

Platypus-QM
プログラム利用マニュアル

β 版 (2012.11.12)

目次

第 1 章	はじめに	1
1.1	本マニュアルの読み方	1
1.1.1	ログインシェル	1
1.1.2	想定ディレクトリ構造	1
1.1.3	表記規則	1
第 2 章	環境設定例と MPI プログラム実行例	5
2.1	環境設定例	5
2.2	MPI プログラムの実行例	5
2.2.1	対話実行例	5
2.2.2	理研 京のバッチ実行例	5
第 3 章	入力データ例と実行例	7
3.1	RHF	7
第 4 章	入力データ	9
4.1	書式の仕様	9
4.2	データグループ一覧	12
4.3	&Atoms	13
4.3.1	書式	13
4.3.2	オプション	13
4.3.3	その他項目	13
4.3.4	記述例	14
4.4	&Basis	15
4.4.1	書式	15
4.4.2	基底関数系ライブラリファイルの書式	15
4.4.3	環境変数	16
4.4.4	オプション	16
4.4.5	その他項目	16
4.4.6	&Atoms の各原子への基底関数系割り当ての優先順位	17
4.4.7	記述例	17
4.5	&PointCahrges	19
4.5.1	書式	19
4.5.2	オプション	19
4.5.3	その他項目	19

4.5.4	記述例	19
4.6	&Control	20
4.6.1	書式	20
4.6.2	オプション	20
4.6.3	記述例	20
4.7	&QM	21
4.7.1	書式	21
4.7.2	オプション	21
4.7.3	記述例	21
4.8	&Integral	22
4.8.1	書式	22
4.8.2	オプション	22
4.8.3	記述例	22
4.9	&RHF	23
4.9.1	書式	23
4.9.2	オプション	23
4.9.3	記述例	23
付録 A	理論概要	25
A.1	HF	25
付録 B	エラーメッセージ	27
付録 C	インストール	29
C.1	Platypus-QM のコンパイル	29

第 1 章 はじめに

1.1 本マニュアルの読み方

1.1.1 ログインシェル

このマニュアルでは、ログインシェルなどを `bash` に限定して説明しています。その他のシェルを用いている場合は、適宜読み替えてください。

1.1.2 想定ディレクトリ構造

実行形式ファイルのインストールディレクトリ

下記のようなディレクトリ構造を想定しています。

`$HOME/local/platypus-qm/bin` Platypus-QM 実行形式ファイルのディレクトリ

ユーザのディレクトリ

下記のようなディレクトリ構造を想定しています。

`$HOME/work/platypus` 実行ディレクトリ

1.1.3 表記規則

このマニュアルの説明では、下記の表記規則を用いています。

(1) タイプライター (Typewriter) 体

このようなタイプライター体で表記された項目は、表記された文字列をそのまま記述する必要があることを表しています。たとえば、入力データにおけるオプションやキーワードの説明や、コマンド、ファイル名、環境変数などの説明で用いています。入力データにおけるほとんどの項目は、大文字と小文字のどちらで記述しても同一とみなされます (同一とみなされない項目については、その都度明記があります) が、コマンド、ファイル名、環境変数などでは、同一とみなされないのでご注意ください。

例 1: マニュアル表記例 : `&RHF`

利用者の記述例 1 : `&RHF`

利用者の記述例 2 : `&rhf`

上記は同一であるとみなされます。

例 2: マニュアル表記例 : `platypus_qm_energy_rhf`

利用者の記述例 : `platypus_qm_energy_rhf`

大文字と小文字は区別されます。

(2) イタリック (*Italic*) 体

イタリック体で表記された項目は、数値や文字列等を与える必要があることを表しています。たとえば、次のものがあります (詳細は後述)。

- *integer* (整数値)
- *real_number* (実数値)
- *digit* (数字: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
- *string* (文字列)
- *file_name* (ファイル名)
- *basis_name* (基底関数系名)
- *symbol* (元素記号)
- *id* (識別番号, 1 個以上の *digit* の並び)

など

例 1: マニュアル表記例 : PRINT = *integer*
 利用者の記述例 1 : PRINT = 1
 利用者の記述例 2 : PRINT = 2

例 2: マニュアル表記例 : BasisSet = *basis_name*
 利用者の記述例 1 : BasisSet = MINI-4
 利用者の記述例 2 : BasisSet = 6-31G

