

基于多组态电子波函数的量子化学计算软件

V1.0

用户手册

1. 简介

基于多组态电子波函数的量子化学计算软件是一款基于多组态（Multi-Configurational，简称 MC）波函数的从头算电子结构计算软件（该软件可简称为 eMC）。其可用于求解复杂电子相关体系的能量和波函数信息，在此基础上进而分析计算体系的分子轨道信息、电子跃迁性质、垂直激发能（荧光光谱信息）等。该软件的主要原理是实现非相对论近似下非定态薛定谔方程的变分求解，其核心的功能模块为多组态自洽场（MC Self-Consistent-Field，简称为 MCSCF）模块和多组态线性响应（MC linear-response，简称为 MCLR）模块，其程序框架如下：

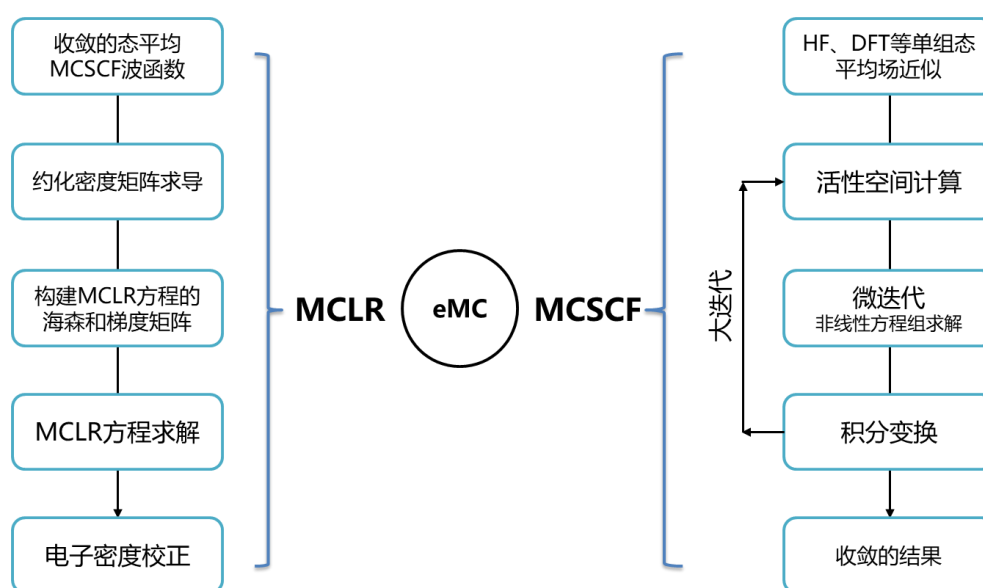


图 1 eMC 两大主体模块流程示意图

eMC 主要包括以下功能：

- ◆ 通用的自洽场迭代(即轨道优化)计算模块,可支持单一组态波函数如 Hartree-Fock 波函数，多组态波函数如组态相互作用波函数、矩阵乘积态波函数。
- ◆ 支持常见的活性空间内轨道与外层轨道之间的轨道旋转，并额外支持活性空间内部的分子轨道旋转。
- ◆ 内含针对阿贝尔点群的积分变换模块。
- ◆ 内含多种轨道优化方案，如简单差分方案、Newton-Raphson 方案、Augmented Hessian 方案、Werner-Meyer-Knowles 方案。
- ◆ 支持子空间迭代加速功能。
- ◆ 多组态线性响应模块，用于耦合微扰来处理电子激发态的密度校正，进而可以获得激发态的能量梯度。
- ◆ 多组态线性响应模块支持组态相互作用和矩阵乘积态波函数。

