2022

Low Scaling Quantum Chemistry (LSQC) Program Users' Manual

LSQC group
ITCC of Nanjing Univ., P. R. China.

2022-1-20

Introduction

Low Scaling Quantum Chemistry (LSQC) program (No. 2006SR09617) is a quantum chemistry package for linear or low scaling electronic structure calculations of large systems, which was developed by the research group of Professor Shuhua Li and Professor Wei Li in Nanjing University. The original version is LSQC-1.0 published on April 20, 2006, and the current version is LSQC-2.5 published on Jan. 20th, 2022. This version supports serial and Intel MPI parallel calculations, and can be run on most Linux systems. The latest information of LSQC program can be found via our website at http://itcc.nju.edu.cn/lsqc or the following QR code:

Content

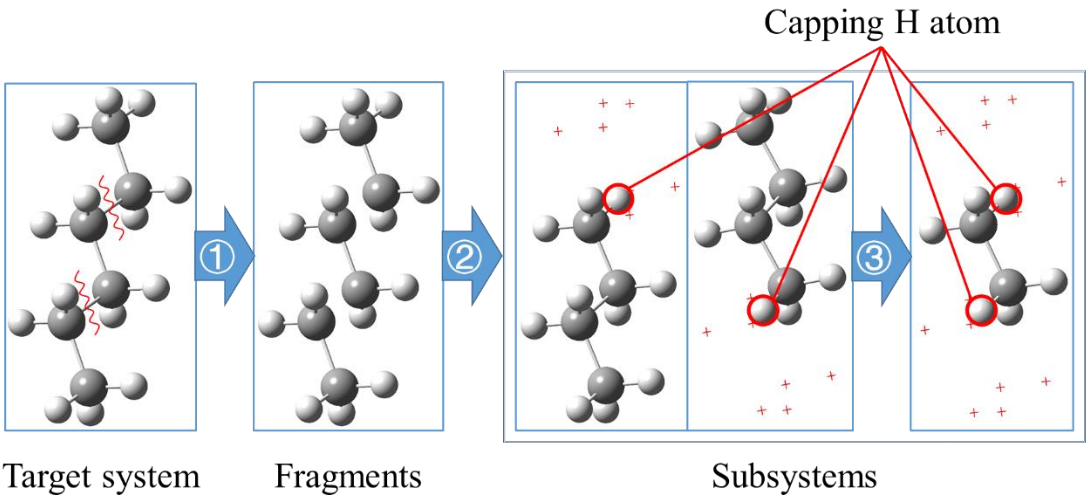
Chapter 1 Introduction to LSQC

1.1 What is LSQC?

Current LSQC program supports two methods for electronic structure calculations of large systems. The first one is the generalized energy-Based fragmentation (GEBF) method and the other one is the cluster-in-molecule (CIM) local correlation method.

1.1.1 GEBF module

The GEBF module in the LSQC package is an efficient implementation of the generalized energy-based fragmentation (GEBF) method.



The brief procedure of the GEBF method is as follows. 1) Divide the target system into several fragments. 2) Combine these fragments into primitive subsystems with capping hydrogen atoms embedded in the background point charges of distant atoms. 3) Generate derivative subsystems according to the inclusion-exclusion principle, which ensures all the atoms in the target system are considered and are not double counting. 4) Get the energies or energy derivatives of those subsystems with traditional quantum chemistry programs. 5) Combine the energies or energy derivatives of these subsystems to get those of the target system.

The final energy of the target system is

$$E_{\mathrm{tot}} = \sum_{m}^{M} C_m \tilde{E}_m - \left[\left(\sum_{m}^{M} C_m \right) - 1 \right] \sum_{A} \sum_{B>A} \frac{Q_A Q_B}{R_{AB}} \tag{1.1}$$

where $\tilde{E}_m$ stands for the energy of the *m*th subsystem (including the self-energy of background point charges), C_m represents the coefficient of the *m*th subsystem, Q_A is the net atomic charge on atom *A*, and *M* is the total number of subsystems.

Then, the fully analytic energy gradients of the target system are

$$\begin{aligned} \frac{\partial E_{\mathrm{tot}}}{\partial \boldsymbol{q}_A} = \sum_{m}^{M} C_m \left(\frac{\partial \tilde{E}_m}{\partial \boldsymbol{q}_A} - F_{m,a} Q_a - \sum_{b} f_{ab} \right) \\ + \left[\left(\sum_{m}^{M} C_m \right) - 1 \right] \sum_{b} f_{ab} \end{aligned} \tag{1.2}$$

where *A* denotes a real atom in a given subsystem, *a* and *b* denote the point-charge centers, $F_{m,a}$ is the electric field generated by the *m*th subsystem on the center *a* (which can be calculated with some existing *ab initio* programs), and f_{ab} represents the Coulomb force between charge on *b* and charge on *a*,

$$f_{ab} = \frac{Q_a Q_b}{|\boldsymbol{q}_a - \boldsymbol{q}_b|^3} (\boldsymbol{q}_a - \boldsymbol{q}_b) \tag{1.3}$$

To reduce the running time, the program uses the simplified equation

$$\frac{\partial E_{\mathrm{tot}}}{\partial \boldsymbol{q}_A} \approx \sum_{m}^{M} C_m \frac{\partial \tilde{E}_m}{\partial \boldsymbol{q}_A} \tag{1.4}$$

to calculate the gradients at atoms and use these gradients for geometry optimization and vibrational frequency calculations. So, the property Ω_{tot} of the system is

$$\Omega_{\mathrm{tot}} = \sum_{m}^{M} C_m \Omega_m \, , \Omega = \mu_i, \alpha_{ij}, \sigma_{ij} \tag{1.5}$$

where Ω_m is the corresponding property of the mth embedded subsystem.

For systems with hundreds and even thousands of atoms, the GEBF-X approach allows full quantum mechanical (QM) calculations at the X level to be accessible on ordinary workstations. This method has been successfully applied to study various large systems including molecular clusters, polypeptides, proteins, foldamers and so on.

The calculations of electrostatically embedded subsystems at various theoretical levels can be done with existing quantum chemistry programs. In this version, only Gaussian program is employed for subsystem calculations.

The single point calculation (SP) at semi-empirical method (AM1, PM3, PM6, etc), HF, DFT and electronic correlation method (MP2, MP3, MP4, CCSD, CCSD(T)) is available in current version. And the geometry optimization (Opt), frequency (Freq), IR intensity, Raman intensity, zero-point energy, enthalpy, Gibbs free energy, dipole moment, static polarizability, hyperpolarizability and NMR are also available in this version.

1.1.2 CIM module

The CIM module in the LSQC package is an efficient implementation of the cluster-in-molecule (CIM) local correlation method. The CIM local correlation method is powerful for post-Hartree–Fock calculations of large systems. In the CIM approach, the electronic correlation energy of a system is expressed as the sum of contributions from all the occupied orbitals:

$$E_{\mathrm{corr}} = \sum_{i}^{N_{\mathrm{occ}}} \Delta E_i \tag{1.6}$$

For example, at the MP2 and CCSD levels, the contribution of a given occupied orbital i is formulated as

$$\Delta E_i = \frac{1}{2} \sum_{j,a<b} V_{ij}^{ab} \tau_{ab}^{ij} \tag{1.7}$$

where $V_{ij}^{ab} = \langle ij||ab \rangle$ is the two-electron integral, $\tau_{ab}^{ij} = t_{ab}^{ij}$ is for MP2, and $\tau_{ab}^{ij} = t_{ab}^{ij} + t_a^i t_b^j - t_a^j t_b^i$ is for CCSD ($t_a^i$and t_{ab}^{ij} are the amplitudes of singly and doubly excited determinants), respectively. Eqs. 1.6 and 1.7 are valid for both canonical molecular orbital (CMO) and localized molecular orbital (LMO) representations.

