peachgk_md Ver. 2.142 manual for getting started (temporary)

Gota KIKUGAWA (菊川豪太)
Institute of Fluid Science, Tohoku University, Japan

1. Compilation

First of all, extract the tar ball archive named peachgk-md-x.xxx.tar.gz (depending on the version). In association with the install script described later, it is preferable to make the peachgk_md/ directory just under your home directory and extract the archive file there as:

> cd
> mkdir peachgk_md
> mv peackgk-md-2.142.tar.gz peachgk_md/ && cd peachgk_md/
> tar xvzf peackgk-md-2.142.tar.gz
> cd peackgk-md-2.142

After that, modify Makefile depending on your system environment as follows.

> edit (vi/emacs etc.) Makefile

…Makefile…

```
LINKER    = gfortran          (gcc Fortran compiler)
#LINKER           = ifort      (Intel Fortran compiler)
#LINKER           = pgfortran  (PGI Fortran compiler)
#LINKER           = mpif77     (MPI compiler depending on MPI library)
#LINKER           = mpif90     (MPI compiler depending on MPI library)
#LINKER           = mpifort    (MPI compiler depending on MPI library)
```

Choose one LINKER description (uncomment) from the above LINKERs. (default is gfortran)

```
CPP = cpp

#FFLAGS           = -O3
FFLAGS    = -O3 -Wall -I.                                  (for gfortran)
#FFLAGS           = -O3 -tpp7 –xW                          (for Intel fortran (old style))
#FFLAGS           = -O3 –xP                                (for Intel fortran)
#FFLAGS           = -O3 –xHost                             (for Intel fortran (recommended))
#FFLAGS           = -O3 -xHost –ipo                        (for Intel fortran with IPO)
#FFLAGS           = -O3 -xHost -mcmodel=large -shared-intel (for Intel fortran using large memory)
#FFLAGS           = -g -traceback –CB                      (for Intel fortran when debugging)
#FFLAGS           = -fastsse -O3 -Mipa=fast,inline –Minfo  (for PGI fortran)
#FFLAGS           = -fastsse -O3 -Mipa=fast,inline -Minfo -mcmodel=medium
```

Choose one optimization option from above.

If you use the moderately new Intel Fortran compiler (Intel Fortran Composer XE), "-O3 –xHost" option is recommended.

When doing the large scale simulation with Intel Fortran, "-mcmodel=large -shared-intel" should be added to the option.

When debugging the source code with Intel Fortran, "-g -traceback –CB" should be enabled.

MPI flag

#MPIFLAGS = -DMPI

#MPIFLAGS = -DMPI -lmpi

#MPIFLAGS = -DMPI -Mmpi=mpich

When performing MPI parallel computation, choose one (uncomment) from above depending on your system environment for MPI execution. In usual cases, "-DMPI" option is sufficient. Default is a serial run.

If you use the super computer system in IFS, Tohoku University, choose LINKER=ifort and MPIFLAGS= -DMPI –lmpi.

By default, the Coulomb interaction is carried out in peachgk_md. When running a parallel simulation with the Coulomb calculation, FFTW library should be installed in the system beforehand and the FFTW flags are enabled as mentioned later.

Optional flags

#FFLAGS2 = -D_LJ_ONLY (skip the Coulomb calculation)

#FFLAGS2 = -D_NOT_SPLINTERP (do not use the spline interpolation for real space calculation of the Ewald method)

#FFLAGS2 = -D_OUTENE_HIGH -D_OUTPRE_HIGH (output energy file with larger digits of decimal number)

#FFLAGS2 = -D_PDB_CONECT

#FFLAGS2 = -D_DOUBLE_OUTPOS -D_DOUBLE_OUTVEL (output position and velocity files with larger digits)

Enable the above option if needed.

If the system does not have any partial charge (no Coulomb calculation), use -D_LJ_ONLY and this makes the MD simulation much faster. In this case, set all ifewald, ifspme, and iffennell flags ".false." in peachgk.ini (see the later section).

Custom potential function flags

CSTMNBFLAGS = -D_CSTMNB_V1

#CSTMNBFLAGS = -D_CSTMNB_V2

#CSTMNBFLAGS = -D_CSTMNB_V2 -D_CSTMNB_V2_ADD_ALL

Setting for custom potential function. In most cases, the default is ok.

For time measurement

#TIMEFLAGS = -DTIME_M

#TIMEFLAGS = -DTIME_MALL

Flags for measurement and report of computational time in detail inside the code. This is compatible with the parallel computation.