通过 eMC 软件系统，用户可以完成针对复杂电子相关体系的电子态自洽场计算和电子激发态密度校正，获得体系中电子的能量和波函数性质。

2. eMC 程序原理与特点

eMC 程序为多组态特色的量子化学计算程序。该程序基于如下的非含时薛定谔方程，其中 H 、 T 、 V 、 U 分别为体系哈密顿算符、动能算符、势能算符、电子-电子相互作用算符， i 、 N 为特定电子以及体系总电子数。

$$\hat{H}\Psi = [\hat{T} + \hat{V} + \hat{U}]\Psi = \left[\sum_i^N \left(-\frac{\hbar^2}{2m_i} \nabla_i^2 \right) + \sum_i^N V(\vec{r}_i) + \sum_{i<j}^N U(\vec{r}_i, \vec{r}_j) \right] \Psi = E\Psi$$

在此基础上，通过变分原理来实现对体系电子性质的模拟。

eMC 作为 post-Hartree-Fock 的方法，可以基于 Hartree-Fock 自洽场优化的体系进行多组态 (MC) 的量子化学计算；计算结果可以直接用于进一步的多参考 (MR) 计算。当前版本的 eMC 没有集成多参考模块，主要用于 MC 计算。当前 eMC 支持多种多组态波函数形式的求解，例如完全活性空间自洽场波函数 (CASSCF)，密度矩阵重整化群自洽场波函数 (DMRG-SCF)，多态密度泛函波函数等等。