We denote occupied and virtual CMOs in the whole system or quasi-CMOs (QCMOs) in the clusters as $i, j, k, \ldots$ and $a, b, c, \ldots$, and meanwhile, the indices $i', j', k', \ldots$ and $a', b', c', \ldots$, refer to occupied and virtual LMOs, respectively. In the CIM framework, for each occupied LMO i' (treated as *central LMO*), a union of this occupied orbital and its spatially neighboring occupied LMOs form the occupied orbital space of a certain cluster P. A subset of virtual orbitals, which are distributed over the similar region as the occupied orbitals, can be constructed as the virtual orbital space of this cluster. The correlation energy contribution of a given occupied LMO can be obtained with the amplitudes from solving MP2 or CC equations within this cluster:

$$\Delta E_{i' \in \{P\}} = \frac{1}{2} \sum_{j' \in \{P\}} \sum_{a'<b' \in \{P\}} V_{i'j'}^{a'b'} \tau_{a'b'}^{i'j'} \tag{1.8}$$

Once all clusters have been constructed, we will transform the LMOs within each cluster into QCMOs, because the use of QCMOs will allow

MP2 or perturbative triple equations to be solved in a non-iterative way. In addition, the use of QCMOs can also accelerate the convergence of solving the CCSD equations. QCMOs can be obtained from LMOs by a unitary transformation **R**, which can be determined by diagonalizing the occupied and virtual blocks of the Fock matrix, separately,

$$\mathbf{F}\mathbf{R} = \mathbf{R}\boldsymbol{\varepsilon} \tag{1.9}$$

$$|i\rangle = \sum_{i' \in \{P\}} R_{i'i} |i'\rangle \tag{1.10}$$

where $\boldsymbol{\varepsilon}$ is a diagonal matrix with the diagonal elements as the corresponding QCMO energies. With the QCMO representation, $\Delta E_{i' \in \{P\}}$ can be rewritten as

$$\Delta E_{i' \in \{P\}} = \frac{1}{2} \sum_{j \in \{P\}} \sum_{a<b \in \{P\}} \bar{V}_{i'j}^{ab} \bar{\tau}_{ab}^{i'j} \tag{1.11}$$

with

$$\bar{V}_{i'j}^{ab} = \sum_{i \in \{P\}} R_{ii'}^{+} V_{ij}^{ab} \tag{1.12}$$

$$\bar{\tau}_{ab}^{i'j} = \sum_{i \in \{P\}} R_{ii'}^{+} \tau_{ab}^{ij} \tag{1.13}$$

where $\mathbf{R}^{+}$ is the transpose of **R**.

The CIM-MP2 and CIM-RI-MP2 calculations are available in the current version. In addition, conventional MP2 and RI-MP2 calculations for medium-sized systems (with/without background charges) are also supported.

1.1.3 PBC-GEBF module

The PBC-GEBF module in the LSQC package is an efficient implementation of the generalized energy-based fragmentation approach for condensed phase systems under periodic boundary condition. In PBC-GEBF, a periodic system is decomposed into several electrostatically embedded subsystems and the construction of subsystems is similar to that for large molecules. All the subsystems are embedded in truncated background charges of finite clusters (*K*) and the dipole moment generated by these background charges are neutralized using the compensation filed scheme. The PBC-GEBF energy per unit cell is

$$E_{\text{tot}} = \sum_{m}^{M} C_m \left(\tilde{E}_m - \sum_{A \in K} \sum_{(B>A) \in K} \frac{Q_A Q_B}{R_{AB}} \right) + E_{\text{Ewald}} \qquad (1.14)$$

where E_{Ewald} is the classical charge-charge interaction per unit cell calculated by Ewald summation.

In the present PBC-GEBF module, the ground-state energy per unit cell for molecular crystals or liquids of small molecules can be readily available at various theoretical levels. For these systems, a molecule is automatically taken as a fragment (by default). Specially, one can obtain PBC-GEBF-X (X=MP2, CCSD, DFT, HF, …) energies for periodic systems mentioned above.

1.2 Skills before using this program

This program is suitable for researchers familiar with theoretical and computational chemistry.

Before using this program, you should be familiar with

- the commands of Linux system;
- the calculations of single point energy, geometry optimization, vibrational frequencies and so on using Gaussian program.

1.3 How this manual be organized?

The manual will help you to understand and use this program freely. The main structure of the manual is.

- Chapter 1 (this chapter), introduction of this program and the manual.
- Chapter 2, installation of this program.
- Chapter 3, preparation and submission of tasks.
- Chapter 4, introduction of GEBF module.
- Chapter 5, introduction of CIM module.
- Chapter 6, introduction of PBC-GEBF module.
- References and contact information.

1.4 Typographic Conventions

Text without special identification in Times New Roman typeface is to state or explain.

<*.xxx> represents the file with suffix ".xxx". The angle brackets are not a part of the file name.

The shaded text in Sitka Text typeface are Linux commands that should be inputted through keyboard. Please type "Enter" after the end of each line.

Virtual box text in Calibri typeface is file's contents, which could be either the input file that you should prepare or the output file that you can check.

Solid box text is screenshot of the Linux system.

Italic thickening text can be deleted according to the personal preferences or the practical condition.

Underlined text can be modified according to the personal preferences or the practical conditions. However, it should NOT be deleted unless ***it is also in bold italic form***.

Text with emphasis call attention to itself.

Text in blue background is to explain file or screenshot. For example, this "screenshot" was forged in a text box, next will be real.

Chapter 2 Installation

In order to ensure the successful installation, **please read the conventions in section 1.4 carefully**, and check whether the following programs are available.

- Python3 and NumPy. (Both of them are included in Anaconda package)
- Gaussian (A quantum chemistry program).
- If you would like to run in parallel on several nodes, please install intel MPI. (version 2019 recommended)
- If you need to calculate on High Performance Computing Center (HPCC) of Nanjing University, please set the routine of Gaussian as a global variable, whose name can only be “g09root” or “g16root”.

With these programs, you can start the installation according to the following steps.

1. Set or create a directory /home/Tom as installation path.

```
cd /home/Tom/
```

2. Get the installation package of LSQC Program (see http://itcc.nju.edu.cn/lsqc), transfer it to your server and decompress it.

```
tar xzf lsqc.2.5.tar.gz
```

List files after decompressing and you will see

```
lsqc-2.5    lsqc-2.5.tar.gz
```

3. Set the following environmental variables:

```
# LSQC main program
export lsroot=/home/Tom/lsqc-2.5
export PATH=$lsroot/bin:$PATH
export LD_LIBRARY_PATH=$lsroot/lib:$LD_LIBRARY_PATH
```

```
# Gaussian for GEBF module and PBC-GEBF module
export GAUSS_SCRDIR=/scratch/Tom/
export LSQC_SCRDIR=$GAUSS_SCRDIR

# for CIM module
export CIM_BASDIR=$lsroot/basis
export KMP_STACKSIZE=100G
export TMPDIR=/scratch/Tom/

# for intel MPI
export impiroot=/home/Tom/intel64
export PATH=$impiroot/bin/:$PATH
export LD_LIBRARY_PATH=$impiroot/lib:$LD_LIBRARY_PATH
```

Scratching path for LSQC

4. Source the shell script file.

```
source ~/.bashrc
```

Now, all files for the GEBF module are ready.

