

SMASH(Scalable Molecular Analysis Solver for High performance computing systems)

ユーザーマニュアル ver. 2.2.0

(青字が ver. 1.1.0 からの追加・変更点)

- ・ 閉殻 MP2 エネルギー微分及び構造最適化計算ルーチンを追加
- ・ B3LYP の相関汎関数(パラメータ)を VWN5 から VWN3 に変更
- ・ 計算精度を制御するキーワード `precision(=high,medium(デフォルト),low)` を追加
- ・ 2 電子積分計算などのデフォルトのカットオフ値を変更
- ・ SCF 収束方法に QC(Quadratically convergent)法を追加
- ・ ダミー原子(点電荷用)とゴースト原子(BSSE 計算用)を追加
- ・ サンプルインプットとアウトプットを追加
- ・ 基底関数が I 関数までサポート(従来は G 関数まで)

1. 可能な計算

Hartree-Fock, B3LYP with VWN5, B3LYP with VWN3: 閉殻・開殻エネルギー及び構造最適化計算 (B3LYP とインプットで指定した場合のパラメータを VWN5 から VWN3 に変更)

MP2: 閉殻エネルギー計算及び構造最適化計算

2. インストール方法

Intel、AMD マシンで MPI を使ってコンパイルする場合、`smash/`ディレクトリ内で `make`

と実行する。

Intel MPI Library (mpiifort)を使用する場合、

`make -f Makefile.mpiifort`

と実行する。

MPI 並列しない場合、

`make -f Makefile.x86_64.noMPI`

と実行する。

Intel コンパイラ、MKL ライブラリ以外を使用する場合、`Makefile`、`Makefile.x86.noMPI` の F90、LIB、OPT を編集する。

京コンピュータ、FX10、FX100 を利用する場合、

`make -f Makefile.fujitsu`

と実行する。

`bin/smash` ができれば、インストール終了。

3. 重要な設定

Linux システムの stack size と環境変数の OMP_STACKSIZE のデフォルト値は小さ

いため、実行前に新しい値を設定する必要がある。

bash の場合： ~/.bashrc ファイルに次の 2 行を追加する。

```
ulimit -s unlimited
```

```
export OMP_STACKSIZE=1G
```

csh,tcsh の場合： ~/.cshrc もしくは ~/.tcshrc ファイルに次の 2 行を追加する。

```
unlimit
```

```
setenv OMP_STACKSIZE 1G
```

もし SCF 計算が開始すぐに止まる場合、OMP_STACKSIZE の値をより大きくする。

4. 実行方法

4.1. ノード内(OpenMP)並列

1 プロセスのスレッド数を決めるため、環境変数 OMP_NUM_THREADS を設定する。

bash の場合： ~/.bashrc ファイルに次の行を追加する。

```
export OMP_NUM_THREADS=(スレッド数)
```

csh,tcsh の場合： ~/.cshrc もしくは ~/.tcshrc ファイルに次の行を追加する。

```
setenv OMP_NUM_THREADS (スレッド数)
```

設定後、次のように計算を実行する。

```
bin/smash < (input ファイル名) > (output ファイル名)
```

4.2. ノード間(MPI)並列

mpirun もしくは mpiexec コマンドにより MPI 並列計算を開始する。

MPI プロセス数は np の後に指定する。

```
mpirun -np (プロセス数) bin/smash < (input ファイル名) > (output ファイル名)
```

MPI ライブラリによっては input ファイルを次のように指定する

```
mpirun -stdin (input ファイル名) -np (プロセス数) bin/smash
```

4.3. ハイブリッド(MPI/OpenMP)並列

1 プロセスあたりのスレッド数(OMP_NUM_THREADS の値)は 1 ノードのコア数と同じ、プロセス数は使用するノード数と同じにすると、ほとんどのシステムで最も効率的に計算できる。

OMP_NUM_THREADS を設定後、MPI 計算を開始する。

5. インプット形式

job, control, scf, opt, dft, mp2 で始まる行で計算内容を指定して、geom の次の行から原子とその座標を指定する。大文字と小文字の区別はない。原子ごとの基底関数、ECP を指定する場合、それぞれ basis、ecp の次の行から指定する。 [点電荷を置く場合、](#)

charge の次の行から指定する。

```
job  runtime=optimize method=b3lyp basis=gen memory=20g
control precision=high
geom
  O   0.0000000    0.0000000    0.1423813
  H   0.0000000    0.7568189   -0.4626257
  H   0.0000000   -0.7568189   -0.4626257
(空行)
basis
O
6-31G(d)
****
H
STO-3G
****
```