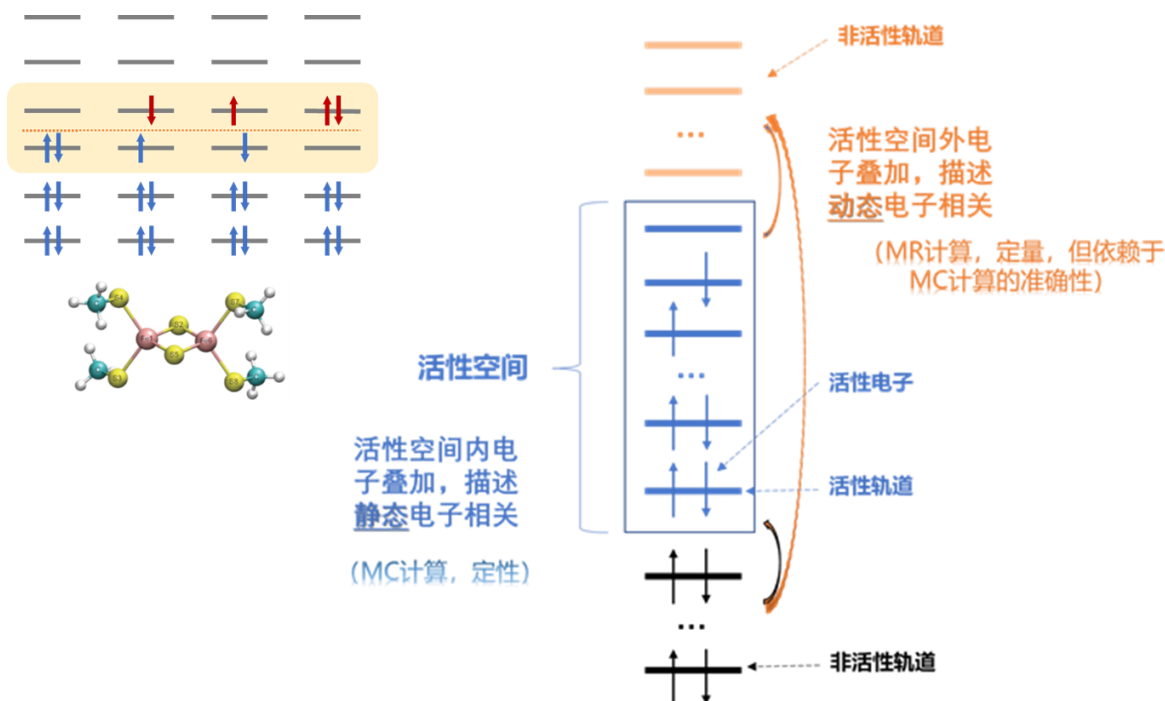


图 2 多组态(MC)/多参考(MR)的量子化学

3. 安装与系统配置

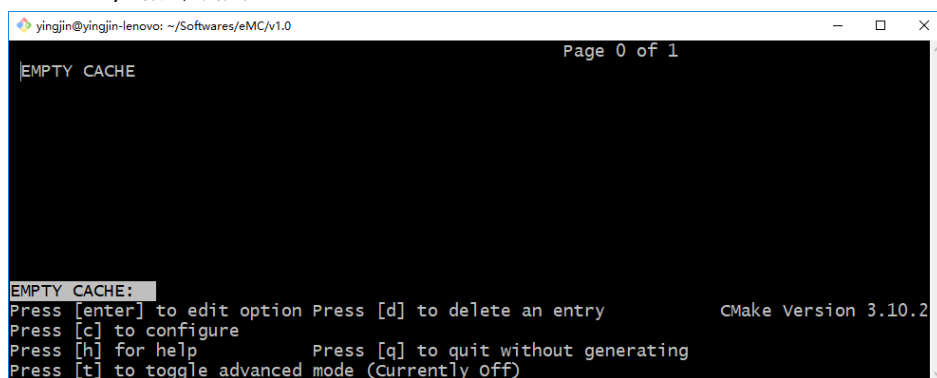
✚ eMC 编译安装需要的环境：

1. GNU gfortran、Intel ifort 等编译器
2. 常见数学函数库，如 Intel MKL、AMD ACML 等
3. Python 运行环境支持
4. CMake 编译系统

✚ eMC 的编译采用 CMake/Make 编译系统，使用外部编译的方式。

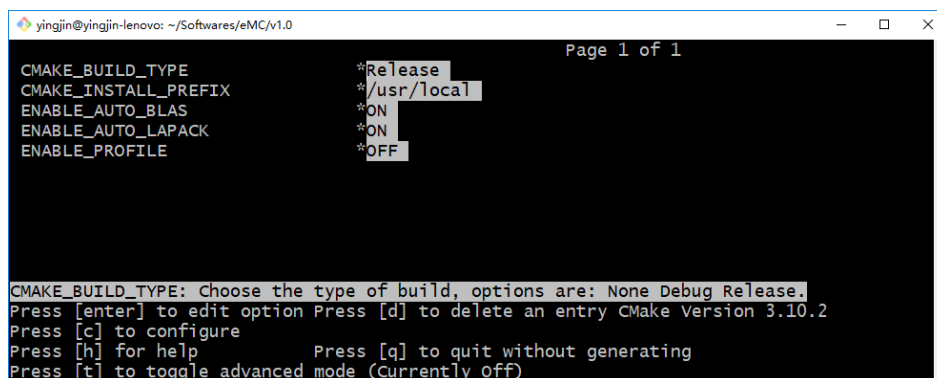
eMC 程序内置先进的环境变量自动配置系统，可以自动配置所需的编译器以及数学函数库（默认按照 MKL；ESSL；OPENBLAS；ATLAS；ACML；SYSTEM_NATIVE 的优先级顺序）。具体编译方式有：

- ✧ CMake-GUI，使用 CMake 图形界面的编译方式（推荐）
 - a. git clone 或者下载源码、解压缩的方式获得程序源代码
 - b. 程序目录下新建编译目录，如 v1.0
 - c. `ccmake ../` 指令调用出 CMake-GUI，



