

反応経路最適化プログラム

# *Reaction plus*

**Pro ver. 2 / Express ver. 2**

for Linux / Windows

マニュアル

© 2015-2024 HPC Systems Inc.



# 目次

概要 .....	2
旧バージョンからの新機能・変更点 .....	3
インストール方法 .....	4
Reaction plus Pro for Linux .....	4
Reaction plus Pro for Windows .....	5
Reaction plus Express for Linux .....	6
Reaction plus Express for Windows .....	7
基本的な使い方 .....	8
反応経路最適化プログラム <b>reactg09, reactg16, reactx, reactux</b> .....	12
コマンド書式 .....	12
ルートセクション（#行）に指定する React キーワードオプション .....	12
追加入力セクション（\$react~\$end）に指定するオプション .....	13
chk ファイルについて（Reaction plus Pro のみ） .....	14
ファイル構成 .....	15
インプットファイルの例 .....	16
ユーティリティプログラム .....	19
ファイルフォーマット .....	20
よくある質問 .....	24
動作環境 .....	26

- 本製品ならびに本製品に付随するマニュアル・チュートリアル著作権は、HPC システムズ株式会社が所有しています。
- 本資料に記載された各社名・各製品名・各ロゴ等は、各社の登録商標または商標です。
- 本製品を使用する権利は、ライセンス数に応じてあらかじめ指定された計算機に対してのみ付与されます。
- 本製品の全部または一部を当社に無断で複製、変更、公開、配布、譲渡、貸出、販売することを禁じます。
- 本製品の使用により生じるいかなる損害についても当社は一切責任を負いません。
- 本ソフトウェア内容のうち、Reaction plus Express 付属の `sqm_rp, cp2k_rp` プログラムは GNU General Public License (GPL) に基づいています。

## 概要

「反応経路が手軽に求められる」

これまで反応経路を求めるには勘と経験が必要であり、特に肝となる遷移状態構造の最適化(TS Opt)がうまく行えず、大変な苦勞をしていました。

Reaction plus Pro および Reaction plus Express (以下、区別しない限りにおいては Reaction plus と表記) は「研究者のセンス」と「シミュレーション技術」をうまく活用したソフトウェアです。反応物と生成物と指定するだけで自動的に反応経路が求まります。ユーザの予想した反応経路がインプットに指定できるため、短時間で自然な反応経路を見つけることができます。

### 主な特徴

(Reaction plus Pro/Express 共通)

- ・ ユーザの想定した反応機構をもとに反応経路を最適化 ※1
- ・ インプットやアウトプットは GaussView 等で作成／可視化が可能

(Reaction plus Pro)

- ・ DFT, MM, ONIOM など Gaussian で利用可能な多くの計算条件に対応

(Reaction plus Express)

- ・ わずか数分程度で反応経路計算が完了 ※2,3



Reaction plus と Gaussian, GaussView を使用した利用の流れ

※1 反応経路最適化に Nudged Elastic Band (NEB)法を一部改良した方法を利用しています。

※2 反応経路計算手法には半経験的手法 PM6 を採用しています。

※3 50 原子程度までの一般的な化学反応系の場合。計算時間は反応系や計算条件、計算機性能に依存します。

## 旧バージョンからの新機能・変更点

### Reaction plus Pro / Express 共通

- 一部の原子座標を固定したまま計算することができるようになりました。
- NGeom のデフォルト値が 1 に変更されました。1 点計算のインプットに複数の座標ファイル (pdb,mol2,xyz) を指定して計算する使い方を基本的な使い方としました。QST2 型や QST3 型のインプットを使用する場合は、それぞれ NGeom=2、NGeom=3 を明示して下さい。
- %xyz および Read オプション、Restart オプションは廃止されました。以前の計算結果から座標を読み込む場合は、追加入力セクション (\$react~\$end セクション) に座標ファイル(pdb,mol2,xyz) を指定して下さい。
- 反応経路最適化のステップ毎に、log ファイル、xyz ファイルに加え、pdb ファイルが出力されるようになりました。
- 反応経路最適化のステップ毎に、得られた遷移状態候補構造から続けて正確な遷移状態構造最適化を行うための Gaussian QST3 入力ファイル (QST-L-M-N.gjf) や、遷移状態構造付近の反応経路をさらに精査するための Reaction plus 入力ファイル (Refine-L-M-N.gjf) が自動生成されるようになりました (L, M, N はビーズ番号)。
- NoAdjust オプションにより、計算開始時の各ビーズ座標生成において原子間距離の微調整を行わないように指定できるようになりました。

### Reaction plus Pro

- ONIOM 計算に対応しました。
- 初期軌道としてユーザの指定した Gaussian Chk ファイルが読み込めるようになりました。開殻系の計算に利用できます。
- Gaussian 追加入力セクションを伴うキーワードとして、従来の Gen、GenECP、Geom=Connectivity に加え、他の多数のキーワード (Charge、ReadIsotope、ReadWindow など) も利用できるようになりました。
- Reaction plus 計算経過のログファイルが、xxx-work ディレクトリ内の log ファイルから、xxx.info ファイルに変更されました (xxx はインプットファイル名)。

### Reaction plus Express

- 開殻反応系に適用できるプログラム reactux を追加しました (reactx は閉殻反応系のみ対応)。

## インストール方法

Reaction plus Pro for Linux

※ `/usr/local/reactg` に Reaction plus Pro をインストールする場合について説明します。

① 環境変数を設定します。

(bash の場合)

ホームディレクトリ内の `.bashrc` ファイルに以下の内容を記載します。

```
export REACTGHOME=/usr/local/reactg
export PATH=$REACTGHOME/bin:$PATH
```

(csh, tcsh の場合)

ホームディレクトリ内の `.cshrc` ファイルに以下の内容を記載します。

```
setenv REACTGHOME /usr/local/reactg
set path=($REACTGHOME/bin $path)
```

② `tgz` ファイルをインストール先に展開します。`tgz` ファイルをインストール先のマシンの適切なディレクトリにコピーした後、`root` 権限で以下のコマンドを実行すれば、`/usr/local/reactg` ディレクトリが生成され、この中に必要なファイルが展開されます。

```
$ tar xvfz reactg-*.tgz -C /usr/local
```

(もし古いバージョンの Reaction plus をお持ちの場合は、それを削除してからインストールを行って下さい。)

③ `/usr/local/reactg/license/` にライセンスファイル `license.txt` を設置して下さい。

```
$ cp license.txt /usr/local/reactg/license/
```

④ Reaction plus プログラムファイルとライセンスファイルのアクセス権限を設定して下さい。

```
$ chmod 755 /usr/local/reactg/bin/*
$ chmod 644 /usr/local/reactg/license/license.txt
```