For the CIM module, in the present version, the Hartree−Fock calculations are performed using the PySCF package. To install PySCF, first download the source codes of Libcint, xcfun, libxc and PySCF.

libcint-4.0.7: https://github.com/sunqm/libcint.git

xcfun-3.5: https://github.com/fishjojo/xcfun.git

libxc-4.3.4: https://gitlab.com/libxc/libxc/-/archive/4.3.4/libxc-4.3.4.tar.gz

pyscf-1.7.6: https://github.com/pyscf/pyscf/releases

These packages may have been updated. If any of the above links doesn’t work, please go to the corresponding Github page for the latest codes. CMake 3.5 or higher is required when this manual was compiling.

The first three should be installed in same directory pathto/cint_and_xc.

To install these packages more easily, we strongly recommend that you should our <Installation_Assistant.sh>, adjust them according to your running environment and then copy and paste them to run these command.

(1) Install Libcint

```
tar -zxf libcint-4.0.7.tar.gz
cd libcint-4.0.7
mkdir build && cd build

cmake -DWITH_F12=1 -DWITH_RANGE_COULOMB=1 \
-DWITH_COULOMB_ERF=1 \
-DMIN_EXPCUTOFF=20 -DKEEP_GOING=1 \
-DCMAKE_INSTALL_PREFIX:PATH=/home/$USER/cint_and_xc \
-DCMAKE_INSTALL_LIBDIR:PATH=lib ..

make && make install
```

Here should be adjusted!

(2) Install libxc

```
tar -zxf libxc-4.3.4.tar.gz
cd libxc-4.3.4/build

cmake -DCMAKE_BUILD_TYPE=RELEASE \
-DBUILD_SHARED_LIBS=1 \
-DCMAKE_INSTALL_PREFIX:PATH=/home/$USER/cint_and_xc \
-DCMAKE_INSTALL_LIBDIR:PATH=lib ..

make && make install
```

Here should add “lib”!

(3) Install xcfun

```
tar -zxf xcfun-cmake-3.5.tar.gz
cd xcfun-cmake-3.5
mkdir build && cd build

cmake -DCMAKE_BUILD_TYPE=RELEASE -DBUILD_SHARED_LIBS=1 \
-DXCFUN_MAX_ORDER=3 \
-DCMAKE_INSTALL_PREFIX:PATH=/home/$USER/cint_and_xc \
-DCMAKE_INSTALL_LIBDIR:PATH=lib ..

make && make install
```

When the above installations are completed, there will be four directories (bin, include, share, and lib) in the directory /home/$USER/cint_and_xc. Define the following environmental variables:

```
    # for LSQC
export LD_LIBRARY_PATH=/home/$USER/cint_and_xc/lib:$LD_LIBRARY_PATH
```

(4) Install PySCF

```
unzip pyscf-master.zip
```

For CIM calculation, one should modify the source code of PySCF.

```
cd pyscf-master/pyscf/scf
```

In the <hf.py> file, near line 222, add “mf.fock = fock” as

```
        mf.fock=fock
        logger.timer(mf, 'scf_cycle', *cput0)
```

```
cd ../../pyscf/lib/
mkdir build && cd build
```

```
cmake -DBUILD_LIBCINT=0 \
-DBUILD_LIBXC=0 -DBUILD_XCFUN=0 \
-DCMAKE_INSTALL_PREFIX:PATH=/home/$USER/cint_and_xc ..
make
```

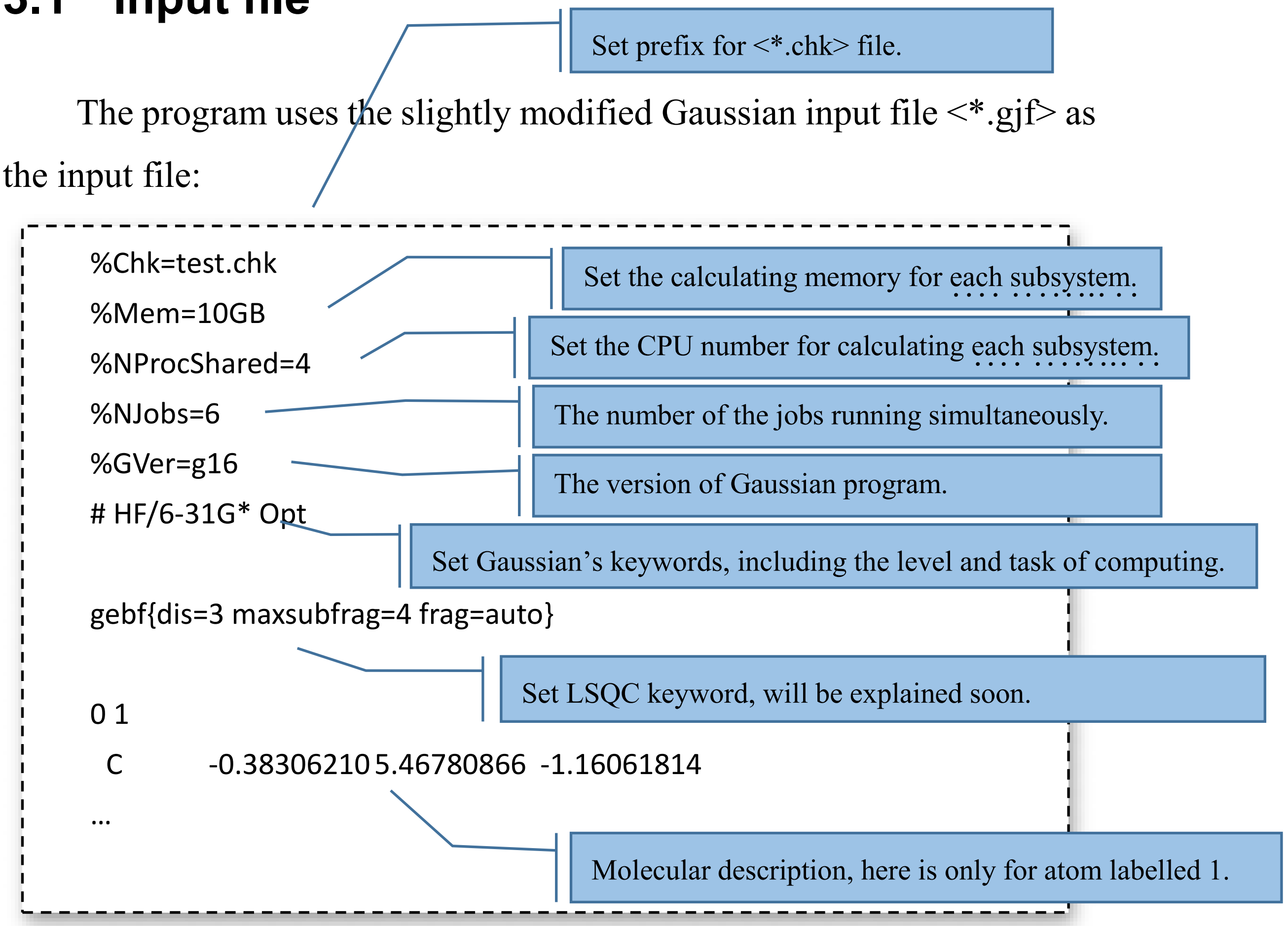
Define the following environmental variable:

```
# for LSQC
export PYTHONPATH=/home/Tom/software/pyscf-master:$PYTHONPATH
```

Chapter 3 How to Run a Job

3.1 Input file

The program uses the slightly modified Gaussian input file <*.gjf> as the input file:



```
%Chk=test.chk
%Mem=10GB
%NProcShared=4
%NJobs=6
%GVer=g16
# HF/6-31G* Opt

gebf{dis=3 maxsubfrag=4 frag=auto}

0 1
 C        -0.38306210 5.46780866  -1.16061814
...
```

Set prefix for <*.chk> file.

