Creating Input Files for Molecular Simulations in LAMMPS Using the Automated Topology Builder and Moltemplate

David J. Keffer
dkeffer@utk.edu
document begun: March 2, 2022
document last updated: March 3, 2022

Table of Contents

I. Objective of this Document

The objective of this document is to provide a working example of creating an input configuration of a system of molecules in LAMMPS.

II. Acquire Initial Structure

We will obtain our initial structure from the NIST Chemistry webbook located at https://webbook.nist.gov/chemistry/. We search by formula for "C2H6" and select "ethane" for this example. On the "ethane" page, located at https://webbook.nist.gov/cgi/cbook.cgi?ID=C74840&Units=SI, we click the text marked "3d SD file" to download the structure file of ethane in sdf format. This file is named "74-84-0-3d.sdf" The initial contents of this file are shown in Table 1.

```
First line is blank!  Must add something here.
  NIST    07011509303D 1   1.00000   -79.83042
Copyright by the U.S. Sec. Commerce on behalf of U.S.A. All rights reserved.
  8  7  0  0  0  0  0  0  0  0999 V2000
    0.5397    1.7666   -0.0025 H   0  0  0  0  0  0  0  0  0  0  0  0
    0.7166    0.8980    0.6425 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.4899    0.0005    0.0551 H   0  0  0  0  0  0  0  0  0  0  0  0
   -0.0078    0.9452    1.4640 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.1541    0.8745    1.1659 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.3313    0.0053    1.8100 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.8785    0.8284    0.3444 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.3805    1.7715    1.7542 H   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  2  4  1  0  0  0  0
  2  5  1  0  0  0  0
  5  6  1  0  0  0  0
  5  7  1  0  0  0  0
  5  8  1  0  0  0  0
M  END
```

Table 1. Initial file contents of 74-84-0-3d.sdf. This content is followed by meta-data describing generation of the document.

For further understanding of the sdf file format, see for example http://www.nonlinear.com/progenesis/sdf-studio/v0.9/faq/sdf-file-format-guidance.aspx.

Importantly, NIST leaves the first comment line blank. The subsequent steps require some content in the first line. I added "ethane" to the first blank line in Table 1.

III. Obtain Script to Generate LAMMPS Configuration and Potential Parameters

To generate the LAMMPS input, we first visit the "Automated Topology Builder" (ATB) at https://atb.uq.edu.au/. You have to first register to use this site. I began by searching for a structure. I chose sdf as the format. I left the charge as *. I pasted the contents of Table 1. (including the altered non-blank first line) into the text box for the structure file. I hit the submit button.

In the search results page, I click the "Show Molecule Page" button for the first result.