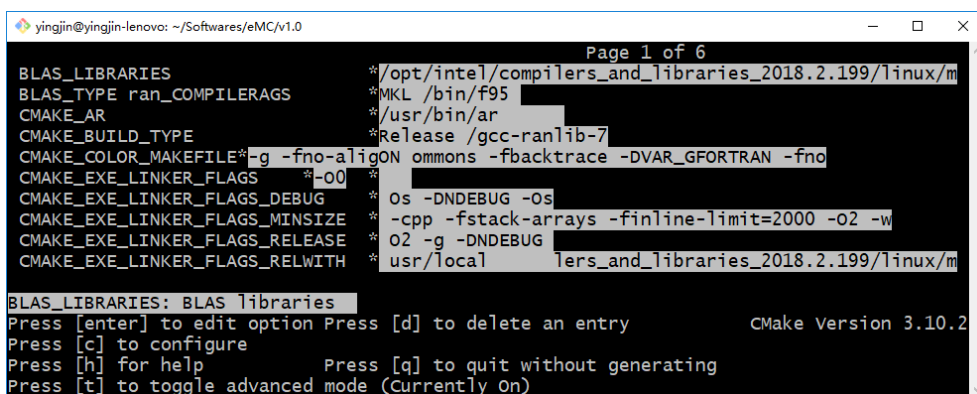
```
yingjin@yingjin-lenovo: ~/Softwares/eMC/v1.0
Page 0 of 1
EMPTY CACHE
EMPTY CACHE:
Press [enter] to edit option Press [d] to delete an entry          CMake Version 3.10.2
Press [c] to configure
Press [h] for help          Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently off)
```

- d. 按照提示键入‘c’，对 eMC 程序进行环境变量的全自动配置



```
yingjin@yingjin-lenovo: ~/Softwares/eMC/v1.0
Page 1 of 1
CMAKE_BUILD_TYPE          *Release
CMAKE_INSTALL_PREFIX      */usr/local
ENABLE_AUTO_BLAS          *ON
ENABLE_AUTO_LAPACK        *ON
ENABLE_PROFILE             *OFF
CMAKE_BUILD_TYPE: Choose the type of build, options are: None Debug Release.
Press [enter] to edit option Press [d] to delete an entry CMake Version 3.10.2
Press [c] to configure
Press [h] for help          Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently off)
```

- e. 键入‘t’，可以进入内置的高级编译选项配置界面



- f. 按需修改高阶的编译参数后，键入‘c’、‘g’可以生成编译所需的 Makefile 文件
 - g. 目录下 make -jN (N 为线程数目)，可开启多线程的程序编译
- ✧ 命令行编译
- a. git clone 或者下载源码、解压缩的方式获得程序源代码
 - b. 程序目录下新建编译目录，如 v1.0
 - c. 采用 FC=gfortran cmake ./ -DENABLE_THREADED_MKL=ON 进行环境变量的自动配置
 - d. 按需编辑修改目录下的 CMakeCache.txt，以此修改高阶的编译参数
 - e. 目录下 make -jN (N 为线程数目)，可开启多线程的程序编译

4. eMC 输入、输出文件结构

4.1 eMC 输入文件结构

当前版本典型的 eMC 输入文件包含有以下的选项卡，分别为\$ORBINT、\$ORBSPC、\$MC_INP、\$MC_SCF、\$MC_LAG，其主要功能描述如下：

选项卡名称	说明	备注
\$ORBINT	积分选项卡，主要功能有： 1. 读入分子轨道下的积分文件 2. 指定积分的点群对称性 3. 给定电子的总数目	详细功能请查看各个选项卡的说明
\$ORBSPC	轨道选项卡，主要功能为划分轨道子空间	
\$MC_INP	计算选项卡，亦可写为 MC_INPUT，主要功能为： 1. 选定活性空间计算所使用的计算程序 2. 指定活性空间计算后的输出文件 3. 指定计算的电子态数目和权重 4. 指定可能的并行调用方式	
\$MC_SCF	自洽场选项卡，主要功能为： 1. 指定采用的自洽场优化方案	

	2. 指定最大迭代步数 3. 指定收敛的判据和阈值	
\$MC_LAG	多态拉格朗日选项卡，亦可写为 MC_LAGRANGE，主要功能为态平均空间中指定电子态波函数的的驰豫，用于后续的梯度求解。	