(3) [A]

[と] で囲まれた項目は、省略するか記述するかを選択が可能であることを表しています。省略した場合は、その項目に対して省略時既定値が与えられます。

例 1: マニュアル表記例 : [PRINT = *integer*]
 利用者の記述例 1 : !記述なし (PRINT には省略時既定値が与えられる)
 利用者の記述例 2 : PRINT = 1

! は、コメント開始を示す文字です (後述)。

(4) { A | B | C }

{ と } で囲まれた項目は、| で区切られた項目のうちのいずれか 1 つを選択する必要があることを表しています。

例 1: マニュアル表記例 : PRUNED = { YES | NO }
 利用者の記述例 1 : PRUNED = YES
 利用者の記述例 2 : PRUNED = NO

(5) ...

この表記は、これの直前の項目の 1 回以上の反復を表しています。また、0 回以上の反復は、[...] のように表しています。

例 1: マニュアル表記例 : A [...]
 利用者の記述例 1 : A
 利用者の記述例 2 : A A A

(6) $A + B$

+ の表記は、前後の項目を連結して記述することを表しています。連結するときには、+ 自体や空白などを含めずに記述してください。

例 1: マニュアル表記例 : $symbol + id$
利用者の記述例 : Mg123

(7) \$

これは、コマンドプロンプトを表しています。

例 1: `$`

(8) コーナーが丸い囲み

これは、端末での操作内容を表しています。

例 1: `$ command [arg [...]]`

(9) コーナーが直角の囲み

これは、ファイルの内容を表しています。

例 1: `line1
line2
line3`

(10) バックスラッシュ (\)

これは、コマンドラインの継続を表しています。

例 1: `$ command arg1 \
arg2`

第 2 章 環境設定例と MPI プログラム実行例

2.1 環境設定例

Platypus-QM を使用するには、./bahrc に次のように入力してください。

```
export PLATYPUSQMDIR=/usr/local/platypus-qm
export PATH="$PLATYPUSQMDIR/bin:$PATH"
```

設定を反映させるためにログインし直してください。

2.2 MPI プログラムの実行例

実行例を示します。

2.2.1 対話実行例

MPI プログラムを実行するには `mpiexec/mpirun` コマンドを使います。 `mpiexec` では並列実行時のプロセス数をオプション `-np` で指定します。たとえば、2 プロセスで実行させる場合は次のようにします。

```
$ mpirun -np 2 program
```

2.2.2 理研 京のバッチ実行例

Flat MPI でのバッチジョブスクリプトファイルの例 (`go_K.sh`) を示します。

```
#!/bin/bash -x
#
#PJM --rsc-list "node=2"
#PJM --rsc-list "elapsed=00:05:00"
#PJM --mpi "rank-map-bynode"
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ./platypus_qm %r:./"
#PJM --stgin "rank=* ./qm.inp %r:./"
#PJM --stgin "rank=* ./BASISFILE %r:./"
#PJM -s
#
. /work/system/Env_base

export OMP_NUM_THREADS=8
LPGPARAM="lpgparm -s 32MB -d 32MB -h 32MB -t 32MB -p 32MB"
```

```
mpiexec ${LPGPARM} ./platypus_qm
```

バッチジョブの投入は、次のようにします。

```
$ pjsub go.sh
```

ジョブの状態表示は、次のようにします。

```
$ pjstat
```

詳細情報を付加するには、次のようにします。

```
$ pjstat -v
```

ジョブをキャンセルするには、キャンセルしたいリクエスト ID (REQID) を指定して、次のようにします。

```
$ pjdel REQID
```

第 3 章 入力データ例と実行例

いくつかの計算例を示します。入力データファイル名は、`qm.inp` の固定名称です。また、以下の説明では、基底関数系ライブラリのファイル名を `BASISFILE` とし、必要な基底関数系が記載されているものとしています。

