1 Introduction

This code is primarily written to ***visualize*** thermodynamic data which LAMMPS output in `log` files for its ***quick analysis***, ***after*** as well as ***during the simulation***. For example, this could be used to visually observe trends in different thermodynamic outputs such as `computes`, `variables`, `fixes`, which can be output through `thermo_style` command. The code works with both `line` and `multi` format of `thermo_style` as well as with `thermo_style custom` in an user friendly manner. The code essentially writes a file `temp`, compatible with Xmgrace and then plots it in Xmgrace in the final stage of its execution.

2 Compilation

```
g++ -O3 -Wno-deprecated thermo_extract.cpp -lm -o therm
```

Move the executable to `/usr/local/bin` or any other directory which is in your path (or add the current directory to your $PATH) so that it can be accessed easily from any directory.

3 Running the program

The syntax to run the executable is

```
therm path_to_logfile
```

To read data from multiple `log` files, the syntax is

```
therm path_to_logfile1 path_to_logfile2 path_to_logfile3 ... ...
```

or if all the log files are in the same directory and are named `log.1, log.2, log.3` etc., they can be read in short as

```
therm log.*
```

3.1 User interface

After successful reading of files, the terminal interface should look like

```
Files Read: ---
Reading file    : log.npt

Select parameters to plot from following: ---
Step            TotEng(te)      KinEng(ke)      Temp(t)         PotEng(pe)      E_bond(be)
E_angle(ae)     E_dihed(de)     E_impro         E_vdwl(vd)      E_coul(ce)      E_long(le)
Press(p)        Volume(v)
--------------

Please enter your parameters in the following format: ---
1. First string should be 'p' or 's' for plotting parameters from different files in
parallel or series
2. Please enter one or more parameters listed above after entering the first string.
3. For certain parameters, short forms may be used as listed in bracket above
   ----- Example: To plot total energy, please type 'p te'.
4. In order to plot one parameter with respect to other, please type 'cross' after
p/s followed by two parameters
5. The independent parameter could be 'Step' as mentioned above.
6. Currently, only two parameters can be plotted with respect to each other.
   ----- Example: To plot temperature vs. pressure p cross t p
--------------

For analysis, please exit and use -analysis flag with filename. The file should contain
at least one of these commands: ---
1. average     all/single_filename startvalue(start) endvalue(end)
2. scale       filename property factor
3. inverse     filename property
```

```
4. write       Under Coding
5. subtract    property1 property2 filename outputpropertyname
6. add         Under Coding
--------------

Enter the parameter(s) of which you want the plot for files, read successfully:
```

Although this interface is self-explanatory by large, brief summary is provided below. First it shows which files have been read successfully through `"Files Read:"` portion. Then, it shows the available parameters which can be plotted (which in current case are generic `thermo_style multi` option parameters). The code automatically scans for `thermo_style` parameters within the `log` file and prints them out for you. Then, it lists the brief description of different options which are available for input.

3.2 Entering required input

The code allows the multiple files to be read either in **series** or in **parallel**

- **Series:** This means that the similar data (like `temperature`) of multiple `log` files will be stored in a single vector in a sequence, corresponding to sequence of input files. This will be plotted as single entity in Xmgrace. For such case an abbreviated letter `"s"` is used as an input. For example, it could be used to plot a parameter from a set of simulations that were run successively and the output was stored in separate files.
- **Parallel:** This means the similar data (like `temperature`) of multiple files will be stored in different vectors and will be plotted as different entities (in different colors) in Xmgrace. For such case an abbreviated letter `"p"` is used as an input.

Before we proceed further, I would like to mention that the code also recognizes abbreviated codes for several popular parameters. These are given in "parenthesis" right next to corresponding parameters. For example, `t`, `p`, `ke`, and `v` can be written instead of `Temp`, `Press`, `KinEng`, and `Volume`, respectively. Here are few examples along with their explanations.

```
p t
```

will plot `temperature` as a function of **number of points (not MD time)** from the log file(s). If two or more files were read, it will plot the `temperature` for all separate files in different colors.

```
p p
```

will plot `pressure` as a function of number of points from the log file(s). If two or more files were read, it will plot the `pressure` for all separate files in different colors.

```
p te vd ke
```

will plot `total energy`, `van der Waals energy` and `kinetic energy` as a function of number of points from the log file(s). For example, if two files are read, it will result in 6 different plots in Xmgrace, i.e., 3 plots for each file.

```
s t
```

will plot `temperature` in a series manner, i.e., if you have more than one file, it will plotted as a single entity with respect to number of points from both files.

```
s te vd
```

will plot `total energy` and `van der Waals energy` in series in two plots.