其中，\$ORBINT、\$ORBSPC、\$MC_INP、\$MC_SCF、为必选选项卡， \$MC_LAG 为可选选项卡。基本程序的输入框架示例如下（以下[]:必填 {}:选填）:

\$ORBINT

```
sym      [Abelian Group/阿贝尔点群]
orbitals [Integrals/积分文件]
electron [N-electrons/体系总电子数]
```

\$END

\$ORBSPC

```
frozen [ frozen orbitals in each irreps/每个不可约表示下的冻结轨道数]
closed [ closed orbitals in each irreps/每个不可约表示下的闭壳层轨道]
occ    [occupied orbitals in each irreps/每个不可约表示下的占据轨道数]
```

\$END

\$MC_INP

```
binary [computing binary/可执行程序]
inputs [input file/输入文件]
reorder [orbital-reordering/轨道重排文件]
output [output file/输出文件]
mpirun [yes | no]
nproc  [nproc/计算核心数]
```

\$END

注: binary 需要外接的基于多组态方法的计算程序，例如 QCMAquis 程序

\$MC-SCF

```
method          {algorithm + iterations/计算方法+迭代步数}
maxcycle        {max iteration for macro-iterations/大：迭代数目上限}
CP_integrals    {transform for operations/采用算符近似以加速收敛}
MPSci_update    {interval for updating operations/更新近似算符间隔}
MAX_iter        { max iteration for micro-iterations/小迭代数目上限}
THRS_energy     {threshold for energy/能量阈值}      默认 1.0d-8
THRS_rotation   {threshold for rotation/旋转阈值}    默认 1.0d-8
THRS_coupled    {threshold for coupled/耦合阈值}     默认 1.0d-5
THRS_gradients  {threshold for gradients/梯度阈值}   默认 1.0d-3
```

act_act_rotation {rotations in active space/活性轨道内部旋转} 默认 无
\$END

注: method 当前版本建议选取为 *augmented Hessian* 方法 (关键字 *augment*)

\$MC-LAG

DMRG-SCF {additional SCF calculation/额外的 SCF 计算} 默认 .false.

DMRG-LR {Energy Lagrangian/能量拉格朗日} 默认 .true.

rlxstate {Relax state/弛豫的电子态} 默认 1

ckp_re-use {checkpoint re-use/中间文件复用} 默认 .true.

irreps {consider irreps/考虑不可约表示} 默认 .true.

\$END

4.2 eMC 输出文件结构

典型的输出文件格式如下

```
-----
[程序说明]
[版本、引用、注意事项说明]
[计算部分]
  1. 闭壳层 Hartree-Fock 结果输出
  2. 态平均信息输出
  3. 轨道信息输出
  4. 多组态自洽场迭代信息输出
  5. 多组态线性响应迭代信息输出
[结果汇总]
-----
```

输出文件示例 (当前版本):

```
-----
[程序说明]
This program is a prototype code for
second-order CASSCF, MCSCF and DMRG-SCF
and
their state-averaged gradients (Lagrange/MCLR part)
[程序说明]
```

[版本、引用、注意事项说明]

Main author:

Y. Ma (CNIC/COSCAR, ETH Zurich and U. Nanjing)

Contributing author:

S. Knecht (ETH Zurich)

ORBOPT version: ORBOPT - version: 0.2
git SHA: 07bbcaabddb7

Nanjing -> Zurich -> Beijing: 2013 – 2018

Using this code in publications necessitates to cite:

Y. Ma, S. Knecht, S. Keller and M. Reiher, J. Chem. Theory Comput., 13, 2522-2549 (2017)

Calculation is running on --> yingjin-lenovo

Content of the input file

Attention :

Current input for TDMs is order sensitive

linux system is ubuntu

[版本、引用、注意事项说明]

[计算部分]

RHF STATE PART:

