Problems, Bugs & Errors

Molecular and Business Modelling

UCL NatSci Innovation Lab 2020

Lau Pak To (Ryan)
September 2020

Contents

For more information, please go to the LAMMPS official website:
http://lammps.sandia.gov/
Please do also check out the relevant documentations on errors:
https://lammps.sandia.gov/doc/Errors.html
If you run into problems, Google your way out. The internet is your best friend.

1 Introduction

1.1 Errors are inevitable

When you are programming using languages like Python or C++, you may encounter different problems and errors. It is extremely common. Running LAMMPS scripts is no different. We, too, struggled a lot during the process. LAMMPS will print an error message and stop, or a warning message and continue the process. These messages will give you ideas on what went wrong and what has to be changed. Do not be afraid to make mistakes. Just like learning any other skills, failing and making mistakes are part of the learning process.

1.2 About this document

To troubleshoot, or to solve a problem, always remember these 3 key steps:
1. Get information; 2. Find the root cause; 3. Find ways to avoid the same problem in the future.
This document will provide you with some ideas on some of the problems, errors, bugs you may encounter while running LAMMPS. It will also introduce and guide you through the resources on LAMMPS errors found on the official website:
https://lammps.sandia.gov/doc/Errors.html

2 Common Problems

https://lammps.sandia.gov/doc/Errors_common.html
This page acts as an introduction to the errors documentation in the LAMMPS website.
The page also provides some useful examples about similar problems that you may encounter while running the simulation.

2.1 Overview

A LAMMPS simulation has two stages: setup and run. Most LAMMPS errors occur during the setup stage, while simulation errors(e.g. bonds stretching too far) will occur in the run stage. The application flags and prints informative error messages. (see part 5 later on)
Do be careful that LAMMPS cannot spot physics or numerical mistakes

2.2 Log files

If you are using Bash or Terminal to run the file, the error will be printed on the interface. Another method to check for errors is to use the log.lammps file. You may find some of the examples of log.lammps file in the "examples" files in the GitHub resouce of LAMMPS: https://github.com/lammps/lammps.git
The logfile will output all the bits and bobs LAMMPS has run, and show any error or warning messages it has encountered. You can read more about logfiles here:
https://lammps.sandia.gov/doc/Run_output.html

3 Bugs

https://lammps.sandia.gov/doc/Errors_bugs.html This page explains the processes if you are confident that you have found a bug, or an unexpected error/result.

3.1 A summary of things you can do

1. Check "New features and bug fixes" section.
   https://lammps.sandia.gov/bug.html
2. Check if the scripts can be run using the latest version of LAMMPS.
3. Check your code with the documentations of LAMMPS.
4. Check the GitHub Issue page(https://github.com/lammps/issues) after reporting the issue and it is still open.
5. Check the GitHub Pull Requests page
   (https://github.com/lammps/pulls) if the bug is already fixed.
6. Check whether your bug has already been discussed in the mailing list archives: https://lammps.sandia.gov/mail.html

If none of these are useful, it is advised that you file a new bug report on the GitHub Issue page(https://github.com/lammps/issues).

3.2 Isolating the problem

A quick tip to help yourself AND developers verify and fix a bug is to try isolate the problem.
For example, try running your script using the smallest number of atoms possible, and try to reproduce the bug. You may be able to identify what went wrong.

4 Crashes

https://lammps.sandia.gov/doc/Errors_debug.html
Following up section 3, this section introduces ways to debug the issue by allowing us to understand the root cause of the crash and fix the problem. Examples, run procedures and guides can be found in the link above.

4.1 Using GDB debugger

Read more on the GDB debugger here:
http://sourceware.org/gdb/current/onlinedocs/gdb/
You can choose to run LAMMPS using the GDB debugger. This tool can help you identify where your errors are in your scripts. It will print the lines the LAMMPS executable is running/running. Once it reaches an error, it will print out a message why the run has stopped.

