LAMMPS Learning Manual

— with other commands like Linux and VMD

Yusi Zhou

March 18, 2018

Contents

1 Getting Started

1.1 LAMMPS Tarball

Download LAMMPS tarball in http://lammps.sandia.gov/ and unzip it by

```
gunzip lammps*.tar.gz
tar xvf lammps*.tar
```

1.2 Compiler Preparation

First, make sure the g++ and FORTRAN77 compiler have been installed in the Linux system. They can be installed by

```
sudo apt-get install g++
sudo apt-get install fort77 (gfortran for Fortran95)
```

1.3 MPICH2 Install

Installing MPICH2 by

```
sudo apt-get install mpich
```

1.4 FFT Install

Fast Fourier transform should also be installed before making LAMMPS. Typing

```
./configure
```

to make Makefile files. Then type

```
make install
```

or

```
make
```

to install the FFT. (The VMD, and other softwares, like PackMol, are installed in the same way)

1.5 LAMMPS Install

After installing MPI, if you want to make with some packages, type

```
make package-status
```

to check the status of packages. Then type

```
make yes-PACKAGE-NAME
```

to install the package like

```
make yes-USER-REAXC
```

Then you can typically use the Makefile.mpi file in src/MAKE by typing the following line (from the ../lammps-version/src dir):

```
make -j P mpi
```

Then a lmp_mpi file will be created in src dir. Copy it to users bin dir to run it by the following command:

```
sudo cp lmp_mpi /usr/local/bin/
```

Then you can run LAMMPS by:

```
lmp_mpi < in.file
mpirun -n P lmp_mpi < in.file
```

where P is the number of processors used.

2 Input Script

3 Data File

4 ReaxFF

4.1 pair_style reax/c

Syntax:

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

cfile = NULL or name of a control file

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 value = factor used for array allocation

mincap value = minimum size for array allocation

Examples:

4.2 fix reax/c/bonds

Syntax:

```
fix ID group-ID reax/c/bonds Nevery filename
```

Description:

Write out the bond information (instantaneous) computed by the ReaxFF potential. The output file format is listed below:

```
ID type nb id_1 ... id_nb mol bo_1 ... bo_nb abo nlp q
```

The *ID* represents the exact ID of atom by read_data command (or other atom created commands), which is the same as *type*. The *nb* is the number of atoms bonding with this atom (the ID). The *id_i* is the ID of atoms bonding. The *mol* is what I don't know now. The *bo_i* is the bond order. The *abo* is the accumulated bond order (i.e. the sum of all individual bond orders). The *nlp* is referring to long pairs (Actually, I don't what this means.).

In ../LAMMPS-version/tools/reax/ dir, the mol_fra.c file provides a C program which can read the output bonds file and export some files contains the same contents as the output file of fix reax/c/species command, using the same cutoff parameters. See more details about *cutoff* in fix reax/c/species command.

4.3 fix reax/c/species

Syntax:

```
fix ID group-ID reax/c/species Nevery Nrepeat Nfreq filename keyword value ...
```

keyword = *cutoff* or *element* or *position*

```
cutoff value = I J Cutoff
   I, J = atom types
   Cutoff = Bond-order cutoff value for this pair of atom types
element value = Element1, Element2, ...
position value = posfreq filepos
   posfreq = write position files every this many timestep
   filepos = name of position output file
```

Examples:

```
fix 1 all reax/c/species 1 100 100 species.out element H O Ni cutoff 3 3 0.9
fix 1 all reax/c/species 1 2 20 species.out position 1000 NiH2O.pos
```

Description:

Write out the chemical species information (either averaged or instantaneous, depending on value of *Nrepeat*) computed by the ReaxFF potential.

5 VMD Commands

5.1 Rotate

Rotate xyz axes

```
rotate x/y/z by DEGREES
```

like

```
rotate y by 90
```

command can rotate y axes by 90° anticlockwise.

6 Linux

6.1 root

Type command

```
sudo su
```

Then enter password.

6.2 Path Environment

Check path environment:

```
echo $PATH
```

There are 3 ways to edit path environment in Linux. (Assuming you want to add path ” /zhouys/bin” to the environment.)

1. `PATH=$PATH: /zhouys/bin`

   This way is only valid in current shell. That is, the path will be reset when you exit the current command window.

2. `vi /etc/profile`

   Then add ”PATH=$PATH: /zhouys/bin” to the file in proper location. This way can change the path environment permanently after you restart the Linux system. This method is strongly recommended.

3. `vi /.bash_profile`

   Edit the PATH line. Unfortunately, I have never succeeded by this method.

6.3 Shell Script

Use shell file to execute a list of commands.

The .sh file must have the first line like ”#! /bin/bash” or ”#! /bin/sh”. This line tells which shell to execute the .sh file. Then in the folder containing this .sh file, type ”source text.sh” to execute this shell file. (There suppose the file name is ”text.sh”.)

For example, the ”text.sh” file contains

```
#!  /bin/bash
cd moltemplate_files
  moltemplate.sh system.lt
  mv -f system.data system.in* ../
cd ../
mpirun -n 16 lmp_mpi < run.in.npt
cp log.lammps log.lammps.npt
mpirun -n 16 lmp_mpi < run.in.nvt
```

Then in the right file, execute the command

```
source text.sh
```

The script will be executed. Find these related example files in the moltemplate examples folder.

6.4 Update gcc Version

Using ppa to update gcc. The method is from link:
blog.csdn.net/gatieme/artical/details/52871438.

1. Add this ppa to your system.

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
```

2. Update gcc.

```
sudo apt-get install gcc-|version| g++-|version|
```

Like,

```
sudo apt-get install gcc-4.9 g++-4.9
```

3. Update db and locate to find installing version of gcc.

```
sudo updatedb && sudo ldconfig
locate gcc | grep -E "/usr/bin/gcc-[0-9]"
locate g++ | grep -E "/usr/bin/g\+\+-[0-9]"
```

4. Switch the default version of gcc. After doing 1-3 steps, the default version of gcc of the system is still old version. Using update-alternatives command to build link gcc command with new version of gcc.

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-|old-v| 20
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-|new-v| 50
```

20 and 50 are priorities of the version. Query the version and check the priority using this command:

```
sudo update-alternatives --query gcc
```

Type same command to update g++ version.