3.1 RHF

入力データ例を示します。

```
&qm
  wftype = rhf
&end

&rhf
  energyconv = 1.d-7
  scfconv = 1.d-04
  maxcyc = 20
&end

&basis
  BasisFile = BASISFILE
  BasisSet = MINI-4
&end

&atoms
  units = au
    O1      0.0      0.0      0.0
    H2      1.43153  0.0      1.10941
    H3      -1.43153 0.0      1.10941
&end
```


第 4 章 入力データ

4.1 書式の仕様

入力データの書式には次の規則があります。

(1) 1 行内のデータ有効範囲

1024 カラムまでがデータの有効範囲です。それ以降の文字は無視されます。

(2) カラム位置

全ての項目は、どのカラム位置に記述しても構いません。

例 1:

```
Print = 1
```

例 2:

```
Print = 1
```

(3) 空行

空行は無視されます。どの位置にも挿入可能です。

例 1:

```
Print = 1
```

例 2:

```
Print = 1
```

(4) コメントの記述

コメント開始文字は、!, #, ; です。これらの文字から行末までは、入力データとして無視されます。これらの文字は、基底関数名称やファイル名などに使用することができませんのでご注意ください。

例 1:

```
Print = 1 ! コメント
```

例 2:

```
Print = 1 # コメント
```

例 3:

```
Print = 1 ; コメント
```

(5) 大文字と小文字の区別

ほとんどの項目は、大文字と小文字のどちらで記述しても同一とみなされますが、例外として、下記の項目は区別されますのでご注意ください。

- コマンドや引数

- ディレクトリ名やファイル名
- 環境変数
- スクリプト

など

例 1: `PRINT = 1 # これらは同じ`

例 2: `Print = 1 # これらは同じ`

(6) データグループの構造

データは、グループ化されています。接頭辞`&`が付いたデータグループの名称で始まり、`&END`で終わります。

例 1: `&RHF
&END`

(7) データグループの外側

データグループの外側には、空白とコメント以外の記述は許されません。

例 1: `&RHF
&END # コメント`

(8) データグループの種別

データグループ内に記述すべき項目の内容で分類すると、次の2種類になります。

- オプションのみである。
- それ以外のデータ項目がある。(オプションを含む場合、オプションとそれ以外のデータ項目を、同一行内に記述できません)

後者のデータグループには、次のものが該当します。

- `&Atoms`
- `&Basis`
- `&PointCharges`
- `&GaussianCharges`

(9) オプションの記述

次のように記述します。

- オプション = 値
 - オプション = (値1 [値2 [...]])
- () の中に値のリストを空白区切りで記述してください。また、改行することはできません。

例 1: `WeightFactors = 1`

例 2: `WeightFactors = (1 1)`

(10) ファイル名、ディレクトリ名

空白やコメント開始文字を含めることはできません。

4.2 データグループ一覧

データグループの一覧は次のとおりです。

(1) オプションおよび、それ以外の書式を含むデータグループ

名称	説明
&Atoms	原子の座標など
&Basis	基底関数系など
&PointCharges	点電荷の座標など

(2) オプションのみのデータグループ

名称	関連する計算
&Control	全部
&QM	QM/MD 連成計算
&Integral	積分
&RHF	RHF, RDFT

4.3 &Atoms

原子の座標を記述します。

4.3.1 書式

```
&Atoms
  [ Units = { ang | au } ]
  atom_name1 x1 y1 z1 [ charge1 ] [ basis_name1 [ basis_library_name1 ] ]
  [ atom_name2 x2 y2 z2 [ charge2 ] [ basis_name2 [ basis_library_name2 ] ] ]
  [ ... ]
&End
```

4.3.2 オプション

オプション	値	説明
Units		長さの単位 [省略時既定値: ang]
	ang	オングストローム (Å)
	au	原子単位 (a.u.)