Set the calculating memory for each subsystem.

Set the CPU number for calculating each subsystem.

The number of the jobs running simultaneously.

The version of Gaussian program.

Set Gaussian’s keywords, including the level and task of computing.

Set LSQC keyword, will be explained soon.

Molecular description, here is only for atom labelled 1.

The parameters for Gaussian program in this file are for subsystems. The default is “%NProc=4” and “%Mem=1GB”.

“%NJobs” and “%GVer” are two keywords for LSQC, which are not accepted by Gaussian program. “%NJobs” is defaulted to be 4. “%GVer” can be “g16” (default) or “g09”. “%NJobs=1” is fixed in CIM module.

Running on several nodes are now supported on Load Sharing Facility (LSF) workload management platform. Set “%Parallel=True” and set the total number of CPU use “TotCPU=*n*” (*n*=48 for default) to activate. This default is run on one node, *i.e.*, “%Parallel=False”.

The title determines the keywords for LSQC.

3.2 Running Command

LSQC running command is

```
lsqc <input file>
```

For example:

```
lsqc water.gjf
```

Chapter 4 GEBF module

4.1 keywords

The title line is used to set the GEBF's parameters, whose format is:

```
gebf {keyword1=parameter1 keyword2=parameter2 and so on}
```

The meaning and the modify parameters of each keyword will be shown as follows.

◆ charge

Set the type of the background point charges. Following parameters are allowed.

- charge=none

No background charges (when you want to use semi-empirical method such as AM1, PM3… you should use the keyword).

- charge=NPA（Default）

use NPA charge.

- charge=ESP

use ESP charge.

- charge=MUL

use MUL charge.

- charge=read

The charges should be provided by user. You can use the <*.cha> file generated by this program or prepare it by yourself. Each line of <*.cha> declares the charge information for one atom, whose format is:

```
“atomic number” “elemental symbol” “charge”
```

For example:

```
      1   C     -0.065739
      2   C     -0.104309
```

- dis=r

Determine the size of the subsystems. The increase of subsystems will lead to the rising in both computational accuracy and computing resource consuming. The unit of *r* is Å and default value is 4.0 (Å).

- frag

Determine the method of fragmentation. The default value is frag=auto. More details about fragmentation will be shown in section 4.2.

- local

Declare only part of the subsystems will be calculated to reduce the calculation. Only supported in NMR calculations with "localnmr" keyword.

```
        N   -7.573   0.102   0.198    localnmr
```

- maxsubfrag=*n*

Help "dis" to determine the size of subsystems. This keyword sets the maximum number of fragments in each subsystem. The default value is 6.

- twofrag

Add additional two-fragment subsystems if the two are within 2**dis*. We recommend this keyword to be used for studying various molecular clusters.

◆ examples for keywords line

Example 1 Use the default parameters

```
gebf {}
```

It means:

```
gebf { charge=NPA dis=4.0 maxsubfrag=6 frag=auto }
```

Example 2 Set one parameter

```
gebf {dis=4.5}
```

It means:

```
gebf { charge=NPA dis=4.5 maxsubfrag=6 frag=auto }
```

Example 3 Set several parameters

```
gebf {dis=4.5 maxsubfrag=8 twofc frag=conn }
```

It means：

```
gebf { charge=NPA dis=4.5 maxsubfrag=8 frag=conn maxsubfrag=6 twofc }
```

4.2 Fragmentation for GEBF

4.2.1 Considerations

Reasonable fragmentation makes computing economic, while unreasonable fragmentation lead to unacceptable results. So we suggest:

- Two non-hydrogen atoms at least should be included in each fragment.
- For the atom in uncommon valence state, you should get fragmentation by user-defined.
- Do not destroy the conjugated systems. Such as benzene ring, peptide bond and so on.

4.2.2 Automatic fragmentation

- frag=auto

Automatic fragmentation. (Default) Each non-hydrogen atom (and its bonded hydrogens) or functional group (such as a double or triple bond) will be recognized. You can get more detail information from reference.

- frag=link

Fragmentation based on natural link determined only by the distance. This method does not cut chemical bonds and is suitable to calculate molecular clusters.

- frag=protein

Automatic fragmentation for protein molecules, DNA/RNA molecules and solvent molecules.

- frag=DNA

The same as frag=protein. This keyword will be discarded.

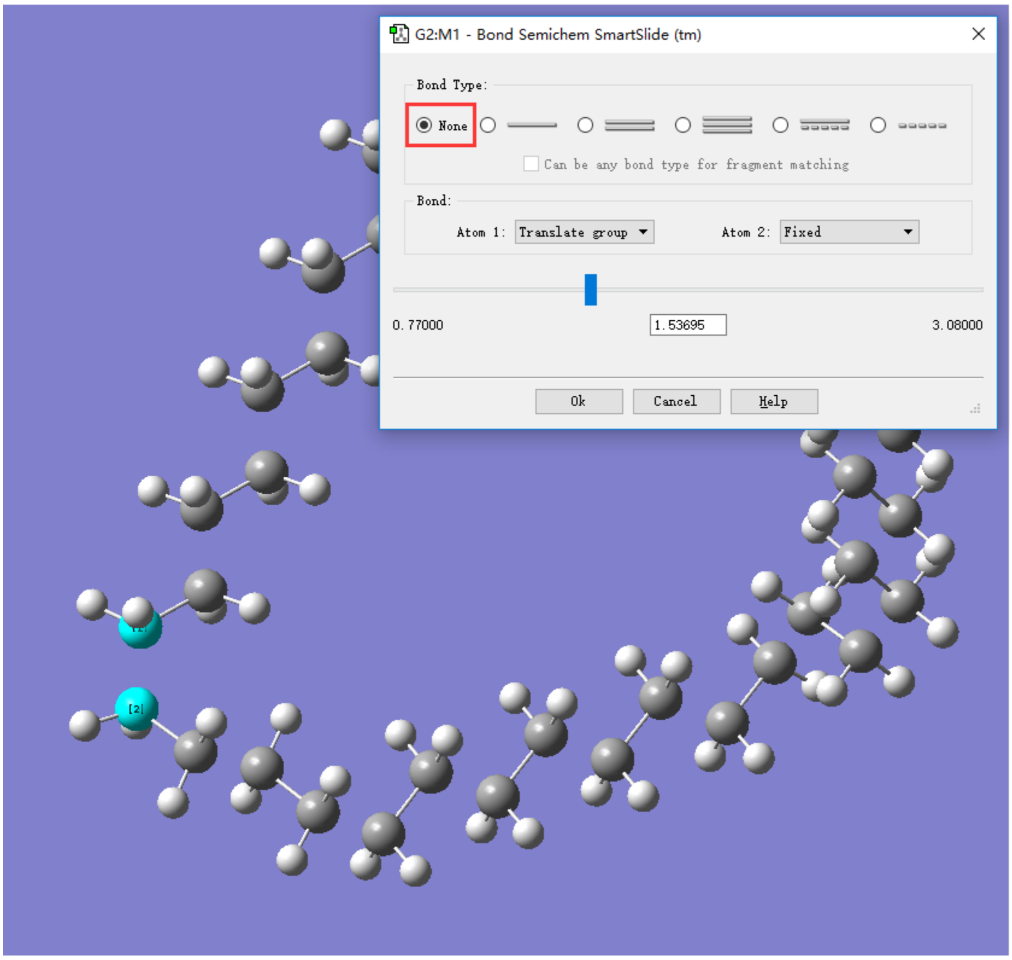
4.2.3 User-defined fragmentation

If you are unsatisfied with the fragmentation, you can fragment the system by yourself. Particularly, when the elements are not in common valence state or metal atoms exists in a system, you need to define fragmentation by yourself.

