Manual for setup_lammps_v2 program.

July 22, 2015

1 Info

Author: Blake Wilson - blake.wilson@utdallas.edu

For questions, comments, or to report a problem, please contact me at the provided e-mail address.

If compiling from source the recommended compilation (using gcc) is:

gcc -O2 setup_lammps_v2.c -o setup_lammps_v2

2 Introduction

setup_lammps_v2 is an updated version of the program setup_lammps, which parses topology files, a parameter database, and a coordinate file and outputs the DATA and PARM files for use with LAMMPS. This manual will go over the usage and features of the setup_lammps_v2 program.

3 Basic Usage

The basic usage is:

setup_lammps_v2 $< topfile_1 >< nmol_1 > [< topfile_2 >< nmol_2 > ..... < topfile_n >< nmol_n >] < param - file >< coordfile - type >< coordfile >$

$< topfile_1 >< nmol_1 > [< topfile_2 >< nmol_2 > ..... < topfile_n >< nmol_n >]$ is the list of topology files (topfile) and nmol is the number of copies of molecules to be assigned that topology. $< param - file >$ is parameter database file, which contains the parameters for all the pair-wise, bonding, angle, dihedral, and improper interactions. $< coordfile - type >$ is the type type of your coordinate file. setup_lammps_v2 can read 3 different coordinate file types including pdb, xyz, and gcd. the The basic usage can also be displayed from the program by executing it without any arguments. NOTE: The order of topology files and copies should match the order of atoms corresponding to those topologies in the coordinate file (or vice versa).

4 Topology Files

The topology file is like a blueprint of your molecular unit which has all the type and connectivity data. The topology file contains multiple sections. These are: atom, bond, angle, dihedral, and improper. The atom section contains atom lines which index the atoms, list the atom types, mass, charges, etc. The format is as follows:

atom (index) (res name) (res type) (atom type) (mass) (charge) (seg id)

The setup program only uses (atom type), (mass), and (charge), so these need to properly set. The atom types should be consistent between the topology and parameter database files. The next section in the topology file is the bond section which contains bond lines. These have format:

bond (index of bond atom 1) (index of bond atom 2)

The angle, dihedral, and improper sections follow a similar format to the bond section. Angle:

angle (index of angle atom 1) (index of angle atom 2) (index of angle atom 3)

Dihedral:

dihedral (index of dihedral atom 1) (index of dihedral atom 2) (index of dihedral atom 3) (index of dihedral atom 4)

Improper:

dihedral (index of improper atom 1) (index of improper atom 2) (index of improper atom 3) (index of improper atom 4)

An example is the water topology:

```
atom        1    OH    OT OT      15.99899959564209 -0.830    OH
atom        2    OH    HT HT      1.0080000162124634 0.415    OH
atom        3    OH    HT HT      1.0080000162124634 0.415    OH
```

```
bond    1    2
bond    1    3


angle    2    1    3
```

5 Parameter Database

The basic parameter database (typically named parm.database) will contain sections listing pair-wise interations, bonds, angles, dihedrals, and impropers with their parameters for all the unique types of these interactions. The commands are: 'pair' command withformat:

pair (atom type 1) (atom type 2) $< parameterargs >$

'bond' command withformat:

bond (atom type 1) (atom type 2) $< parameterargs >$

'angle' command withformat:

angle (atom type 1) (atom type 2) (atom type 3) $< parameterargs >$

'dihedral' command withformat:

dihedral (atom type 1) (atom type 2) (atom type 3) (atom type 4) $< parameterargs >$

'improper' command withformat:

improper (atom type 1) (atom type 2) (atom type 3) (atom type 4) $< parameterargs >$

A simple example is the database for CHARMM water (note that this corresponds to the water topology example and that the atom types match the definitions in the topology):

```
pair  OT    OT     0.1020   3.1880
pair  HT    OT    -0.0000   1.5940
```

```
pair  HT    HT     0.0000   0.0000

bond   OT   HT       450.00 0.9572

angle  HT   OT   HT  55.000 104.52 0.0 0.0
```

In addition to the basic commands, the parsing of the parameter database has been updated and some additional database commands added. The LAMMPS styles can now be explicitly defined in the paramter database using the new style commands :

