

The Fragment Molecular Orbital Approach

Zoe Seeger

Contents

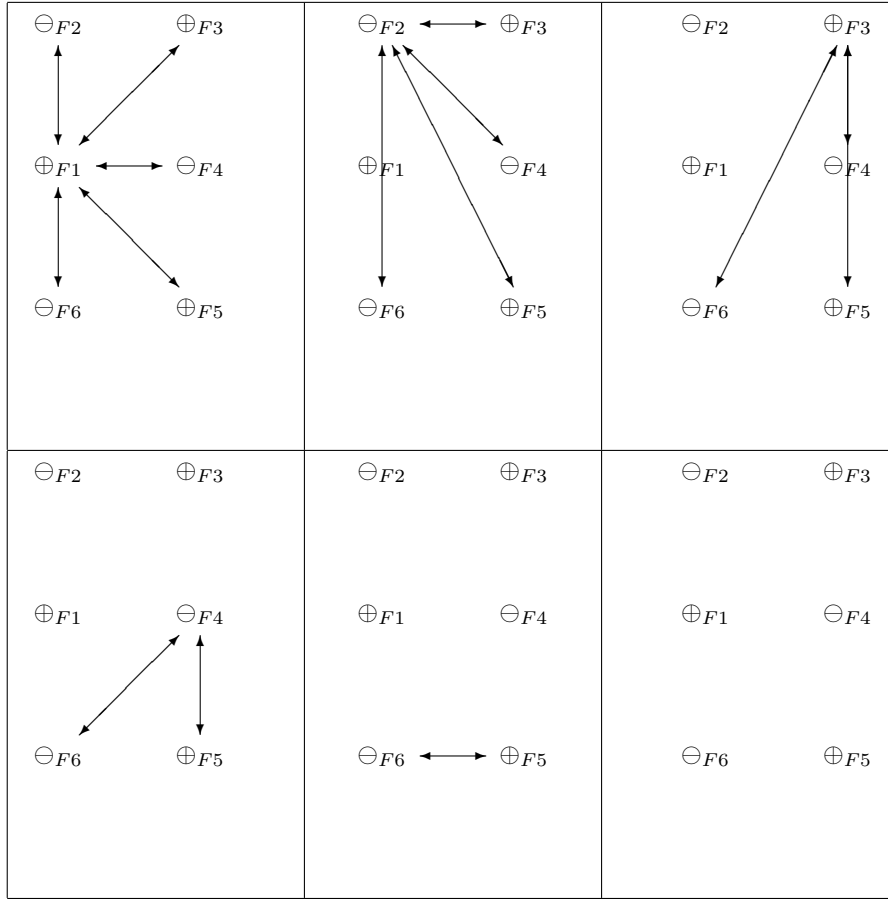
| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 2 | Theory | 4 |
| 3 | Input file | 5 |
| 3.1 | MWORDS | 6 |
| 3.2 | RUNTYP | 6 |
| 3.3 | NGROUP | 7 |
| 3.4 | GBASIS | 7 |
| 3.5 | NFRAG | 7 |
| 3.6 | NBODY | 7 |
| 3.7 | MPLEVL | 7 |
| 3.8 | INDAT | 7 |
| 3.9 | ICHARG | 7 |
| 3.10 | RESDIM, RCORSD | 8 |
| 3.11 | SCSPT, SCSOPO, SCSPAR | 8 |
| 3.12 | Title card | 8 |
| 3.13 | Atoms | 8 |
| 3.14 | Coordinates | 8 |
| 3.15 | CODE | 8 |
| 3.16 | MEMDDI | 9 |
| 3.17 | RITRIM | 9 |
| 4 | Job file | 9 |
| 4.1 | Raijin | 9 |
| 4.2 | Magnus | 10 |
| 5 | Output | 10 |
| 6 | Using PPQC to generate input files | 11 |