I next click on the "Molecular Dynamics (MD) Files" tab. I set the format to LAMMPS. I download two files from this page: "GROMOS_54A7_ATB moltemplate forcefield file", which I save as GROMOS_54A7_ATB.lt and All-Atom moltemplate file (optimized geometry), which I save as "ethane_allatom_optimizedgeometry.lt". The forcefield file contains the parameters for the GROMOS interaction potential and is not specific to ethane. The configuration file is specific to the molecule. In Table 2, we show the contents of the structure file, excluding the initial contents. This file contains the information regarding atom position, charge, bonds, angles and dihedral torsional angles.

```
TBEC inherits GROMOS_54A7_ATB {
    write("Data Atoms"){
      $atom:H06    $mol:... @atom:HC      0.016000 -1.1637645986E+00  5.4624780310E-01  8.6335948810E-01
      $atom:C01    $mol:... @atom:C      -0.048000 -7.6523644520E-01 -2.3093000000E-06  4.1210700000E-05
      $atom:H04    $mol:... @atom:HC      0.016000 -1.1640659528E+00  4.7451593380E-01 -9.0459336720E-01
      $atom:H05    $mol:... @atom:HC      0.016000 -1.1641697310E+00 -1.0207261997E+00  4.1221629300E-02
      $atom:C02    $mol:... @atom:C      -0.048000  7.6525137550E-01  1.6809600000E-05 -7.9896900000E-05
      $atom:H01    $mol:... @atom:HC      0.016000  1.1640938687E+00 -4.7437709060E-01  9.0461228690E-01
      $atom:H02    $mol:... @atom:HC      0.016000  1.1637946045E+00 -5.4624980750E-01 -8.6337852290E-01
      $atom:H03    $mol:... @atom:HC      0.016000  1.1642139073E+00  1.0207321415E+00 -4.1346282800E-02
    }
    write("Data Bonds"){
        $bond:b1      @bond:g3     $atom:H06    $atom:C01
        $bond:b2      @bond:g3     $atom:C01    $atom:H04
        $bond:b3      @bond:g3     $atom:C01    $atom:H05
        $bond:b4      @bond:g27    $atom:C01    $atom:C02
        $bond:b5      @bond:g3     $atom:C02    $atom:H01
        $bond:b6      @bond:g3     $atom:C02    $atom:H02
        $bond:b7      @bond:g3     $atom:C02    $atom:H03
    }
    write("Data Angles"){
        $angle:a1      @angle:g43    $atom:H06    $atom:C01    $atom:H04
        $angle:a2      @angle:g43    $atom:H06    $atom:C01    $atom:H05
        $angle:a3      @angle:g44    $atom:H06    $atom:C01    $atom:C02
        $angle:a4      @angle:g43    $atom:H04    $atom:C01    $atom:H05
        $angle:a5      @angle:g44    $atom:H04    $atom:C01    $atom:C02
        $angle:a6      @angle:g44    $atom:H05    $atom:C01    $atom:C02
        $angle:a7      @angle:g44    $atom:C01    $atom:C02    $atom:H01
        $angle:a8      @angle:g44    $atom:C01    $atom:C02    $atom:H02
        $angle:a9      @angle:g44    $atom:C01    $atom:C02    $atom:H03
        $angle:a10     @angle:g43    $atom:H01    $atom:C02    $atom:H02
        $angle:a11     @angle:g43    $atom:H01    $atom:C02    $atom:H03
        $angle:a12     @angle:g43    $atom:H02    $atom:C02    $atom:H03
    }
    write("Data Dihedrals"){
        $dihedral:d1      @dihedral:glj14  $atom:H06    $atom:C01    $atom:C02    $atom:H01
        $dihedral:d2      @dihedral:glj14  $atom:H06    $atom:C01    $atom:C02    $atom:H02
        $dihedral:d3      @dihedral:glj14  $atom:H06    $atom:C01    $atom:C02    $atom:H03
        $dihedral:d4      @dihedral:g34    $atom:H04    $atom:C01    $atom:C02    $atom:H01
        $dihedral:d5      @dihedral:glj14  $atom:H04    $atom:C01    $atom:C02    $atom:H02
        $dihedral:d6      @dihedral:glj14  $atom:H04    $atom:C01    $atom:C02    $atom:H03
        $dihedral:d7      @dihedral:glj14  $atom:H05    $atom:C01    $atom:C02    $atom:H01
        $dihedral:d8      @dihedral:glj14  $atom:H05    $atom:C01    $atom:C02    $atom:H02
        $dihedral:d9      @dihedral:glj14  $atom:H05    $atom:C01    $atom:C02    $atom:H03
    }
    write("Data Impropers"){
    }
}
```

Table 2. File contents of ethane_allatom_optimizedgeometry.lt, excluding the initial contents.

IV. Install MolTemplate

The scripts generated by ATB are intended to be used with the software MolTemplate, located at https://www.moltemplate.org/. On ISAAC, users must install their own version of MolTemplate. Here are the instructions for the one time installation of MolTemplate.

1) logon to ISAAC
2) to download moltemplate, type command:
   git clone https://github.com/jewettaij/moltemplate moltemplate
3) to load python, type command:
   module load python3/3.7.9
4) to create a python virtual environment, type command:
   python3 -m venv moltemplate-env
5) to activate python virtual environment, type command:
   source moltemplate-env/bin/activate
6) to install moltemplate, type command:
   pip3 install .

Table 3. Commands to install MolTemplate on ISAAC.