※Reaction plus Proは Gaussian がインストールされているフォルダの中にインストールします。ここでは、`C:\G09W\reactg` に Reaction plus Pro をインストールする場合について説明します。(パスに日本語やスペースが含まれないようにご注意ください。)

- ① zip ファイルをインストール先に展開します。フォルダ構成は下記ようになります。

```
C:\G09W\reactg\
```

```
  bin\
```

```
    reactg09.exe  (Gaussian 09 環境用)
```

```
    reactg16.exe  (Gaussian 16 環境用)
```

```
  examples\
```

```
  license\
```

- ② Gaussian 09 環境であれば `reactg09.exe` の、Gaussian 16 環境であれば `reactg16.exe` のショートカットをデスクトップなどに作成します。
- ③ `C:\G09W\reactg\license` にライセンスファイル `license.txt` を設置します。

※ /usr/local/reactx に Reaction plus Express をインストールする場合について説明します。

① 環境変数を設定します。

(bash の場合)

ホームディレクトリ内の.bashrc ファイルに以下の内容を記載します。

```
export REACTXHOME=/usr/local/reactx
export PATH=$REACTXHOME/bin:$PATH
```

(csh, tcsh の場合)

ホームディレクトリ内の.cshrc ファイルに以下の内容を記載します。

```
setenv REACTXHOME /usr/local/reactx
set path=($REACTXHOME/bin $path)
```

② **tgz** ファイルをインストール先に展開します。**tgz** ファイルをインストール先のマシンの適切なディレクトリにコピーした後、**root** 権限で以下のコマンドを実行すれば、/usr/local/reactx ディレクトリが生成され、この中に必要なファイルが展開されます。

```
$ tar xvfz reactx-*.tgz -C /usr/local
```

(もし古いバージョンの Reaction plus をお持ちの場合は、それを削除してからインストールを行ってください。)

③ /usr/local/reactx/license/ にライセンスファイル **license.txt** を設置して下さい。

```
$ cp license.txt /usr/local/reactx/license/
```

④ Reaction plus プログラムファイルとライセンスファイルのアクセス権限を設定して下さい。

```
$ chmod 755 /usr/local/reactx/bin/*
$ chmod 644 /usr/local/reactx/license/license.txt
```

※ **C:¥reactx** に **Reaction plus Express** をインストールする場合について説明します。(パスに日本語やスペースが含まれないようにご注意ください。)

① zip ファイルをインストール先に展開します。フォルダ構成は下記のようになります。

**C:¥reactx¥**

**bin¥**

**reactx.exe** (閉殻反応系計算用)

**reactux.exe** (開殻反応系計算用)

**sqm\_rp.exe**

**cp2k\_rp.exe**

**examples¥**

**license¥**

**extra¥**

② **reactx.exe**、**reactux.exe** のショートカットをデスクトップなどに作成します。

③ **C:¥reactx¥license** にライセンスファイル **license.txt** を設置します。

## 基本的な使い方 (詳しくはチュートリアルをご利用下さい)

Reaction plus Pro、Reaction plus Express は、コマンド名 (Windows 版ではドラッグ&ドロップ先のアイコン) がそれぞれ異なるだけで、基本的な使い方は同じです。

① インプットファイル(\*.gjf)を作成します。

インプットファイルの書き方には、Gaussian QST2/QST3 入力フォーマットに倣った方法と、外部ファイルから座標データを読み込む方法の2種類があります。

(例1 : Gaussian QST2 入力フォーマットに倣った書き方)

```
%nprocshared=16
%mem=16GB
#p b3lyp/6-31g(d) react=(ngeom=2,nbeads=16,fitaxis)
```

ここに Reaction plus の計算  
オプションを入力します

Wittig reaction

0 1				
C	-2.07681900	-0.18930700	-0.00017300	
O	-1.94044900	-1.39974600	-0.00140400	始状態の分子構造
C	1.62195100	1.62306900	0.00073100	
H	1.25708200	2.03884800	0.93302500	
H	1.25579100	2.03949900	-0.93076700	
P	2.32694500	0.10124900	-0.00034600	
H	3.17759200	-0.06296000	1.11737500	
H	3.17536100	-0.06224400	-1.11984500	
C	-2.13528600	0.60932500	-1.28948900	
H	-1.22721000	1.22258500	-1.36267900	
H	-2.99209600	1.29325900	-1.29910700	
H	-2.18722900	-0.06646700	-2.14585300	
C	-2.13465400	0.60676400	1.29075300	
H	-2.99129600	1.29088000	1.30205500	
H	-1.22639700	1.21965400	1.36483000	
H	-2.18643000	-0.07072400	2.14578600	
C	1.49289700	-1.56192700	0.00003900	
H	2.24804900	-2.35915400	-0.00092400	
H	0.85474900	-1.66596700	0.88284300	
H	0.85290400	-1.66550900	-0.88147400	

Wittig reaction

0 1				
C	2.11591200	0.22024100	0.00000000	
O	-1.04783900	-1.22467800	0.00000000	終状態の分子構造
C	2.02650100	1.55618100	0.00000000	
H	1.99738000	2.12786900	-0.92504600	
H	1.99738000	2.12786900	0.92504500	
P	-1.63635300	0.15398300	0.00000000	
H	-1.28949900	0.98697800	-1.09488600	
H	-1.28949900	0.98697800	1.09488600	
C	2.15876100	-0.58363800	1.27538800	
H	3.06423700	-1.20470000	1.31737100	
H	1.29851800	-1.26348000	1.31733000	
H	2.14383400	0.05711000	2.16325500	
C	2.15876100	-0.58363900	-1.27538800	
H	1.29851800	-1.26348000	-1.31733000	
H	3.06423700	-1.20470000	-1.31737100	
H	2.14383400	0.05711000	-2.16325500	
C	-3.45996300	0.26345500	0.00000000	
H	-3.77776600	1.31278200	-0.00000100	
H	-3.86149600	-0.23412700	-0.88784300	
H	-3.86149600	-0.23412600	0.88784400	

(例2：外部ファイルから座標データを読み込む書き方)

```
%nprocshared=16
%mem=16GB
#p b3lyp/6-31g(d) react=(nbeads=16,fitaxis)
```