- frag=conn

Fragmentation based on Gaussian's keyword geom=connectivity. The <*.gjf> file must contain connectivity information, which can be obtained using GaussView.

For example, $C_{30}H_{62}$ is divided into 15 fragments based on connectivity.



G2:M1 - Bond Semichem SmartSlide (tm)

Bond Type: ● None ○ ○ ○ ○ ○

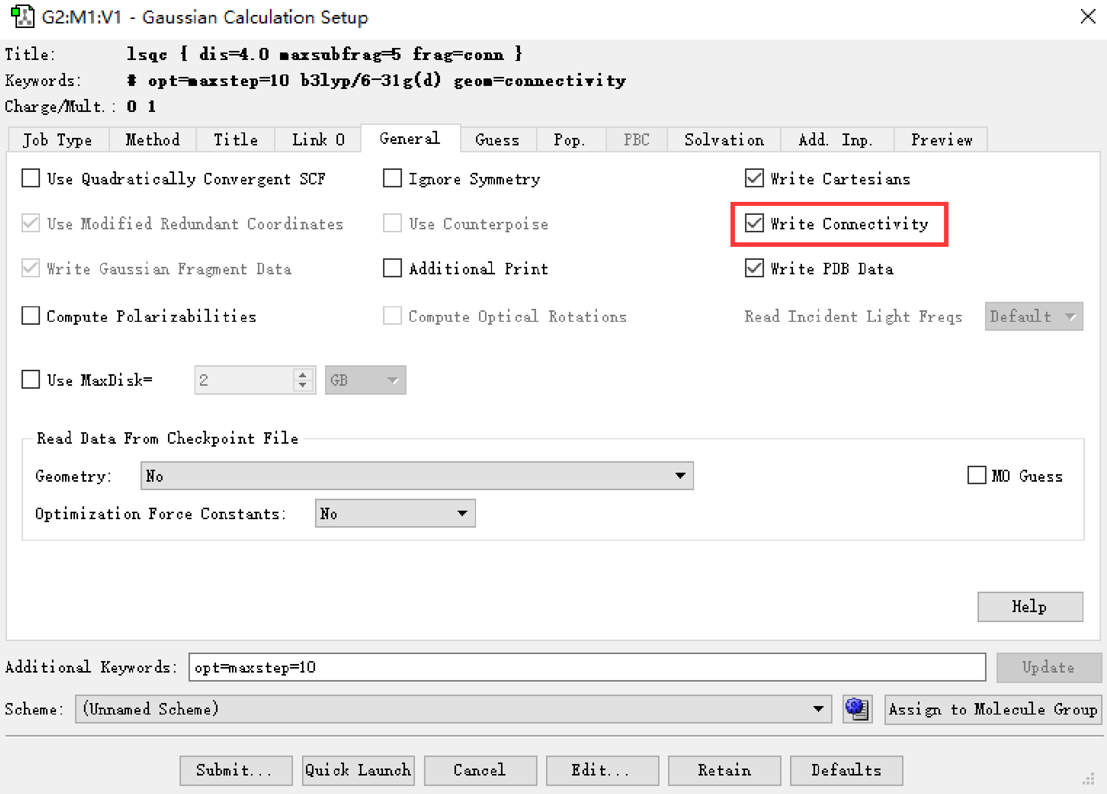
☐ Can be any bond type for fragment matching

Bond:

Atom 1: Translate group

Atom 2: Fixed

0.77000 1.53695 3.08000



G2:M1:V1 - Gaussian Calculation Setup

Title: lsqc { dis=4.0 maxsubfrag=5 frag=conn }
Keywords: # opt=maxstep=10 b3lyp/6-31g(d) geom=connectivity
Charge/Mult.: 0 1

☐ Use Quadratically Convergent SCF
☑ Use Modified Redundant Coordinates
☑ Write Gaussian Fragment Data
☐ Compute Polarizabilities
☐ Ignore Symmetry
☐ Use Counterpoise
☐ Additional Print
☐ Compute Optical Rotations
☑ Write Cartesians
☑ Write Connectivity
☑ Write PDB Data
Read Incident Light Freqs: Default
☐ Use MaxDisk= 2 GB

Read Data From Checkpoint File
Geometry: No
Optimization Force Constants: No
☐ MO Guess

Additional Keywords: opt=maxstep=10
Scheme: (Unnamed Scheme)

- frag=read

If you’d like to reuse the fragmentation scheme of previous work, used this option and copy the <*.frg> to your working path. The <*.frg> can also be prepared by yourself.

Each line in < *.frg> declares one fragment, whose format is:

serial number spin multiplicity (non-hydrogen atoms number) ***charge***

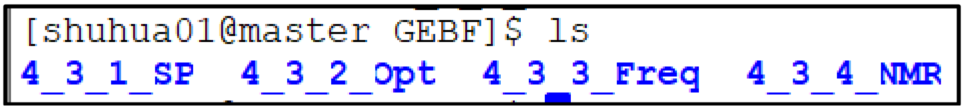
Use “,” to separate non-hydrogen atom, use ‘-’ to omit consecutive numbers; the default of charge value is 0. For example,

1 1 (1, 4) +1
2 1 (8-10) -1

“+” is OK.

4.3 Examples

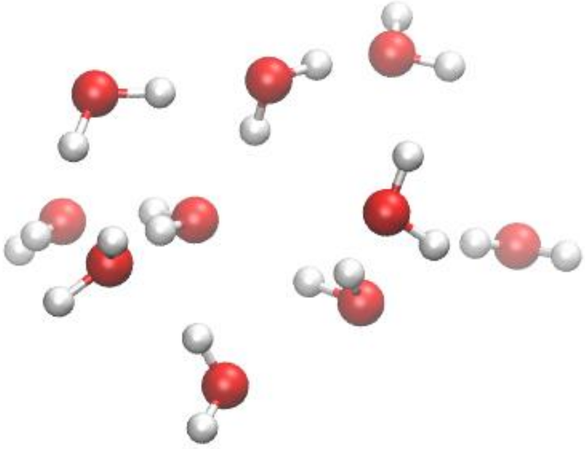
All the examples in this section will be found in directory $lsroot/example/GEBF.



```
[shuhua01@master GEBF]$ ls
4_3_1_SP   4_3_2_Opt   4_3_3_Freq   4_3_4_NMR
```

4.3.1 Single point energy

Example: calculate the single point energy of water cluster $(H_2O)_{10}$ at GEBF(3.0,4)-CCSD/cc-pVTZ level



Input file: <h2o_ccsd.gjf>

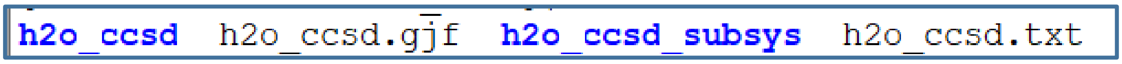
```
%chk=h2o_ccsd.chk
%nproc=4
%njobs=6
%Gver=g16
%mem=20gb
# ccsd/cc-pvtz

gebf { dis=3.0 maxsubfrag=4 frag=link }
```

```
0 1
O     0.083    -1.617     0.245
…
```

Submit this work. The LSQC program will create a working directory named “h2o_ccsd” for this work. And the main information is provided in <h2o_ccsd.txt>.

```
……
Read parameters from .keys file: h2o_ccsd.keys
Read geometry from .xyz file: h2o_ccsd.xyz
……
The largest subsystem is No.1 with 232 basis functions.
  GEBF(3.0,4)-CCSD/cc-pvtz energy:
  SCF =         -760.530426
……
  CCSD =         -763.258113    E(CORR) =             -2.727688
Dipole Moment (field-independent basis, Debye):
  X=    -0.8332    Y=    -2.7750    Z=    4.3232    Tot=     5.2043
……
Real time: 0:10: 20.615857
```

The details about the parameters are saved in directory “h2o_ccsd” and the details about the subsystems are saved in directory “h2o_ccsd_subsys”.