pstyle	The pstyle command is used to explicitly define the LAMMPS pair_style that is to be used. e.g. 'pstyle lj/cut 2.5' will generate the generate the line 'pair_style lj/cut 2.5' in the PARM.FILE. DEFAULT='lj/cut/coul/long 10'
bstyle	The bstyle command is used to explicitly define the LAMMPS bond_style that is to be used. e.g. 'bstyle harmonic' will generate the generate the line 'bond_style harmonic' in the PARM.FILE. DEFAULT='harmonic'
astyle	The astyle command is used to explicitly define the LAMMPS angle_style that is to be used. e.g. 'astyle harmonic' will generate the generate the line 'angle_style harmonic' in the PARM.FILE. DEFAULT='charmm'
dstyle	The dstyle command is used to explicitly define the LAMMPS dihedral_style that is to be used. e.g. 'dstyle harmonic' will generate the generate the line 'dihedral_style harmonic' in the PARM.FILE. DEFAULT='charmm'
istyle	The istyle command is used to explicitly define the LAMMPS improper_style that is to be used. e.g. 'istyle harmonic' will generate the generate the line 'improper_style harmonic' in the PARM.FILE. DEFAULT='harmonic'
kstyle	The kstyle command is used to explicitly define the LAMMPS kspace_style that is to be used. e.g. 'kstyle pppm 0.0001' will generate the generate the line 'kspace_style pppm 0.0001' in the PARM.FILE. DEFAULT='ewald 0.0001'
sbonds	The sbonds command is used to explicitly define the LAMMPS special_bonds style that is to be used. e.g. 'sbonds lj/coul 0 1 1 extra 2' will generate the generate the line 'special_bonds lj/coul 0 1 1 extra 2' in the PARM.FILE. DEFAULT1(if number of impropers=0)='lj 0.0 0.0 0.0' DEFAULT2(if number of impropers>0)='lj 0.0 0.0 1.0'

The number of parameter arguments for each type of interaction (pair, bond, etc.) can now also be explicity defined (with a maximum of 10 arguments) using the new commands:

npairargs	npairargs is used before a set of 'pair' listings to set the number of parameter arguments to be used for that set of pair interactions. e.g. 'npairargs 4' sets the number of parameter arguments to 4. All subsequent 'pair' definitions will be expected to have 4 parameters listed. Multiple npairargs commands can be defined throughout the pair listings, which could be used for hybrid systems that require different numbers of parameters for the different pair interaction styles that are used. DEFAULT=2
nbondargs	nbondargs is used before a set of 'bond' listings to set the number of parameter arguments to be used for that set of bond interactions. e.g. 'nbondargs 3' sets the number of parameter arguments to 3. All subsequent 'bond' definitions will be expected to have 3 parameters listed. Multiple nbondargs commands can be defined throughout the bond listings, which could be used for hybrid systems that require different numbers of parameters for the different bond styles that are used. DEFAULT=2
nangleargs	nangleargs is used before a set of 'anlge' listings to set the number of parameter arguments to be used for that set of anlge interactions. e.g. 'nanlgeargs 2' sets the number of parameter arguments to 2. All subsequent 'angle' definitions will be expected to have 2 parameters listed. Multiple nanlgeargs commands can be defined throughout the angle listings, which could be used for hybrid systems that require different numbers of parameters for the different bond styles that are used. DEFAULT=4
ndihedralargs	ndihedralargs is used before a set of 'dihedral' listings to set the number of parameter arguments to be used for that set of dihedral interactions. e.g. 'ndihedralargs 3' sets the number of parameter arguments to 3. All subsequent 'dihedral' definitions will be expected to have 3 parameters listed. Multiple ndihedralargs commands can be defined throughout the dihedral listings, which could be used for hybrid systems that require different numbers of parameters for the different dihedral styles that are used. DEFAULT=2
nimproperargs	nimproperargs is used before a set of 'improper' listings to set the number of parameter arguments to be used for that set of improper interactions. e.g. 'nimproperargs 3' sets the number of parameter arguments to 3. All subsequent 'improper' definitions will be expected to have 3 parameters listed. Multiple nimproperargs commands can be defined throughout the improper listings, which could be used for hybrid systems that require different numbers of parameters for the different improper styles that are used. DEFAULT=2

The water example can be updated to use some of the new commands :

```
npairargs 2
```

```
pair  OT    OT     0.1020   3.1880
pair  HT    OT    -0.0000   1.5940
pair  HT    HT     0.0000   0.0000

nbondargs 2
bond   OT   HT       450.00 0.9572
nanagleargs 2
angle  HT   OT   HT  55.000 104.52
```

This water example is updated to use hybrid pair interactions :

```
pstyle hybrid lj/cut/coul/long 10 table linear 1000

npairargs 5
pair  OT    OT  lj/cut/coul/long   0.1020   3.1880  10.0 15.0
npairargs 3
pair  HT    OT  table   tableHT_OT.tab ENTRY1
pair  HT    HT  table   tableHT_HT.tab ENTRY1

nbondargs 2
bond   OT   HT       450.00 0.9572
nanagleargs 2
angle  HT   OT   HT  55.000 104.52
```

Note that in the hybrid system the style should be listed as the first parameter argument.

6 Coordinate Files

setup_lammps_v2 is able to read three different coordinate file types. They are pdb, xyz, and gcd. The pdb and xyz formats are standard formats. However, the gcd is a file with only coordinates. It is the simplest coordinate file type with format:

```
x1 y1 z2
x2 y2 z2
........
........
xn yn zn
```

Only the pdb format contains readable box size data. If the box sizes are not defined in the pdb file, or if another coordinate type is used, setup_lammps_v2 uses a simple routine to guess the box sizes using the provided coordinates plus some additional padding.