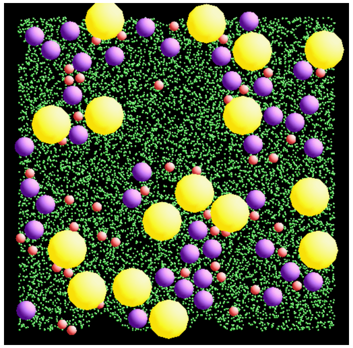
For nanoparticles and solvent:

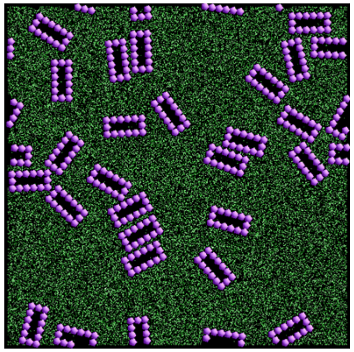
- Particle/particle interactions
  - pair gayberne, resquared, colloid, yukawa/colloid, vincent
  - pair brownian, lubricate, lubricateU (implicit)
  - pair hybrid/overlay for DLVO models
- Stochastic rotation dynamics
  - cheap solvent for nanoparticles
  - no solvent-solvent interactions
  - enables simulations of micron-size particles for seconds
  - with Jeremy Lechman (Sandia) and Pieter in't Veld (BASF)
- Fast lubrication dynamics (FLD package)
  - implicit solvent
  - fast variant of Stokesian Dynamics
  - enables simulations of micron-size particles for seconds
  - Amit Kumar and Jon Higdon (U Illinois)