1 Introduction

The newly developed Fragment Molecular Orbital (FMO) approach (FMOA) allows for accurate models of condensed systems such as ILs to be assessed. Originally designed to observe protein interactions, it is almost exclusively used for biological systems. [1] The FMOA separates the system into fragments which are treated individually with a method of choice. Application of correlated *ab initio* methods to large systems is extremely time-consuming, and as such, a compromise in thoroughness must be found that does not substitute accuracy. The FMOA is a process that virtually fragments the system and then combines the energy of the partitions with inter-fragment corrections to find the total properties of large structures. When applied to optimising large protein-like molecules, the FMO fragments were created by breaking the bonds between atoms and assigning the electrons to one partition creating ions. In this case ghost atoms need to be included for the missing atoms involved in the bonding such that basis functions are included to correct for the delocalisation of the electrons across the bond. For ionic liquids, clear divisions exist between ions and therefore fragmentation is straightforward, more so than the biological systems it was initially applied to. The partition of a cluster into many parallelisable (one ion per node) calculations creates an overall calculation that scales linearly with molecular size. [2] Previously Hartree-Fock and density functional theory (DFT) calculations have been used for large structures, but the cost and resources scale too steeply with system size, rendering them too expensive for application. [3] This is overcome by the development of FMOA. FMO theory is specialised by how many fragments or body interactions are being considered at once. The two-body (FMO2) and three-body (FMO3) calculations look at all the two and three fragment combinations respectively. This idea is displayed in *Example 1* as six fragments, ($n = 6$), are shown to have fifteen FMO2 combinations.

Example 1.1. All possible combinations of two fragments in a three ion pair system (six ions) where

$$E_{ij} = E_{ji} \text{ for fragments } i \text{ and } j :$$



In general, for a system of n -fragments, the number of two-fragment combinations is $n(n-1)/2$. For

$n = 6$ the possible fragment combinations is 15.

Example 1.2 provides all FMO3 combinations of six fragments, again $n = 6$. It is clearly seen that FMO3 will have exponentially more fragments than FMO2 for a given n . The inter-fragment interactions are very important in large-scale structures and even more so in ionic liquids whose electrostatic and dispersion forces have a large impact on the overall properties. In general, condensed systems such as water, require the inclusion of 2, 3 and 4-body interactions to accurately determine the overall energy of the system. [4] It was shown by our group that FMO3-MP2 provides excellent accuracy of within 0.2 kJ mol^{-1} in full clusters of ILs. [5]

Example 1.2. List of all possible combinations of three combinations in a six-fragment system:

$F_1F_2F_3$ $F_2F_3F_4$ $F_3F_4F_5$ $F_4F_5F_6$
 $F_1F_2F_4$ $F_2F_3F_5$ $F_3F_4F_6$
 $F_1F_2F_5$ $F_2F_3F_6$ $F_3F_5F_6$
 $F_1F_2F_6$ $F_2F_4F_5$
 $F_1F_3F_4$ $F_2F_4F_6$
 $F_1F_3F_5$ $F_2F_5F_6$
 $F_1F_3F_6$
 $F_1F_4F_5$
 $F_1F_4F_6$
 $F_1F_5F_6$

In general, for a system of n -fragments, the number of three-fragment combinations is $\binom{n}{3} = \frac{n!}{6(n-3)!}$.

2 Theory

Initially fragmentation needs to be decided. The most benefit is gained from the FMOA when the system is fragmented into its smallest constituents as less basis functions need be considered at a time remembering that these scale exponentially. However this in turn impacts the energy. On the other hand, having larger fragments results in the full quantum-mechanical description of larger areas, which is more accurate but also more expensive. It is indeed possible to include two molecules in a fragment to boost accuracy.

It is then important to consider the number of interacting body effects that the calculation should account for. As previously stated, the strong hydrogen bonding in water required the inclusion of four-body effects: the inclusion of four fragment combinations. Fragments are called monomers, fragment pairs, called dimers, and fragment triple are called trimers. As is common for optimisations to use lower levels of theory for the location of stationary points and higher levels of theory for single point energy calculations, the same energy enhancement can be obtained by using FMO2 for optimisations and FMO3 for the single point energy calculations.

The fragment interactions are performed in a Coulomb field exerted by the remaining fragments. This field is known as the electrostatic potential or Coulomb bath. In this way the energy of say a dimer will be unequal to the same dimer being calculated separately to the cluster.

The two-body FMO expansion is of the form:

$$E^{FMO2} = \sum_I^N E_I + \sum_{I>J}^N (E_{IJ} - E_I - E_J)$$

Here, the first term sums the energies of the monomers of the system. The second term finds the total energy of each dimer and subtracts the energy of each monomer of the pair such that the monomers are not

counted twice. In this way the two-body effects can be isolated.

