MOLCAS version 8.2

Quickstart Guide for Molcas

Serrano-Andrés L.,
Veryazov V., Lindh R., Malmqvist P.-Å., Widmark P.-O.

Contents

Section 1

Introduction to MOLCAS

1.1 MOLCAS Quantum Chemistry Software

MOLCAS is a Quantum Chemistry software package developed by scientists to be used by scientists. It is neither a commercial product nor is it sold for significant profit gain by its owner, Lund University. The authors of MOLCAS have assembled their collected experience and knowledge in computational Quantum Chemistry to produce a research product which is used as a platform by the scientists in the MOLCAS network to develop new and improved computational tools in Quantum Chemistry. Several of the codes in the MOLCAS software have newly developed leading-edge features. Along with these new capabilities, users should not be surprised to ocasionally discover bugs when using MOLCAS.

The basic philosophy behind MOLCAS is to develop methods that allow accurate *ab initio* treatment of very general electronic structure problems for molecular systems in both ground and excited states which is not an easy task. Nowadays, knowledge about how to obtain accurate properties for single-reference dominated ground states is well developed, and MOLCAS contains a number of codes that can perform such calculations (MP2, CC, CPF, DFT etc). All these methods treat the electron correlation starting from a single determinant (closed or open shell) reference state. Such codes are today's standard in most Quantum Chemistry program.

However, MOLCAS is to be able to treat, highly degenerate states, such as those occurring in excited states, transition states in chemical reactions, diradicaloid systems, heavy metal systems, as well as other chemically important problems, all at the same level of accuracy. This is a much more difficult problem, since a single-determinant approach does not work well in these cases. The key feature of MOLCAS is the multiconfigurational approach. MOLCAS contains codes for general and effective multi-configurational SCF calculations at the Complete Active Space (CASSCF) level, but also employs more restricted MCSCF wave functions such as the Restricted Active Space, RASSCF, and the Generalized Active Space, GASSCF. It is also possible using either CASSCF or RASSCF to employ optimization techniques and obtain equilibrium geometries, transition-state structures, force fields, and vibrational energies using gradient techniques.

Although the RASSCF approach is known to give reasonable structures for degenerate systems both in ground and excited states, in general it is not capable of recovering more than a fraction of the correlation energy. Therefore, it becomes necessary to supplement the multi-configurational SCF treatment with a calculation of dynamic correlation effects. In the earliest version of MOLCAS, this was achieved by means of the multi-reference (MR) CI method. This method has severe limitations in the number of electrons that can be correlated and the size of the reference space which limits study to excited states of small molecules. However, the MRCI code in MOLCAS does have the capacity to produce very accurate wave functions and potential energy surfaces, and is used by many groups for this purpose. In fact, it is also possible to run the COLUMBUS MRCI code together with MOLCAS.

During the period of 1986-90, a new method called CASPT2 was developed, which computes dynamic electron correlation effects for multi-configurational wave functions based on second order perturbation theory and was included into the second version of MOLCAS. From the beginning it was not clear whether or not the CASPT2 method would be sufficiently accurate to be useful in practice, but it turned out to be surprisingly accurate in a number of different types of chemical applications. The CASPT2 approach has become especially important in studies of excited states and spectroscopic properties of large molecules, where no other *ab initio* method has, so far, been applicable. Since the CASPT2 method is based on second order perturbation theory and has, therefore, limitations in accuracy, the error limits have been investigated in a large number of applications. The relative energy errors are small in almost all cases leading to results which can be used for conclusive predictions of molecular properties in ground and excited states. Important application areas for the CASPT2 method are potential energy surfaces for chemical reactions, photochemistry, transition metal chemistry, and heavy element chemistry.