Viscosity for rigid-bodies in stochastic rotation dynamics fluid





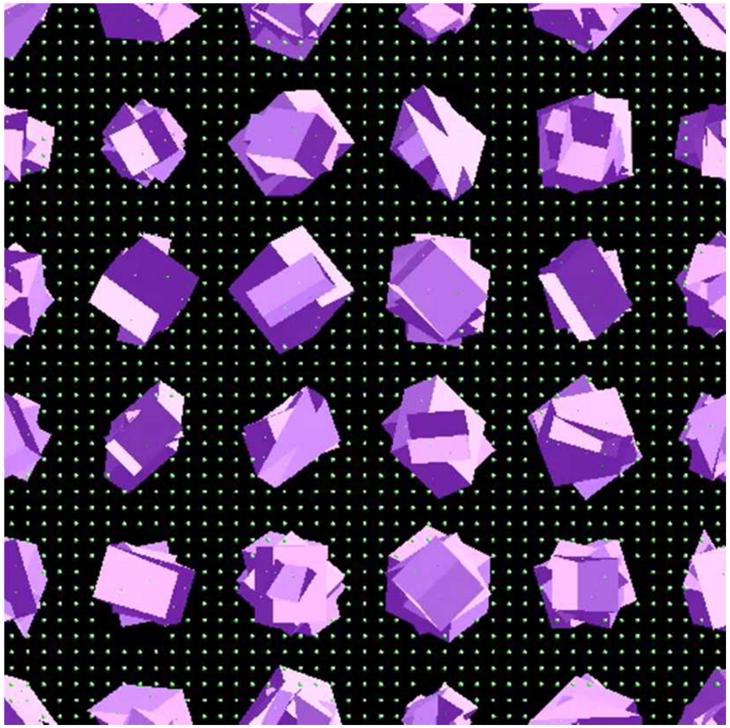


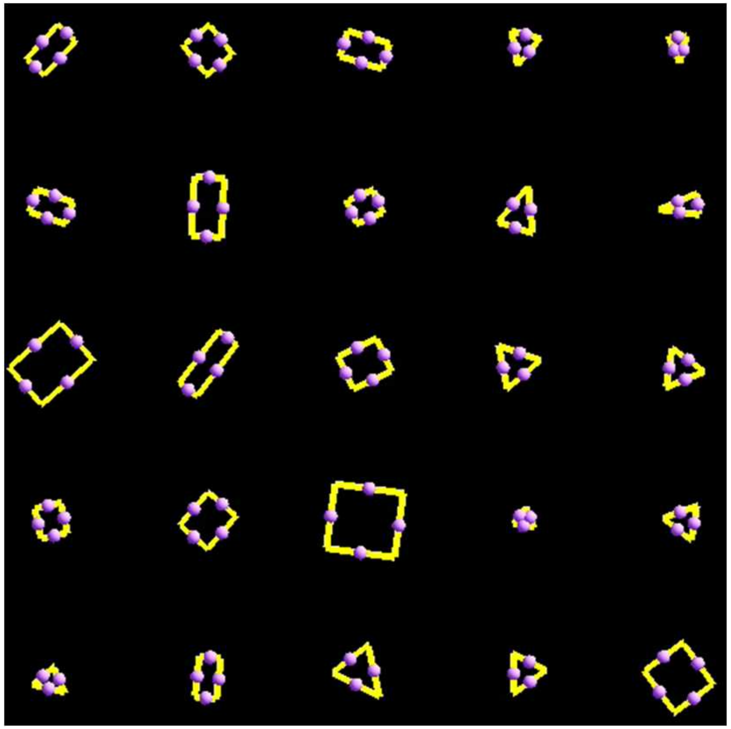


Coarse-graining via aspherical particles

- 2 formulations already exist
  - ellipsoids and Gayberne-like potentials
  - rigid-body collections of point-particles and spheroids
- Atom styles: line (2d), triangle (3d)
  - can build rigid bodies out of them
  - potentials for line/line and tri/tri interactions
- Atom style body for generalized asphericals
- Body particles store internal state
  - sub-particles, facets, etc.
  - customizable Body class

Triangle and line particle examples





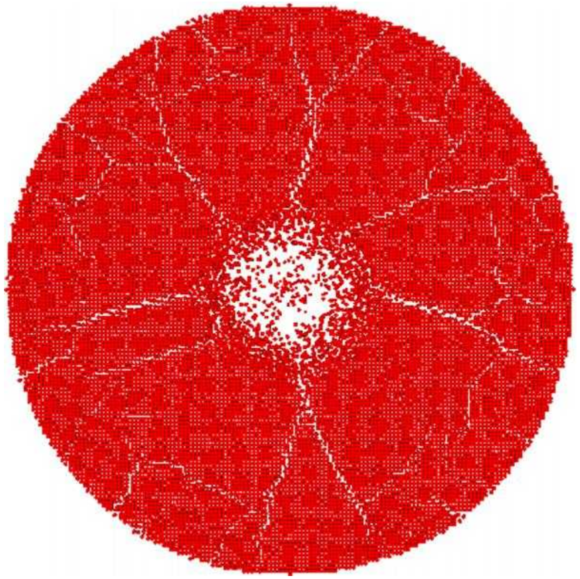
Coarse-graining via mesoscale particles

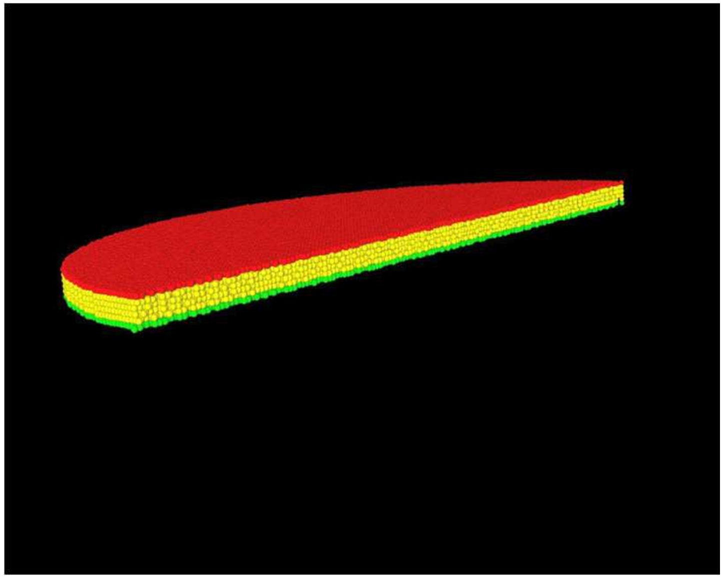
- DPD = dissipative particle dynamics
- PD = peridynamics
- SPH = smoothed particle hydrodynamics
- Granular = normal & tangential friction

Peridynamics

PERI package, Mike Parks and Stuart Silling (Sandia)

- Particle-based meshless non-local continuum model
- Hi-deformation impact & fracture
- Constitutive models encoded in pairwise interactions & bonding



