peachgk_md Ver. 2.142 スタートマニュアル（暫定版）

菊川豪太
東北大学流体科学研究所

1. コンパイル

tar アーカイブを展開する．後述するインストールスクリプトに関連して，ホームディレクトリ直下に peachgk_md/ディレクトリを作成し，そこでアーカイブを展開することが望ましい．

> 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

環境に合わせて Makefile を修正
> edit (vi/emacs 等) Makefile
…Makefile 内…
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)
上記から環境に合わせて LINKER を一つ選ぶ.

CPP = cpp

#FFLAGS = -O3
FFLAGS = -O3 -Wall -I.
#FFLAGS = -O3 -tpp7 -xW
#FFLAGS = -O3 -xP
#FFLAGS = -O3 -xHost
#FFLAGS = -O3 -xHost -ipo
#FFLAGS = -O3 -xHost -mcmodel=large -shared-intel
#FFLAGS = -g -traceback -CB
#FFLAGS = -fastsse -O3 -Mipa=fast,inline -Minfo
#FFLAGS = -fastsse -O3 -Mipa=fast,inline -Minfo -mcmodel=medium
上記から最適化フラグを一つ選ぶ.

比較的新しい Intel コンパイラの場合，-O3 –xHost がお勧め.
また，大規模計算の場合 -mcmodel=large -shared-intel を入れること.
デバッグを行う場合は，-g -traceback –CB を有効にするとよい.

MPI flag
#MPIFLAGS = -DMPI
#MPIFLAGS = -DMPI -lmpi
#MPIFLAGS = -DMPI -Mmpi=mpich
MPI 計算をする場合，上記から 1 つ選ぶ．通常は，-DMPI を選択.
流体研スパコンを利用する場合，LINKER=ifort として，MPIFLAGS= -DMPI –lmpi を選択すること.
デフォルトではクーロン計算を行うが，並列計算の場合， FFTW をシステムにインストールした上で後述の FFTW フラグを有効化する必要あり.

Optional flags
#FFLAGS2 = -D_LJ_ONLY (Coulomb 計算をスキップ)
#FFLAGS2 = -D_NOT_SPLINTERP (実空間計算にスプライン補間を使用しない)
#FFLAGS2 = -D_OUTENE_HIGH -D_OUTPRE_HIGH (エネルギーファイルの高精度出力)
#FFLAGS2 = -D_PDB_CONECT
#FFLAGS2 = -D_DOUBLE_OUTPOS -D_DOUBLE_OUTVEL (位置・速度ファイルの高精度出力)
上記から必要に応じてオプションフラグを有効にする.
LJ 計算のみの場合-D_LJ_ONLY を入れると MD 計算が速い．この場合，peachgk.ini で ifewald, ifspme，iffennell を全て.false.に設定すること.

Custom potential function flags
CSTMNBFLAGS = -D_CSTMNB_V1
#CSTMNBFLAGS = -D_CSTMNB_V2
#CSTMNBFLAGS = -D_CSTMNB_V2 -D_CSTMNB_V2_ADD_ALL
カスタムポテンシャル関数に関する設定．通常は変更する必要はない.

For time measurement
#TIMEFLAGS = -DTIME_M
#TIMEFLAGS = -DTIME_MALL
コード内部で計算時間の測定，レポートを行うためのフラグ．並列計算にも対応している.

FFT FLAGS

#FFTOPT = -D_FFTW3
#FFTINCLUDE = -I/opt/fftw/include
#FFTLIBS = -lfftw3 -L/opt/fftw/lib
クーロン計算を含む MPI 並列計算をする場合は必須（LJ_ONLY を有効にした場合はコメントアウトしておいてよい）．システムに合わせて FFTW ライブラリの場所を適宜変更すること（2 行目および 3 行目）．
…Makefile ここまで…

以上の設定の後 Make する．
> make

[補足 1]
FFTW のコンパイル
システムに FFTW ライブラリ (http://www.fftw.org/) がインストールされていない，もしくはシステムにプリインストールされている FFTW を使用せず自分でコンパイルする場合，以下のように行う．

まず，最新バージョンの FFTW のソースコードをダウンロードする（少なくとも Ver. 3 以降が必要）．
任意の場所で tar ball アーカイブを展開する．
> tar xvzf fftw-3.x.x.tar.gz
> cd fftw-3.x.x

