Advanced System Building Features in LAMMPS

Dr. Axel Kohlmeyer

LAMMPS Core Developer

Research Professor, Department of Chemistry
Associate Director, Institute for Computational Science
High-Performance Computing Team Lead

Temple University
Philadelphia PA, USA

a.kohlmeyer@temple.edu

ICMS
Institute for Computational Molecular Science

TEMPLE UNIVERSITY®

Overview

- System setup and data model in LAMMPS
- Entering simulation system info into LAMMPS incrementally with:
  - create_atoms
  - read_data
- Commands to assist in building topologies
- Using the molecule command
- Using multiple force fields (hybrid styles)

How LAMMPS stores Topology Data

- LAMMPS uses a distributed data model
  → topology data is attached to atoms
  → as atoms move between sub-domains, so does the topology information
  → per sub-domain info (e.g. bondlist) and atom migration is part of neighbor list build
  → requires to know the sub-domain layout
  => can only be done after the box is defined
- Definition of the simulation cell locks is certain system properties (e.g. number of types)

Pre-Box Commands

- Certain commands in LAMMPS are only allowed before the simulation box is defined (via create_box or read_data or read_restart)
- Examples: atom_style, boundary, dimension, package, processors, newton, units
- When the box is created, most per-atom arrays are defined and then properties like number of types (atom/bond/angle/dihedral) and dimensioning of per-atom arrays (e.g. for storing topology data with atoms) are locked in

Simple Systems == No Problems

- Thus to set up a simple atomic system, you only set the number of atom types (create_box) and then once or more call create_atoms
- Or when using external tools, all topology info is provided in a single data file (cf. header section)

  LAMMPS can then infer the per-atom array dimensions from this data. it reads the file a first time for that purpose and then creates the box and then fills it with atoms and their associated info and topology data

Multiple Lattice Commands

- When using create_atoms multiple times and the lattice info is changed, some care is needed
- The lattice information is no system information, but simply provides a set of length measures that simplify setting up lattice based geometries
- Lattice information is applied when used and not retroactively altered with a new lattice
- Thus building systems with two components having different lattices settings, lengths have to be commensurate to avoid overlaps etc.

Building Complex Systems with External Build Tools

- Often the preferred option, as the options and tools in LAMMPS are limited by its nature of a parallel program with distributed data
- Examples:
  - charmm2lammps.pl (via psfgen / charmm)
  - msi2lmp (via Material Studio)
  - moltemplate (standalone, importing fragments)
  - TopoTools (via packmol, OpenBabel, scripting)

Advanced System Building Inside of LAMMPS

- Need to reserve space for additional type and topology data when creating the box:
  create_box 2 bond/types 1 angle/types 1 &
  extra/bond/per/atom 2 extra/angle/per/atom 1&
  extra/special/per/atom 4
- Similar options exist for read_data:

  read_data data.wrapped extra/atom/types 1 &
  extra/bond/types 1 extra/angle/types 1 &
  extra/dihedral/types 1 extra/bond/per/atom 9 &
  extra/angle/per/atom 10 ...

Simple Case: read_data plus create_atoms

- For example to create a metal slab system with a molecular system layered on top of it you can:
  - Read in the molecular system via read_data
  - Reserve space for the additional (metal) atom type
  - Create the metal slab atoms with create_atoms (usually restricted to a specific region)

Using read_data Multiple Times

- It is possible to use read_data multiple times
- **All** system info must be set with the **first** call
- Subsequent calls **must** use the “add” keyword
- There are different options to define how atom IDs and various types are determined
- **No** check is made for overlaps
- Added systems can be shifted around
- Final box is the combined maximum, care must be taken with respect to image flags

How to Handle Overlaps

- Avoid them by choosing box boundaries to be shifted from lattice grid positions
  → whether atoms are included or not is no longer ambiguous
- Avoid them by using box dimensions that are commensurate with multiple chosen lattices. N.b. this also applies to rotated lattices
- Use the delete_atoms overlap command
  it can delete topology data, but cannot use compress option with molecular data (reset_ids)