FFT FLAGS

#FFTOPT = -D_FFTW3

#FFTINCLUDE = -I/opt/fftw/include

#FFTLIBS = -lfftw3 -L/opt/fftw/lib

These flags must be enabled when using both MPI parallel computation and Coulomb calculation (not needed when -D_LJ_ONLY is enabled). Depending on your system, change the description for the location of FFTW library (-I… -L… options at the second and third line).

…Makefile until here…

After finishing these settings, just make:

> make

[Side note 1]

Compile FFTW

If you do not have the FFTW library (http://www.fftw.org/) in the system, or if you do not want to utilize the pre-installed FFTW library and want to compile it by yourself, follow the instruction below.

First of all, get the source code of the latest version of FFTW (at least ver. 3 and above is required).

Then, extract the tar ball at any location as:

> tar xvzf fftw-3.x.x.tar.gz

> cd fftw-3.x.x

Define some environmental variables in the current shell as follows. (in the case of the bash shell)

> export CC=icc (when using Intel C compiler)

> export F77=ifort (when using Intel Fortran compiler)

> export CFLAGS="-O3 -xHost" (when using Intel C compiler and this optimization option)
> export FFLAGS="-O3 -xHost" (when using Intel Fortran compiler this optimization option)
See another environmental variables by entering
> ./configure --help

Then, do the configure:
> ./configure --prefix=/opt/fftw/ --enable-sse2
In the above arguments, --prefix is used to specify the location where the FFTW library is installed. If the CPU of your system supports the SSE2 SIMD extensions, use --enable-sse2 (and --enable-avx if supported). See ./configure --help for more detailed information.

If there is no problem in the configuration procedure, type
> make && make install

The installed directory used in the above procedure should be put in Makefile of peachgk_md.

[Side note 2]
If the large system such as over 30000 atoms is treated, tweak config.h which defines the maximum values for array size before building peachgk_md.

2. Perform MD Simulations (Demo Run)

The install shell script, which copy all the files needed to start a MD run, is prepared in the source directory as
inst.sh
Copy this file to your working directory (let the working directory be ~/work/ here).
For example,
> cp ~/peachgk_md/peachgk-md-2.142/inst.sh ~/work/

If the peachgk_md source files were not extracted to ~/peachgk_md/peachgk-md-x.xxx/, modify the following line seen at the beginning of the copied inst.sh script:
SRCDIR=$HOME/peachgk_md/peachgk-md-2.142
to your installed directory of peachgk_md.

Then, execute the inst.sh script.
> ./inst.sh

By doing so, all required script files and executable file are copied to the working directory.
The executable file is “peachgk_md.out” and the initial script for setting MD simulation conditions is “peachgk.ini”.

How to run a job is completely depending on the system environment and how you want to run. For example, run peachgk_md in the serial computation with taking a log, briefly do
> ./peachgk_md.out > log_md &

However, in order to avoid errors concerning memory allocation, it is strongly recommended that you use the run script like run_single.sh and run_opmpi.sh placed under the misc/ directory in the peachgk_md source.

When running a serial job, copy run_single.sh to the working directory and do
> ./run_single.sh

When running a MPI parallel job, copy run_opmpi.sh. Here, OpenMPI (http://www.open-mpi.org/) library is supposed to be installed. But even if you have other MPI library, it would be sufficient to modify this script very slightly according to your using MPI library. The run script should be modified to fit your system as follows.
…run_opmpi.sh…
MACHINEFILE="/opt/tools/openmpihosts" (Location of the machine file for OpenMPI)
NPROCS=8 (Number of Processes)
…run_opmpi.sh until here…

Then, execute
> ./run_opmpi.sh
When you run MD simulations via the run script, all logs are output to the “log_md” file.

If you want to restart the MD simulation at the same working directory, you need to remove the output files in the previous run. After taking backup of the previous run in advance, do
> ./rmout.sh
remove output file ? (y/[n]): y
Then, the script automatically removes all the files which inhibit restarting of a MD run.
After that, execute (restart) a MD simulation as described above.

When running your jobs on the supercomputer at IFS, Tohoku University, you need an extra care. Please

refer to the user manual of AFI supercomputer system.

