Manual 01 for MD++

Introduction to MD++

Keonwook Kang and Wei Cai

June 21, 2006

1 Overview

MD++ is a molecular dynamics simulation package written in C++. It is originally developed by Wei Cai during his graduate at MIT and still being updated to include more commands and functions. MD++ is developed to run in the Unix/Linux environment. However, if you have cygwin[1] installed in your Windows machine, you can also use MD++ there. This code is designed for studying the defect behavior in solid materials at molecular level. In the long run, we plan to apply MD++ to other applications such as polymer/fluid interface, or bio-material properties. This manual was written for those who do not have any background knowledge of MD simulations and focuses on the explanation of how to use MD++. These explanations are reinforced with many simple examples about solids.

2 Installation

The latest version of MD++ code can be downloaded from the coursework web site, http://coursework.stanford.edu/. By the way, this manuals asuumes readers have already known basic Unix/Linux shell commnads.[2] Before you download MD++ code, it is recommended that you make the following directory in your home directory.

```
$ mkdir Codes
$ cd Codes
```

Save the MD++ package in this directory(∼/Codes) and unzip the downloaded MD++ code. If the downloaded file name is md++-2005-10-07.tar.gz,

[1] For download and installation, visit http://www.cygwin.com/. Notice that you must download full packages of cygwin.

[2] For those who are not familiar with Unix/Linux, I strongly recommend "A practical guide to Linux commands, editors and shell programming" written by Mark G. Sobell. Just for the shell commands, there are several websites you can easily find.

```
$ tar -zxvf md++-2005-10-07.tar.gz
```

This command will make "MD++" directory and extract all the files and subdirectories under that directory. Make a "runs" directory if it does not exist.

```
$ cd MD++; mkdir runs
```

To compile/build as release(R) mode in Linux, type

```
$ make all build=R
```

If you are on Unix machine, you add `SYS=gcc` to the above command.[3]

```
$ make all SYS=gcc build=R
```

You can also compile MD++ code on Windows machine by typing

```
$ make all SYS=cygwin build=R
```

After compilation, you have execution files like *fs_gpp*, *eam_gpp*, and *sw_gpp* in the 'bin' directory.[4] The release mode compiles the executibles with the -O3 option, which runs faster but takes longer time to compile. Unless you specfy the build mode, the default is a debugging(D) mode, which runs slower because it is compiled with -g option for debuggers. If you want to compile a specific single binary file, e.g. *fs_gpp*, you designate it like

```
$ make fs build=R
```

The name of each execution file stands for what potential you are using in the MD simulation. For example, *fs_gpp* means Finnis-Sinclair potential[1], *lj_gpp* does Lennard-Jones pair potential[2], *eam_gpp* does embedded-atom method potential[3], and *sw_gpp* does Stillinger-Weber potential[4][5]. Each potential has its own category of atoms for MD simulation. For example, body-centered-cubic (BCC) materials like Mo, Ta, and W atoms belong to the FS potential class, and you have to run *fs_gpp* if you want MD simulation of molybdenum atoms. The binary file, *md_gpp* can be used for any atoms but it only can display the crystal structure and does not give any physical data like potential energy, stress, or temperature.
When you need to recompile after you modifiy the source code, we recommend you clean the previous binary files and then rebuild.

```
$ make clean; make all build=R
```

[3] For more `SYS` values, see the ∼/Codes/MD++/src/Makefie.base.

[4] In the 'cygwin' platform, the file name would be fs_cygwin.exe, eam_cygwin.exe, and sw_cygwin.exe.

To see more help on `make`, type

```
$ make help
```

To check if the installation process is properly done, run this example script.

```
$ bin/fs_gpp scripts/Examples/example01-mo.script
```

If you want to stop MD++ while running, press `ctrl+c`.

3 MD++ Directories

After the installation of MD++, you will get the following directories under 'MD++' directory.[5]

```
$ ls
bin        Doc        makefile   potentials   scripts   Tools
CSD-book   Examples   matlab     README       src
```

src/	All the source codes are in this 'src' directory.
bin/	Binary (or execution) files will be built in the 'bin' directory.
Doc/	MD++ manual is in the 'Doc' directory.
scripts/	You save your script file in this directory.
runs/	The outputs are stored in the subdirectory of this directory.
potentials/	Potential files are in this directory.
Examples/	'Examples' directory has some example script files.
Tools/	Some tools like 'atomeye' for MD++ simulation.
matlab/	Matlab codes for the postprocessing.

4 Script File

To execute a molecular simuation, the MD++ code reads a script file which contains a series of commands and variables for the simulation. To make or edit a script file, it is recommended to use *vi*, *emacs*, or *nedit* editor in Unix/Linux environment. For the usage of these editors, search the mannuals on-line. Basically, to edit a script file, type

```
$ vi scripts/example01-mo.script
```

or

```
$ emacs scripts/example01-mo.script &
```

[5] The new version of MD++ may have different directories, but the main structure is still almost same.

or

```
$ nedit scripts/example01-mo.script &
```

If you open an example script file, you will see what a script file has in it. Script files have the common head part:

```
# -*-shell-script-*-
setnolog                    # No log file saved
setoverwrite                # Overwrite a log file if existed
dirname = runs/mo-example       # Specify output directory
```

The first line is there so that *Emacs* will use the shell script mode to display the file. In a script file, you can commnet out any line or a part of a line by starting it with the pound sign, #. You can use # to leave your own comment or explanation anywhere in the script. The `setnolog` command will make no log file saved. Instead, it will be displayed on your screen. If `setnolog` is commented out, the log file will be saved as "A.log.gz" in the output directory. All the other output files are also to be stored in the output directory designated in the head.
The `quit` command stops the MD++ simulation. You can use `quit` alone or use it together with `sleep`.

```
quit                    # quit the simulation
```

or

```
sleep quit              # System gets sleep for 600 sec before quit.
```

The `sleep` command is very useful especially when you want to look at the graphic window carefully or display it for some time.

5 Graphic Window Control

When you run MD simulation, usually a new window pops up and you can see the motion of atoms in that window. We will refer to this window as the 'graphic window'. There are several short keys to handle this window. For example, to translate the object press 't' and then use arrow keys. To rotate it, use 'r' and arrow keys. To see the following help, press F1.

```
Mouse drag to rotate
Hot Keys:
F1        : display this message
Up        : rotate up
Down      : rotate down
Left      : rotate left
```

```
Right     : rotate right
PgUp      : rotate counterclockwise
PgDn      : rotate clockwise
Home      : back to initial viewpoint
Space     : stop rotate
p         : toggle pause
t         : translation
s         : scaling
d         : move projection infinity point
r         : rotation
f         : toggle pbc enableness
m         : toggle drawframe
g         : pbc glide
x         : pbc shift in x
y         : pbc shift in y
z         : pbc shift in z
w         : print window specification
F9        : output gif
F10       : output postscript
```

References

[1] M. W. Finnis and J. E. Sinclair, *Philos. Mag. A.* **50** 45 (1984)

[2] J. E. Lennard-Jones, "Cohesion" *Proceedings of the Physical Society* **43** 461-482 (1931)

[3] M. S. Daw and M. I. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals", *Phys. Rev. B* **29** 6443 (1984)

[4] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon", *Phys. Rev. B* **31** 5262 (1985)

[5] H. Balamane, T. Halicioglu, and W. A. Tiller, "Comparative study of silicon empirical interatomic potentials", *Phys. Rev. B* **46** 2250 (1992)