ここに Reaction plus の計算  
オプションを入力します

Wittig reaction

```
0 1
C          -2.07681900   -0.18930700   -0.00017300
O          -1.94044900   -1.39974600   -0.00140400
C           1.62195100    1.62306900    0.00073100
H           1.25708200    2.03884800    0.93302500
H           1.25579100    2.03949900   -0.93076700
P           2.32694500    0.10124900   -0.00034600
H           3.17759200   -0.06296000    1.11737500
H           3.17536100   -0.06224400   -1.11984500
C          -2.13528600    0.60932500   -1.28948900
H          -1.22721000   -1.22258500   -1.36267900
H          -2.99209600    1.29325900   -1.29910700
H          -2.18722900   -0.06646700   -2.14585300
C          -2.13465400    0.60676400    1.29075300
H          -2.99129600    1.29088000    1.30205500
H          -1.22639700    1.21965400    1.36483000
H          -2.18643000   -0.07072400    2.14578600
C           1.49289700   -1.56192700    0.00003900
H           2.24804900   -2.35915400   -0.00092400
H           0.85474900   -1.66596700    0.88284300
H           0.85290400   -1.66550900   -0.88147400
```

原子ラベル情報

(原子ラベル情報を読み込むために  
必要な入力ストリーム。座標情報は  
実際の計算では無視され、外部ファ  
イルから読み込まれます。)

```
$react
reactant.pdb
product.pdb
$end
```

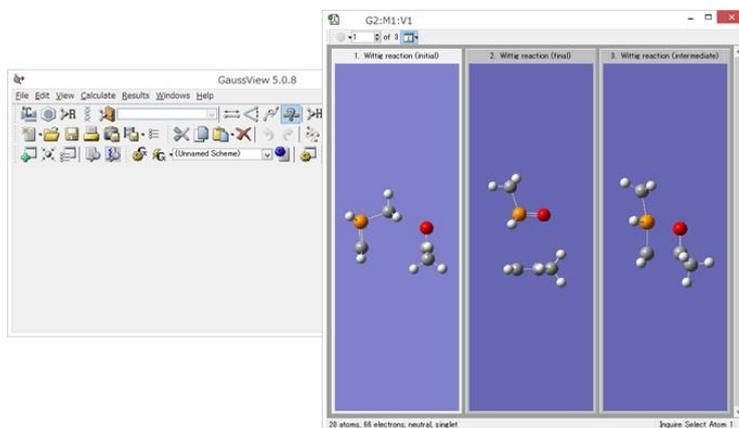
外部ファイル (始状態の分子構造)  
外部ファイル (終状態の分子構造)

※Reaction plus Express では以下の仕様で計算が実行されます。

- ・指定した並列数 (%NProcShared=\*\*) だけビーズが並行に計算されます。
- ・指定したメモリ容量 (%Mem=\*\*) は無視されます。
- ・指定した量子化学計算条件 (B3LYP/6-31G など) は無視され、常に PM6 での計算となります。  
(#行の内容は、React=( )のみが有効となります。)

※GaussView でインプットファイルを作成する場合は、次のような手順となります。

1. 2つの構造 (始状態、終状態) または 3つの構造 (始状態、終状態、中間状態) を作成。
2. Gaussian Calculation Setup から Job Type → Optimization TS(QST2) / TS(QST3) を選択。
3. Additional Keywords に react で始まる Reaction plus 計算条件を入力。
4. Retain をクリックして Gaussian Calculation Setup ウィンドウを終了。
5. gjf ファイルとして保存。



② Reaction plus を実行します。

Linux 版では、以下のコマンドにより計算を実行します。

```
$ reactg16 wt.gjf      (Reaction plus Pro, Gaussian16 環境)
$ reactg09 wt.gjf      (Reaction plus Pro, Gaussian09 環境)
$ reactx wt.gjf        (Reaction plus Express, 閉殻反応系)
$ reactux wt.gjf       (Reaction plus Express, 開殻反応系)
```

Windows 版では、プログラムのアイコン（またはショートカット）にインプットファイル(\*.gjf)をドラッグ&ドロップすると計算が始まります。

③ 計算結果を確認します。

1. 次のファイルが作成されていることを確認します。

(wt.gjf を実行した場合、以下のファイル・ディレクトリが自動生成されます。)

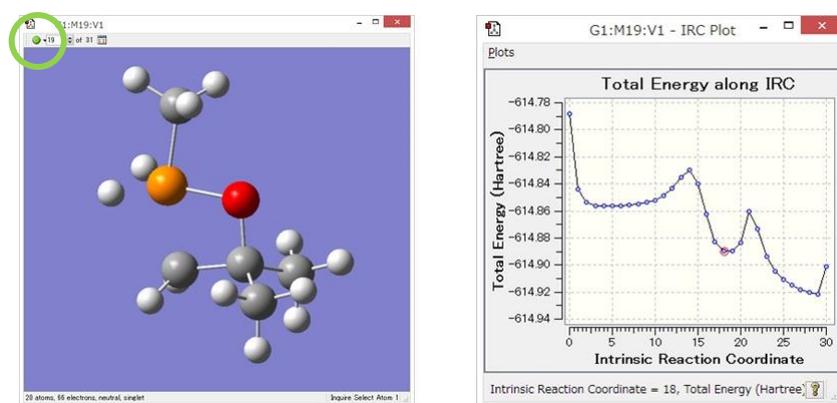
wt.info	計算ログファイル
wt.log	Gaussian IRC 形式のアウトプットファイル
wt.pdb	各ビーズの分子座標ファイル (PDB 形式)
wt.xyz	各ビーズの分子座標ファイル (XYZ 形式)
wt.dat	最適化過程のエネルギー情報ファイル (タブ区切りデータ)
wt-work/	中間データディレクトリ

2. \*.info ファイルは、適当なテキストエディタで閲覧できます。

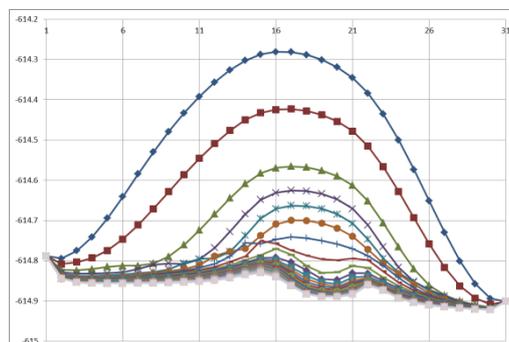
```
@ 1 cycles ( 30 qc-runs )
-----
Bead  Energy(Hartree)
  0    -0.04896696   *
  1    -0.02851917   *
  2     0.03632257   *
  3     0.13705962   *
  4     0.25472920   *
  5     0.37494193   *
  6     0.48663763   *
  7     0.57103322   *
  8     0.61381079   *
  9     0.59753263   *
 10     0.49756934   *
 11     0.35831597   *
 12     0.20205252   *
 13     0.05245279   *
 14    -0.05705642   *
 15    -0.09220173   *
-----
Emean   0.24723212
Gmax    0.26221697 (0.00450000) False
Grms    0.00150478 (0.00300000) True
Xmax    0.25103655 (0.01800000) False
Xrms    0.00120143 (0.01200000) True

@ 2 cycles ( 44 qc-runs )
-----
Bead  Energy(Hartree)
  0    -0.04896696   *
```