A multi-state version of CASPT2 is available, which allows for the simultaneous study of several electronic states, including their interaction to second order. This code is especially useful in cases where two or more energy surfaces are close in energy. An analytical CASPT2 gradient code is in the process of development, but this work is as yet unfinished. In place of the analytical gradient capability, the present version (8.2) includes a numerical procedure, which allows automatic geometry optimization at the CASPT2 level of theory. It is applicable to all states and systems for which the CASPT2 energy can be computed including the calculation of vibrational frequencies. It is important to note that the CASPT2 method is under constant development.

In the present version (8.2) the Multiconfiguration Pair-Density Functional Theory, MC-PDFT, is also available to treat dynamical correlation.

If only a few electrons are correlated, the active space can be quite large without too many configurations being generated, but in most cases the number of active electrons is comparable to the number of active orbitals. Occasionally, a larger active space would be preferred, but would result in too many configurations (more than a few million CSF's). The more general RASSCF scheme can be useful, at the price of less efficient calculations, and the risk of bad convergence properties in the orbital optimization. The CASPT2 program handles also such wave functions, but will not include correlation within the active space, i.e., interaction with states that would have been included in the full CASSCF but are excluded by the RASSCF restrictions. The RASSCF wave function is regarded as an accurate approximation to the full CASSCF wave function, and the CASPT2 program evaluates only dynamic correlation that involves at least one non-active orbital.

MOLCAS not only contains the ability to produce various types of wave functions, but also can compute molecular properties using formulas of expectation values or finite perturbation theory using the RASSI program. The RASSI program has the capacity to compute the interaction between several RASSCF wave functions based on different orbitals which are generally non-orthonormal (i.e. a non-orthogonal CI). RASSI is routinely used to compute transition dipole moments in spectroscopy, to study electron transfer, and to obtain eigenstates of a relativistic Hamiltonian with inclusion of spin-orbit interaction.

Scalar, i.e. spin-averaged, relativistic effects are typically included in any calculations by using Douglas-Kroll-Hess transformation of one-electron integrals. The standard basis set library ANO-RCC [1, 2, 3, 4, 5] is optimized for use with these integrals and to include correlation of semi-core orbitals, and to to have uniform quality across the periodic system up to element 96, Curium. For heavier elements, typical calculations include the spin-orbit interaction by using CASSCF wave functions as a many-electron basis set, letting RASSI compute a Hamiltonian matrix over the set of all spin-components of these functions, correct for dynamic correlation using CASPT2 and include a one-electron spin-orbit Hamiltonian. This procedure has been shown to give accurate results in a number of studies for actinides and other heavy atom systems [6].

It is also possible to model solvent effects by adding a reaction field Hamiltonian (PCM). A new QM/MM model is also included in 8.2.

The release of MOLCAS-7 leads to many important enhancements. The sizes of the systems that can be treated with MOLCAS were previously restricted because of limitations in storing two-electron integrals for large basis sets. This system size limitation has been substantially reduced by the introduction of a Cholesky decomposition of the two-electron integrals. This feature is used in MOLCAS-7 at all levels of theory [7, 8, 9] and speeds up calculations by orders of magnitude, extending the size of basis sets that can be used. Accuracy can be controlled by the threshold used in the decomposition. The same approach can be used to generate RI auxiliary basis sets on the fly, allowing the calculation of energy derivatives for HF, MP2, DFT, and CASSCF levels of theory.

It is important to emphasize that important problems in Quantum Chemistry cannot be solved by simply applying ***black box*** techniques. Nor is MOLCAS a ***black box*** tool. A typical MOLCAS user should be someone with a high degree of chemical insight, who has some knowledge of different Quantum Chemical models in use today, and, most importantly, is able to apply these models to the appropriate chemical problem while understanding the inherent accuracy of these methods. The typical MOLCAS user should also apply critical analysis of results, take nothing for granted, and always check that the results are consistent with the model that was used. The skill to use MOLCAS effectively will not come immediately, but the user has several resources including this manual and examples which explain how different key projects were solved using MOLCAS . Users are certain to find them helpful in their own attempts to master the software for use in chemical applications. The MOLCAS group also arranges regular workshops, which provide a more intimate environment on learning how to use MOLCAS .