One-electron energy : -739.80377707200000

Two-electron energy : 276.63874959139639

Core & Nuclear energy : 165.96769587700001

[- RHF state energy] : -297.19733160360363

State-averaged DMRG-SCF with weights:

0.500 0.500

MCSCF Part (orbital optimization):

frozen orbitals: 0

closed orbitals: 17

occupied orbitals: 21

active orbitals: 4

external orbitals: 6

Iter | -- RDM_ENERGY -- | sum|R| | Method


```

1      XXXXXXXX      XXX      augment
      .....
N      XXXXXXXX      XXX      augment
===== MCSCF optimization finished =====
===== time required:          0.16

-----
Entering the linear-response solver
-----

The 1-th state will be relaxed
allocated mps info.
Finish read-in all properties for every states
Print the lowest 5 eigenvalues of full-SA Hessian :
-0.00000000  -0.00000000   0.00000000   0.00000000   0.00613110
Entered the PCG solver
Finished the orbital-only PCG solver
Start the Solver_CG_SA
==> Entering the Solver_CG_SA solver <==
    Notice the CG_SA threshold is r^2=1.0e-14
==> Finished the Solver_CG_SA solver <==

```

[计算部分]**[结果汇总]**

```

== The SCF results were saved in Step_XXX folders ==
== The orb-lag   was written into orb_lag.txt   ==
== The mps-lag  was written into RDM_lag.txt   ==
----- Normal Termination -----

```

[结果汇总]

5. eMC 计算示例

5.1 CASSCF 方法简单计算示例

这里我们使用最为简单的 H₂ 分子（键长为 0.75Å），采用 6-31G 的基组，选取价层的电子和轨道作为活性空间（即 2e2o）。此时 eMC 的 CASSCF 计算输入文件如下所示：

```

$ORBINT
  sym c1
  orbitals FCIDUMP
  electron 2
$END

```

```

$ORBSPC
  frozen, 0
  closed, 0
  occ, 2
$END
$MC_INP
  binary      maquis
  inputs      dmrg-input1
  reorder     MASORB.orb
  output      dmrg.out
  mpirun      no
  nproc       1
$END
$MC_SCF
  method      augment 20 (AH)
  THRS_energy 1.0d-8
$END

```

其中 FCIDUMP 为体系的积分文件，eMC 支持文本或者二进制形式的积分文件。示例的 H₂ 体系单、双电子积分如下所示：

```

&FCI NORB= 4
  ORBSYM= 1, 1, 1, 1,
  ISYM=1
&END
  0.651702246201      1  1  1  1
-0.327093570732E-09  2  1  1  1
  0.177321245966     2  1  2  1
  0.651203577103     2  2  1  1
-0.747701960494E-09  2  2  2  1
  0.695241739730     2  2  2  2
-0.166719704673     3  1  1  1
-0.752580255525E-09  3  1  2  1
-0.171882767861     3  1  2  2
  0.108112503059     3  1  3  1
-0.199925536322E-08  3  2  1  1
-0.809274370242E-01  3  2  2  1
-0.190494415682E-08  3  2  2  2
  0.195827936061E-08  3  2  3  1
  0.702371501376E-01  3  2  3  2
  0.530164263062     3  3  1  1
  0.271050219312E-08  3  3  2  1