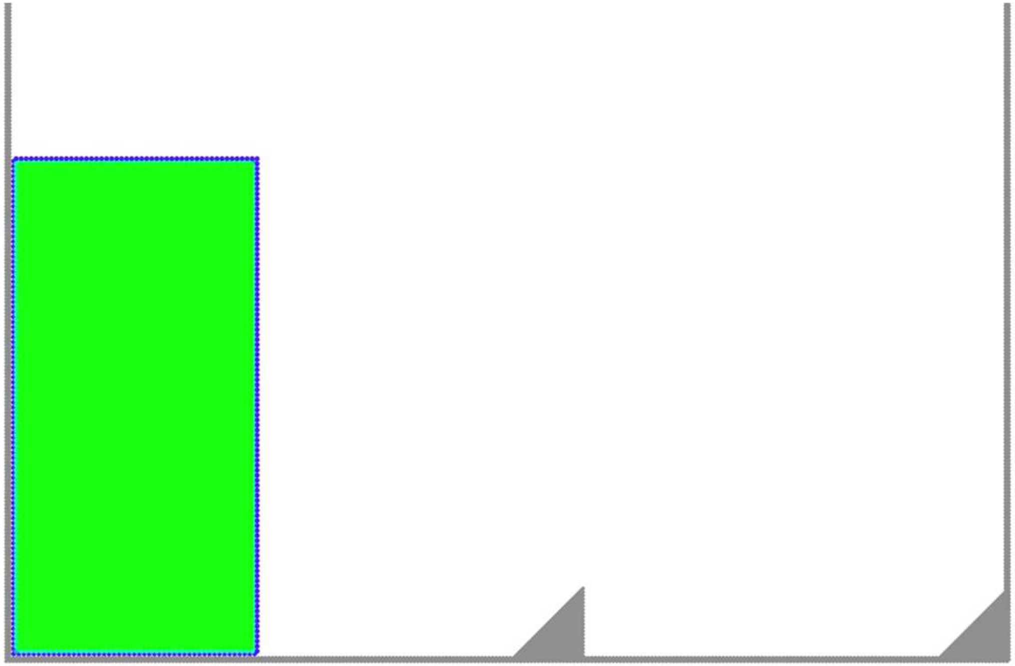


Smoothed particle hydrodynamics

USER-SPH package

Georg Ganzenmüller (Franhofer-Institute, EMI, Germany)