1.2 The MOLCAS Manual

1.2.1 Manual in Four Parts

This manual is designed for use with the *ab initio* Quantum chemistry software package MOLCAS 8.2 developed at the by the world-wide MOLCAS team where its base and origin is the Department of Theoretical Chemistry, Lund University, Sweden. MOLCAS is designed for use by Theoretical Chemists and requires knowledge of the Chemistry involved in the calculations in order to produce and interpret the results correctly. The package can be moderately difficult to use because of this 'knowledge requirement', but the results are often more meaningful than those produced by ***blackbox*** packages which may not be sufficiently chemically precise in either input or output.

The MOLCAS manual is divided in four parts to facilitate its use.

1. The **MOLCAS *Installation Guide*** describes simple and more complex aspects on how to install, tailor, and control the MOLCAS package.
2. The ***Short Guide to* MOLCAS** is a brief introductory guide which addresses the needs of the novice and intermediate users and is designed for all those who want to start using MOLCAS as soon as possible. Only basic environment definitions, simple input examples, and minimal description of output results are included in the short guide.

   Two types of introductory tutorials are given in the short guide: problem-based and program-specific.

   (a) Problem-based tutorials are exercises focused on solving a simple Quantum Chemical project and contain all the required input files. Examples include computing electronic energy of a molecule at different levels of theory, optimizing the geometry of a molecule, calculating the transition state in the ground state of a chemical system, and computing an excited state. The input files for this section can be found in the directory $MOLCAS/doc/samples/problem_based_tutorials. These examples are also employed in MOLCAS workshops that the MOLCAS team has organized in recent years.

   (b) Another type of tutorial is designed for the first-time user to provide an understanding of program modules contained in MOLCAS include simple, easy-to-follow examples for many of these modules.

   The systems covered in the short guide are not necessarily calculated with most suitable methods or produce highly significant results, but provide both several tips for the beginner and actual input file formats.

   The ***Short Guide to* MOLCAS** can be independently printed as a booklet.
3. The **MOLCAS *User's Guide*** contains a complete listing of the input keywords for each of the program modules and a information regarding files used in each calculation. Here the user will find all keywords that can be used together with a specific program and thus how to set up the input for a MOLCAS run.
4. ***Advanced Examples*** and ***Annexes*** include outlines of actual research performed using MOLCAS.

   The approach to a research project is outlined including input files and shell scripts. More importantly, however, the value of the calculations is evaluated and advanced features of 8.2 are used and explained to improve the value of the results.

The complete manual is available on the net in HTML and PDF formats
(*`http://www.molcas.org`*).

1.2.2 Notation

For clarity, some words are printed using special typefaces.

- Keywords, i.e. words used in input files, are typeset in the small-caps typeface, for example EndOfInput.
- Programs (or modules) are typeset in the teletype typeface. This will eliminate some potential confusion. For example, when discussing the RASSCF method, regular uppercase letters are used, while the program will look like `RASSCF`.
- Files are typeset in the slanted teletype typeface, like *`InpOrb`*.
- Commands, unix or other, are typeset in a sans serif typeface, like ln -fs.
- Complete examples, like input files, shell scripts, etc, are typeset in the teletype typeface.

1.3 Acknowledgment

The Molcas project is carried out by the Lund University quantum chemistry group supported by the Swedish Science Research Council (VR).

The MOLCAS group is a member of the Linnaeus project *Organising Molecular Matter, OMM.*

The MOLCAS group acknowledges the contributions from the scientists in the MOLCAS Network, who are members of the MOLCAS developers team.

The Lund University Center for Scientific and Technical Computing (LUNARC) has provided computer resources for the project.