The FMO3 energy is found in similar fashion where the monomer and dimer contributions are found first:

$$E^{FMO3} = E^{FMO2} + \sum_{I>J>K}^N (E_{IJK} - E_I - E_J - E_K) - (E_{IJ} - E_I - E_J) - (E_{IK} - E_I - E_K)$$

3 Input file

The FMO approach has been implemented in the GAMESS (The General Atomic and Molecular Electronic Structure System) software package where the input for GAMESS and information for the supercomputer must be held in separate files. [6,7] A labelled example of an input file is given in *Example 3.1* and the highlighted keywords are explained in more detail. Each parameter that is set has a group that it is assigned to. A group is denoted by a \$ and ends with \$END. The first group in the example file is \$SYSTEM where MWORDS is the only defined parameter in the group before \$END. Parameters can be written in any order, as long as they are contained within the correct group. More detailed explanations can be found at reference [9], the website of Mark Gordon's Quantum Theory Group who develop the GAMESS code. This is an important reference when attempting new calculations.

All lines in the input file have one space before the \$group except for the title card and symmetry. The symmetry is generally left as C1 as only simple molecules without interaction obtain higher symmetry.

Most parameters have default values when they are not exclaimed in the input file. Some of these values disable the parameter from being used. It is important when attempting new calculations that the possible parameters enabled are checked such that the default parameters are suitable (see RITRIM).

Note: This is a FORTRAN code and characters written past character 73 do not get read.

Example 3.1. Example input file with important keywords numbered and explained further in the following chapter corresponding to the number they are given (1-14). A quick index is given on the right hand side.

```

$SYSTEM MWORDS=300 $END
$CONTRL SCFTYP=RHF RUNTYP=OPTIMIZE MAXIT=200 ISPHER=1 $END
$GDDI NGROUP=4 $END
$SCF DIRSCF=.TRUE. FDIFF=.FALSE. DIIS=.TRUE. $END
$BASIS GBASIS=CCT $END
$FMO
  NFRAG=4 NBODY=3
  MPLEVL(1)=2
  INDAT(1)= 0,1,-1,
            0,2,-2,
            0,3,-3,
            0,4,-4,
            0
  ICHARG(1)=1,-1,1,-1
  RESPAP=0 RESPPC=-1 RESDIM=100 RCORSD=50
$END
$MP2 CODE=IMS SCSPT=SCS SCSOPO=0.69 SCSPAR=1.32 $END
$DATA
optimisation NaCl FMO3-SRS-MP2/cc-pVTZ
C1
Na 11.0
Cl 17.0
$END
$FMOXYZ
Na      11.0      3.000123      4.093222      -1.093736
Cl      17.0     -4.876367     -1.763248       7.826361
Na      11.0      0.324656      0.723547       0.862346
Cl      17.0     -1.435839      1.893475       0.234782
$END

```

- 1: Memory
- 2: Calculation type
- 3: Number of nodes
- 4: Basis set
- 5: Number of fragments
- 6: Level of FMO
- 7: Level of MPn
- 8: List of fragment atoms, indices correspond to the placement of the fragment in the coordinates list under \$FMOXYZ
- 9: List of charges in order of fragments in coordinates list under \$FMOXYZ
- 10: Distance at which FMO2 combinations are not considered
- 11: MP2 scaling coefficients for opposite and same spin
- 12: Title card
- 13: All atoms and numbers
- 14: Coordinates and atomic numbers

3.1 MWORDS

- Keyword of \$SYSTEM group
- The maximum memory which your job can use on each core
- Where M = mega and a word = 64 bits
- If you know the number of gigabytes per CPU that you are using decide the mwords with:

$$MWORDS = GBs \text{ per CPU} \times 1024/8$$

- Therefore the total memory that you are using is (included in .job file):

$$Total\ GBs = GBs \text{ per CPU} \times CPUs$$

or

$$Total\ GBs = MWORDS \times CPUs \times 8/1024$$

3.2 RUNTYP

- Keyword of \$CONTRL group
- Type of calculation to be performed
- Single point energy calculation ENERGY
- Optimisation OPTIMIZE
- Gradient GRADIENT
- Transition state SADPOINT

3.3 NGROUP

- Keyword of \$GDDI group
- Determines how parallelisable your calculation will be
- Number of calculations to split your system into
- Maximised when equal to number of fragments
- However two fragments can be calculated on one group of CPUs - two fragments are calculated together in a full *ab initio* calculation
- Must be equal to or smaller than number of nodes specified in .job file

