The xyz.in input file

From GPUMD
Jump to navigationJump to search

Contents

Brief descriptions

- This file defines the **simulation model**, such as the **initial conditions** and **boundary conditions**.

File format

- This file should have one of the following formats (empty lines and comments are **not** allowed):

Format A

```
N M cutoff 0 has_velocity number_of_grouping_methods
pbc_x pbc_y pbc_z L_x L_y L_z
t0 x0 y0 z0 m0 [vx0] [vy0] [vz0] [group0_0] [group1_0] [group2_0]...
t1 x1 y1 z1 m1 [vx1] [vy1] [vz1] [group0_1] [group1_1] [group2_1]...
t2 x2 y2 z2 m2 [vx2] [vy2] [vz2] [group0_2] [group1_2] [group2_2]...
...
```

Format B

```
N M cutoff 1 has_velocity number_of_grouping_methods
pbc_a pbc_b pbc_c a_x a_y a_z b_x b_y b_z c_x c_y c_z
t0 x0 y0 z0 m0 [vx0] [vy0] [vz0] [group0_0] [group1_0] [group2_0]...
t1 x1 y1 z1 m1 [vx1] [vy1] [vz1] [group0_1] [group1_1] [group2_1]...
t2 x2 y2 z2 m2 [vx2] [vy2] [vz2] [group0_2] [group1_2] [group2_2]...
...
```

Line 0 (we prefer to count from 0)

- `N` is the **number of atoms**.
- `M` is the **maximum number of neighbors** any atom can ever has. It is the responsibility of the user to use a large enough value of `M`. If some atom can have more than `M` neighbors during the simulation, unexpected error may occur and there may be no useful error message. If `M` is too large, it just wastes

some memory. For programmers: the Verlet neighbor list will be allocated a memory chunk of size `sizeof(int) * N * M`. If you really don't know how to choose the value of `M`, you can simply set it to the maximal allowed value: 1024.
- `cutoff` is the **initial cutoff distance used for building the neighbor list**. If you really don't know how to choose the value of `cutoff`, you can simply set it to cutoff distance of the maximal force cutoff in your system plus 1.0 A. You may want to have a look at the neighbor keyword.
  - The next number can be either 0 (**Format A**) or 1 (**Format B**). If it is 0, the simulation box will be **orthogonal**; if it is 1, the simulation box will be **triclinic**.
- `has_velocity` can be 1 or 0, which means this file contains or does not contain the **initial velocities**, respectively.
- `number_of_grouping_methods` can be 0 or any positive integer, which is the **number of grouping methods** that need to be defined in this file.

Line 1

- If the box is **orthogonal**, the first three items, `pbc_x`, `pbc_y`, and `pbc_z`, can only be 1 or 0. If `pbc_x` is 1, it means that periodic boundary conditions will be applied to the x direction; if `pbc_x` is 0, it means that free boundary conditions will be applied to the x direction. Similar descriptions apply to the other two directions. The next three items, `L_x`, `L_y`, and `L_z`, are the initial lengths of the (rectangular) simulation box along the x, y, and z directions, respectively.
- If the box is **triclinic**, the first three numbers correspond to `pbc_a`, `pbc_b`, and `pbc_c`, which have similar meanings as `pbc_x`, `pbc_y`, and `pbc_z`, but for the $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ directions as specified by the remaining nine numbers in this line:

$$\boldsymbol{a} = a_x\boldsymbol{e}_x + a_y\boldsymbol{e}_y + a_z\boldsymbol{e}_z;$$

$$\boldsymbol{b} = b_x\boldsymbol{e}_x + b_y\boldsymbol{e}_y + b_z\boldsymbol{e}_z;$$

$$\boldsymbol{c} = c_x\boldsymbol{e}_x + c_y\boldsymbol{e}_y + c_z\boldsymbol{e}_z;$$

These vectors define the simulation box and there is no restriction on the values of the vector components, but the user is responsible for the compatibility between the initial atom coordinates and the simulation box.