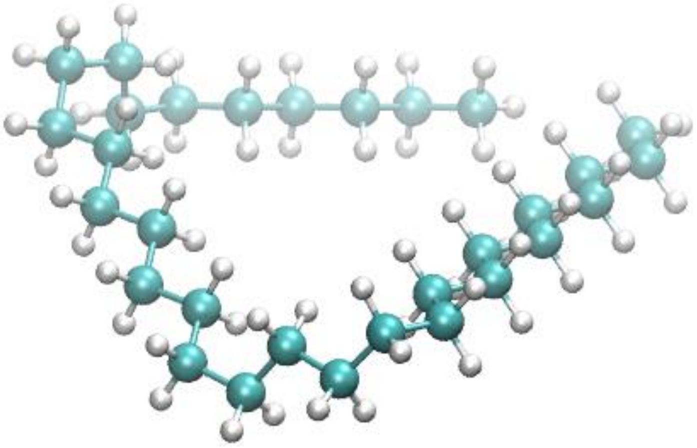


```
h2o_ccsd   h2o_ccsd.gjf   h2o_ccsd_subsys   h2o_ccsd.txt
```

4.3.2 Geometry optimization

In this program, the Gaussian program uses the forces from the GEBF approach to optimize the system, so you can adjust the keyword of Gaussian according to your experience.

Example: Optimize the *n*-$C_{30}H_{62}$ at GEBF(4.0,5)-HF/6-31G level



The input file is <c30h62_40-5.gjf>.

```
%mem=10GB
%nprocshared=2
%njobs=12
# hf/6-31g opt

gebf { dis=4.0 maxsubfrag=5 frag=conn }


0 1
  C            0.10719300      8.13647800      0.96036200
…
```

The main output of this task is <c30h62_40-5.log>.

```
 …
           Item                Value      Threshold   Converged?
 Maximum Force            0.003245      0.000450     NO
 RMS        Force              0.000381      0.000300     NO
 Maximum Displacement 2.865172      0.001800     NO
 RMS        Displacement 0.572324      0.001200     NO
 Predicted change in Energy=-7.676849D-04
…
           Item                Value      Threshold   Converged?
 Maximum Force            0.000015      0.000450        YES
 RMS        Force              0.000002      0.000300        YES
 Maximum Displacement 0.000642      0.001800        YES
 RMS        Displacement 0.000144      0.001200        YES
 Predicted change in Energy=-2.606358D-09
 Optimization completed on the basis of negligible forces.
    -- Stationary point found.
…
Normal termination of Gaussian 16 at Sat Oct 26 17:04:43 2019.
```

4.3.3 Frequency calculations

Example: Frequency (IR) calculation of the optimized *n*-$C_{30}H_{62}$ at GEBF- HF/6-31G level (The structure in the result of 4.3.2 is used)

The input file is <c30h62_40-5_opted.gjf>.

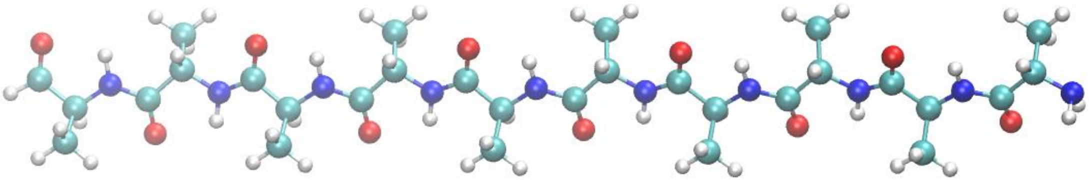
```
%mem=10GB
%nprocshared=6
# hf/6-31g freq

gebf { dis=4.0 maxsubfrag=5 frag=conn }

0 1
 C            -0.97866700       8.45496000      -0.58587800
...
```

The main output of this task is <c30h62_40-5_opted .log>.

```
...
                          1            2            3
                          A            A            A
 Frequencies --      2.7657       5.3066       8.8183
 Red. masses --      4.0254       4.1561       4.1877
 Frc consts    --    0.0000       0.0001       0.0002
 IR Inten       --   0.0000       0.0005       0.0005
...
Normal termination of Gaussian 16 at Sat Oct 26 17:17:24 2019.
```

4.3.4 NMR calculations

Example: NMR calculations of the optimized (Ala)$_{10}$ at GEBF(3.0,3)-B3LYP-D3(BJ)/6-31G* level. (This structure has been optimized at B3LYP-D3(BJ)/6-31G* level.)



The input file is prepared as <ala10_30-3.gjf>.

```
%chk=mol.chk
%nprocshared=6
%gver=g16
%mem=30GB
# b3lyp/6-31g* em=gd3bj nmr int=(ultrafine,acc2e=10) scf=(tight,xqc)

gebf { dis=3.0 maxsubfrag=3 frag=protein }

0 1
  N         -14.73925100       0.18468300       0.12108600
…
```

The magnetic shielding tensor will be reported in <ala10_30-3.txt>.

```
…
SCF GIAO Magnetic shielding tensor (ppm):
        1       N Isotropic = 149.684185
…
```

Chapter 5 CIM module

5.1 Keywords

- **disl**: distance threshold (in Å) for MO domain construction. The default value is 5.5 (Å) for CIM-MP2 and 6.0 (Å) for CIM-RI-MP2.
- **aubs**: set the auxiliary basis set for CIM-RI-MP2 calculation. Auxiliary basis set MUST be specified.
- **virt**: set the PAO projection threshold. The default value is 0.05. Users usually do not need to set this value. Only when the energy of HOMO is higher than LUMO in a certain cluster, one must set this threshold to a smaller value, like 0.03.

CIM-MP2 example:

```
%mem=10GB
%nprocshared=20
# mp2/cc-pvdz

cim{disl=6}

0 1
  C 0.0 0.0 0.0
  O 0.0 0.0 1.15
```

keywords for CIM module

CIM-RI-MP2 example:

```
%mem=10GB
%nprocshared=20
# rimp2/cc-pvdz
```

```
cim{disl=6 aubs=cc-pvdz-ri}

0 1
  C 0.0 0.0 0.0
  O 0.0 0.0 1.15
```

5.2 Conventional MP2 and RI-MP2 calculations

5.2.1 MP2 parameters

➢ **nofr**: no frozen core orbitals and all occupied orbitals are correlated. In default, orbitals with energies lower than −3.0 are set as frozen core orbitals.