3. \*.log ファイルは GaussView で閲覧できます。分子描画ウィンドウ左上の緑ボタンをクリックすると、反応経路に沿った分子構造変化のアニメーションを見ることができます。また、Results → IRC/Path... メニューから、反応経路のポテンシャルエネルギーを確認することができます。



4. \*.pdb ファイル、\*.xyz ファイルは VMD などの可視化ソフトウェアで見ることができます。
5. \*.dat ファイルを表計算ソフトウェアでグラフ化すると、反応経路の最適化過程が閲覧できます。



## 反応経路最適化プログラム reactg09, reactg16, reactx, reactux

反応経路を最適化するプログラムです。

### コマンド書式

reactg16 [FILE]	(Reaction plus Pro, Gaussian16 環境)
reactg09 [FILE]	(Reaction plus Pro, Gaussian09 環境)
reactx [FILE]	(Reaction plus Express, 閉殻反応系)
reactux [FILE]	(Reaction plus Express, 開殻反応系)

(例)

```
$ reactg16 input.gjf
```

### ルートセクション (#行) に指定する React キーワードオプション

※ [ ] はデフォルト値。インプットで省略した場合、デフォルト値が使われます。また大文字小文字は区別されません。

NGeom	インプットに指定する分子構造の数。[NGeom=1]
NBeads	ビーズ (経路上の分子構造) の数。始状態から終状態へ至る経路上にこの数の分子構造を配置します。[NBeads=15]
KSpring	隣り合うビーズをつなぐバネの定数。[KSpring=0.1]
MaxCycles	最適化ステップ数の上限値。[MaxCycles=100]
MaxStep	最適化1ステップにおける移動サイズ。[MaxStep=0.5]
Loose / Normal / Tight / VeryTight	反応経路の収束判定条件。[Loose]
FitAxis	計算開始時にインプット構造の分子が重なるように座標軸を合わせます。また Freeze (一部の原子座標を固定させる) 指定とは併用できません。[無指定]
NoAdjust	計算開始時にビーズを発生させますが、その際デフォルト (NoAdjust 無指定時) では各ビーズの原子間距離を微調整しています。このオプションを指定するとこの処理を行いません。[無指定]
OptReactant	始状態の分子構造を最適化させる (固定しない)。[無指定]
OptProduct	終状態の分子構造を最適化させる (固定しない)。[無指定]

(例)

```
# B3LYP/6-31G(d) React=(NGeom=2, NBeads=20, FitAxis)
```

※いくつかのオプションについては短縮名が利用できます。

```
# B3LYP/6-31G(d) React=(NGeom=2, NBeads=15, MaxCyc=300, NoAdj)
```

```
# B3LYP/6-31G(d) React=(NGeom=3, NBeads=15, Fit)
```

```
# B3LYP/6-31G(d) React=(NGeom=3, Nbeads=15, OptR, OptP)
```

## 追加入力セクション (\$react~\$end) に指定するオプション

<外部ファイル入力>

ファイル名を指定すると、指定した順に構造を読み込みます (通常の座標指定とは異なり、始状態→中間状態→終状態の順となります)。ファイル拡張子.pdb, .mol2, .xyz に対応しています。複数の構造を含むファイルにも対応しています。(gjf ファイルに初めから書かれている座標は無視されます。)

(例：始状態、中間状態、終状態をそれぞれ molecule-1.pdb~molecule-3.pdb から読み込む)

```
# B3LYP/6-31G(d) react=(NBeads=20,FitAxis)
Title
0 1
C          -0.65595819   -0.25292166    0.00072516
H          -1.22050819    0.07108534   -0.88321384
H          -0.65579419   -1.35376966    0.00074316
H          -1.22052119    0.07110734    0.88464716
C           0.73850181    0.30065234    0.00072416
H           0.81146881    1.40040034    0.00073016
O           1.72266281   -0.40277366    0.00072516

$react
molecule-1.pdb      始状態の座標
molecule-2.pdb      中間状態の座標
molecule-3.pdb      終状態の座標
$end
```

<原子固定機能>

インプットファイル(\*.gjf)の最後に指定するオプション。**\$react** 行から**\$end** 行の間に指定します。

**Freeze** 指定した番号の原子の座標を固定する。(※初期構造の座標に固定されます。また **FitAxis** オプションとは併用できません。)

(例：1 番目、3 番目、5~7 番目の原子座標を固定する)

```
# B3LYP/6-31G(d) react=(NGeom=2,NBeads=20)
Reactant
0 1
C          -0.65595819   -0.25292166    0.00072516
H          -1.22050819    0.07108534   -0.88321384
H          -0.65579419   -1.35376966    0.00074316
H          -1.22052119    0.07110734    0.88464716
C           0.73850181    0.30065234    0.00072416
H           0.81146881    1.40040034    0.00073016
O           1.72266281   -0.40277366    0.00072516

product
0 1
C          -0.65595819   -0.25292166    0.00072516
H          -1.49022095    0.21523760   -0.00014828
H          -0.65687544   -1.37120035   -0.00276556
H           1.77526628   -1.27299075   -0.00159379
C           0.73850181    0.30065234    0.00072416
H           0.81146881    1.40040034    0.00073016
O           1.72266281   -0.40277366    0.00072516

$react
Freeze 1,3,5-7      コンマとハイフンを用いて番号を指定
$end
```

## chk ファイルについて (Reaction plus Pro のみ)

%chk 行を指定しない場合、初期反応経路の計算では、ビーズ 0 の計算で得られた chk ファイルをビーズ 1 の初期軌道 (initial guess) 読み込み用の chk ファイルとして利用し、ビーズ 1 の計算で得られた chk ファイルをビーズ 2 の初期軌道ファイルとして利用し…というように、次々に chk ファイルを引き継ぎながら計算を行います。

%chk 無指定時の初期反応経路計算における chk ファイルの取り扱い (NGeom=3, NBeads=7 の場合 : 以下の例でも同様)

Reaction plus の インプット	構造 1 (始状態)			構造 3 (中間状態)			構造 2 (終状態)
ビーズ	ビーズ 0	ビーズ 1	ビーズ 2	ビーズ 3	ビーズ 4	ビーズ 5	ビーズ 6
インプット chk	なし	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk
アウトプット chk	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk	bead-6.chk