4.3.3 その他項目

項目	説明
<i>atom_name</i>	原子の名称 <i>symbol</i> [+ <i>id</i>]
<i>symbol</i>	元素記号 (H-Ba) (例: H, Mg)
<i>id</i>	識別番号 &Basis における基底関数系の割り当てに用いる。 <i>digit</i> [+ <i>digit</i> [...]] (例: H1, Mg12)
<i>x</i>	X 座標 <i>real_number</i>
<i>y</i>	Y 座標 <i>real_number</i>
<i>z</i>	Z 座標 <i>real_number</i>
<i>charge</i>	電荷 (単位: a.u.) <i>real_number</i> [省略時既定値: 原子番号の値]

(continued)

項目	説明
<i>basis_name</i>	この原子に対して割り当てる基底関数系 <i>string</i> [省略時既定値: &Basis で定義されたもの]
<i>basis_library_name</i>	基底関数系 <i>basis_name</i> を記述してある基底関数系ライブラリファイル <i>string</i> 絶対パスあるいは PLATYPUS_QM_BASIS_DIR からの相対パス [省略時既定値: &Basis で定義されたもの]

4.3.4 記述例

1. オプション `units` で原子単位 (a.u.) を指定

```
&atoms
  units = au
    O1      0.0      0.0  0.0
    H2      1.43153  0.0  1.10941
    H3     -1.43153  0.0  1.10941
&end
```

2. 基底関数系を割り当てる

```
&atoms
  units = au
    O1      0.0      0.0  0.0      MINI-4
    H2      1.43153  0.0  1.10941  MINI-4
    H3     -1.43153  0.0  1.10941  MINI-4
&end
```

4.4 &Basis

&Atoms で記述した各原子へ、どの基底関数系を割り当てるか指定します。

4.4.1 書式

```
[
&Basis
  [ BasisFile = basis_library_name ]
  [ BasisSet = basis_name ]
  [
  # basis_set_definition_block
  def symbol = basis_name
    # shell_definition_block
    shell_type number_of_primitives [ scale_factor ]
    # primitive_function_definition_line
    orbital_exponent coefficient [ coefficient ]
    [ ... ]
    [ ... ]
  [ ... ]
  ]
  [
  # basis_set_assignment_line
  set { symbol | symbol+id } = basis_name [ basis_library_name ]
  [ ... ]
  ]
&End
]
```

次の書式の基底関数系ライブラリファイルを作成しておき、その中から引用して割り当てる方法があります。引用は、*symbol* と *basis_name* の組を指定して行います。

4.4.2 基底関数系ライブラリファイルの書式

&Basis の基底関数系定義ブロックと同じ書式です。

```
[
# basis_set_definition_block
def symbol = basis_name
  # shell_definition_block
  shell_type number_of_primitives [ scale_factor ]
  # primitive_definition_line
  orbital_exponent coefficient [ coefficient ]
  [ ... ]
  [ ... ]
  [ ... ]
]
```

4.4.3 環境変数

環境変数	値	説明
PLATYPUS_BASISDIR	<i>string</i>	基底関数系ライブラリファイルのディレクトリ [省略時既定値: ./]
PLATYPUS_BASISFILE	<i>string</i>	基底関数系ライブラリファイルの名称 PLATYPUS_BASISDIR からの相対パスあるいは絶対パス [省略時既定値: BASISFILE]
PLATYPUS_BASISSET	<i>string</i>	基底関数系の名称 [省略時既定値: STO-3G]

4.4.4 オプション

オプション	値	説明
BasisFile	<i>string</i>	基底関数系ライブラリファイルの名称 絶対パスか、PLATYPUS_BASISDIR からの相対パス [省略時既定値: 環境変数 PLATYPUS_BASISFILE の値]
BasisSet	<i>string</i>	基底関数系の名称 [省略時既定値: 環境変数 PLATYPUS_BASISSET の値]

4.4.5 その他項目

項目	説明
def	基底関数系を定義するブロックを開始するキーワード
set	基底関数系の割り当て行を開始するキーワード
<i>atom_name</i>	基底関数系を割り当てる原子の名称 <i>symbol</i> [+ <i>id</i>]
<i>symbol</i>	元素記号 (例: H, Mg)
<i>symbol+id</i>	元素記号 + 識別番号 (例: H1, Mg12)
<i>id</i>	識別番号 <i>digit</i> [+ <i>digit</i> [...]]
<i>basis_name</i>	基底関数系の名称 <i>string</i>
<i>basis_library_name</i>	<i>basis_name</i> を記述してある基底関数系ライブラリファイルの名称 絶対パスあるいは PLATYPUS_BASISDIR からの相対パス <i>string</i> [省略時既定値: BasisFile で指定されたもの]

(continued)