➢ **thre**: integral threshold for two-electron integral. This is in the pH form. thre=11 will set the integral threshold as 10^{-11}. The default value is 10^{-12}.

example:

```
%mem=10GB
%nprocshared=20
# mp2/cc-pvdz

{nofr}

0 1
  C 0.0 0.0 0.0
  O 0.0 0.0 1.15
```

5.2.2 RI-MP2 parameters

➢ **aubs**: set auxiliary basis set for RI-MP2 calculation. There is no default value for this option, thus the auxiliary basis set **MUST** be specified.

- **nofr**: the same as that in MP2 module.

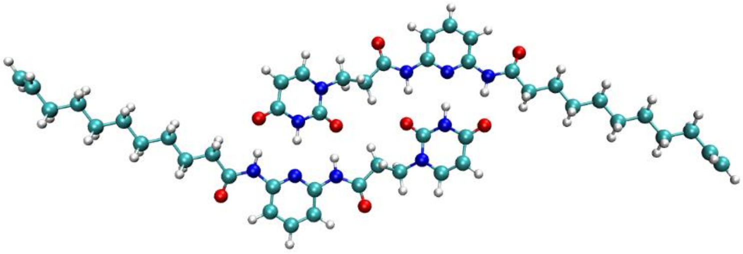
example:

```
%mem=10GB
%nprocshared=20
# rimp2/def2-tzvp

{aubs=def2-tzvp-ri}


0 1
  C 0.0 0.0 0.0
  O 0.0 0.0 1.15
```

5.3 Examples

Most of the examples in this section using smaller basis sets can be found in $lsroot/example/CIM.

5.3.1 CIM-RI-MP2 calculation



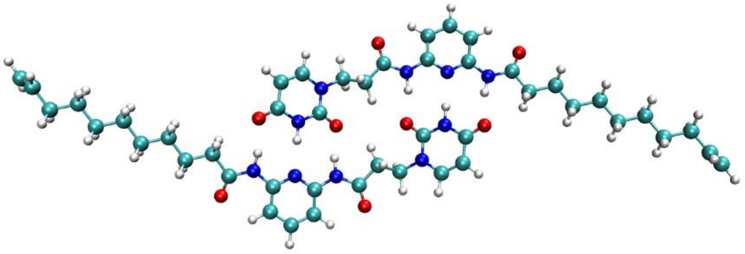
Input file <HPi.gjf>

```
%mem=25GB
%nprocshared=16
# rimp2/cc-pvdz

cim { aubs=cc-pvdz-ri virt=0.03 }

0 1
  C                    4.90011200     -2.45613900      0.26795800
…
```

Output file <HPi.out>

```
…
Number of atoms: 126
Nuclear repulsion energy:     8426.5072926896
…
Max(atm) =     63 in cluster       1
Max(bas) =    684 in cluster       1
Max(cen) =     15 in cluster      33
Max(occ) =     70 in cluster       1
Max(vir) =    402 in cluster       1
…
E(SCF)                 =    -2921.9089102855
E(CIM-5.5)             =        -9.2356726766
E(noncen-5.5)          =        -0.0053896019
E(corr-5.5)            =        -9.2410622785
E(Total)               =    -2931.1499725640
…
Job elapsed time:       6.83 min
Job completed at Tue Jan 18 19:37:41 2022
…
```

5.3.2 CIM-MP2 calculation



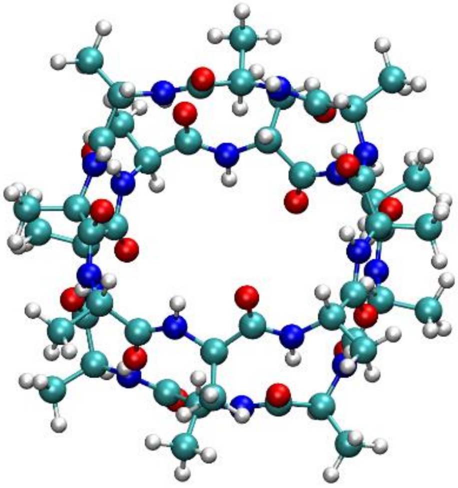
Input file <HPi.gjf>

```
%mem=25GB
%nprocshared=16
# mp2/cc-pvdz

cim{}
…
```

Output file <HPi.out>

```
…
Number of atoms: 126
…
E(SCF)    = -2921.9089102918
…
E(Corr) =      -9.2361548856
E(Total)= -2931.1450651774

Total elapsed time for correlation energy:     79.54 min
…
```

5.3.3 RI-MP2 calculation



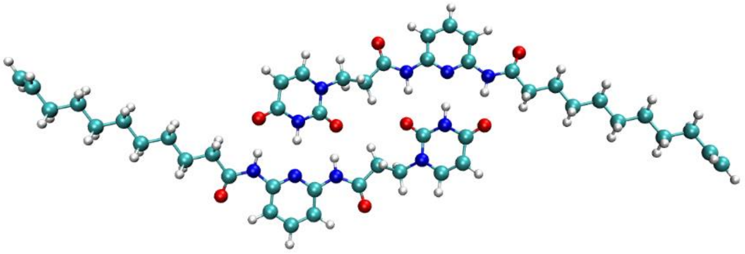
Input file <twopep.gjf>

```
%mem=160GB
%nprocshared=24
# rimp2/cc-pvdz

{aubs=cc-pvdz-ri}

0 1
  O                1.63457100      3.46400800      3.87096600
…
```

Output file <twopep.out>

```
…
Number of atoms: 160
Nuclear repulsion energy:    16119.3339631789

==== Basis Set Information ====
Basis set data read from existing env file.
Number of shells                  :      640
Number of basis functions:    1520

==== Canonical RI-MP2 Module ====
Auxiliary basis set: cc-pvdz-ri
Number of aux basis functions :    5600

Number of occupied orbitals       :      304
Number of correlated orbitals :      224
Number of virtual orbitals        :    1216

Elap time for 3C2E int and tran:        1.18 min
Elap time for MO int and energy:       22.60 min

Hartree-Fock energy             :    -3934.0190238022
RI-MP2 correlation energy:        -12.1537414101
Total RI-MP2 energy             :    -3946.1727652123

Elap time for RI-MP2:       25.13 min

Job summary:
============
Job ran on 24 threads
Job elapsed time:         25.13 min
Job completed at Sat Nov    2 19:27:57 2019
```

5.3.1 MP2 calculation



Input file <HPi.gjf>

```
%mem=50GB
%nprocshared=16
# mp2/cc-pvdz

{}


0 1
  C          4.90011200     -2.45613900       0.26795800
...
```

Output file <HPi.out>

```
…
Number of atoms: 126
Nuclear repulsion energy:     8426.5072926896

==== Basis Set Module ====
Basis set data read from existing env file.
Number of shells                  :      506
Number of basis functions:    1206

==== Canonical MP2 Module ====
Number of basis functions          :    1206
Number of occupied orbitals       :      236
Number of frozen core orbitals:        64
Number of correlated orbitals :      172
Number of virtual orbitals         :      970

Integral threshold:      1.0000E-10

Direct MP2 will be done in 4 batches.
…
Hartree-Fock energy      :    -2921.9089102918
MP2 correlation energy:          -9.2427410773
Total MP2 energy            :    -2931.1516513691

Elap time for MP2:    263.88 min

Job summary:
============
Job ran on 16 threads
Job elapsed time:      263.88 min
Job completed at Sat Nov    2 17:04:38 2019
```

Chapter 6. PBC-GEBF module

6.1 Keywords

- dis=*r*

Determine the size of the subsystems. The increase of subsystems will lead to the rising in both computational accuracy and computing resource consuming. The unit of *r* is Å and default value is 4.0 (Å).