一方、Reaction plus のインプットファイルに %chk 行を指定した場合、以下のようになります。

(a) %chk=aaa.chk と指定し、aaa.chk というファイルが存在するとき :

ビーズ 0 のインプット chk に aaa.chk を指定して計算を行います。

初期反応経路計算における chk ファイルの取り扱い

Reaction plus の インプット	構造 1 (始状態)			構造 3 (中間状態)			構造 2 (終状態)
ビーズ	ビーズ 0	ビーズ 1	ビーズ 2	ビーズ 3	ビーズ 4	ビーズ 5	ビーズ 6
インプット chk	<b>aaa.chk</b>	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk
アウトプット chk	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk	bead-6.chk

(b) %chk=aaa.chk と指定し、aaa-0.chk ~ aaa-6.chk (0 からの連番) ファイルが存在するとき :

追加入力セクションに xyz ファイルを指定することにより、x 番目のビーズに aaa-x.chk ファイルをそれぞれ割り当てて計算を行います。この場合は、aaa-\*.chk ファイルの数とビーズの数は一致していなければなりません。

初期反応経路計算における chk ファイルの取り扱い

Reaction plus の インプット	構造 0	構造 1	構造 2	構造 3	構造 4	構造 5	構造 6
ビーズ	ビーズ 0	ビーズ 1	ビーズ 2	ビーズ 3	ビーズ 4	ビーズ 5	ビーズ 6
インプット chk	<b>aaa-0.chk</b>	<b>aaa-1.chk</b>	<b>aaa-2.chk</b>	<b>aaa-3.chk</b>	<b>aaa-4.chk</b>	<b>aaa-5.chk</b>	<b>aaa-6.chk</b>
アウトプット chk	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk	bead-6.chk

なお、2 回目以降の反応経路計算サイクルでは、%chk 行の有無にかかわらず、前のサイクルで得られた chk ファイルをそれぞれのビーズごとに引き継いで計算します。

2 回目以降のサイクルの反応経路計算における chk ファイルの取り扱い

Reaction plus の インプット	構造 1 (始状態)			構造 3 (中間状態)			構造 2 (終状態)
ビーズ	ビーズ 0	ビーズ 1	ビーズ 2	ビーズ 3	ビーズ 4	ビーズ 5	ビーズ 6
インプット chk	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk	bead-6.chk
アウトプット chk	bead-0.chk	bead-1.chk	bead-2.chk	bead-3.chk	bead-4.chk	bead-5.chk	bead-6.chk

## ファイル構成

### Reaction plus Pro の場合

input.gjf	ユーザが作成したインプットファイル
input.info	Reaction plus 計算経過を記録したログファイル
input.log	Gaussian IRC 形式のアウトプットファイル
input.pdb	PDB 形式のビーズ座標ファイル
input.xyz	XYZ 形式のビーズ座標ファイル
input.dat	最適化過程のエネルギー情報 (タブ区切りデータ)
input-work/	中間データ用ディレクトリ
bead-x.gjf	x 番目のビーズの Gaussian gjf ファイル
bead-x.log	x 番目のビーズの Gaussian log ファイル
bead-x.pdb	x 番目のビーズの PDB ファイル
bead-x.chk	x 番目のビーズの Gaussian chk ファイル
bead-x.fchk	x 番目のビーズの Gaussian fchk ファイル
path-n.log	最適化サイクル(n サイクル目)の Gaussian IRC 形式のアウトプットファイル
path-n.pdb	最適化サイクル(n サイクル目)の PDB 形式のビーズ座標ファイル
path-n.xyz	最適化サイクル(n サイクル目)の XYZ 形式のビーズ座標ファイル
path.dat	最適化過程のエネルギー情報 (タブ区切りデータ)
QST-L-M-N.gjf	ビーズ番号 L-M-N についての Gaussian QST3 計算用ファイル (見つかった反応障壁の数だけ生成されます。)
Refine-L-M-N.gjf	ビーズ番号 L-M-N についての Reaction plus 精査計算用ファイル (見つかった反応障壁の数だけ生成されます。)
:	

### Reaction plus Express の場合

input.gjf	ユーザが作成したインプットファイル
input.info	Reaction plus 計算経過を記録したログファイル
input.log	Gaussian IRC 形式のアウトプットファイル
input.pdb	PDB 形式のビーズ座標ファイル
input.xyz	XYZ 形式のビーズ座標ファイル
input.dat	最適化過程のエネルギー情報 (タブ区切りデータ)
input-work/	中間データ用ディレクトリ
bead-x.sqmin	x 番目のビーズの sqm インプットファイル
bead-x.sqmout	x 番目のビーズの sqm アウトプットファイル
bead-x.pdb	x 番目のビーズの PDB ファイル
path-n.log	最適化サイクル(n サイクル目)の Gaussian IRC 形式のアウトプットファイル
path-n.pdb	最適化サイクル(n サイクル目)の PDB 形式のビーズ座標ファイル
path-n.xyz	最適化サイクル(n サイクル目)の XYZ 形式のビーズ座標ファイル
path.dat	最適化過程のエネルギー情報 (タブ区切りデータ)
QST-L-M-N.gjf	ビーズ番号 L-M-N についての Gaussian QST3 計算用ファイル (見つかった反応障壁の数だけ生成されます。)
Refine-L-M-N.gjf	ビーズ番号 L-M-N についての Reaction plus 精査計算用ファイル (見つかった反応障壁の数だけ生成されます。)
:	

## インプットファイルの例

(一般的注意)

インプットとなる分子構造の数は、**Opt=QST2**、**Opt=QST3** キーワードではなく、**React** キーワード中の **NGeom** オプションによって決定されます。**NGeom** を指定しない場合は **NGeom=1** と解釈され、追加入力セクションで外部ファイルから分子構造を読み込まない限り、計算が走りませんので、ご注意下さい。

(**GaussView** での操作が容易となるため、**Opt=QST2**、**Opt=QST3** キーワードの指定をお勧め致しますが、計算実行時にはこれらのキーワードは無視されます。)

また、**Force** キーワード、**Freq** キーワードを指定すると **Reaction plus** が走りませんので、これらのキーワードは指定しないようお願い致します。

(例 1 : 始状態・終状態の 2 点の構造を指定して計算)

```
%NProcShared=16
# Opt=QST2 B3LYP/STO-3G React=(NGeom=2,NBeads=7)           NGeom=2 を指定

Reactant
0 1
O          0.00000000    0.00000000    0.00000000
H          0.70000000    0.70000000    0.00000000
H         -0.70000000    0.70000000    0.00000000

Product
0 1
O          0.00000000    0.00000000    0.00000000
H          0.70000000   -0.70000000    0.00000000
H         -0.70000000   -0.70000000    0.00000000
```