1.4 Citation for Molcas

The recommended citations for Molcas Version 8.2 are:

Molcas 8: F. Aquilante, J. Autschbach, R. K. Carlson, L. F. Chibotaru, M. G. Delcey, L. De Vico, I. Fdez. Galván, N. Ferré, L. M. Frutos, L. Gagliardi, M. Garavelli, A. Giussani, C. E. Hoyer, G. Li Manni, H. Lischka, D. Ma, P. Å. Malmqvist, T. Müller, A. Nenov, M. Olivucci, T. B. Pedersen, D. Peng, F. Plasser, B. Pritchard, M. Reiher, I. Rivalta, I. Schapiro, J. Segarra-Martí, M. Stenrup, D. G. Truhlar, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, V. P. Vysotskiy, O. Weingart, F. Zapata, R. Lindh, *Journal of Computational Chemistry*, **37**, 506, (2016). DOI: 10.1002/jcc.24221

Molcas 7: F. Aquilante, L. De Vico, N. Ferré, G. Ghigo, P.-Å. Malmqvist, P. Neogrády, T. B. Pedersen, M. Pitonak, M. Reiher, B. O. Roos, L. Serrano-Andrés, M. Urban, V. Veryazov, R. Lindh, *Journal of Computational Chemistry*, **31**, 224, (2010). DOI: 10.1002/jcc.21318

Code development: V. Veryazov, P.-O. Widmark, L. Serrano-Andrés, R. Lindh, B. O. Roos, *International Journal of Quantum Chemistry*, **100**, 626 (2004). DOI: 10.1002/qua.20166

Molcas 6: G. Karlström, R. Lindh, P.-Å. Malmqvist, B. O. Roos, U. Ryde, V. Veryazov, P.-O. Widmark, M. Cossi, B. Schimmelpfennig, P. Neogrády, L. Seijo, *Computational Material Science*, **28**, 222 (2003). DOI: 10.1016/S0927-0256(03)00109-5

The following persons have contributed to the development of the MOLCAS software:

K. Andersson, F. Aquilante, M. Barysz, A. Bernhardsson, M. R. A. Blomberg, J. Boström, Y. Carissan, L. Chibotaru, D. L. Cooper, M. Cossi, M. Delcey, A. Devarajan, L. De Vico, I. Fdez. Galván, N. Ferré, M. P. Fülscher, A. Gaenko, L. Gagliardi, G. Ghigo, C. de Graaf, S. Gusarov, B. A. Hess, D. Hagberg, J. M. Hermida-Ramón, A. Holt, G. Karlström, J. W. Krogh, R. Lindh, P.-Å. Malmqvist, T. Nakajima, P. Neogrády, J. Olsen, T. B. Pedersen, D. Peng, M. Pitonak, J. Raab, M. Reiher, B. O. Roos, U. Ryde, I. Schapiro, B. Schimmelpfennig, M. Schütz, L. Seijo, L. Serrano-Andrés, P. E. M. Siegbahn, J. Stålring, B. Suo, P. Sushko, T. Thorsteinsson, T. Tsuchiya, L. Ungur, S. Vancoillie, V. Veryazov, V. Vysotskiy, and P.-O. Widmark.

1.5 Web Site

Please contact on the web at :
URL: `http://www.molcas.org`

1.6 Disclaimer

MOLCAS is shipped on an "as is" basis without warranties of any kind. The authors of MOLCAS-8 therefore assume no responsibility of any kind from the use of the supplied material. Permission is granted to use the package, but not to reproduce or redistribute any part of this material by any means. Material in the software may not be included in any commercial product. The authors reserve the right to change plans and existing material without notice.

Section 2

Quickstart Guide for Molcas

2.1 Introduction

Running Molcas 8.2 requires a small number of operations. This section of the manual, entitled **Quickstart Guide for Molcas** is aimed at those users who want to immediately run a simple Molcas calculation in order to become familiar with the program. Basic hints are included which set the proper environment, build simple input files, run a calculation, and subsequently extract information from the resulting output.