collapse of a water column



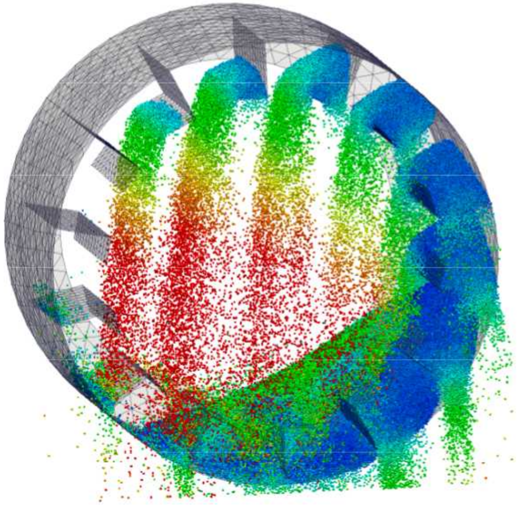
Granular modeling

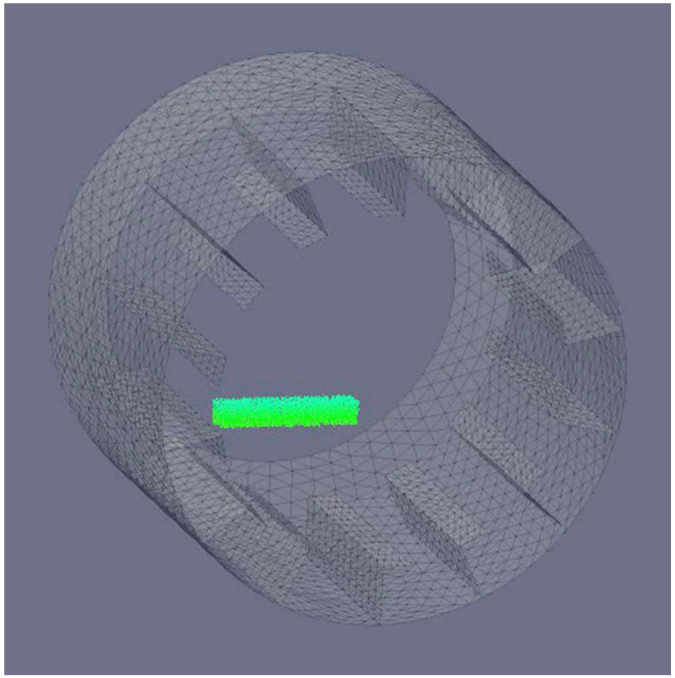
GRANULAR package

Christoph Kloss group (JKU) created add-on LIGGGHTS package

www.cfdem.com

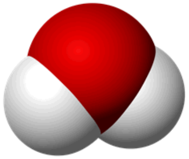
- particles + CAD mesh





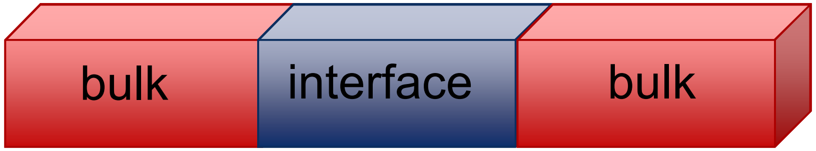
Long-Range Electrostatics

- Truncation doesn’t work well for charged systems due to long-ranged nature of Coulombic interactions
- Use Kspace style to add long-range electrostatics:
  - PPPM—usually fastest, uses FFTs
  - Ewald—potentially most accurate, but slow for large systems
  - MSM—multigrid method that also works for non-periodic systems
- Usually specify a relative accuracy (1e-4 or 1e-5 typically used)
- Example syntax (for periodic systems): kspace_style pppm 1.0e-4
- Use pair_style *coul/long such as lj/cut/coul/long



2D Slab Geometry with Kspace

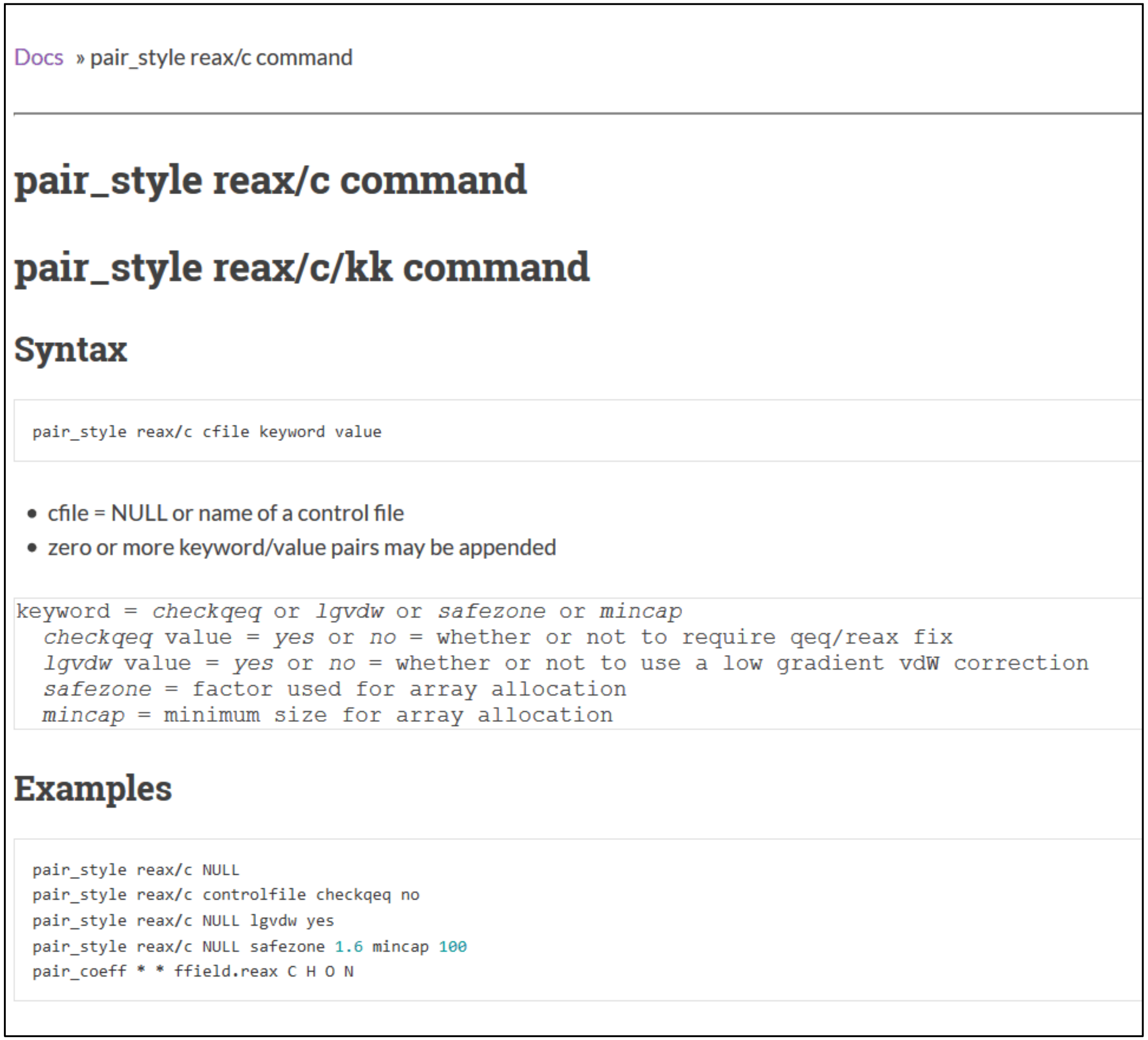
- The *slab* keyword allows a Kspace solver to be used for a systems that are periodic in x,y but non-periodic in z
- Must use a boundary setting of “boundary p p f”
- Actually treats the system as if it were periodic in z, but inserts empty volume between atom slabs and removing dipole inter-slab interactions so that slab-slab interactions are effectively turned off
- Example syntax: kspace_modify slab 3.0
- May need to use reflecting walls in the z-dimension