コンパイルに使用する現在のシェルで以下の環境変数を定義する（ここでは bash シェルの場合）．
> export CC=icc (Intel C compiler)
> export F77=ifort (Intel Fortran compiler)
> export CFLAGS=”-O3 -xHost” (Intel C compiler を用い，この最適化オプションを使用する場合)
> export FFLAGS=”-O3 -xHost” (Intel Fortran compiler を用い，この最適化オプションを使用する場合)
その他の環境変数については，以下を入力して確認すること．
> ./configure --help

上記設定が終わったら，configure を実行する．
> ./configure --prefix=/opt/fftw/ --enable-sse2
上記コマンドの引数において，--prefix は FFTW ライブラリをインストールする場所を指定する．使用するシステムの CPU が SSE2 SIMD 拡張命令をサポートしているなら，--enable-sse2 を加え

ること（AVX がサポートされているなら--enable-avx も加える）. ./configure –help でその他のオプションを参照できる.

configure の実行で特にエラーがなければ,
> make && make install

上記で指定した FFTW のインストールディレクトリを peachgk_md の Makefile 内部で使用する.

[補足 2]
粒子数が 30000 原子を超える場合など, 大規模計算を行う場合は config.h に記載されている上限値を修正した上でコンパイルすること.

2. 計算実行

ソースディレクトリにある
inst.sh
は，実行に必要なファイルをコピーするためのシェルスクリプトである.
このファイルを，別途用意した実行ディレクトリ（ここでは~/work/とする）にコピーする.
例えば,
> cp ~/peachgk_md/peachgk-md-2.142/inst.sh ~/work/

1 節冒頭で推奨したディレクトリ以外の場所に peachgk_md を展開した場合，コピーした inst.sh スクリプト内の冒頭にある
SRCDIR=$HOME/peachgk_md/peachgk-md-2.142
この部分を，自分が peachgk_md のソースを展開したディレクトリに合わせて修正する.

次に，実行ディレクトリにて，inst.sh を実行する.
> ./inst.sh
これによって，実行ファイルや実行に必要なスクリプトが全てコピーされる.
例えば, 実行ファイルの本体は, peachgk_md.out であり, 計算条件を指定する入力スクリプトは, peachgk.ini である.

計算実行は，環境等によるが，例えばスカラージョブを実行しながらログをとる場合,

> ./peachgk_md.out > log_md &
などとすればよい.
ただし，メモリー関連などのエラーを避けるため，ソースディレクトリ下の mics/ディレクトリ

にある実行用スクリプト run_single.sh や run_opmpi.sh を実行ディレクトリにコピーして使用した方がよい.

スカラージョブの場合は,

> ./run_single.sh

MPI 並列ジョブの場合は run_opmpi.sh を使用する. OpenMPI (http://www.open-mpi.org/)ライブラリがインストールされていることを前提としているが, その他のライブラリの場合でもスクリプトの小修正で対応可能と考えられる. スクリプト内冒頭の以下の部分について修正を行い実行する.

…run_opmpi.sh 内…

MACHINEFILE="/opt/tools/openmpihosts"　　OpenMPI 用のマシンファイルの場所

NPROCS=8　　並列プロセス数

…run_opmpi.sh ここまで…

実行は,

> ./run_opmpi.sh

実行スクリプト経由で実行した場合, log_md ファイルに全てのログが出力される.

同じ実行ディレクトリで計算をリスタートする場合は, 前のジョブで出力された計算結果のファイルを削除する必要がある. 予め（前の）計算結果をバックアップした後, 実行ディレクトリにて,

> ./rmout.sh

remove output file ? (y/[n]): y

とすれば, リスタートを妨げるファイル群を全て削除できる. その後, 上述した方法によって計算を実行すればよい.

流体研スパコンを利用する場合には別途実行方法に注意が必要だがここでは省略する.