(例 2 : 始状態・終状態・中間状態の 3 点の構造を指定して計算。分子構造の指定順に注意！)

```
%NProcShared=16
# Opt=QST3 B3LYP/STO-3G React=(NGeom=3,NBeads=7,MaxCyc=20) NGeom=3 を指定

Reactant
0 1
O          0.00000000    0.00000000    0.00000000      始状態
H          0.70000000    0.70000000    0.00000000
H         -0.70000000    0.70000000    0.00000000

Product
0 1
O          0.00000000    0.00000000    0.00000000      終状態
H          0.70000000   -0.70000000    0.00000000
H         -0.70000000   -0.70000000    0.00000000

TS
0 1
O          0.00000000    0.00000000    0.00000000      中間状態
H          0.70000000    0.00000000    0.00000000
H         -0.70000000    0.00000000    0.00000000
```

(例 3 : 中間状態を 2 点指定し、計 4 点の構造を用いて計算。分子構造の指定順に注意！)

```
%NProcShared=16
# B3LYP/STO-3G React=(NGeom=4,NBeads=7,MaxCyc=20)         NGeom=4 を指定

Reactant
0 1
```

O	0.00000000	0.00000000	0.00000000	始状態
H	0.70000000	0.70000000	0.00000000	
H	-0.70000000	0.70000000	0.00000000	
Product				
0 1				
O	0.00000000	0.00000000	0.00000000	終状態
H	0.70000000	-0.70000000	0.00000000	
H	-0.70000000	-0.70000000	0.00000000	
TS1				
0 1				
O	0.00000000	0.00000000	0.00000000	1つ目の中間状態
H	0.70000000	0.20000000	0.00000000	
H	-0.70000000	0.20000000	0.00000000	
TS2				
0 1				
O	0.00000000	0.00000000	0.00000000	2つ目の中間状態
H	0.70000000	-0.20000000	0.00000000	
H	-0.70000000	-0.20000000	0.00000000	

(例 4 : xyz ファイルから反応経路上の分子構造を読み込んで計算)

```
%NProcShared=16
# B3LYP/STO-3G React=(NBeads=30)

Reactant

0 1
O      0.00000000  0.00000000  0.00000000
H      0.70000000  0.70000000  0.00000000
H     -0.70000000  0.70000000  0.00000000

$react
mol.xyz
$end
```

座標情報は xyz ファイルから読み込まれますが、形式的に分子構造の入力ストリームが必要です (実際計算ではこの座標情報は無視されます)

座標を読み込む xyz ファイル名

(例 5 : 一部の原子の座標を固定して計算)

```
%NProcShared=16
# B3LYP/STO-3G React=(NGeom=2,NBeads=15)

Reactant

0 1
O      0.00000000  0.00000000  0.00000000
H      0.70000000 -0.70000000  0.00000000
H     -0.70000000  0.70000000  0.00000000

Product

0 1
O      0.00000000  0.00000000  0.00000000
H      0.70000000 -0.70000000  0.00000000
H     -0.70000000 -0.70000000  0.00000000

$react
Freeze 1
$end
```

始状態

終状態

1 番目の原子 (この例では酸素原子) の座標を固定

(例 6 : 外部ファイルから座標を読み込み、ONIOM 計算)

```
%NProcShared=16
# ONIOM(B3LYP/6-31G(d):PM6) React=(NBeads=15)

Title
```

```

0 1 0 1 0 1
C          0.08333900    0.00000000    0.01731300
H          0.01937200    0.00000000    1.12248600
O          1.18332700    0.00000000   -0.60902400
C          -1.32684400    0.00000000   -0.77350100 L H 1
F          -1.11952500    0.00000000   -2.12650700 L
F          -2.06087200   -1.11090000   -0.44065700 L
F          -2.06087200    1.11090000   -0.44065700 L

$react
  file-1.pdb          始状態の座標
  file-2.pdb          1つ目の中間状態の座標
  file-3.pdb          2つ目の中間状態の座標
  file-4.pdb          3つ目の中間状態の座標
  file-5.pdb          終状態の座標
$end

```

(例7: chk ファイルを読み込んで計算。mol.xyz には複数の構造が書かれており、対応する初期分子軌道を guess ディレクトリ内の bead-0.chk,bead-1.chk,bead-2.chk,...から読み込む。)

```

%NProcShared=16
%chk=guess/bead.chk          このようなディレクトリ指定も可能
# B3LYP/STO-3G Guess=Read React=(NBeads=15)

Reactant

0 1
O          0.00000000    0.00000000    0.00000000
H          0.70000000   -0.70000000    0.00000000
H          -0.70000000    0.70000000    0.00000000

$react
  mol.xyz
$end

```

座標情報は xyz ファイルから読み込まれますが、形式的に分子構造の入力ストリームが必要です(実際計算では無視されます)

※Reaction plus Pro で計算を行うと、work ディレクトリ内に bead-N.chk (N はビーズ番号) という名前の chk ファイルが自動的に生成されます(本マニュアルの 15 ページ参照)。したがって、求めた各ビーズの分子軌道を次の計算の初期分子軌道として利用したい場合は、bead-N.chk を別のディレクトリ(この場合 guess ディレクトリ)内にコピーして、上記のように指定すると便利です。

## ユーティリティプログラム

### ONIOM 計算支援ツール makeoniom (Linux 版 Reaction plus Pro 付属)

AMBER や GROMACS のファイルから ONIOM 1 点計算用 Gaussian インプットのひな形を生成する支援ツールです。指定した番号の原子を High layer に指定したインプットファイルを作成します。

このツールは、それ単独で正常に ONIOM 計算が動作するファイルが生成できることを保証するものではありません。スピン多重度の値やリンク原子に対するパラメータ、力場情報など、適宜修正して使用して下さい。

コマンド書式

```
makeoniom [AMBER トポロジーファイル] [AMBER 座標ファイル] "[QM 原子の番号]"  
makeoniom [GROMACS トポロジーファイル] [GROMACS 座標ファイル] "[QM 原子の番号]"
```

(例)

```
$ makeoniom protein.prmtop protein.inpcrd "52-120,134-147" > oniom.gjf  
$ makeoniom ligand.top ligand.gro "13-28" > oniom.gjf
```

### 計算停止ツール kill-react.bat (Windows 版 Reaction plus Pro 付属)

実行中の Reaction plus 関連のプロセスを全て停止させるツールです。

使用方法

Reaction plus 実行中に kill-react.bat のアイコンをダブルクリックします。(停止処理が終了すると自動的にウィンドウが閉じられます。)