項目	説明
<i>shell_type</i>	シェルタイプ { S P D SP }
<i>number_of_primitives</i>	シェルを構成するプリミティブの個数 <i>integer</i> ここで指定した個数分のプリミティブ定義行を記述する
<i>scale_factor</i>	スケール因子 <i>real_number</i> [省略時既定値: 1.0]
<i>orbital_exponent</i>	軌道指数 <i>real_number</i>
<i>coefficient</i>	短縮係数 <i>real_number</i> SP は、 <i>coefficient</i> を 2 つ、それ以外は 1 つ記述する。

補足 1. *basis_library_name* の中に、def で定義された *basis_name* と同じ名称のものがあつた場合は、def で定義されたものが優先されます。

4.4.6 &Atoms の各原子への基底関数系割り当ての優先順位

上位のものが優先的に割り当てられます。

順位 1. &Atoms の中での基底関数系の指定

その原子に対して、指定された基底関数系を割り当てる。

順位 2. `set symbol+id = basis_name [basis_library_name]` での指定

同じ `symbol+id` の原子名をもつ原子 (`symbol+id`) に対して、指定された基底関数系を割り当てる。

順位 3. `set symbol = basis_name [basis_library_name]` での指定

上記までで割り当てられなかった原子に対して、同じ `symbol` をもつ原子 (`symbol`、`symbol+id`) に対して、指定された基底関数系を割り当てる。

順位 4. 指定なし

上記までで割り当てられなかった原子に対して、BasisSet で指定された基底関数系を割り当てる。

4.4.7 記述例

1. 基底関数系ライブラリファイルからの引用し、全ての原子に対して同じ基底関数系を指定

```
&Basis
  BasisFile = ../BasisLib
  BasisSet  = 6-31G
&End
```

2. 基底関数系ライブラリファイルからの引用し、元素毎に基底関数系を指定

```
&Basis
  BasisFile = ../BasisLib
  set H = 6-31G
  set O = 6-31G**
&End
```

3. H の基底関数系を定義して、H に対して割り当てる

```
&Basis
  BasisFile = ../BasisLib
  set H = 6-31G
  set O = 6-31G**

  def H = 6-31G
    S 3
      18.7311370    0.03349460
      2.8253937   0.23472695
      0.6401217   0.81375733
    S 1
      0.1612778   1.0000000
  &End
```

4. 基底関数系ライブラリファイルの例

```
def H = 6-31G
  S 3
    18.7311370    0.03349460
    2.8253937   0.23472695
    0.6401217   0.81375733
  S 1
    0.1612778   1.0000000
```

4.5 &PointCharges

点電荷の座標と電荷を記述します。

4.5.1 書式

```
[
  &PointCharges
    [ Units = { ang | au } ]
      x1 y1 z1 charge1
    [ x2 y2 z2 charge2 ]
    [ ... ]
  &End
]
```

4.5.2 オプション

オプション	値	説明
Units		長さの単位 [省略時既定値: ang]
	ang	オンGSTローム (Å)
	au	原子単位 (a.u.)

4.5.3 その他項目

項目	説明
<i>atom_name x</i>	X 座標 <i>real_number</i>
<i>y</i>	Y 座標 <i>real_number</i>
<i>z</i>	Z 座標 <i>real_number</i>
<i>charge</i>	電荷 (単位: a.u.) <i>real_number</i>

4.5.4 記述例

```
&PointCharges
# x y z charge
  1.0 0.0 0.0 1.1
  2.0 0.0 0.0 2.1
&End
```

4.6 &Control

4.6.1 書式

```
[  
  &Control  
    [ PRINT = integer ]  
  &End  
]
```

4.6.2 オプション

オプション	値	説明
PRINT	<i>integer</i>	プリントレベル [省略時既定値: 0]

4.6.3 記述例

```
&Control  
  print = 1  
&End
```

4.7 &QM

4.7.1 書式

```
[
  &QM
    [ WFTYPE = { RHF } ]
    [ Check = { NO | Energy } ]
  &End
]
```

4.7.2 オプション

オプション	値	説明
WFTYPE		波動関数タイプ [省略時既定値: RHF]
	RHF	RHF 波動関数
Check		QM 計算確認のための終了位置指定 [省略時既定値: NO]
	NO	しない(力の計算まで全て実行)
	Energy	エネルギー計算までで終了