Syntax specifying advanced force field input

- Look in the LAMMPS documentation (i.e. http://lammps.sandia.gov/doc/pair_reax.html)
- Shows pair_style name and accelerator variants
- Shows syntax with required and optional keywords
- Gives examples of command use
- Gives a description of the keywords
- Also gives restrictions, related commands, and default values of optional keywords

ReaxFF Syntax



Docs » pair_style reax/c command

pair_style reax/c command

pair_style reax/c/kk command

Syntax

```
pair_style reax/c cfile keyword value
```

- cfile = NULL or name of a control file
- zero or more keyword/value pairs may be appended

```
keyword = checkqeq or lgvdw or safezone or mincap
  checkqeq value = yes or no = whether or not to require qeq/reax fix
  lgvdw value = yes or no = whether or not to use a low gradient vdW correction
  safezone = factor used for array allocation
  mincap = minimum size for array allocation
```

Examples

```
pair_style reax/c NULL
pair_style reax/c controlfile checkqeq no
pair_style reax/c NULL lgvdw yes
pair_style reax/c NULL safezone 1.6 mincap 100
pair_coeff * * ffield.reax C H O N
```

ReaxFF Forcefield Files

- LAMMPS provides several different versions of ffield.reax in /potentials, each called potentials/ffield.reax.<label>. These are documented in /potentials/README.reax. The default ffield.reax contains parameterizations for the following elements: C, H, O, N
- More ReaxFF parameterizations are available from Adri van Duin's group at PSU
- They are continuously deriving and updating parameterizations for different classes of materials
- Can submit a contact request at the Materials Computation Center (MCC) website https://www.mri.psu.edu/materials-computation-center/connect-mcc, describing the material(s) you are interested in modeling with ReaxFF. They can tell you what is currently available or what it would take to create a suitable ReaxFF parameterization

EAM, SW, and Tersoff Syntax

- pair_style eam
- pair_coeff * * cuu3
- pair_coeff 1*3 1*3 niu3.eam

- pair_style sw
- pair_coeff * * si.sw Si
- pair_coeff * * GaN.sw Ga N Ga

- pair_style tersoff
- pair_coeff * * Si.tersoff Si
- pair_coeff * * SiC.tersoff Si C Si

- Can use forcefield files in /potentials folder

COMB Syntax

COMB

- pair_style comb
- pair_coeff * * ../potentials/ffield.comb Si
- pair_coeff * * ../potentials/ffield.comb Hf Si O

COMB3

- pair_style comb3 polar_off
- pair_coeff * * ../potentials/ffield.comb3 O Cu N C O

- Can use forcefield file in /potentials folder

Data Files

- The data file contains basic information such as:
  - size of the problem to be run
  - the initial atomic coordinates
  - molecular topology
  - and (optionally) force-field coefficients
- Example syntax for reading in a data file: read_data data.lj
- Example syntax for writing out a data file: write_data data.lj
- Data files are text files
- Restart files are binary files which save a system configuration
- Can convert a restart file to a data file using the “-restart” command line option

Data File Rules

- Blank lines are important. After the header section, new entries are separated by blank lines
- Indentation and space between words/numbers on one line is not important except that keywords (e.g. Masses, Bond Coeffs) must be left-justified and capitalized
- The header section (thru box bounds) must appear first in the file, the remaining entries (Masses, various Coeffs, Atoms, Bonds, etc) can come in any order
- These entries must be in the file: header section, Masses, Atoms
- These entries must be in the file if there are a non-zero number of them: Bonds, Angles, Dihedrals, Impropers. Force field coefficients can be specified in the input script, so do not have to appear in the data file
- The Nonbond Coeffs entry contains one line for each atom type. These are the coefficients for an interaction between 2 atoms of the same type. The cross-type coeffs are computed by the appropriate class I or class II mixing rules, or can be specified explicitly using the "nonbond coeff" command in the input command script