```

0.532425279787	3	3	2	2
-0.117667025476	3	3	3	1
-0.270506729929E-08	3	3	3	2
0.460258300775	3	3	3	3
0.187555328653E-08	4	1	1	1
-0.521661713626E-01	4	1	2	1
0.218157624569E-08	4	1	2	2
-0.626727715742E-09	4	1	3	1
0.530682734490E-01	4	1	3	2
0.565171913199E-09	4	1	3	3
0.420361269737E-01	4	1	4	1
-0.152645934871	4	2	1	1
0.137577022027E-08	4	2	2	1
-0.165739623031	4	2	2	2
0.935975123594E-01	4	2	3	1
0.270067506453E-09	4	2	3	2
-0.109687829322	4	2	3	3
-0.171656503209E-08	4	2	4	1
0.929253989916E-01	4	2	4	2
-0.607166959877E-09	4	3	1	1
0.114843938575	4	3	2	1
-0.593146693776E-09	4	3	2	2
-0.271758947736E-09	4	3	3	1
-0.442636269815E-01	4	3	3	2
0.182717893800E-08	4	3	3	3
-0.235405156796E-01	4	3	4	1
0.967353474723E-09	4	3	4	2
0.930029437519E-01	4	3	4	3
0.446110214232	4	4	1	1
-0.278106841595E-08	4	4	2	1
0.467540161142	4	4	2	2
-0.780718395153E-01	4	4	3	1
0.746419328085E-10	4	4	3	2
0.397235163859	4	4	3	3
0.141935673604E-08	4	4	4	1
-0.692925355692E-01	4	4	4	2
-0.218930528077E-08	4	4	4	3
0.377785255732	4	4	4	4
-1.24183752116	1	1	0	0
0.410125765158E-09	2	1	0	0
-0.484739594039	2	2	0	0
0.157623888111	3	1	0	0

-0.273108972377E-08	3	2	0	0
-0.170872011139	3	3	0	0
-0.183128466379E-08	4	1	0	0
-0.298394027779	4	2	0	0
0.421638988712E-08	4	3	0	0
0.142829069165	4	4	0	0
0.705569614227	0	0	0	0

该示例程序的计算输出中可以得到以下信息：

linux system is redhat

RHF STATE PART:

One-electron energy : -2.4823477403199998
Two-electron energy : 0.65030907960100004
Core & Nuclear energy : 0.70556961422700004
[- RHF state energy] : -1.1264690464919997

MC-SCF Part (orbital optimization):

frozen orbitals: 0
closed orbitals: 0
occupied orbitals: 2

active orbitals: 2

external orbitals: 2

Iter	-- RDM_ENERGY --	sum R	Method
1	-1.14629359	0.0275	augment
2	-1.14633259	0.0033	augment
3	-1.14633314	0.0005	augment
4	-1.14633315	0.0001	augment
5	-1.14633315	0.0000	augment

Macro iterations converged!

Orbital optimization has successfully finished

Final ENERGY : -1.146333153

可以得知，对于 H_2 体系，eMC 仅使用 4 次优化即可以收敛，CASSCF 优化的体系电子能量为 -1.146333153 Hartree。此结果与各种商业的量子化学软件（如 Gaussian、Molpro 等）完全一致。

5.2 DMRG-CASSCF/MCSCF 方法计算示例

该示例计算选取过渡金属 Cr_2 分子(键长 2.8\AA)，使用 Douglas-Kroll-Hess (DKH2) 哈密顿和 ANO-RCC-VTZP 基组。活性空间由 Cr_2 分子的 3d、4s、4p、4d 成份的分子轨道构成，即 12 电子 28 轨道的活性空间，同时内层面子采用冻结轨道近似来处理。具体轨道选取的方式是按照轨道的不可约表示 ($a_g, b_{3u}, b_{2u}, b_{1g}, b_{1u}, b_{2g}, b_{3g}, a_u$) 来选取，即选择 5,2,2,0,5,2,2,0 为内层轨道，内层轨道之上选择 6,3,3,2,6,3,3,2 为活性轨道。