3.4 GBASIS

- Keyword of \$BASIS group
- Using standard basis set
- E.g. CCT for cc-pVTZ
- E.g. CCD for cc-pVDZ

3.5 NFRAG

- Keyword of \$FMO group
- The number of FMO fragments
- Must be consistent with number of INDAT fragments

3.6 NBODY

- Keyword of \$FMO group
- Defines the expansion of FMO
- E.g. FMO2 including monomer and dimers 2
- E.g. FMO3 including monomers, dimers and trimers 3

3.7 MPLEVL

- Keyword of \$FMO group
- MPn level
- E.g. MP2 2

3.8 INDAT

- Keyword of \$FMO group
- Assigns atoms in \$FMOXYZ to fragments where the first atom is 1
- E.g. If the first 3 lines are H₂O to be treated as a fragment then the first line of INDAT is written as 0,1,-3,
- The first character of a fragment and last line of INDAT is always 0

3.9 ICHARG

- Keyword of \$FMO group
- Charges of fragments in order of INDAT assignment

3.10 RESDIM, RCORSD

- Keyword of \$FMO group
- Cutoff for dimers SCF and MP2 calculations respectively
- Unitless cutoff
- If closest two atoms of a dimer exceed cutoff the calculation is not performed
- Set very large to include all
- $\text{RESDIM} \geq \text{RCORSD}$

3.11 SCSPT, SCSOPO, SCSPAR

- Keyword of \$MP2 group
- SCSPT initialises the use of SCSOPO and SCSPAR
- SCSOPO and SCSPAR are scaling factors for opposite spin and same spin respectively
- SRS coefficients for ionic liquids are cc-pVDZ = 1.752, cc-pVTZ = 1.640, where in both cases SCSPAR = 0 [9]

3.12 Title card

- Part of \$DATA group
- Irrelevant but necessary
- No space before title

3.13 Atoms

- Part of \$DATA group
- List of atoms in the system with atomic number

3.14 Coordinates

- Part of \$FMOXYZ group
- List of atoms in groups of fragments, atomic number and cartesian coordinates
- Do not forget atomic numbers if obtaining geometry from .xyz

3.15 CODE

- Keyword of \$MP2 group
- The algorithm used within GAMESS to run
- Accounts for different software and RUNTYP
- The default for closed-shell parallel runs is IMS
- For RUNTYP UHF and ROHF use DDI and enable MEMDDI

3.16 MEMDDI

- Keyword of \$SYSTEM group
- Is specified when CODE is not IMS
- The maximum memory used to ‘talk’ between CPUs
- In megawords where a word = 64 bits
- If you know the number of gigabytes per CPU that you are using decide the MWORDS and MEMDDI with:

$$GBs\ per\ CPU = (MWORDS + MEMDDI/CPUs) \times 8/1024$$

and

$$Total\ GBs = (MWORDS \times NCPUS + MEMDDI) \times 8/1024$$

3.17 RITRIM

- Keyword of \$MP2 group
- List of four thresholds for trimer calculations
- Written as: RITRIM(1)=50.0,50.0,50.0,50.0
- Default values have been shown to be too short for ionic liquids

4 Job file

To make the job file for an input file the total memory should be calculated using the equations in section 3.1 or 3.16. The number of CPUs must also be calculated as the number of CPUs per node (often 12 or 16) multiplied by the number of groups the calculation will be divided into (generally equal to the number of fragments). This can be simplified to:

$$nodes = fragments$$

$$CPUs = fragments * CPUs\ per\ node$$

4.1 Raijin

Raijin has 16 CPU nodes with the majority of CPUs containing 2 GBs of memory although 4 GB CPUs also exist.

```
#!/bin/sh
#PBS -P k96
#PBS -l mem=230gb
#PBS -l ncpus=64
#PBS -l jobfs=200gb
#PBS -l walltime=48:00:00
#PBS -l wd

module unload openmpi/1.6.3
module load openmpi/1.8.4
/short/k96/apps/gamess16-srs/rungms.rika inputfile.inp $PBS_NCPUS outputfile.log
```

To be able to use the SCSOPO/SCSPAR parameters the path to GAMESS must be set to the correct version. Additionally, this version was created with a different version of openmpi than the currently loaded one in the Raijin environment and this must be specified in the job script.