2.2 Molcas Environment Setup

The environment variable (MOLCAS) and Molcas driver (*`molcas`*) must be defined in order to run Molcas. The MOLCAS environment variable points to the root directory of the Molcas installation and can be defined by the bash shell command

```
export MOLCAS=/home/molcas/molcas.version
```

The location of the Molcas driver is defined at installation time and is typically located in *`/usr/local/bin`* or *`$HOME/bin`*. Check to ensure that this directory is included in your path. Otherwise, the path can be extended by the following command:

```
export PATH=$PATH:$HOME/bin
```

In addition, the variable MOLCAS_NPROCS is needed to run Molcas in parallel. This specifies the number of MPI processes that will be used.

It may be also convenient to define environment variables such as WorkDir which points to a directory for intermediate files and Project to define the name of a project:

```
export Project=MyMolecule
```

Molcas will provide default values if they are not explicitly defined. For a discussion of other Molcas environment variables, please see the section on **Environment and EMIL Command** . All environment variables can either be defined explicitly or entered in a shell script which can be subsequently executed.

2.3 Customization of MOLCAS Execution

MOLCAS has flexible control of organizing filenames and directories used during a calculation. The default values used for customization can be altered either by shell variables or a resource file ***molcasrc*** which is more preferable. A command molcas setuprc provides guided help if to create such file.

The terminology used in this chapter:

- `LOG`: the output and error files produced by MOLCAS .
- `ProjectName`: the Project name used for file naming.
- `RUNFILE`: a file used in a calculation will be named as `ProjectName`.Runfile,
- `WorkDirName`: the WorkDir name used as the directory for temporary/binary files produced by MOLCAS .
- `Scratch`: the scratch disk area which provides a path to a parent directory for `WorkDirName`s.

  The ***WorkDir*** variable used in the MOLCAS manual is constructed as `Scratch`/`WorkDirName`,
- `CurrDir`: the submit directory where the MOLCAS command was issued.

  Note, that in this tutorial, it is assumed that the input file is located in `CurrDir`,
- `OutputDir`: the output directory which is used for storage of extra output files, such as Orbital files and molden files.

It is quite important to understand, that if a user performs two consecutive runs of molcas, using the same scratch area (WorkDir) and project name, MOLCAS will try to reuse intermediate data, e.g. integrals and orbitals, in order to make a restart of a calculation. This can save time, but can also be can be dangerous if two consecutive calculations are not compatible.

Assuming that ***molcasrc*** does not exist, and no environment is set, the command molcas inputfile will use the following defaults:

- `LOG` is printed to the screen,
- `OutputDir` and `CurrDir` are defined to be the same directory,
- `ProjectName` is s taken as the name of ***inputfile*** by removing the suffix (before the last . (dot) character),
- `Scratch` is defined as ***/tmp/***,
- and `WorkDirName` is defined from the `ProjectName` plus a random suffix.

For example, when a user issues the following commands:

```
cd /home/joe/projects/water
vi H2O.DFT.input
molcas H2O.DFT.input
```

the following files will be generated:

```
/home/joe/projects/water/H2O.DFT.ScfOrb
/home/joe/projects/water/H2O.DFT.scf.molden
...
/tmp/H2O.DFT.15014/H2O.DFT.RunFile
...
```

If a flag -f is used in a MOLCAS command, `LOG` files will be stored in the `CurrDir` directory with a name `ProjectName`.log and `ProjectName`.err.

`ProjectName` can either be set in a shell script running MOLCAS or included directly into the MOLCAS command:

```
molcas Project=water H2O.DFT.input
```

will change the default value for `ProjectName` to water.

If the `MOLCAS_WORKDIR` environment variable is set either as part of MOLCAS command or is included in the ***`molcasrc`*** file), the name of WorkDir will NOT be random, but determined by the `ProjectName`.