为了节省计算资源，该示例程序将冻结轨道部分的电子积分以有效势的形式整合到活性和外层虚轨道对应的轨道空间内。此时的 FCIDUMP 积分文件已经包含了内层电子的作用，所以在 \$ORBSPC 选项卡里面仅需要填写 occ 轨道即可。

```
$ORBINT
  sym d2h
  orbitals FCIDUMP
  electron 6,2,2,2,0,0,0,0
$END

$ORBSPC
  frozen, 0,0,0,0,0,0,0,0
  closed, 0,0,0,0,0,0,0,0
  occ, 6,3,3,2,6,3,3,2
$END

$MC_INP
  binary   maquis
  input    dmrg-input
  reorder  MASORB orb
  output   dmrg.out
  mpirun   yes
  nproc    8
$END

$MC-SCF
  method      WMKUBAR  20
  maxcycle           20
  CP_integrals       0
  MAX_iter           100
```

```

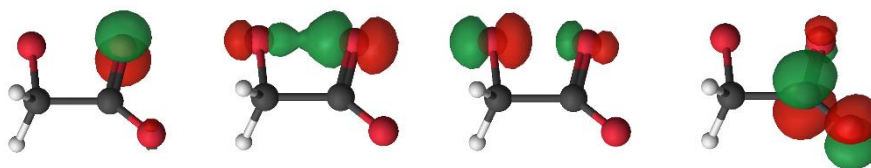
! ===== Used in macro-iter =====
THRS_energy      1.0d-8
THRS_rotation    1.0d-8
! ===== Used in micro-iter =====
THRS_coupled     1.0d-4
THRS_gradients   1.0d-3
!act_rotation
$END

```

此算例中，初始的体系能量为-2099.32179049 (hartree)，自洽场收敛后，体系能量为-2099.41178916。收敛大概需要 11 次迭代。如果\$MC-SCF 选项卡中将 CP_integrals 参数设置为 2 (启用近似的算符变化来加速收敛)，则 6 次迭代即可达到收敛。

5.3 DMRG-Lagrange 计算示例 (测试)

该示例计算中，使用荧光素的前驱分子，在其锥形交叉点附近对 $\sigma \rightarrow \sigma^*$ 电子态就行能量弛豫。能量弛豫在由 $\sigma \rightarrow \sigma^*$ 电子态和 $n \rightarrow \sigma^*$ 电子态构成的活性空间中进行，对应的活性空间由如下 4 个活性轨道所构成。



活性轨道 1

活性轨道 2

活性轨道 3

活性轨道 4

```

$ORBIT
  sym c1
  orbitals FCIDUMP
  electron 38
$END

$ORBSPC
  frozen, 0
  closed, 17
  occ, 21
$END

$MC_INP
  binary      maquis
  nstates     2 1

```

```
inputs      dmrq-input1 dmrq-input2
tdm_inputs  dmrq-input_TDM_1_2 dmrq-input_TDM_2_1
weight      1 1
reorder     MASORB.orb
output      dmrq.out
mpirun      yes
nproc       4
$END

$MC-LAG
  DMRG-SCF  .false.
  DMRG-LR   .true.
  rlxstate  1
  irreps    .true.
  ckp_re-use .true.
$END

$MC-SCF
  method  augment  20 (AH)
  maxcycle          20
$END
```

计算之后可在计算目录得到 `orb_lag.txt` 以及 `xRDM_lag.txt`，分别为轨道和组态部分的拉格朗日参数，可以用于后续的激发态分析、构型优化等计算。

6. 与 MOLCAS、MOLPRO 联用

跟其他基于 `post-Hartree-Fock` 的方法类似，当前 `eMC` 的计算需要使用分子轨道下的积分文件。这里我们建议可以使用 `MOLCAS` 或者 `MOLPRO` 来获取。内嵌积分的 `eMC` 版本也在开发之中，将于之后发布。

`MOLCAS` 提取积分的输入文件可参考以下（此积分亦用于 5.1 的计算示例）：

```
&GATEWAY
  coord=HH.xyz
  basis=6-31g
  group=NOSYM
```

```
&SEWARD
&SCF
```

```
&RASSCF
```

```
inactive = 0
  RAS2 = 4
  DMRG
  RGinput
    nsweeps = 20
    max_bond_dimension = 100
  endRG
DUMP
```

该输入文件会得到 H₂ 的积分文件，可用于 eMC 的后续计算。