Data File Rules

- In the Atoms entry, the atoms can be in any order so long as there are N entries. The atom_style determines format (see later slide)
- Atom velocities are initialized to 0.0 if there is no Velocities entry. In the Velocities entry, the atoms can be in any order so long as there are N entries. The 1st number on the line is the atom-tag (number from 1 to N) which is used to identify the atom which the given velocity will be assigned to.
- Entries for Velocities, Bonds, Angles, Dihedrals, Impropers must appear in the file after an Atoms entry.
- For simulations with periodic boundary conditions, xyz coords are remapped into the periodic box (from as far away as needed), so the initial coordinates need not be inside the box
- The number of coefficients specified on each line of coefficient entries (Nonbond Coeffs, Bond Coeffs, etc) depends on the "style" of interaction. This must be specified in the input command script before the "read data" command is issued, unless the default is used

Data file and molecule file formats

```
LAMMPS Description          (1st line of file)

100 atoms          (this must be the 3rd line, 1st 2 lines are ignored)
95 bonds                (# of bonds to be simulated)
50 angles               (include these lines even if number = 0)
30 dihedrals
20 impropers

5 atom types            (# of nonbond atom types)
10 bond types           (# of bond types = sets of bond coefficients)
18 angle types
20 dihedral types      (do not include a bond,angle,dihedral,improper type
2 improper types              line if number of bonds,angles,etc is 0)

-0.5 0.5 xlo xhi       (for periodic systems this is box size,
-0.5 0.5 ylo yhi        for non-periodic it is min/max extent of atoms)
-0.5 0.5 zlo zhi       (do not include this line for 2-d simulations)
```

Data file and molecule file formats

```
Masses

  1 mass
  ...
  N mass                    (N = # of atom types)

Nonbond Coeffs

  1 coeff1 coeff2 ...
  ...
  N coeff1 coeff2 ...          (N = # of atom types)

Bond Coeffs

  1 coeff1 coeff2 ...
  ...
  N coeff1 coeff2 ...          (N = # of bond types)         (N = # of angle types)
```

Data file and molecule file formats

Atoms

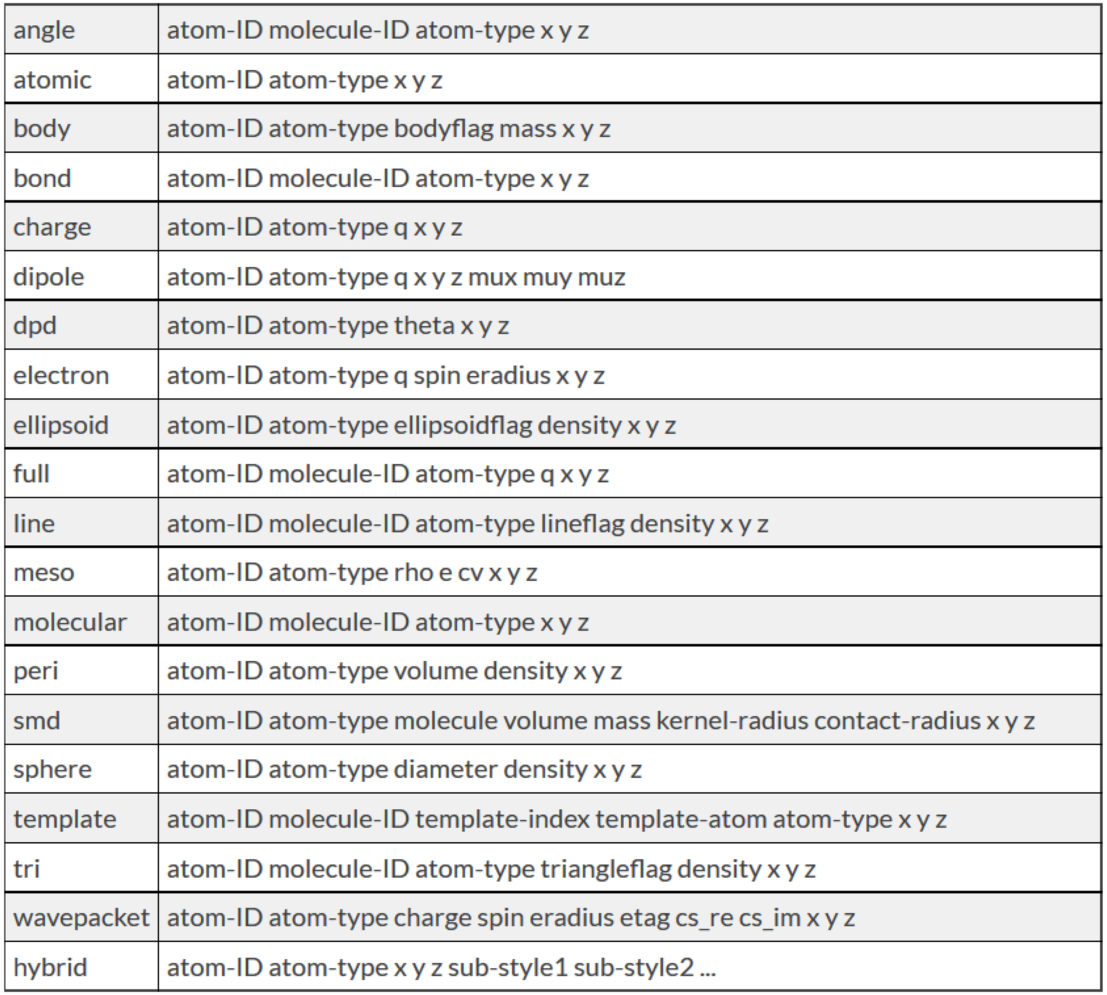
```
 1 molecule-tag atom-type q x y z nx ny nz  (nx,ny,nz are optional -
 ...                                  see "true flag" input command)
 ...
 N molecule-tag atom-type q x y z nx ny nz  (N = # of atoms)
```

