Manual 05 for MD++

MD++ Tour with Sample Scripts

Keonwook Kang and Wei Cai

May 11, 2007

1 Finite Temperature Simulation

In manual M02, we learned how to create a perfect BCC, FCC or DC crystal by `makecrystal` which gives only information of the initial positions of all the atoms. For nonzero finite temperature simulation or molecular dynamics (MD) simulation, we need additional information such as initial velocity, numerical integrator, time step and temperatue. In this manual, we aim to be familiar with MD++ commands and variables related with MD simulations.
MD simulations can be categorized as microcanonical (NVE), canonical (NVT), isoenthalpic-isobaric (NPH) or constant pressure/temperature (NPT) ensemble type according to what physical quantities are expected to be constant. For example, microcanonical ensemble has total energy (E), system volume (V) and the number of atoms (N) to be conserved during MD simulation. You need to choose proper ensemble for your own purpose of simulation.

1.1 Microcanonical(NVE) Ensemble

The easiest MD simulation is the one with microcanonical, or NVE ensemble. The following example script is rewritten as Tcl version of the script introduced in Manual M04, which shows how to run typical NVE MD simulation at 300 K from the perfect BCC molybdenum (Mo) structure. You can run the script by typing

```
$ bin/fs_gpp scripts/mo_NVE.tcl
```

The content of `mo_NVE.tcl` is

```
# -*-shell-script -*-
# run NVE MD simulation of perfect crystal of Mo.
#
#*****************************************
# Definition of procedures
#*****************************************
```

```
proc initmd { } { MD++ {
#setnolog
setoverwrite
dirname = runs/mo-example
zipfiles = 1   # zip output files
#
#-------------------------------------------
# Create Perfect Crystal
#
element0 = Mo
crystalstructure = body-centered-cubic
latticeconst = 3.1472          # in Angstrom for Mo
latticesize = [ 1 0 0 5
                0 1 0 5
                0 0 1 5 ]
} }

#-------------------------------------------------------------
proc readMoPot { } { MD++ {
# Read the potential file
potfile = ~/Codes/MD++/potentials/mo_pot readpot
} }

#-------------------------------------------------------------
proc openwindow { } { MD++ {
# Plot Configuration
atomradius = 1.0 bondradius = 0.3 bondlength = 2.725
atomcolor = orange highlightcolor = purple
bondcolor = red backgroundcolor = gray70
plotfreq = 10  rotateangles = [ 0 0 0 1.25 ]
openwin  alloccolors rotate saverot eval plot
} }

#-------------------------------------------
proc setup_md { } { MD++ {
#MD settings
T_OBJ = 300       # (in Kelvin) Desired Temperature
atommass = 95.94  # (g/mol)
timestep = 0.0005 # (ps)
totalsteps = 2001
saveprop = 1 savepropfreq = 10
savecn = 0 savecnfreq = 100
writeall = 1
DOUBLE_T = 1 randseed = 12345 srand48
ensemble_type = "NVE" integrator_type = "VVerlet"
} }
```

```
#*****************************************
# Main program starts here
#*****************************************
initmd
readMoPot

MD++ makecrystal finalcnfile = perf.cn writecn
openwindow
#-----------------------------------------
# run MD
setup_md
MD++ initvelocity finalcnfile = init.cn writecn
MD++ {output_fmt = "curstep EPOT KATOM Tinst HELMP"}
MD++ outpropfile = 300K_NVE.out  openpropfile
MD++ run closepropfile
MD++ finalcnfile = 300K-5X5X5.cn writecn eval
MD++ sleep quit
```