At this point, MolTemplate is installed. In the future, to use MolTemplate, you only have to type the following 2 commands in order to run MolTemplate.

```
module load python3/3.7.9
source moltemplate-env/bin/activate
```

Table 4. Commands to use MolTemplate after initial installation

V. Generate LAMMPS input files for simulation of a single molecule

Create a new directory to run this example. Place in this directory three files:

1) "ethane_allatom_optimizedgeometry.lt" (generated by ATB)
2) "GROMOS_54A7_ATB.lt" (generated by ATB)
3) system.lt

You must create the system.lt file. It is a text file. Sample contents are shown in Table 5. It is essential that the four letter sequence, TBEC in this example, that appears in the last line of system.lt (Table 5)

mol1 = new **TBEC**.move(0.0,0.0,0.0)

match exactly the same sequence that appears in the geometry file, in this example ethane_allatom_optimizedgeometry.lt, (Table 2)

TBEC inherits GROMOS_54A7_ATB {

```
# system.lt
# simulation of a single molecule in a box of size 50x50x50 Angstroms^3

write_once("Data Boundary") {
   -25.0 25.0 xlo xhi
   -25.0 25.0 ylo yhi
   -25.0 25.0 zlo zhi
}

write_once("In Init") {
  # a variable named `cutoff` is required by GROMOS_54A7_ATB.lt
  variable cutoff equal 14.0 # Angstroms
  boundary p p p
}

# import the forcefield file
import "GROMOS_54A7_ATB.lt"
# import molecule building block file
import "ethane_allatom_optimizedgeometry.lt"

# create a single copy of this molecule at position 0,0,0
mol1 = new TBEC.move(0.0,0.0,0.0)
```

Table 5. File contents of "system.lt" for a single molecule.

To run moltemplate, type the command:

```
moltemplate.sh -atomstyle "full" system.lt
```

You have now created the following files:
1) system.in
2) system.data
3) system.in.init
4) system.in.settings

"system.in" is the main LAMMPS input file. "system.in.init" is a module containing initial simulation settings. "system.in.settings" contains the forcefield parameters. "system.data" contains the configuration file with atom positions and molecule connectivity.

Note: Unfortunately, MolTemplate includes the potential parameters for all types of atoms, including those not found in the molecule. We shall address this issue later. As it is this simulation will run.

VI. Run a LAMMPS simulation of a single molecule

Copy the four files created in the previous step to a new directory for simulation.

1) system.in
2) system.data
3) system.in.init
4) system.in.settings

As created, the "system.in" file does not actually run a simulation. To run a simulation, append some content to the end of the "system.in" file. See Table 6. for an example of such commands.

```
variable tem      equal 300.0
variable timestep equal 1.0
variable tdampfac      equal 100.0
variable tdamp      equal ${timestep}*${tdampfac}
velocity    all create ${tem} 87287 dist gaussian
fix     1 all nvt temp ${tem} ${tem} ${tdamp}
thermo 10
dump 1 all atom 10 dump.lammpstrj
run 1000
```

Table 6. Example file contents to be appended to the end of "system.in".

Run LAMMPS as usual. An example sequence of commands to be issued after opening an interactive queue (sufficient for this toy example) is shown in Table 7.

```
module load lammps/30Apr19
mpirun -n 1 lmp_mpi < system.in > lammps.out
```

Table 7. Example commands to run LAMMPS.

At this point, you have a run a LAMMPS simulation of a single molecule.

VII. Generate LAMMPS input files for simulation of a system of molecules

This process is very similar to what was described for a single molecule. We repeat all the steps here to have this section be self-contained. Create a new directory to run this example. Place in this directory three files:

1) "ethane_allatom_optimizedgeometry.lt" (generated by ATB)
2) "GROMOS_54A7_ATB.lt" (generated by ATB)
3) "system.lt"

You must create the system.lt file. It is a text file. Sample contents are shown in Table 8. The contents of this file are different than what was used for the single molecule.