Velocities

```
 1 vx vy vz
 ...
 ...
 N vx vy vz                        (N = # of atoms)
```

Atoms Format

Depends on atom_style:



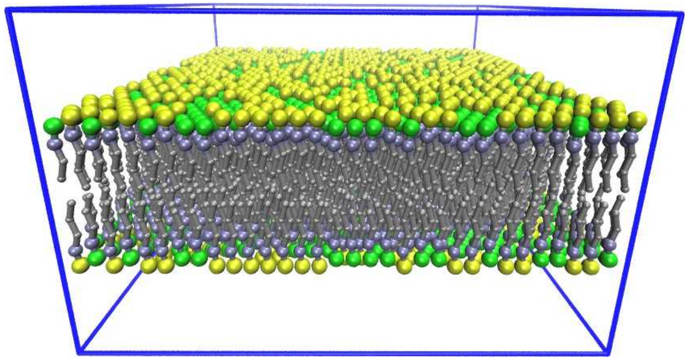
angle	atom-ID molecule-ID atom-type x y z
atomic	atom-ID atom-type x y z
body	atom-ID atom-type bodyflag mass x y z
bond	atom-ID molecule-ID atom-type x y z
charge	atom-ID atom-type q x y z
dipole	atom-ID atom-type q x y z mux muy muz
dpd	atom-ID atom-type theta x y z
electron	atom-ID atom-type q spin eradius x y z
ellipsoid	atom-ID atom-type ellipsoidflag density x y z
full	atom-ID molecule-ID atom-type q x y z
line	atom-ID molecule-ID atom-type lineflag density x y z
meso	atom-ID atom-type rho e cv x y z
molecular	atom-ID molecule-ID atom-type x y z
peri	atom-ID atom-type volume density x y z
smd	atom-ID atom-type molecule volume mass kernel-radius contact-radius x y z
sphere	atom-ID atom-type diameter density x y z
template	atom-ID molecule-ID template-index template-atom atom-type x y z
tri	atom-ID molecule-ID atom-type triangleflag density x y z
wavepacket	atom-ID atom-type charge spin eradius etag cs_re cs_im x y z
hybrid	atom-ID atom-type x y z sub-style1 sub-style2 ...

Data file and molecule file formats

```
Bonds

  1 bond-type atom-1 atom-2
  ...
  N bond-type atom-1 atom-2        (N = # of bonds)

Angles

  1 angle-type atom-1 atom-2 atom-3  (atom-2 is the center atom in angle)
  ...
  N angle-type atom-1 atom-2 atom-3  (N = # of angles)

Dihedrals

  1 dihedral-type atom-1 atom-2 atom-3 atom-4  (atoms 2-3 form central bond)
  ...
  N dihedral-type atom-1 atom-2 atom-3 atom-4  (N = # of dihedrals)

Impropers

  1 improper-type atom-1 atom-2 atom-3 atom-4  (atom-2 is central atom)
  ...
  N improper-type atom-1 atom-2 atom-3 atom-4  (N = # of impropers)
```

