



GAMESS ユーザーマニュアル



目次

1	GAMESS について	2
2	GAMESS 実行方法	3
2.1	GAMESS の並列計算手法	3
2.2	rungms の編集方法	3
2.3	GAMESS バイナリの実行方法	7
3	既知の問題点	8
付録 A	9
A.1	HPC システムズ お問い合わせ先	9

1 GAMESS について

GAMESS は、非経験的分子軌道法/密度汎関数理論計算プログラムです。開発の過程で複数の系統に枝分かれしており、著名なものとして以下が挙げられます。

1. Iowa 州立大の M.S.Gordon を中心とした研究グループが中心となって¹開発及び配布が行われている系統（通称、GAMESS(US)）
2. イギリスの CFS によって開発および配布が行われている GAMESS-UK
3. Moscow 大学の Alex A. Granovsky らによって開発された Firefly（旧：PC GAMESS）

本ドキュメントで記載する内容は 1.についてであり、これを以下単に GAMESS と記載します。

GAMESS は使用者、使用グループが公式ホームページに登録する事により、無償で入手できます。ライセンスの詳細については、GAMESS のホームページにあるライセンス規約（https://www.msg.chem.iastate.edu/games/License_Agreement.html）を参照ください。

本マニュアルでは GAMESS の実行方法をご案内します。より詳細な内容については以下の GAMESS の公式 HP をご確認ください。

<https://www.msg.chem.iastate.edu/games>

¹ 日本では、大阪府立大の小関らが開発に携わっている

2 GAMESS 実行方法

2.1 GAMESS の並列計算手法

GAMESS は、バイナリ作成時に、DDI(Distributed Data Interface) 版か MPI(Message Passing Interface)版という 2 つの並列手法の選択を選択します。

この選択は、計算原理そのものの選択ではなく、GAMESS の並列計算時における、複数の CPU 間（プロセス間）の通信手法の選択です。

前者の DDI 版は、昔から GAMESS が標準としてきた通信手法です。ソースコード中にある、通信用のライブラリを使用する手法で、代々の GAMESS において使用されてきた実績があります。複数 Node をまたいだ計算にも対応していますが、基本的には Gigabit Ethernet による接続を想定しており、最近 HPC 用計算機で採用が多い InfiniBand 等の高速インターコネクトでは十分な性能が得られません。

後者の MPI 版は、比較的最近 GAMESS が公式にサポートした手法です。名前の通り、並列計算において現在主流である、MPI 通信を使用したバイナリになります。公式サポートしてからの期間は長くはありませんが、高速インターコネクトの性能を引き出す事ができます。

本ドキュメントにおいては、前者の DDI 版の実行方法について以下解説を行っていきます。また、本ドキュメントにおいては、GPU を用いた計算については考慮していません。

2.2 rungms の編集方法

GAMESS は、rungms というスクリプトを実行環境に応じて編集し、実行することで動作します。rungms は GAMESS のインストールディレクトリ内に存在します。rungms は、GAMESS の Version によって内容が変化しますので、必ず使用する GAMESS のソースコード内にある rungms を使用してください。

なお、以下の記述は、GAMESS September 30, 2020 R2 Version のものとなるので、異なる Version の rungms では内容や行数が異なる可能性があります。

rungms で修正する必要がある箇所は大きく分けて以下の部分となります²。

- インストールディレクトリ、出力ディレクトリの指定

GAMESS のインストールディレクトリ、スクラッチファイルの出力ディレクトリを環境に合わせて修正する必要があります。修正箇所は rungms の 64 行目からの以下の部分です。

```
set SCR=/scr1/$USER
set USERSCR=~ /gamess-devv
set GMSPATH=~ /gamess-devv
```

設定している変数は以下の意味となります。

- SCR : 計算途中に作成されるスクラッチファイルが作成されるディレクトリ。このディレクトリに実行ユーザーがファイルを書き込めない場合、GAMESS は正常に実行されませんので、該当ディレクトリには実行ユーザーの書き込み権限が必要となります。
- USERSCR : 計算後に一部の出力ファイルが作成されるディレクトリ。SCR の指定と同じく、実行ユーザーの書き込み権限が必要となります。また、該当ディレクトリに、新たに作成される出力ファイルと同名のファイルが存在している場合、GAMESS は正常に実行されません。
- GMSPATH : GAMESS がインストールされているディレクトリ。HPCS が設定のお手伝いをしている場合、設定済となります。