To run moltemplate, type the command:

```
moltemplate.sh -atomstyle "full" system.lt
```

You have now created the following files:
1) system.in
2) system.data
3) system.in.init
4) system.in.settings

```
# system.lt
# simulation of a single molecule in a box of size 50x50x50 Angstroms^3

write_once("Data Boundary") {
   -25.0 25.0 xlo xhi
   -25.0 25.0 ylo yhi
   -25.0 25.0 zlo zhi
}

write_once("In Init") {
  # a variable named `cutoff` is required by GROMOS_54A7_ATB.lt
  variable cutoff equal 14.0 # Angstroms
  boundary p p p
}

# import the forcefield file
import "GROMOS_54A7_ATB.lt"
# import molecule building block file
import "ethane_allatom_optimizedgeometry.lt"

# Create grid of molecules

grid = new TBEC [5].move(10.0,   0,     0 )
                [5].move(0.0, 10.0,   0 )
                [5].move(0.0, 0.0, 10.0)

# Optional: Center the grid at the origin.

grid[*][*][*].move(-25.0, -25.0, -25.0)
```

Table 8. File contents of "system.lt" for a system of 125 (5^3) molecules.

VIII. Run a LAMMPS simulation of a system of molecules

This step is exactly the same as what was done for a single molecule. Copy the four files created in the previous step to a new directory for simulation.

1) system.in
2) system.data
3) system.in.init
4) system.in.settings

As created, the "system.in" file does not actually run a simulation. To run a simulation, append some content to the end of the "system.in" file. See Table 6. for an example of such commands. See Table 7. for the commands to run LAMMPS.

IX. Getting Rid of Unnecessary Atom Types and Potential Information

MolTemplate has a tool to clean up the input files that it generates, removing unnecessary atom types and potential information. The script to execute this command is

cleanup_moltemplate.sh

It should be executed after the moltemplate.sh script.

There are two problems in using the cleanup_moltemplate.sh script with files generated by the automated topology builder. Below we identify the problems and provide work arounds for both issues.

First, the clean up does not allow variable declarations as appear in the "system.in.init" file.

The old system.in.init file is shown in Table 9.

```
# a variable named `cutoff` is required by GROMOS_54A7_ATB.lt
variable cutoff equal 14.0 # Angstroms
boundary p p p
    units real
    atom_style      full
    bond_style      harmonic
    angle_style     harmonic
    dihedral_style  harmonic
    improper_style  harmonic
    #pair_style      lj/cut/coul/cut ${cutoff}  # for non-period sims
    pair_style      lj/cut/coul/long   ${cutoff}
    kspace_style    pppm 0.0001
    special_bonds lj 0.0 0.0 0.5 coul 0.0 0.0 1.0 angle yes dihedral yes
```

Table 9. File contents of "system.in.init" as generated.

The new system.in.init file is shown in Table 10. We commented out the variable cutoff line and hard-coded in the value in the pair_style command.

```
# a variable named `cutoff` is required by GROMOS_54A7_ATB.lt
# variable cutoff equal 14.0 # Angstroms
boundary p p p
    units real
    atom_style      full
    bond_style      harmonic
    angle_style     harmonic
    dihedral_style  harmonic
    improper_style  harmonic
    #pair_style      lj/cut/coul/cut ${cutoff}  # for non-period sims
    pair_style      lj/cut/coul/long   14.0
    kspace_style    pppm 0.0001
    special_bonds lj 0.0 0.0 0.5 coul 0.0 0.0 1.0 angle yes dihedral yes
```

Table 10. File contents of "system.in.init" after modification.

With these changes, the clean up script will run

cleanup_moltemplate.sh

This script modifies the four input files and outputs them with the same file names:

1) system.in
2) system.data
3) system.in.init
4) system.in.settings

This process eliminates the definition of the masses. Currently, we manually fix the problem by adding the mass statements to the end of the "system.in". We added two lines to the beginning of the file shown in Table 6. resulting in the file shown in Table 11.

```
    mass 1          12.011000000
    mass 2          1.008000000

variable tem      equal 300.0
variable timestep equal 1.0
variable tdampfac      equal 100.0
variable tdamp      equal ${timestep}*${tdampfac}
velocity    all create ${tem} 87287 dist gaussian
fix     1 all nvt temp ${tem} ${tem} ${tdamp}
thermo 10
dump 1 all atom 10 dump.lammpstrj
run 1000
```

Table 11. File contents of "system.in".

These resulting four files are ready to be run in LAMMPS.