4.2 Magnus

Magnus has 12 and 24 CPU nodes. Total memory does not need to be declared.

```
#!/bin/bash --login
#SBATCH --nodes=8
#SBATCH --account=pawsey0197
#SBATCH --time=24:00:00
#SBATCH --export=NONE

module use /group/pawsey0197/software/cle52up04/modulefiles
module load gamess/2016
rungms alt-interchange.inp 00 96 12
```

The lines beginning with ‘module’ are setting the path to the version of GAMESS containing SCS parameters. The total number of CPUs and CPUs per node are specified by ‘96 12’, respectively.

5 Output

Equilibrium geometry:

***** EQUILIBRIUM GEOMETRY LOCATED *****

One body properties:

One-body FMO properties.

Two body properties:

Two-body FMO properties.

Three body properties:

Three-body FMO properties.

Summary of energies:

| One-Body | Two-Body | Three-Body |
|-----------|----------|------------|
| ----- | | |
| Total HF | | |
| MP2 | | |
| Total SCS | | |
| Corr MP2 | | |
| Corr SCS | | |

6 Using PPQC to generate input files

The pre-processor of quantum chemistry (PPQC) is a program created by Samuel Tan adapted from Jason Rigby's QCPP. The original program was designed to make Gaussian and GAMESS input files. The adapted version, PPQC, includes a template file system where the template directed to in the generation file (.gen) can be created and edited by the user as needed.

The generation file:

```
task=gen_cluster_intE
nfrag=8
guess_charges=true
using=geometryfile.xyz
template=/group/pawsey0197/apps/source/ppqc/templates/fmo.template
```

The path to the template file can be altered such that the user can direct PPQC to use their own template file. The task can be changed to create an array of different input syntaxs, however gen_cluster_intE is the best imitation of a FMO job. An example of a template file is:

```
$SYSTEM MWORDS=200 $END
$CONTRL SCFTYP=RHF RUNTYP=ENERGY MAXIT=200 ISPHER=1
      ICHARG=1 $END
$SCF DIRSCF=.TRUE. FDIFF=.FALSE. DIIS=.TRUE. $END
$BASIS GBASIS=CCT $END
$MP2 CODE=IMS SCSPT=SCS SCSOP0=1.05 SCSPAR=0.68 $END
$DATA
SCS energy cation
C1
xyz_data
$END
```

In this example, PPQC replaces xyz_data with the geometry collected from the .xyz file specified in .gen and therefore xyz_data must be included. To run the PPQC the name of the .gen file must be included following the path to the program.

Path to PPQC:

Raijin: /short/k96/apps/bin/ppqc

Magnus: /group/pawsey0197/apps/bin/ppqc

An example of running PPQC on Raijin:

```
/short/k96/apps/bin/ppqc file.gen > inputfile.inp
```

References

- [1] K. Kitaura and D. G. Fedorov, The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems, **2009**.
- [2] E. I. Izgorodina, U. L. Bernard and D. R. MacFarlane, Ion-Pair Binding Energies of Ionic Liquids: Can DFT Compete with Ab Initio-Based Methods? *J. Phys. Chem. A*, **2009**, 113, 7064-7072.
- [3] A. Szabo and N. Ostlund, Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory, *Dover Publications*, **1996**.
- [4] A. J. Stone, Intermolecular Potentials, **2008**, *Science*, 321, 787-789.
- [5] E.I. Izgorodina, J. Rigby and D.R. MacFarlane, Large-scale Ab Initio Calculations of Archetypical Ionic Liquids, **2012**, *Chem. Commun.*, 48, 1493-1495.
- [6] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, *et al.*, General Atomic and Molecular Structure System, **1993**, *J. Comput. Chem*, 14, 13471363.
- [7] D. G. Fedorov and K. Kitaura, The Importance of Three-body Terms in the Fragment Molecular Orbital Method, **2004**, *J. Chem. Phys.*, 120, 6832-6840.
- [8] <http://www.msg.ameslab.gov/gamess/documentation.html>
- [9] S. Tan, S. Barrera Acevedo and E. I. Izgorodina, Generalized Spin-Ratio Scaled MP2 Method for Accurate Prediction of Intermolecular Interactions for Neutral and Ionic Species, **2017**, *J. Chem. Phys.*, 14, 146.