[補足]

inst.sh 実行によってコピーされるファイル：

C7H15OH_cor.dat	ヘプタノールの座標ファイル
C7H15OH_top.dat	ヘプタノールの結合情報ファイル
C5H11OH_cor.dat	ペンタノールの座標ファイル
C5H11OH_top.dat	ペンタノールの結合情報ファイル
H2O_topOB.dat	SPC/E（水分子）の結合情報ファイル
para_bond.dat	分子内のポテンシャルパラメータ
para_vdw_s.dat	分子間のポテンシャルパラメータ

para_const.dat	結合長を拘束する結合を記述するファイル
peachgk_md.out	MD 実行ファイル
peachgk.ini	MD の制御ファイル
rmout.sh	実行後に出力ファイルを削除するシェルスクリプト

3. ソースコードの構造

そのうち真面目に書く予定

param/	MD 計算に必要なパラメータファイルを格納しているディレクトリ
cstmnb/	カスタムポテンシャル関数（ユーザーが自由にポテンシャルを導入できる）が格納されたディレクトリ
misc/	peachgk.ini のバージョン更新のためのパッチファイルや計算実行時の run スクリプト，createcor.F90 の例などが格納されたディレクトリ
inst.sh	計算実行に必要なファイルを実行ディレクトリにインストールするシェルスクリプト
rmout.sh	計算で発生する出力ファイルを削除するシェルスクリプト
Versioninfo	peachgk-md のバージョン情報
Makefile.std	通常計算用の Makefile のバックアップ
Makefile.hf	熱流束・運動量流束計算用の Makefile のバックアップ
Makefile	Makefile
peachgk.ini	MD 制御スクリプト
config.h	配列変数の上限を決定するヘッダーファイル
md_global.inc	md_global.h を生成するインクルードファイル. MD に用いる変数をまとめたグローバル変数が記述されている.
spme_global.inc	smpe_global.h を生成するインクルードファイル. SPME 関連の変数をまとめたグローバル変数が記述されている.
mpi_global.inc	mpi_global.h を生成するインクルードファイル. MPI 計算に用いられる変数をまとめたグローバル変数が記述されている.
*.F90	ソースファイル群. F が大文字になっているのは，CPP（C プリプロセッサ）をかけた上で，Fortran コンパイルさせるため.

4. peachgk.ini の解説

peachgk_md は基本的に peachgk.ini によって全てコントロールされる．したがって，この初期スクリプトファイルの書き方を述べる．なお，peachgk_md のバージョンによって peachgk.ini の仕様が異なるため，必ず同じバージョンに含まれている peachgk.ini を利用すること．

詳細は別ファイル”peachgk_md 入力ファイルマニュアル”を参照すること．

5. ポストプロセス

5.1. VMD によるアニメーション作成について（そのうち真面目に書く予定）

peachgk_md の計算結果は，VMD（http://www.ks.uiuc.edu/Research/vmd/)を用いると手軽にアニメーション可視化できる．このために計算によって出力された pdb ファイルと out_pos.dat を使い，後処理プログラム pos2dcd を用いて DCD ファイル（トラジェクトリ）を作成する．pos2dcd のソースコードは，process_data ソースツリー下の process_data/pos2dcd/src/ディレクトリに格納されている．

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

例えば，
> pos2dcd –pdb out_pdb.pdb –pos out_pos.dat –o out_pos.dcd
とすると，out_pos.dcd が得られ，VMD で pdb を読み込んだ後，
その molecule に DCD を Load すると簡単にアニメーションが得られる．

6. peachgk_md における NVE 計算（Ewald 法）の大まかな流れ（moldyn に分岐）

前処理	
rdscript	peachgk.ini を読み込み MD 実行の際の制御パラメータを読み込む．
init_gen_rand	乱数の初期化（SFMT）
openfile	入力ファイル，出力ファイルをオープン
calbase	系の無次元化．無次元量の算出．
rdcor	***_cor.dat ファイルを読み込む．H_2O の座標を生成．
rdtop	***_top.dat ファイルを読み込む．
rdpara	para_bond.dat，para_vdw_s.dat ファイルを読み込む．
rdconst	para_const.dat を読み込む．
mkmolept	ある分子にどの原子が属するかというポインタを生成する．
createcor	初期座標配置を決定する．

rdstarec	初期座標配置，初期速度を読み込む.
mkexcl	排除ペアを登録する．（分子内で vdW，Coulomb 相互作用を無視するペア）
linkbond	全原子に質量と部分電荷を与え，全結合に index を与える.
mkconst	全結合から，結合長拘束するものを登録する.
createvel	初期速度を与える（速度 0）.
prepmd	その他 MD の前処理を行う.
prepewk	エワルド法の波数空間の計算に用いる波数ベクトルを計算する.
wrsumm	MD の制御パラメータの出力

MD 計算部

moldyn	NVE 計算，速度スケーリング（温度制御）計算の MD ルーチン

後処理

wrsta	座標，速度の出力