The potential keyword

From GPUMD
Jump to navigationJump to search

Contents

Purpose

- This keyword is used to specify one potential for the system or part of the system. For a simple system, a single `potential` keyword is sufficient, but for a complicated system, one might need to use multiple `potential` keywords.

Grammar

- This keyword needs at least one parameter, `potential_filename`, which is the filename (including relative or absolute path) of the potential file to be used.

- If the potential specified in `potential_filename` is **not** a Lennard-Jones (LJ) potential, the grammar of the `potential` keyword is

```
potential potential_filename [list_of_types]
```

Here, `list_of_types` gives the list of atom types that will participate in the potential.

- If the potential specified in `potential_filename` **is** an LJ potential, the grammar of the `potential` keyword can be one of the following two:

```
potential potential_filename
potential potential_filename grouping_method
```

Here, `grouping_method` is a grouping method used to exclude "intra-material" LJ interactions. The meaning of "intra-material" will be illustrated in the following examples.

- If your system can be completely described by one potential, just use one `potential` keyword.

- If your system consists of more than one part and each part can be described by one potential and you also want to add an LJ potential between the different parts, you can use one `potential` keyword for one part of your system and an extra `potential` keyword for the LJ potential.

- To fully understand the usage of this keyword, it is helpful to check the following examples.

Examples

Silicon crystal

For a silicon crystal system described by a Tersoff potential, the potential for this system can be specified by using a single `potential` keyword:

```
potential Si_Tersoff.txt 0
```

Here, `Si_Tersoff.txt` should be a potential file for the Tersoff potential and `0` means that the potential applies to atoms with type `0`. The Tersoff potential defined in `Si_Tersoff.txt` can be a single-element Tersoff potential, or a multi-element Tersoff potential. In both cases, type `0` defined in `Si_Tersoff.txt` should correspond to type `0` in `xyz.in`.

Silicon Carbide

For a SiC crystal system described by a two-element Tersoff potential, the potential for this system can be specified by using a single `potential` keyword:

```
potential SiC_Tersoff.txt 0 1
```

Here, `SiC_Tersoff.txt` should be a potential file for the Tersoff potential and the numbers `0 1` mean that the potential applies to atoms with types `0` and `1`. The Tersoff potential defined in `SiC_Tersoff.txt` can be a two-element Tersoff potential, or a Tersoff potential with more elements. In both cases, types `0` and `1` defined in `SiC_Tersoff.txt` should correspond to types `0` and `1` in `xyz.in`.

MoS_2 + SiO_2

For a hybrid MoS_2 + SiO_2 system with LJ interactions between MoS_2 and SiO_2, the potentials can be specified as follows:

```
potential SiO_Tersoff.txt 0 1
potential MoS_REBO.txt 2 3
potential Si_O_Mo_S_LJ.txt 1 # The number 1 here is not atom type, but grouping method
```

This means:

- The Si and O atoms have atom types 0 and 1 respectively. The Tersoff potential defined in `SiO_Tersoff.txt` will be applied to this material.
- The Mo and S atoms have atom types 2 and 3 respectively. The REBO potential defined in `MoS_REBO.txt` will be applied to this material. There is a type conversion here: types 2 and 3 will be converted to `2-2=0` and `3-2=1`, before applying the potential defined in `MoS_REBO.txt` to MoS_2.
- An LJ potential is defined in `Si_O_Mo_S_LJ.txt`. **It must be an LJ potential defined for all the 4 atom types**. The number `1` after `Si_O_Mo_S_LJ.txt` means that grouping method `1` (should be defined in `xyz.in`) will be used to exclude "intra-material" LJ interactions. In this grouping method, one can put Si and O atoms into group 0 and put Mo and S atoms into group 1. Then, LJ interactions will be excluded for Si-Si, Si-O, and O-O pairs in SiO_2 and similarly for Mo-Mo, Mo-S, and S-S pairs in MoS_2.

- Of course, in this example, one can also simply set the LJ cutoff distance for these pairs to 0 in `Si_O_Mo_S_LJ.txt` without specifying a grouping method, but using a grouping method results in better performance (this is specific to the implementation of `GPUMD` and the user does not need to figure out why). While the use of a grouping method in this example is optional, it is mandatory in the next example.

Carbon peapod

Carbon peapod is formed by encapsulating fullerene molecules (such as C60) into a carbon nanotube (CNT). We have recently studied thermal transport properties of this material in [Dong 2020], where we have considered 40 C60 molecules. If we want to apply a Tersoff potential to the covalent C-C bonds and an LJ potential to the C-C pairs in which the two atoms are not both from the CNT and not from the same C60 molecule, we can use the following two commands:

```
potential C_Tersoff.txt 0
potential C_LJ.txt 1 # The number 1 here is not atom type, but grouping method
```

This means:

- All the carbon atoms are of type `0` in `xyz.in` and the Tersoff potential defined in `C_Tersoff.txt` will be applied to all the atoms. Note that due to the short cutoff distance (2.1 A) of the Tersoff potential, there will be no Tersoff interactions between the CNT and any C60 molecule or between two C60 molecules.

- An LJ potential is defined in `C_LJ.txt`. The name of this file suggests that there is a single entry, which is for the C-C pairs.

- The number `1` after `C_LJ.txt` means that grouping method `1` will be used to exclude "intra-material" LJ interactions based on grouping method `1` (should be defined in `xyz.in`). In this grouping method, atoms in the CNT should be in one group and atoms in each C60 molecule should also be in its own group. That is, there should be 41 groups in this grouping method. Based on this grouping method, two carbon atoms from the same "material" (the CNT or any C60 molecule) will have no LJ interactions.

Pure LJ systems

The user must have realized that the LJ potential is kind of special in `GPUMD`. This is true and we find it beneficial to treat it as a special potential. Our special rules for the LJ potential are:

- One does not need (and cannot) specify atom type(s) for this potential. **It is always assumed that the LJ potential applies to all the atom types defined in `xyz.in`**.

- Related to the above rule, **the LJ potential cannot be specified more than once in `run.in`**.

Base on the above rules, the potential for an argon system could be specified as

```
potential Ar_LJ.txt
```

If one writes

```
potential Ar_LJ.txt 0
```

`GPUMD` will misunderstand you by assuming that you want to exclude "intra-material" LJ interactions based on grouping method `0`.

Similarly, the potential for an argon-krypton system could be specified as

```
potential Ar_Kr_LJ.txt
```

If one writes

```
potential Ar_Kr_LJ.txt 0 1
```

`GPUMD` will response more cleverly by reporting an error message complaining that there are too many parameters for the `potential` keyword.

Caveats

- It is very important to make sure that the atom types in the xyz.in file are consistent with the potential files.

References

- [Dong 2020] Haikuan Dong, Zheyong Fan, Ping Qian, Tapio Ala-Nissila, and Yanjing Su, *Thermal conductivity reduction in carbon nanotube by fullerene encapsulation: A molecular dynamics study* (https://doi.org/10.1016/j.carbon.2020.01.114), Carbon **161**, 800-808 (2020).

Retrieved from "https://gpumd.zheyongfan.org/index.php?title=The_potential_keyword&oldid=21526"

- This page was last edited on 27 August 2020, at 03:18.