I will skip some of commands in the above script that are already explained in MD++ manual M04. `zipfiles = 1` will compress output files. For example, the configuration file `init.cn` and the thermodynamic data file `300K_NVE.out` will be zipped as `init.cn.gz` and `300K_NVE.out.gz`, respectively. If we do not specify `outputpropfile`, the default file name is `prop.out`. The format of output file is determined by `output_fmt` and every line of the file `300K_NVE.out.gz` has current step (`curstep`), potential energy (`EPOT`), kinetic energy (`KATOM`), temperature (`Tinst`) and Helmholtz energy (`HELMP`) as defined in `output_fmt`. If we do not specify `output_fmt`, the default array of thermodynamic data output is current step (`curstep`), kinetic energy (`KATOM`), potential energy (`EPOT`), pressure (`PRESSURE`), $-\sigma_{xy}$ (`TSTRESS_xy`), $-\sigma_{yz}$ (`TSTRESS_yz`), $-\sigma_{zx}$ (`TSTRESS_zx`), `HELM`, `HELMP`, thermodynamic frction coefficient (`zeta`), `dEdlambda` and system volume (`OMEGA`). Units are given as eV for energy and eV/$\mathring{A}^3$ for stress.
Atom species can be specifed like `element0 = Mo`. We define a Tcl function `readMoPot` to read the interatomic potential of Mo. Commands and variables related with visualization in `openwindow` will be explained in a separate manual in detail.
A configuration `.cn` file always has information of number of atoms (`NP`) in the first line and 3 by 3 matrix **H** in the last three lines. Inbetween, `writeall=1` stores `NP` lines of the following information: scaled coordinates, sacaled velocities, individual potential energy, whether-or-not fixed, topology, atom species and atom groups. Without declearation of `writeall=1`, only scaled coordinates will be saved at each line. You can check this differences by comparing two configuration files `perf.cn` and `init.cn`. Because `perf.cn` is written before setting `writeall = 1`, it has only number of atoms, scaled coordinates of each atom and the box matrix **H**. Alternatively, `writevelocity = 1` can be used, which stores scaled coordinates and velocities.

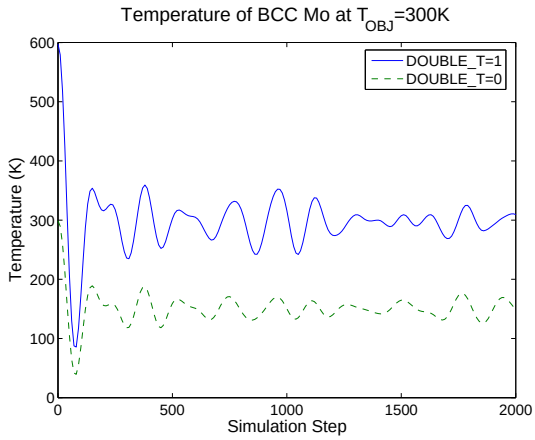


Figure 1: MD simulation of microcanonical(NVE) ensemble of BCC Mo. To obtain the desired temperature (300K), the initial temperature needs to be doubled or 600K.

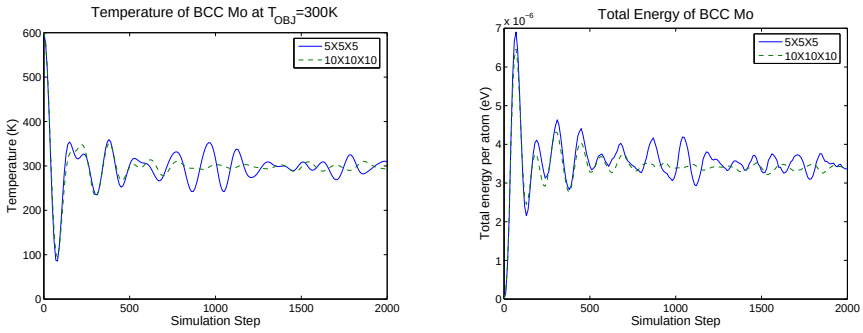
As long as it is guaranteed the total energy of the system is conserved, the timestep (`timestep`) is usually taken as big as possible. Total time step (`totalsteps`) is typically chosen long enough for the system to reach its dynamical steady state. The parameter `DOUBLE_T=1` makes the initial temperature doubled. The reason why the initial temperature is doubled is followed. Since the initial atomic configuration at 0 K is in local energy minimum state, part of kinetic energy always goes to potential energy during finite temperature simulation and this portion is same as half of the kinetic energy. ($\Delta E_{pot} = \Delta E_{kin} = \frac{3}{2} N k_B T$ for 3D case) due to the equipartition theorem. Thus, to set the final equilibrium temperature to be the desired one, the initial temperature is taken as twice high as the objective temperature. (See Fig.1.) The effect of system size on the temperature or energy fluctuation is shown in Fig. 2.

1.2 Canonical(NVT) Ensemble

In the following example script, the molybdenum at 300 K will be simulated using Nose-Hoover thermostat[1]. Run the script by typing

```
$ bin/fs_gpp scripts/mo_NVT.tcl
```

```
# -*-shell-script -*-
# run NVE MD simulation of perfect crystal of Mo.
#
#*****************************************
# Definition of procedures
#*****************************************
```



(a) (b)