6. サンプルインプット

[smash/example/](#)ディレクトリにサンプルインプットとアウトプットファイルがある。

7. キーワード内容

7.1. job

runtime	計算実行方法 energy: エネルギー計算 (default) gradient: エネルギー微分計算 optimize: 構造最適化計算
method	計算方法 HF: Hartree-Fock 計算 (default) b3lyp: B3LYP 計算 with VWN3 b3lyp5: B3LYP 計算 with VWN5 (Version1.1.0 までの b3lyp) mp2: MP2 計算
basis	基底関数 sto-3g (default), 6-31g, 6-31g(d), cc-pvdz, cc-pvtz, cc-pvqz, d95v, lanl2dz gen: 元素ごとに指定(参照 7.8 basis)
memory	1 ノード当たりメモリ使用量 (default 8GB) 使用可能単位 : B, KB, MB, GB, TB

charge	系の電荷 (default ± 0) 7.10 の charge で指定された点電荷の値は含まない
multi	スピン多重度 1: singlet (default) 2: doublet 3,4,...: triplet, quartet, ...
scftype	波動関数種類 RHF: Closed-shell (default) UHF: Open-shell
ecp	ECP (Effective Core Potential) none (default) lanl2dz gen: 元素ごとに指定(参照 7.9 ecp)

7.2. control

precision	計算精度の制御 (原子軌道 2 電子積分計算のカットオフ、SCF 収束判定、DFT の grid 点数) 以下の変数が個別に設定された場合、数値は上書きされる high: cutint2=1.0D-12, dconv=5.0D-6, threshrho=1.0D-6, threshdfock=1.0D-5, threshdftao=1.0D-4, nrad=150, nleb=590 medium(default): cutint2=1.0D-11, dconv=5.0D-6, threshrho=1.0D-5, threshdfock=1.0D-4, threshdftao=1.0D-3, nrad=96, nleb=302 low: cutint2=1.0D-10, dconv=1.0D-5, threshrho=1.0D-4, threshdfock=1.0D-4, threshdftao=1.0D-2, nrad=72, nleb=302
spher	Spherical Harmonics もしくは Cartesian 基底指定 .true.: Spherical (5d, 7f,...) (default) .false.: Cartesian (6d, 10f,...)
guess	初期波動関数 huckel: 拡張 Huckel 計算 (default) check: チェックポイントファイル参照
check	チェックポイントファイル名 (default:空白)
cutint2	2 電子積分のカットオフ値 1.0D-11 (default) (Version 1.1.0 までは 1.0D-12)
bohr	距離の単位 .false.: Å (default)

	.true. : bohr
iprint	出力制御 1 : 最少出力 2 : 通常出力 (default) 3 : 詳細出力
threshover	線形従属判定、閾値以下の重なり行列の固有値の数だけ分子軌道数を削減 1.0D-6 (default)
threshatom	原子間距離判定、閾値以下なら計算ストップ 2.0D-1 (default)

7.3. scf

scfconv	SCF 収束方法指定 DIIS : DIIS 法 (default) SOSCF: Approximated Second-orderSCF 法 QC: Quadratically convergent SCF 法
maxiter	最大 SCF 回数 150 (default)
dconv	SCF での電子密度収束判定 5.0D-6 (default)
maxdiis	最大 DIIS 回数 20 (default)
threshdiis	DIIS 開始判定 6.0D-1 (default)
maxsoscf	最大 SOSCF 回数 20 (default)
threshsoscf	SOSCF 開始判定 2.5D-1 (default)
maxqc	最大 QC 回数 15 (default)
maxqcdiag	最大 Davidson 対角化回数 100 (default)
maxqcdiagsub	Davidson 対角化の最大試行ベクトル数 10 (default)
threshqc	QC 収束判定 1.0D-5 (default)

7.4. opt

nopt	最大 Opt 回数 100 (default)
optconv	Opt での Force 収束判定 1.0D-4 (default)
cartesian	構造最適化時の座標系 .true. : Cartesian coordinate .false. : Redundant coordinate (default)

7.5. dft

nrad	汎関数数値積分の動径点数 96 (default)
nleb	汎関数数値積分の Lebedev グリッド角度点数 302 (default)
threshweight	grid 点の重みの閾値 1.0D-8 (default)
thresrho	grid 点の電子密度の閾値 1.0D-5 (default)
threshdfock	grid 点の Fock 行列成分の閾値 1.0D-4 (default)
threshdftao	grid 点の基底関数値の閾値 1.0D-3 (default)
bqrad(9)	ghost 原子の原子半径(Å) 1.06D+0 (default, Bohr 半径の 2 倍)