4.2 Using Valgrind

Read more on Valgrind here: https://valgrind.org/
Similarly, Valgrind can help you pinpoint mistakes in your LAMMPS code. Valgrind causes the execution to run slower, but it gives you a more specific hint of what went wrong. It is also recommended to redirect the valgrind output to another *.txt file, similar to producing a log file.

5 Error Messages

Error messages documentation:
https://lammps.sandia.gov/doc/Errors_messages.html
As from section 2.1, the LAMMPS executable prints the last input script command that it was processing, and prints error messages, then stops the running process. The error messages are very informative and will tell you what went wrong.

The linked documentation is a whole alphabetical list of ERROR messages LAMMPS prints out, as well as their respective reasons.
Do note that error messages which come from the user-contributed packages(https://lammps.sandia.gov/doc/Packages_user.html) are not listed here.

5.1 Self-explanatory messages

As it says, the error message explains itself. For example, [Invalid keyword in dump cfg command]; [Invalid shape in set command]; [Compute ID for fix ave/atom does not exist]; [Temperature control can not be used with fix nph/body]; [Fix rigid file has no lines]; and many more.

5.2 Other messages

Some messages are not self-explanatory. The errors documentation will come in handy. For example:

Error message	Explanation
Fix nve/body requires bodies	This fix can only be used for particles that are bodies.
Could not find change_box group ID	Group ID used in the change_box command does not exist.
Fix_modify temperature ID does not compute temperature	The compute ID assigned to the fix must compute temperature.
Improper_coeff command when no impropers allowed	The chosen atom style does not allow for impropers to be defined.
Per-atom energy was not tallied on needed timestep	You are using a thermo keyword that requires potentials to have tallied energy, but they did not on this timestep. See the variable doc page for ideas on how to make this work.

6 Warning Messages

Warning messages documentation:
https://lammps.sandia.gov/doc/Errors_warnings.html
As we have seen, LAMMPS will also produce warning messages. The program will continue to run, unlike when errors occur.
Just like section 5, the warning messages documentation provides a whole alphabetical errors list and their explanations.
Do also note that error messages which come from the user-contributed packages(https://lammps.sandia.gov/doc/Packages_user.html) are not listed here.

6.1 Self-explanatory messages

As it says, the error message explains itself. For example, [Angle style in data file differs from currently defined angle style]; [Charges are set, but coulombic solver is not used]; [KIM Model does not provide 'forces'; Forces will be zero]; [Simulations might be very slow because of large number of structure factors]; [Too many inner timesteps in fix ttm]; and vice versa.

6.2 Other messages

Some messages are not self-explanatory. The errors documentation will come in handy. For example:

Warning message	Explanation
Restart file used different newton pair setting, using input script value.	The input script value will override the setting in the restart file.
One or more respa levels compute no forces.	This is computationally inefficient.
More than one compute centro/atom.	It is not efficient to use compute centro/atom more than once.
Fix srd particles may move > big particle diameter.	This may cause accuracy problems.
Angles are defined but no angle style is set.	The topology contains angles, but there are no angle forces computed since there was no angle_style command.

7 Errors we encountered

While we are trying to create and run the data file, we encountered some errors. The table below shows the errors we ran into, its explanations and how we tackled it. Input and data files are generated using the C++ code produced by our teammate, Lukas. Be sure to check that out.

Error message	Explanation	Solution(s)
Bond atoms %d %d missing on proc %d at step %ld	The second atom needed to compute a particular bond is missing on this processor. Typically this is because the pairwise cutoff is set too short or the bond has blown apart and an atom is too far away.	Increasing the neighbor cutoff
Lost atoms: original %ld current %ld	Lost atoms are checked for each time thermo output is done. Lost atoms usually indicate bad dynamics.	N/A
Out of range atoms - cannot compute PPPM	It is not efficient to use compute centro/atom more than once.	In the LAMMPS script, using the following code: neigh_modify delay 0 every 1 check yes
Cannot use non-periodic boundaries with PPPM	For kspace style pppm, all 3 dimensions must have periodic boundaries unless you use the kspace_modify command to define a 2d slab with a non-periodic z dimension.	Can be fixed by either not using kspace, or switching to periodic boundary conditions, which is more practical.