上記の3つの変数を、実行環境に応じて編集します。

複数 Node で GAMESS を実行する場合、上記3変数で指定するディレクトリは、計算を実行するすべての Node から参照できる必要があります。

- 実行 Node の指定

GAMESS を実行する Node と、それぞれに割り当てる計算コア（並列数）の指定を行います。該当箇所は rungms の 358 行目からの以下の部分となります。

² rungms は csh のシェルスクリプトとなっているため、csh の文法に従って任意に修正可能

```

if ($NCPUS == 1) then
    set NNODES=1
    set HOSTLIST=(`hostname`)
endif
#
# 3. This is an example of how to run on a multi-core SMP enclosure,
# where all CPUs (aka COREs) are inside a -single- NODE.
# At other locations, you may wish to consider some of the examples
# that follow below, after commenting out this ISU specific part.
if ($NCPUS > 1) then
    switch (`hostname`)
        case se.msg.chem.iastate.edu:
        case sb.msg.chem.iastate.edu:

                (中略)

        default:
            echo " "
            echo Assuming a single but multicore node.
            echo " "
            set NNODES=1
            set HOSTLIST=(`hostname`:cpus=$NCPUS)
    endsw
endif

```

設定している変数は以下の意味となります。

NNODES : 計算に使用する Node 総数

HOSTLIST : 計算に使用する Node と、それぞれの Node に割り当てる並列コア数。

書式例 : NodeA で X 並列、NodeB で Y 並列、NodeC で Z 並列の場合

```
NNODES=3
```

```
HOSTLIST = ( NodeA:cpus=X NodeB:cpus=Y NodeC:cpus=Z )
```

基本的に、実行する Node、並列数に応じて、2つの変数を修正する必要がありますが、`rungms` を実行する Node のみで計算を実行する場合に限り、以下の内容にすることで並列数によらず共通の `rungms` で計算を実行する事が可能です。

```
NNODES=1
HOSTLIST = ( `hostname` :cpus=$NCPUS )
```

以上の内容を踏まえ、具体的な修正内容は以下となります。

(修正中、赤字で書かれている部分は修正内容およびその解説となります。)

```
# if ($NCPUS == 1) then                                ←先頭に#を追加
set NNODES=1                                           ←実行環境に応じて修正
set HOSTLIST=( `hostname` :cpus=$NPUCS)              ←実行環境に応じて修正
# endif                                               ←先頭に#を追加
#   以下、行の先頭に#が無い行について、行頭に#を追加
# 3. This is an example of how to run on a multi-core SMP enclosure,
# where all CPUs (aka COREs) are inside a -single- NODE.
# At other locations, you may wish to consider some of the examples
# that follow below, after commenting out this ISU specific part.
# if ($NCPUS > 1) then
#   switch ( `hostname` )
#     case se.msg.chem.iastate.edu:
#     case sb.msg.chem.iastate.edu:
#
#           (中略)
#
#     default:
#       echo " "
#       echo Assuming a single but multicore node.
#       echo " "
#       set NNODES=1
#       set HOSTLIST=( `hostname` :cpus=$NCPUS)
#     endsw
#   endif
```

2.3 GAMESS バイナリの実行方法

まず、実行ディレクトリに `rungms` とインプットファイルをコピーします。GAMESS が `/usr/local/games` にインストールされており、インプットファイルが `/usr/local/games/tests/standard/exam01.inp` の場合、以下のコマンドになります。

```
$ cp /usr/local/games/rungms .  
$ cp /usr/local/games/tests/standard/exam01.inp .
```

この状態で、前節で解説した、`rungms` の編集を行います。

編集が終了し、トータル N 並列で計算を実行する場合、

```
$ ./rungms exam01 00 N >& output.log
```

というコマンドで GAMESS が実行され、`output.log` という名前でアウトプットファイルが出力されます。(00 は数字のゼロ 2 つです。)

3 既知の問題点

2022年4月28日現在、GAMESS で明らかになっている問題点はありません。

付録A

A.1 HPC システムズ お問い合わせ先



弊社ホームページ http://www.hpc.co.jp/support_index.html

サポート案内やお問い合わせの多い内容など様々な情報を掲載しております。
是非ご活用ください。

HPC システムズ株式会社

〒108-0022 東京都港区海岸 3-9-15 LOOP-X 8 階

HPC 事業部



【営業】 03-5446-5531 【サポート】 03-5446-5532

お電話によるサポート受付は祝日、弊社指定休日を除く月曜日から金曜日の 9:30～17:30
とさせていただきます。



【FAX】 03-5446-5550



【電子メール】 hpcs_support@hpc.co.jp