7.6. mp2

ncore	Frozen core 軌道数 自動的に計算 (default)
nvfz	Frozen virtual 軌道数 0 (default)
maxmp2diis	MP2 エネルギー微分計算における CPHF 計算の最大 DIIS 回数 20 (default)
maxmp2iter	MP2 エネルギー微分計算における CPHF 計算の最大収束回数 100 (default)
threshmp2cphf	MP2 エネルギー微分計算における CPHF 計算の収束判定

	1.0D-10 (default)
--	-------------------

7.7. geom

“geom”の次の行から分子構造の読み込みが始まる。1 行に 1 原子で、元素記号と xyz 座標を書く。空行もしくはファイルの最後で分子構造の読み込みが終了する。
 チェックポイントファイルから読むときは、初めの行を **geom=check** と書く。
 点電荷のための **dummy atom** は”X”で指定する。BSSE など基底や ECP を置いたり、DFT の **grid** 点を作成する **ghost atom** は”Bq1(=Bq)”から”Bq9”まで利用可能。

7.8. basis

元素ごとに指定する。
 関数名(6-31G(d)、cc-pVDZ、LANL2DZ など)でも関数の直接記入でも可。
 関数名と関数の併用も可(LANL2DZ に d 関数追加など)。
 元素ごとの区切りは****で、basis 指定の最後は空行。
 関数のフォーマットは次の通り。
 (元素記号)
 (軌道角運動量 (S,P,D,F,G,H,I,SP)) (primitive 関数の数)
 (Gauss 関数の指数) (短縮係数) (P 関数の短縮係数(SP 関数の場合のみ))
 (Gauss 関数の指数) (短縮係数) (P 関数の短縮係数(SP 関数の場合のみ))
 ...primitive 関数の数繰り返し
 ...関数指定の繰り返し

例)

```
basis
Se
LanL2DZ
D    1
      0.384    1.0
****
C H
6-31G(d)
****
```

7.9. ecp

元素ごとに指定する。
 関数名(LANL2DZ)でも関数の直接記入でも可。
 ecp 指定の最後は空行。

関数のフォーマットは次の通り。

(元素記号)

(関数名(任意)) (最大軌道角運動量) (Core 電子数)

(タイトル(任意))

(Gauss 関数の数)

(R の次数) (Gauss 関数の指数) (Gauss 関数の係数)

(R の次数) (Gauss 関数の指数) (Gauss 関数の係数)

...Gauss 関数の数繰り返し

...Gauss 関数指定の繰り返し

例)

```
ecp
Au
LanL2DZ
Cl
Cl-ECP      2      10
d-ul potential
  5
1      94.8130000      -10.0000000
2      165.6440000       66.2729170
2      30.8317000      -28.9685950
2      10.5841000      -12.8663370
2       3.7704000      -1.7102170
s-ul potential
  5
0      128.8391000       3.0000000
1      120.3786000      12.8528510
2       63.5622000     275.6723980
2       18.0695000     115.6777120
2        3.8142000     35.0606090
p-ul potential
  6
0      216.5263000       5.0000000
1       46.5723000       7.4794860
2      147.4685000     613.0320000
2       48.9869000     280.8006850
2       13.2096000     107.8788240
2        3.1831000     15.3439560
```


7.10. charge

原子の核電荷(小数点も可)を指定する。

点電荷を置く場合は、`dummy atom` を `geom` セクションで `X` を使って指定する。

Counterpointse 計算などで基底関数や ECP を置く場合は、`ghost atom` を `Bq(=Bq1)` から `Bq9` までを使って指定する。

電荷指定のフォーマットは次の通り。

(`geom` で書いた原子の番号) (電荷)

(例)

geom			
O	0.0	0.0	0.0
H	1.0	0.0	0.0
H	0.0	1.0	0.0
X	0.0	0.0	1.5
X	0.0	0.0	-2.0
charge			
4	1.0		
5	-0.5		

8. 可視化

`smash/visual/`ディレクトリにチェックポイントファイルから `vtk` ファイルと `cube` ファイルを作成するソースコードがある。`visual` ディレクトリで `make` を実行すると、`vtk-generator` と `cube-generator` ファイルが作られる。使い方は、`vtk-generator_manual` と `cube-generator_manual` に書かれている。