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．H2O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