4.7.3 記述例

```
&QM
  wftype = rhf
&End
```

4.8 &Integral

4.8.1 書式

```
[  
  &Integral  
    [ ERITHRES = real_number ]  
  &End  
]
```

4.8.2 オプション

オプション	値	説明
ERITHRES	<i>real_number</i>	ERI しきい値 [省略時既定値: 1.0E-8]

4.8.3 記述例

```
&Integral  
  erithres = 1.0d-8  
&End
```

4.9 &RHF

4.9.1 書式

```
[
&RHF
  [ ION = integer ]
  [ MAXCYC = integer ]
  [ SCFCONV = real_number ]
  [ ENERGYCONV = real_number ]
  [ DIISIZE = integer ]
&End
]
```

4.9.2 オプション

オプション	値	説明
ION	<i>integer</i>	分子の電荷 [省略時既定値: 0]
MAXCYC	<i>integer</i>	最大繰り返し回数 [省略時既定値: 30]
SCFCONV	<i>real_number</i>	収束判定値、前回の密度行列との差の RMS [省略時既定値: 1.0E-6]
ENERGYCONV	<i>real_number</i>	収束判定値、前回のエネルギーとの差 [省略時既定値: 1.0E-10]
DIISIZE	<i>real_number</i>	DIIS において過去何回分格納するか [省略時既定値: 8]

- SCFCONV と ENERGYCONV の両方が満たされたとき収束したとみなされます。

4.9.3 記述例

```
&RHF
  MAXCYC      = 20
  SCFCONV     = 1.0E-4
  ENERGYCONV = 1.0E-8
&END
```


付録 A 理論概要

ここでは、実装している計算機能の概要を説明します。詳細については、参考文献を参照ください。
分子軌道法における基本式は、シュレーディンガー方程式です。

$$\hat{H}\Psi = E\Psi \quad (\text{A.1})$$

\hat{H} はハミルトニアン演算子、 Ψ は波動関数、 E はエネルギーを表しています。

実際は、近似的に解かざるを得ないため、各問題に適した様々な手法が提案されています。Platypus-QM では、以下の手法が利用可能です。

A.1 HF

HF (Hartree-Fock) 法では、波動関数 Ψ を単一のスレーター行列式 Φ_0 とした手法です。

$$\Psi = \Phi_0 \quad (\text{A.2})$$

スレーター行列式は、MO (分子軌道; Molecular Orbital) ϕ_i から組み立てられたものです。どの MO に、電子が占有しているかを表現しています。 N 電子系のスレーター行列式をスピン軌道で、書くと次式のようになります。

$$\Phi_0 = |\phi_1(1)\phi_2(2)\dots\phi_N(N)| \quad (\text{A.3})$$

空間 MO ϕ_i に α スピンの電子が占有しているものを ϕ_i 、 β スピンの電子が占有しているものを $\overline{\phi_i}$ と表し、

$$\Phi_0^{RHF} = |\phi_1(1)\overline{\phi_1(2)}\dots\phi_{N/2}(N-1)\overline{\phi_{N/2}(N)}| \quad (\text{A.4})$$

のように、同一空間 MO に 2 電子占有している形式を、RHF (Restricted Hartree-Fock) 法と呼びます。電子数が偶数の場合のみ、計算可能です。奇数の場合は、1 電子占有 MO 数を N_{open} として、high spin 状態を、

$$\Phi_0^{ROHF} = |\phi_1\overline{\phi_1}\dots\phi_{(N-N_{open})/2}\overline{\phi_{(N-N_{open})/2}}\phi_{(N-N_{open})/2+1}\dots\phi_{(N-N_{open})/2+N_{open}}| \quad (\text{A.5})$$

ROHF (Restricted Open shell Hartree-Fock) 法と呼びます。

α スピンの電子が占有している MO と、 β スピンの電子が占有している MO を区別した形式

$$\Phi_0^{UHF} = |\phi_1(1)\phi_2(2)\dots\phi_{N_\alpha}(N_\alpha)\overline{\phi_{N_\alpha+1}(N_\alpha+1)}\dots\overline{\phi_{N_\alpha+N_\beta}(N_\alpha+N_\beta)}| \quad (\text{A.6})$$