[Side note]

Files copied by doing inst.sh script:

C7H15OH_cor.dat	Coordinate file for 1-heptanol
C7H15OH_top.dat	Topology file for 1-heptanol
C5H11OH_cor.dat	Coordinate file for 1-pentanol
C5H11OH_top.dat	Topology file for 1-pentanol
H2O_topOB.dat	Topology file for water modeled by SPC/E
para_bond.dat	Potential parameter lists for bonded interaction
para_vdw_s.dat	Potential parameter lists for vdW interaction
para_const.dat	File which defines the bond restraint (SHAKE/RATTLE)
peachgk_md.out	MD executable file
peachgk.ini	Initial script to control MD simulation
rmout.sh	Script to remove output files

3. Structure of the Source Code

To be updated.

param/	Directory including the parameter files needed for the MD simulation
cstmnb/	Directory including the source codes for custom potential functions (user can freely implement their own interaction type)
misc/	Directory including the patch files to update the version of peachgk.ini, run scripts to start MD simulation, and examples of createcor.F90
inst.sh	Shell script to copy all the needed files to run a MD simulation to the working (current) directory
rmout.sh	Shell script to remove output files from a MD simulation
Versioninfo	Version information for peachgk_md
Makefile.std	Backup file of Makefile used for a usual MD simulation
Makefile.hf	Backup file of Makefile used for measuring heat and momentum flux
Makefile	Makefile
peachgk.ini	Initial script to control MD simulation conditions

config.h	Header file to define the maximum values of arrays
md_global.inc	File to generate md_global.h which involves global variables used for a MD simulation
spme_global.inc	File to generate spme_global.h which involves global variables particularly for the SPME method
mpi_global.inc	File to generate mpi_global.h which involves global variables for the MPI procedure
*.F90	Source files of peachgk_md. Capital F in filename is important because the source files are processed by CPP (C language pre-processor) beforehand and then compiled by a Fortran driver

4. Explanation of peachgk.ini

Basically, peachgk_md is controlled by peachgk.ini. So, user should know how to configure this initial script as described below. Note that the different version of peachgk_md uses the different version of the peachgk.ini script. Therefore, use the peachgk.ini script included in the source code of the corresponding peachgk_md.

The detail information is available in the separate manual of "peachgk_md Input Script Manual".

5. Post-Process

5.1. Visualize MD trajectory by VMD

To be updated.

The result of peachgk_md can be visualized by VMD (http://www.ks.uiuc.edu/Research/vmd/) quite briefly. To this end, the post-process program named "pos2dcd" is executed to generate DCD formatted trajectory file using a PDB file and out_pos.dat from MD simulation. The source code of pos2dcd is located at the process_data/pos2dcd/src/ directory under the top of the source tree of peachgk_md.

Usage: pos2dcd -pdb [pdb filename] -pos [pos filename]
-o [output dcd filename]
[-init [initial step]] [-last [last step]]
[-box [xcel] [ycel] [zcel]]

Let out_pdb.pdb be the PDB file obtained from the MD simulation. For example, type
> pos2dcd –pdb out_pdb.pdb –pos out_pos.dat –o out_pos.dcd

Then, the out_pos.dcd file is created.

Boot up the VMD and read the PDB file. Once the DCD file is loaded on the molecule, the animation is available by pushing the play button on the VMD main window.

6. Brief Framework of NVE MD Simulation (with Ewald Method) in peachgk_md

To be updated.

Pre-process

rdscript	Read peachgk.ini script and store parameters to control MD simulation
init_gen_rand	Initialization of random number generator (SFMT)
openfile	Open input and output files
calbase	Non-dimensionalize the quantities and calculate the reference value for (non-)dimensionalization
rdcor	Read ***_cor.dat coordinate files
rdtop	Read ***_top.dat topology files
rdpara	Read para_bond.dat and para_vdw_s.dat files
rdconst	Read para_const.dat file
mkmolept	Generate pointers to indicate which atoms are belonging to a certain molecule
createcor	Generate the initial configuration of the MD system
rdstarec	Read the coordinate and velocity data from the restart file
mkexcl	Register the excluded pairs (pairs for which intramolecular vdW and Coulomb interactions are not evaluated)
linkbond	Impose mass and charge to all the atoms and give indexes to all the covalent bonds.
mkconst	Register the restraint bonds for SHAKE/RATTLE methods
createvel	Give the initial velocities
prepmd	Other pre-process of MD simulation
prepewk	Calculate wave vectors used in the Ewald reciprocal space calculation
wrsumm	Output the summary of MD control parameters

MD routine

moldyn	MD routine for NVE simulation and velocity scaling to control temperature

Post-process

wrsta	Output the coordinates and velocities to restart the MD simulation