※本ツールは、実行されている reactg16.exe, g16.exe, l502.exe など全てを停止させるプログラムです。本ソフトウェアとは関係のない同名のプログラムも停止されますので注意して下さい。例えば、Reaction plus とは別に単独で走らせた Gaussian も停止されます。不都合のある場合は本プログラム (bat ファイル) を適宜編集して使用して下さい。

## ファイルフォーマット

### インプットファイル (\*.gif)

インプットファイル形式は、Gaussian の 1 点計算や QST2, QST3 計算のインプットに類似した形式を採用しています。

1 点計算のファイルをベースに Reaction plus のインプットを作成する場合は、ファイルの最後に \$react~\$end セクションを設け、座標ファイルの名前 (xyz、pdb、mol2 形式) を反応進行順に羅列します。QST2 計算のファイルをベースにインプットを作成する場合は、分子構造を始状態→終状態の順に指定します。QST3 に対しては、始状態→終状態→中間状態の順に指定します。分子構造を 4 点以上指定する場合は、始状態→終状態→中間状態 1→中間状態 2→…の順に指定します。

### ○Link 0 コマンド

- %NProcShared      並列数を指定します。
- %Mem                メモリ容量を指定します。(Reaction plus Pro のみ)
- %Chk                Gaussian チェックポイント (chk) ファイルを指定します。過去の計算で収束した分子軌道を初期軌道 (initial guess) として利用したい場合に指定します。(Reaction plus Pro のみ)

### ○ルートセクション (#行)

Reaction plus Pro では、反応経路最適化の計算条件 (React キーワード) の他、B3LYP/6-31G(d)や EmpiricalDispersion=GD3 のような量子化学計算条件が指定できます。ただし、Opt、NoSymm、Force キーワードは無視されます。また、反応経路最適化プロセスと両立しないキーワードを指定した際の動作は保証致しません。

Reaction plus Express では、React 以外のキーワードは無視され、常に PM6 で計算が実行されます。

### ○タイトルセクション (始状態)

始状態のタイトルを指定します。

### ○分子情報セクション (始状態)

始状態構造の電荷、スピン多重度、分子構造を指定します。分子構造については、

```
原子 1  x 座標  y 座標  z 座標
原子 2  x 座標  y 座標  z 座標
原子 3  x 座標  y 座標  z 座標
      :
```

という Cartesian 座標形式のみに対応しています。

### ○タイトルセクション (終状態)

終状態のタイトルを指定します。

○分子情報セクション (終状態)

終状態構造の電荷、スピン多重度、分子構造を指定します。

○タイトルセクション (中間状態)

中間状態のタイトルを指定します。

○分子情報セクション (中間状態)

中間状態構造 (遷移状態や中間安定状態の候補構造) の電荷、スピン多重度、分子構造を指定します。

○追加入力セクション \$react~\$end

一部の原子座標を固定する時や、外部ファイルから座標を読み込む時に指定します。

(例 1)

%NProcShared=16	Link 0 コマンド
# B3LYP/STO-3G React=(NGeom=2,NBeads=30)	ルートセクション
Reactant	タイトルセクション
0 1	分子情報セクション
O	0.00000000 0.00000000 0.00000000
H	0.70000000 -0.70000000 0.00000000
H	-0.70000000 0.70000000 0.00000000
Product	タイトルセクション
0 1	分子情報セクション
O	0.00000000 0.00000000 0.00000000
H	0.70000000 -0.70000000 0.00000000
H	-0.70000000 -0.70000000 0.00000000
\$react	追加入力セクション
Freeze 1,2	
\$end	

(例 2)

%NProcShared=16	Link 0 コマンド
# B3LYP/STO-3G React=(NGeom=2,NBeads=30)	ルートセクション
Title	タイトルセクション
0 1	分子情報セクション
O	0.00000000 0.00000000 0.00000000
H	0.70000000 -0.70000000 0.00000000
H	-0.70000000 0.70000000 0.00000000
\$react	追加入力セクション
molecule-1.xyz	
molecule-2.xyz	
\$end	

## Gaussian IRC 形式のアウトプットファイル (\*.log)

Gaussian IRC 計算のアウトプットファイル(\*.log)を模した形式のファイルです。GaussView を使用して、反応経路に沿った各分子構造のエネルギーや反応のアニメーションを確認することができます。

## ビーズ座標ファイル (\*.pdb)

各ビーズの座標を出力したファイルです。フォーマットは PDB 形式です。

(例)

```
REMARK      Energy: -0.0607709343218 Hartree
MODEL
1
ATOM       1 C          -0.578   0.455   0.410           C
ATOM       2 H          -0.563   1.003   1.361           H
ATOM       3 H          -1.522  -0.111   0.377           H
ATOM       4 H          -0.619   1.200  -0.395           H
ATOM       5 C           0.613  -0.447   0.271           C
ATOM       6 H           1.593   0.056   0.296           H
ATOM       7 O           0.516  -1.645   0.140           O
ENDMDL
REMARK      Energy: -0.0606977454264 Hartree
MODEL
2
ATOM       1 C          -0.578   0.462   0.416           C
ATOM       2 H          -0.466   1.191   1.226           H
ATOM       3 H          -1.507  -0.099   0.593           H
ATOM       4 H          -0.723   1.023  -0.518           H
ATOM       5 C           0.604  -0.456   0.306           C
ATOM       6 H           1.590   0.026   0.418           H
ATOM       7 O           0.497  -1.644   0.104           O
ENDMDL
REMARK      Energy: -0.0605417417882 Hartree
MODEL
3
ATOM       1 C          -0.577   0.476   0.417           C
ATOM       2 H          -0.382   1.350   1.046           H
ATOM       3 H          -1.468  -0.037   0.807           H
ATOM       4 H          -0.849   0.837  -0.586           H
ATOM       5 C           0.594  -0.458   0.335           C
ATOM       6 H           1.581  -0.000   0.518           H
ATOM       7 O           0.480  -1.635   0.081           O
ENDMDL
:
```

## ビーズ座標ファイル (\*.xyz)

各ビーズの座標を出力したファイルです。フォーマットは XYZ 形式です。

(例)

```
7
Comment
C -0.606267591892 0.475214675944 0.413376315277
H -0.453036595356 1.18791813676 1.21957788655
H -1.51348725183 -0.0931600235451 0.596000294483
H -0.717079049365 1.0285388928 -0.517163692548
C 0.603596388378 -0.467952407339 0.307094395027
H 1.58426568244 0.0259936922539 0.408699184127
O 0.527530264217 -1.66765803924 0.115563221315
7
comment
C -0.604462900522 0.481730628535 0.409236977147
H -0.366404070111 1.33464112967 1.03832425592
H -1.47255105667 -0.0359546729396 0.808312970264
H -0.843650237171 0.841648408657 -0.589778700423
C 0.59969688427 -0.474582009388 0.335098426921
H 1.58313832368 -0.00821193989382 0.505673287554
O 0.519039380061 -1.66521201691 0.0951991194189
7
comment
```