3.3 Few other features

- The code plots the datapoints not only from finished runs but also from simulations currently being run. In a half-complete file (*simulation being run*), it rejects the last line of the `log` file and plots the rest.
- The code also puts out appropriate legend information for each data set in the Xmgrace automatically. For "parallel" case, it is `"filename:parameter"`. For "series" case it is just `"parameter"`.

- Another functionality of the code is in `"cross"` option at input line. This option can be used to plot one quantity with respect to another to observe some sort of possible correlations. Lets say if we want to plot pressure vs volume (although both are fluctuating significantly during the simulation), we will write at user input line,

  ```
  p cross p v
  ```

  This command will plot all the simulation data points of `pressure` (ordinate) with respect to `volume` (abscissa).

- As it was mentioned before that the X-axis is not `"time"` but `"number of points"`, it can easily be changed to time within Xmgrace graphical user interface by rescaling X-axis appropriately though `Data` → `Transformations` → `Geometric transform` → `Scale X` option.

3.4 Optional commands

Currently the codes have two optional command-line parameters that could be put in for special cases.

- **-leave option:** This option is used when certain sections of the `logfile` need to be skipped. For example, if the logfile contains data from a) minimization run followed by b) NVT simulation run followed by c) NPT simulation for X steps. If you want to plot `volume` in Xmgrace which only comes at NPT stage, it following command should be issued.

  `"therm -leave 2 logfile"` followed by `"p v"` when it asks for required input after it reads the `logfile`. This will skip first two runs from plotting and will only plot the 3^{rd} NPT run.

- **-anal option:** This option can be used to do some preliminary analysis on data, stored in `logfile`. The analysis, in principle, can also be done in Xmgrace itself (after code plots the data) using Xmgrace options through its graphical user interface. This part is not fully coded yet. In order to use this option, one needs to create a file with input analysis information stored in following syntax. These are all separate examples commands and can be written in any order or only one of them. Lets call the file `"analysis"`.

  ```
  average     all start end
  average     all 5000 end
  average     logfile 5000 10000

  scale       logfile Volume 0.001

  inverse     logfile Volume
  subtract    TotEng KinEng logfile Diff
  ```

  On the terminal, the command to be executed should be
  `therm -anal analysis logfile`

  When executed, the following set of commands in the `analysis` file will result in the following.

  - `average all start end`: This will print out average as well as standard deviation of all quantities (which are mentioned in input) for all files on the terminal. Here, the average will be done from the `start` to the `end` of the simulation data.
  - `average all 5000 end`: This will print out average as well as standard deviation of all quantities (which are mentioned in input) for all files on the terminal. However, in this case, the code will skip first 5000 data points and will printout average and standard deviation for the rest of the data.
  - `average logfile 5000 10000`: This will print out average as well as standard deviation of all quantities (which are mentioned in input) for `logfile` on the terminal. In this case, the code will only average data points from 5000 to 10000. This command used repeatedly could be employed for calculating block averages of various quantities.
  - `scale logfile Volume 0.001`: This command will scale the `Volume` data points in `logfile` by a factor of 0.001 and will plot the resultant quantity in Xmgrace.

- `inverse logfile Volume`: This command will inverse the `Volume` data points in `logfile` and plot the resultant quantity in Xmgrace.
- `subtract TotEng KinEng logfile Diff`: This command will take the difference `TotEng - KinEng` in `logfile` and plot the resultant quantity `Diff` in Xmgrace.

4 Changing the formatting of data in Xmgrace

The Xmgrace offers several plotting styles for fonts, data points representation, lines, axes representations, overall graph representations. The current code overwrites some of these representation features for a visual representation which I like better. However, one could easily change these properties by changing `symbol` and `line` properties in `plot_xmgrace()` function.

Contact Information:
In case of any questions, please contact Vikas Varshney @ vv0210_at_gmail_dot_com