Line 2

- `t0` is the type of atom 0. The atom type will be used to determine which potential parameters to use. **We use integers to record the atom types and the indices start from 0.**
- `x0`, `y0`, and `z0` are the coordinates of this atom.
- `m0` is the mass of this atom.
- if `has_velocity` is 1, the next 3 items are the initial velocity components of this atom.
- if `number_of_grouping_methods` is larger than zero, the next items are the group labels of this atom in these grouping methods. That is, `group0_0` specifies which group this atom belongs to using grouping method 0; `group1_0` specifies which group this atom belongs to using the grouping method 1; `group2_0` specifies which group this atom belongs to using the grouping method 2. **Both grouping methods and group labels within a given grouping method are represented as integers and start from 0.** Some keywords in the `run.in` file will only use grouping method 0 and there is no need (or no way) to choose a grouping method for these keywords.

The remaining lines

- Similarly, the $(m+2)$th line gives the information for the mth atom. This file should have `(N+2)` lines.

Units

- The mass should be given in units of the unified **atomic mass unit** (amu).
- The cutoff distance, box lengths and atom coordinates should be given in units of **angstrom**.
- Velocities should be in the natural units adopted in GPUMD, i.e., eV $^{1/2}$ amu$^{-1/2}$. This is usually guaranteed, because one will probably use the dump_restart keyword to produce an `xyz.in` file which contains the velocity data. Otherwise, one would probably not prepare velocity data in the `xyz.in` file.

An example

Assume we have the following `xyz.in` file:

```
10 2 1.5 0 0 3
1 0 0 4 1 1
0 0 0 0 1 0 0 0
1 1 0 0 1 0 1 0
0 2 0 0 1 0 2 0
1 3 0 0 1 0 3 0
0 4 0 0 1 0 4 0
1 5 0 0 1 1 5 0
0 6 0 0 1 1 6 0
1 7 0 0 1 1 7 0
0 8 0 0 1 1 8 0
1 9 0 0 1 1 9 0
```

It means:

- There are 10 atoms (or particles)
- The maximum number of neighbors is set to 2, which is acceptable if we are sure that one atom can have at most 2 neighbors.
- The initial cutoff distance for building the neighbor list is 1.5 A.
- The simulation box is orthogonal.
- There are no velocity data in this file.
- There are 3 grouping methods defined in this file (see below).
- Use periodic boundary conditions in the x direction and free boundary conditions in the other directions.
- The box lengths in the three directions are respectively 4 A, 1 A, and 1 A.
- Atoms 0, 2, 4, 6, and 8 are of type 0 and atoms 1, 3, 5, 7, and 9 are of type 1.
- The 10 atoms are located along a line in the x direction with equal spacing, from 0 A to 9 A.
- All the 10 atoms have mass 1.
- In grouping method 0, atom 0 to atom 4 have group label 0 (which means they are in group 0), and atom 5 to atom 9 have group label 1 (which means they are in group 1).
- In grouping method 1, atom m ($0 \leq m \leq 9$) has group label m. That is, each group consists of a single atom.
- In grouping method 2, all the atoms have group label 0. That is, all the atoms are in the same group.

Tips

- A valid `xyz.in` file is a valid XYZ file (https://en.wikipedia.org/wiki/XYZ_file_format) that can be visualized by VMD (https://www.ks.uiuc.edu/Research/vmd/). Before this, one has to rename the file, giving it a suffix of `.xyz`.
- There are no functionalities for building simulation models within `GPUMD`, but the Python package thermo (https://github.com/AlexGabourie/thermo) developed by Alexander J. Gabourie can be used to pre-process and post-process data related to `GPUMD`.
- Recently, ASE (https://gitlab.com/ase/ase) also supported reading and writing the `xyz.in` input file for `GPUMD`. Check the `read_gpumd` and `write_gpumd` functions in the `io` module of `ASE`.

Retrieved from "https://gpumd.zheyongfan.org/index.php?title=The_xyz.in_input_file&oldid=21166"

- This page was last edited on 20 August 2020, at 20:28.