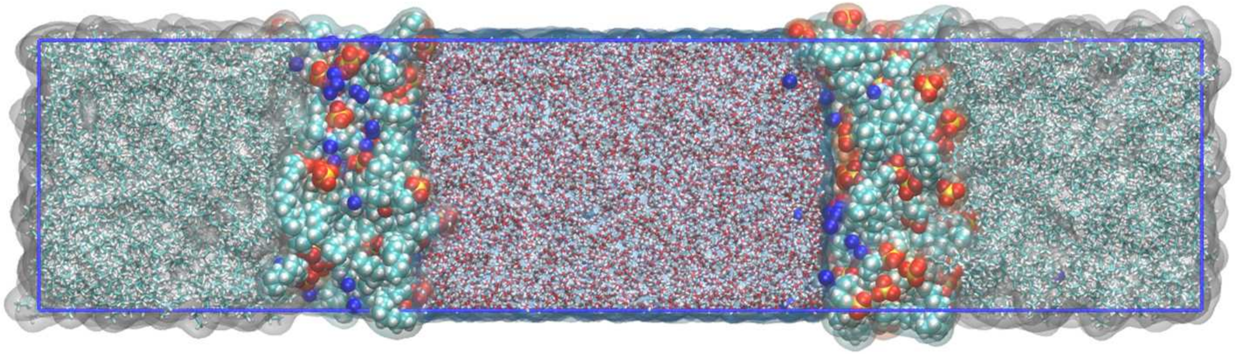
Molecule Builders: Moltemplate

- Developed and maintained by Andrew Jewett (UCSB)
- Distributed with LAMMPS in the tools/moltemplate directory
- Designed for building coarse-grained biomolecular models
- Can create both: lammps DATA files (containing geometry and topology), and lammps INPUT scripts (containing force-fields, fixes, and groups)
- Gives users access to all of the force-fields available in LAMMPS
- Molecules can be used as building blocks for bigger molecules



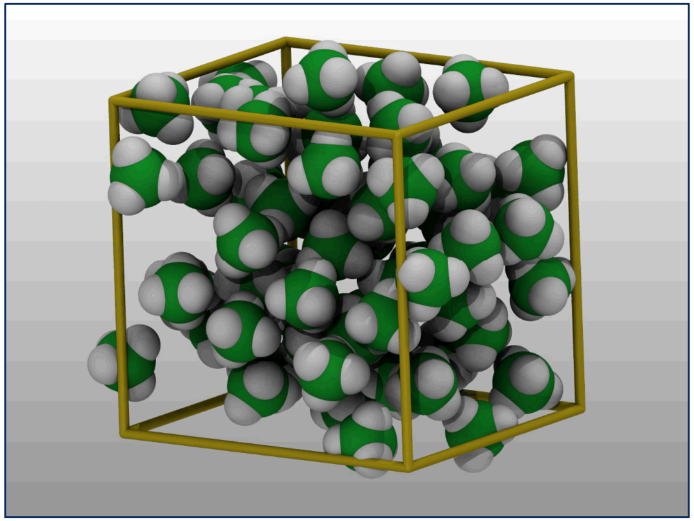
Molecule Builders: Enhanced Monte Carlo (EMC)

- Developed and maintained by Pieter J. in 't Veld (BASF)
- Provides an environment for creating and manipulating input structures for particle simulations using COMPASS, CHARMM, OPLS, Martini, DPD, or colloidal force fields
- Provides manipulation of molecular or coarse-grained structures through SMILES strings
- Applies Monte Carlo principles to unoverlap atoms
- Provides output ports to LAMMPS, PDB, and XYZ formats
- A compiled version for Linux, MacOS, or Windows can be found at montecarlo.sourceforge.net



Molecule Builders: VMD TopoTools

- Developed and maintained by Axel Kohlmeyer (Temple U)
- See http://sites.google.com/site/akohlmey/software/topotools for more details
- Leverages the power of VMD and TCL to create lammps DATA files and convert them to and from other formats
- Has two components: a middleware script which can extract and manipulate topology information, and several high-level applications built on top of it, which, for example can enable it to read/write data files, replicate and merge systems.
- Together with VMD, topotools can infer topology from PDB files, PSF files, and atom pair distances and solvate a protein



Other Molecule Builders

Avogadro: advanced molecule editor and visualizer designed for cross-platform use in computational chemistry, molecular modeling, bioinformatics, materials science, and related areas. It offers flexible high quality rendering and a powerful plugin architecture.

Packmol: creates an initial point for molecular dynamics simulations by packing molecules in defined regions of space. The packing guarantees that short range repulsive interactions do not disrupt the simulations.

Atomsk: aims at creating, manipulating, and converting atomic systems. It supports many file formats, which makes it easy to convert files for ab initio calculations, classical potential simulations, or visualization. Additionally, atomsk can also perform some simple transformations of atomic positions, like rotation, deformation, inserting dislocations.

OCTA: consists of simulation engines (Molecular Dynamics, Rheology simulation, Self Consistent Field Theory, Finite Element Method, etc.) and a GUI for modeling soft matter systems. OCTA also provides an environment for the collaborative usage of several kinds of simulators, i.e. multi-physics and multi-scale simulations.

Questions?