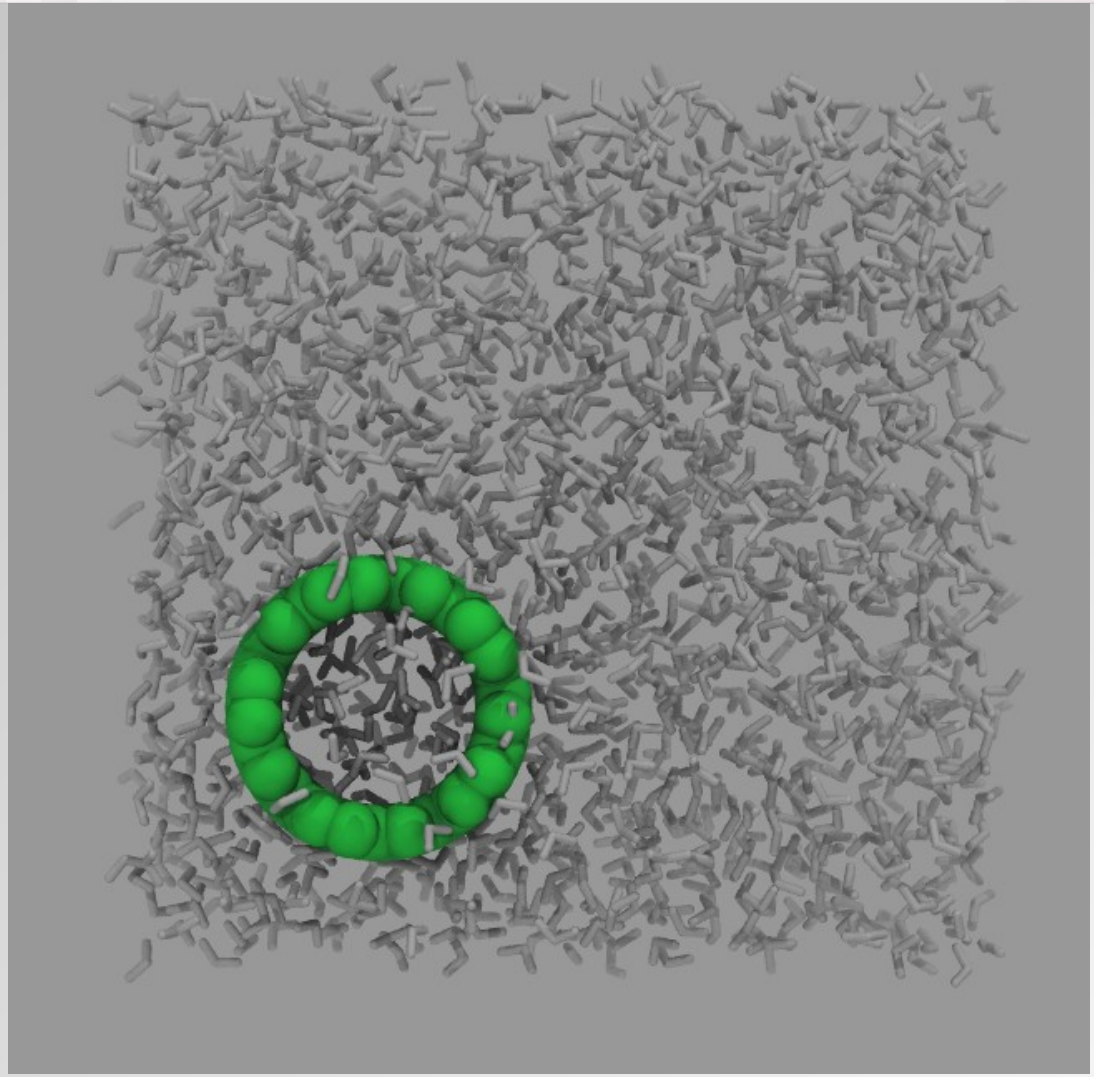
Molecule Command and Files

- Molecule command allows to store information for a group of atoms, e.g. a molecule
- Information is read from a file similar (but not the same) as a data file
- Molecule information is not part of system Identified with a Molecule ID string
- Several commands that can insert atoms into the system, can also insert molecules similar to read_data type offsets can be used
- Space must be available for topology data

Using Multiple Force Fields

- LAMMPS has provisions to do simulations with having **different parts** of the system represented by **different** force fields
- Convenient, but has implications that can easily render the resulting total forces **bogus**
- Best when different **parts** are **separate** entities
- Contact interactions between different force fields **must** be pair-wise additive
- Must not be used to “mix” manybody potentials

Partial Computation of Forces



Advanced System Building Features in LAMMPS

Dr. Axel Kohlmeyer

LAMMPS Core Developer

Research Professor, Department of Chemistry
Associate Director, Institute for Computational Science
High-Performance Computing Team Lead

Temple University
Philadelphia PA, USA

a.kohlmeyer@temple.edu

ICMS
Institute for Computational Molecular Science

TEMPLE UNIVERSITY®