- maxsubfrag=*n*

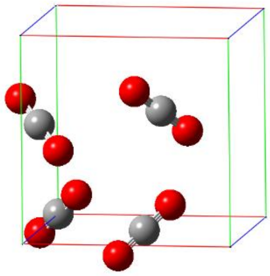
Help “dis” to determine the size of subsystems. This keyword sets the maximum number of fragments in each subsystem. The default value is 6.

- moresubs=2

Add additional two-fragment subsystems. This keyword is recommended for higher accuracy.

6.2 Examples

6.2.1 Cell energy of CO_2 crystal



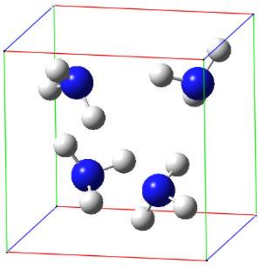
Input file <CO2.gjf>

```
%njobs=4
%nproc=6
%gver=g16
%mem=10gb
#p m062x/cc-pvdz

pbc-gebf { frag=link dis=4.0 maxsubfrag=4 moresubs=2 }

0 1
   C      0.00000     0.00000     0.00000
   O      0.60804     0.60804     0.60804
   O     -0.60804    -0.60804    -0.60804
……
   Tv      5.63000     0.00000     0.00000
   Tv      0.00000     5.63000     0.00000
   Tv      0.00000     0.00000     5.63000
```

Output file <CO2.txt>

```
……
 PBC-GEBF Energy per unit cell:
 M062X          =            -753.933597
  U + P*V    =            -753.933597 +            0.000000
 ENERGY          =            -753.933597
……
 PBC-GEBF task over on: Tue Jan 18 15:59:23 2022
 Total CPU time:     0 days    0 hours    0 minutes 12.4 seconds.
 Total Wall time:    0 days    0 hours    9 minutes       8 seconds.
 Normal Termination of PBC-GEBF Program!
```

6.2.2 Cell energy of NH_3 crystal (with dispersion correction)



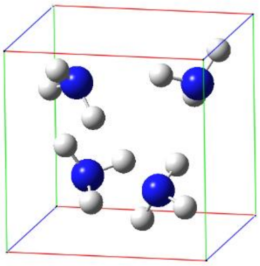
Input file <NH3.gjf>

```
%njobs=8
%nproc=3
%mem=10gb
#p B3LYP/cc-pVDZ Em=GD3BJ

pbc-gebf { frag=link dis=4.0 maxsubfrag=6 moresubs=2 }


0 1
    N      1.1267048196      1.1271942484      1.1285585896
    N      3.8227096921      1.5659496295      4.2643998108
    N      4.2624152569      3.8231134160      1.5673257746
    N      1.5652153709      4.2633107579      3.8246075354
...
   Tv     5.3934570030     -0.0012317719     -0.0009019286
   Tv    -0.0012317719      5.3935431374     -0.0005585776
   Tv    -0.0009019286     -0.0005585776      5.3933018998
```

Output file <NH3.txt>

```
……
  PBC-GEBF Energy per unit cell:
  B3LYP          =            -226.297858
    U + P*V    =            -226.297858 +            0.000000
  ENERGY          =            -226.297858
  ENER(1)    =            -226.297858
  ENER(2)    =            -226.297858
……
  PBC-GEBF task over on: Tue Jan 18 10:36:01 2022
  Total CPU time:     0 days    0 hours    0 minutes 17.7 seconds.
  Total Wall time:    0 days    0 hours    8 minutes      45 seconds.
  Normal Termination of PBC-GEBF Program!
```

6.2.3 Cell energy of NH$_3$ crystal (MP2 level)



Input file <NH3.gjf>

```
…
#p MP2(fulldir)/cc-pVDZ

pbc-gebf { frag=link dis=4.0 maxsubfrag=6 moresubs=2 }

0 1
    N       1.1267048196       1.1271942484       1.1285585896
…
```

Output file <CO2.txt>

```
……
PBC-GEBF Energy per unit cell:
 RHF       =     -224.801239
 MP2       =     -225.588747          E(CORR) =     -0.787508
   U + P*V    =            -225.588747 +           0.000000
 ENERGY          =            -225.588747
 ……
 PBC-GEBF task over on: Tue Jan 18 10:36:23 2022
 Total CPU time:     0 days    0 hours    0 minutes 17.0 seconds.
 Total Wall time:   0 days    0 hours    6 minutes      28 seconds.
 Normal Termination of PBC-GEBF Program!
```

References

References for GEBF module

All publications calculated with GEBF module should acknowledge at least:

[1] W. Li, H. Dong, J. Ma, S. Li. *Acc. Chem. Res.* **2021**, *54*, 169–181.

[2] S. Li, W. Li, J. Ma. *Acc. Chem. Res.* **2014**, *47*, 2712–2720.

[3] S. Li, W. Li, T. Fang. *J. Am. Chem. Soc.* **2005**, *127*, 7215–7226.

[4] S. Li, W. Li, Y. Jiang, J. Ma, T. Fang, W. Hua, S. Hua, H. Dong, D. Zhao, K. Liao, W. Zou, Z. Ni, Y. Wang, X. Shen, B. Hong, *LSQC Program*, Version 2.5. Nanjing University, Nanjing **2022**; see https://itcc.nju.edu.cn/lsqc/.

The license of Gaussian program should be available and Gaussian should be cited. See http://www.gaussian.com/ for more information.

References for CIM module

The following references should be cited if the CIM approach is used:

[1] S. Li, J. Shen, W. Li, Y. Jiang. *J. Chem. Phys.* **2006**, *125*, 74109.

[2] S. Li, J. Ma, Y. Jiang. *J. Comput. Chem.* **2002**, *23*, 237–244.

[3] S. Li, W. Li, Y. Jiang, J. Ma, T. Fang, W. Hua, S. Hua, H. Dong, D. Zhao, K. Liao, W. Zou, Z. Ni, Y. Wang, X. Shen, B. Hong, *LSQC Program*, Version 2.5. Nanjing University, Nanjing **2022**; see https://itcc.nju.edu.cn/lsqc/.

Here Hartree-Fock calculations are performed by the PySCF package and the electron integrals are obtained by Libcint electron integral library. The corresponding references should be cited. See https://pyscf.org/ and https://github.com/sunqm/libcint for more information.

References for PBC-GEBF module

All publications calculated with PBC-GEBF module should acknowledge at least:

[1] W. Li, H. Dong, J. Ma, S. Li. *Acc. Chem. Res.* **2021**, *54*, 169–181.

[2] T. Fang, W. Li, F. Gu, S. Li. *J. Chem. Theory Comput.* **2015**, *11*, 91.

[3] S. Li, W. Li, Y. Jiang, J. Ma, T. Fang, W. Hua, S. Hua, H. Dong, D. Zhao, K. Liao, W. Zou, Z. Ni, Y. Wang, X. Shen, B. Hong, *LSQC Program*, Version 2.5. Nanjing University, Nanjing **2022**; see https://itcc.nju.edu.cn/lsqc/.

The license of Gaussian program should be available and Gaussian should be cited. See http://www.gaussian.com/ for more information.

Contact

LSQC Group

Address: Institute of Theoretical and Computational Chemistry, School of Chemistry and Chemical Engineering, Nanjing University, Nanjing, 210023, P. R. China.

Email: lsqc@nju.edu.cn

Website: http://itcc.nju.edu.cn/lsqc or scan the following QR Code

Please feel free to contact us via our email if you get some problems or have any suggestions for running LSQC program.