原子数  
コメント  
元素、座標(Å)

```
C -0.601891332982 0.487276305409 0.400160441733
H -0.29907523412 1.44038878472 0.823008068824
H -1.40240344581 0.0630643027687 1.00167456392
H -0.978897721683 0.649958739367 -0.607576346906
C 0.596843260058 -0.479652508369 0.357193558939
H 1.57543465037 -0.0385587942148 0.602527956383
O 0.516516589399 -1.65988016384 0.0706028040931
:
```

### 最適化過程のエネルギー情報 (\*.dat)

反応経路上のエネルギー曲線の最適化過程を記録したファイルです。1行目はビーズの番号、2行目以降は各ビーズのエネルギー値が最適化サイクルごとに出力されています。データはタブ区切りで出力されており、表計算ソフトで読み込むことにより、簡単にグラフ化することができます。

(例)

```
1      2      3      4 ...
-150.944059796 -150.894922863 -150.749934491 -150.609383907 ...
-150.944059796 -150.916729194 -150.877929958 -150.848885367 ...
-150.944059796 -150.927729824 -150.90335675 -150.89397191 ...
-150.944059796 -150.9380063 -150.928546698 -150.923830681 ...
:
```

## よくある質問

### アカデミック利用において制限はありますか？

- ソフトウェアの機能において制限はありませんが、研究成果を論文等で公開の際には本ソフトウェアを利用した旨を記載して下さい。

(例) Reaction plus Pro ver. 2 を使って TS 候補構造を求め、それを初期値にしてソフトウェア AAA による TS 計算を行った場合

... The molecular geometries for the transition states were first estimated by Reaction plus Pro (ver. 2) software package [1], based on the nudged elastic band (NEB) method [2], and were subsequently re-optimized using AAA software package. ...

[1] Software to optimize reaction paths along the user's expected ones, HPC Systems Inc., <http://www.hpc.co.jp/chem/react2.html> (written in Japanese)

[2] H. Jónsson, G. Mills, K. W. Jacobsen, Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions, in Classical and Quantum Dynamics in Condensed Phase Simulations, Ed. B. J. Berne, G. Ciccotti and D. F. Coker, 385 (World Scientific, 1998); G. Henkelman and H. Jónsson, Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points, J. Chem. Phys. 113, 9978 (2000)

### 動作しないのですが？

- Reaction plus が正しくインストールされていることをご確認下さい。
- プログラムのインストール先やインプットファイルのパスに全角文字や半角カナ等が含まれていないか、ご確認下さい。
- ライセンスファイルが正しいディレクトリに置かれていることをご確認下さい。

### 実行直後に終了してしまうのですが？

- ルートセクションのキーワードが正しく与えられているか、今一度ご確認下さい。  
特に「Force」「Freq」キーワードが指定されていると、Reaction plus が走りません。これらが指定されている場合は、当該キーワードを削除してから改めて計算実行をお試し下さい。
- 読み込むべき chk ファイルがないのに、%chk 行が指定されていませんか？ chk ファイルを初期軌道として利用しない場合は、%chk 行を削除して計算を行って下さい。特に、GaussView でインプットファイルを作成すると、デフォルトでは%chk 行が指定されてしまうので、ご注意下さい。
- NGeom オプションが正しく指定されていることをご確認下さい。NGeom 無指定時は NGeom=1 と解釈され、追加入力セクションで外部からファイルを読み込まない限り、計算が走りません。
- 分子構造のフォーマットをご確認下さい。タイトル→座標→タイトル→座標→…のようにタイトルと座標を交互に入力する必要があります。分子構造は Cartesian 座標で与えて下さい。

### 実行後しばらくしてエラー終了してしまうのですが？

- ユーザが与えた計算手法や分子構造が適切でないと、エネルギーが収束せず、エラー終了することがあります。
- 本プログラムは、始状態・終状態の間に遷移状態があることを前提としています。エネルギーポテンシャルを単調に登るだけ、下るだけの反応の場合、正しく動作しないことがあります。
- 本プログラムは、入力した始状態・中間状態・終状態の座標を線形に補間してビーズを生成させています。補間の際に原子間距離が近すぎる構造が生成されると、正しく動作しないことがあります。

### 計算を停止させたいのですが？

- (Linux 版 Pro) reactg16 → g16 または reactg09 → g09 の順に停止させて下さい。
- (Linux 版 Express) reactx → sqm\_rp または reactux → cp2k\_rp の順に停止させて下さい。
- (Windows 版) 計算停止プログラム kill-react.bat を使用して停止させて下さい。

### 計算結果が思わしくありません

- QST3 形式インプットの分子構造を始状態・中間状態・終状態の順に指定していませんか？ この形式での正しいフォーマットは始状態・終状態・中間状態の順です。ただし、追加入力セクションで構造を与える場合は、始状態・中間状態・終状態の順になります。
- 始状態・終状態・中間状態の分子構造において原子のラベル番号の対応が適切でないと、不適当な反応経路や遷移状態構造を与えることがあります。
- 始状態・終状態・中間状態の座標軸が適切でないと、不適当な反応経路や遷移状態構造を与えることがあります。FitAxis オプションを利用するか、手動で座標軸を合わせて下さい。
- Freeze オプションを利用する場合、固定したい原子座標を始状態～終状態の全ての構造で一致させた上、FitAxis オプションを指定せずに計算して下さい。
- 想定外の初期経路が得られてしまった場合、初期経路における中間構造座標の生成が上手くいっていない可能性があります。そのような場合、NoAdjust オプションを試してみると、うまくいくかもしれません。
- 本プログラムでは、ビーズの動きが十分小さくなったと判断した時に収束したとみなし、最適化が終了します。その場合、化学的には収束が十分でないことがあります。もし収束が十分でないと思われる時は、より厳しい収束閾値を用いて再計算を行うか、インプットの分子構造を修正して計算を行ってみて下さい。

## 動作環境

### Linux 版

OS	Red Hat Enterprise Linux or CentOS 7.x, 6.x
----	---

### Windows 版

OS	Windows 10, Windows 8.1, Windows 7 (いずれも 64bit 版)
----	---

※ すべての環境について動作を保証するものではありません。動作報告例については弊社ウェブページをご覧ください。

※ インプット作成やアウトプット表示については、GaussView 5.0.9 for Windows にて動作確認を行っております。