を、UHF (Unrestricted Hartree-Fock) 法と呼びます。

HF 法は、分子軌道法の基準となる近似であるため、最初の計算として行われます。計算結果として、エネルギー E 、MO 係数 $\{C\}$ 、軌道エネルギー $\{\epsilon\}$ などが得られます。

さらに近似レベルを上げた計算が必要な場合は、CI 法や CASSCF 法による計算を行います。これらの HF 計算から近似レベル上げた手法は、ひとまとめにしてポスト HF 法と呼ばれています。

分子軌道法では、MO ϕ_i を、AO (原子軌道; Atomic Orbital) χ_μ の重ね合わせで表現します。AO は、基底関数 (basis function) とも呼ばれます。

$$\phi_i = \sum_{\mu}^{N_{AO}} C_{i\mu} \chi_{\mu} \quad (\text{A.7})$$

どのような基底関数を選択するかによって、計算結果に影響がおよびます。選択ガイドラインは、参考文献にゆだねます。一般に、計算結果の信頼度を向上するためには、大きな基底関数系 (Basis set) を用います。そのため、基底関数の個数 N_{AO} が大きくなり、計算コストが大きくなります。このような場合では、並列数を大きくすることにより、計算時間を短縮できる可能性があります。

HF 法では、次式のような一般化固有値問題を解いています。

$$FSC = C\epsilon \quad (\text{A.8})$$

F は Fock 行列、 S は重なり積分、 C は MO 係数、 ϵ は軌道エネルギーです。

式 (A.8) の Fock 行列 F は、求めたい MO 係数行列 C を含んでいるため、初期 MO 係数 C^{init} を与えて、 C が収束するまで式 (A.8) を反復して解く必要があります。どのような初期 MO 係数 C^{init} を与えるかによって、収束までの反復回数が増減します。

付録 B エラーメッセージ

コード	メッセージ / 説明
200	File Not Found ファイルが存在しない
201	File Open Error ファイルオープンエラー
202	File Close Error ファイルクローズエラー
203	File Rrewind Error ファイルリワインドエラー
204	File Read Error ファイルリードエラー
205	File Write Error ファイルライトエラー
300	Group Not Specified データグループが存在しない
301	Group Begin データグループの開始エラー
302	Group End データグループの終了エラー
303	Group EOF データグループが終了していない
304	Invalid Group Name データグループの名称エラー
305	Duplicate Group 同一データグループが複数存在する
311	Incorrect Option Syntax オプション書式エラー
312	Invalid Option Name オプション名称エラー
313	Duplicate Option 同一オプションが複数存在する
321	Invalid Value 不正な値である
322	Out Of Range 値が許されない範囲である

B. エラーメッセージ

コード	メッセージ / 説明
323	Inconsistent Value オプションに不整合がある
331	Invalid Format 不正な書式である
341	Invalid Name 不正な名称である
351	Duplicate 二重定義がある
352	No Atoms 原子の指定がない
355	No Basis Set 基底関数の指定がない
999	Internal Error 内部エラー

付録 C インストール

Platypus-QM のインストールには、下記のコンパイラやライブラリなどが必要です。

- (1) コンパイラ
 - Fortran
- (2) MPI ライブラリ
- (3) 数学ライブラリ
 - BLAS, LAPACK

例として、下記マシン環境でのコンパイル手順を示します。

- (1) 理化学研究所 京コンピュータ

C.1 Platypus-QM のコンパイル

1. 配布ソースコードの展開

```
$ tar xvf platypus-qm-0.63-K.tar.bz2
```

2. コンパイル

```
$ ./runConfigure_K  
$ make
```

3. 実行確認

- (1) ジョブ投入可能な適切なディレクトリへ移動し、
- (2) コンパイルしたディレクトリから、下記をコピー

- src/platypus_qm
- data/qm.inp
- data/BASISFILE
- data/go_K.sh

- (3) go_K.sh の内容を適宜変更

- (4) ジョブ投入

```
$ pjsub go_K.sh
```

- (5) 結果を data/qm_C6H6_H2O_PC.out_ref と比較し、一致していれば、コンパイル成功。