Figure 2: By changing `latticesize` in the "mo_NVE.script", 10x10x10 BCC Mo (N=2000) is also simulated at T_{OBJ} =300(K) and (a) temperature and (b) total energy per atom are plotted as time step together with that of 5X5X5 BCC Mo (N=250). It is observed that the bigger system shows smaller fluctuation at the equilibrium state.(σ_T = 17.7 K for 5X5X5 and σ_T = 5.51 K for 10X10X10.)

```
proc initmd { } { MD++ {
#setnolog
setoverwrite
dirname = runs/mo-example
zipfiles = 1    # zip output files
#
#-------------------------------------------
# Create Perfect Crystal
#
element0 = Mo
crystalstructure = body-centered-cubic
latticeconst = 3.1472          # in Angstrom for Mo
latticesize = [ 1 0 0 5
                0 1 0 5
                0 0 1 5 ]
} }

#-------------------------------------------------------------
proc readMoPot { } { MD++ {
# Read the potential file
potfile = ~/Codes/MD++/potentials/mo_pot readpot
} }

#-------------------------------------------------------------
proc openwindow { } { MD++ {
```

```
# Plot Configuration
atomradius = 1.0 bondradius = 0.3 bondlength = 2.725
atomcolor = orange highlightcolor = purple
bondcolor = red backgroundcolor = gray70
plotfreq = 10  rotateangles = [ 0 0 0 1.25 ]
openwin  alloccolors rotate saverot eval plot
} }

#-----------------------------------------
proc setup_md { } { MD++ {
#MD settings
T_OBJ = 300       # (in Kelvin) Desired Temperature
atommass = 95.94  # (g/mol)
timestep = 0.0005 # (ps)
totalsteps = 2001
saveprop = 1 savepropfreq = 10
savecn = 0 savecnfreq = 100
writeall = 1
DOUBLE_T = 1 randseed = 12345 srand48
ensemble_type = "NVT" integrator_type = "VVerlet"
implementation_type = 0 vt2 = 1e28
} }

#*****************************************
# Main program starts here
#*****************************************
initmd
readMoPot

MD++ {
  incnfile = 300K-5X5X5.cn
  atommass = 95.94 timestep = 0.0005
  readcn
}
openwindow
#-----------------------------------------
# run MD
setup_md

MD++ {output_fmt = "curstep EPOT KATOM Tinst HELMP"}
MD++ outpropfile = 300K_NVT.out  openpropfile
MD++ run closepropfile
MD++ finalcnfile = 300K-5X5X5-NVT.cn writecn eval
MD++ sleep quit
```

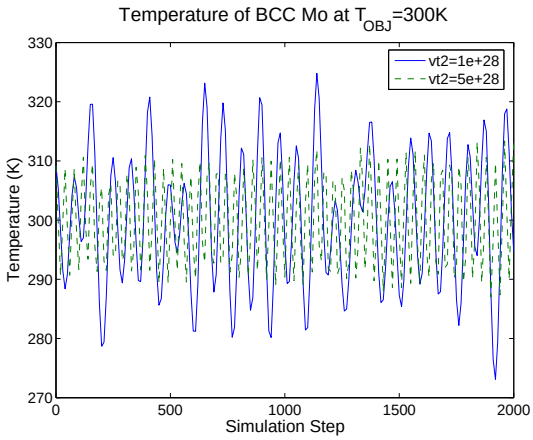


Figure 3: Temperature vs. timestep using Nose-Hoover thermostat. (a) `vt2 = 1e+28` (b) `vt2 = 5e+28`

Instead of making BCC Mo crystal, the configuration file "300K_5X5X5.cn" is read by the command `readcn`. When you read `.cn` file which was generated in the previous simulation, the timestep should be same. It is safe that you specify `timestep` and `atommass` when you read `.cn` file. Setting `ensemble_type = "NVT"` activates Nosé-Hoover thermostat. In MD++, there are three different implementations of NVT integrator defined through `implementation_type`. If it is zero, the implicit integrator will be chosen. If it is one, the explicit integrator based on Störmer-Verlet method will work. If it is two, another explicit integrator based on Liouville formulation will be activated.
Thermal coefficient `vt2` is related with thermal mass Q of Nosé-Hoover thermostat as

$$Q = \frac{3Nk_BT_{\mathrm{OBJ}}}{\texttt{vt2}} \times (1\mathrm{e}{+24})$$

where N is number of atoms and k_B is Boltzmann constant.(8.617343e-5 eV/K) When N=250 and T=300 K, `vt2 = 1e28` corresponds to $Q = 1.9389e\text{-}03$. The bigger `vt2` is, the faster the response of the system is to control the temperature. The effect of `vt2` on temperature fluctuation is shown in Fig. 3.

References

[1] S. Nose, "A Molecular Dynamics Method for Simulations in the Canonical Ensemble", Molecular Physics, **52** 255 (1984)