Example:

```
cd /home/joe/projects/water
vi H2O.DFT.input
molcas MOLCAS_WORKDIR=/tmp Project=water -f H2O.DFT.input
```

will generate the following files:

```
/home/joe/projects/water/water.log
/home/joe/projects/water/water.ScfOrb
...
/tmp/water/water.RunFile
...
```

For More options to control the behavior of MOLCAS , run the command molcas setuprc script. The file ***`molcasrc`*** can be used to set global preferences for the MOLCAS package and/or to set user preferences. The ***`setuprc`*** script creates a ***`molcasrc`*** file (***`HOME/.Molcas`***) in a users home directory.

The following ***`molcasrc`*** file for uses the ***`/scratch`*** area as a parent for WorkDirs and Project name generated for the the name of the input file, then removes WorkDir before a calculation followed by subsequent retains of this file when the calculation finished:

```
# Version 1.0
MOLCAS_MEM=256
MOLCAS_WORKDIR=/scratch
MOLCAS_NEW_WORKDIR=YES
MOLCAS_KEEP_WORKDIR=YES
MOLCAS_PROJECT=NAME
```

Once the ***`molcasrc`*** is created, it is usually not necessary to use shell script or environment variables to run MOLCAS.

2.4 MOLCAS Command-Line Help System

Just by typing molcas help you get access to MOLCAS Command-Line Help System. There are different options:

- molcas help produces a list of available programs and utilities.
- molcas help module yields the list of keywords of the program `MODULE`.
- molcas help module keyword offers the detailed description of the keyword.
- molcas help -t text displays a list of keywords that contain the text word in their description.

2.5 Input Structure and EMIL Commands

MOLCAS has a modular program structure. The easiest way to run calculations is to prepare an input file in which the different programs are executed sequentially when the the module name (&module) is provided. The keywords of module name then follow, with each entry on a separate line or several entries on one line, separated by ;. In addition to specific program module keywords, MOLCAS incorporates certain commands (See section on EMIL Commands.) that allow operations such as looping over the modules, allowing partial execution, changing variables, and substituting certain Unix commands.

2.6 Basic Examples

2.6.1 Simple Calculation on Water

Start by preparing a file containing the cartesian coordinates of a water molecule.

```
 3
Angstrom
 O       0.000000  0.000000  0.000000
 H       0.758602  0.000000  0.504284
 H       0.758602  0.000000 -0.504284
```

which is given the name ***`water.xyz`***. In the same directory we prepare the input for the MOLCAS run. We can name it ***`water.input`***:

In addition to using an editor to insert atomic coordinates into a file, a coordinate file can be obtained by using a graphical interface program, for example, the `LUSCUS` module as shown later in this guide.

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
```

The **GATEWAY** program module combines the molecular geometric of water (In this case, from the external file, water.xyz) and the basis set definition. The **SEWARD** program module then computes the integrals, and **SCF** program modules completer the calculation by computing the Hartree-Fock wave function.

To run the calculation, the following command is used:

```
molcas water.input -f
```

The file ***water.log*** now contains output from the calculation, and the ***water.err*** includes any error messages. In the same directory, other files, including ***water.scf.molden*** or ***water.lus*** (if the keyword GRID˙IT is added at end of input file) that help to analyze the results graphically with the external graphical viewer **LUSCUS** or **MOLDEN** program. Examples of their use are demonstrated below.

In the case of an open-shell calculation (UHF or UDFT), the **SCF** program is again used. Below, two examples are shown:

(a) A UDFT calculation yielding an approximate doublet by setting the charge to +1, even if they are not pure spin functions:

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
 charge=+1
 uhf; ksdft=b3lyp
```

(b) A triplet state (using keyword ZSPIN to specify that there are two more α than β electrons) states

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
 zspin=2
 uhf; ksdft=b3lyp
```

2.6.2 Geometry Optimization

In the next example, a DFT/B3LYP geometry optimization is performed on the ground state of the water molecule. Notice that, after &gateway has defined the coordinates and basis set definition, the EMIL commands >>> Do while and >>> EndDo are employed to form a loop with the **SEWARD**, **SLAPAF**, and **SCF** programs until convergence of geometry optimization is reached. Program **SEWARD** computes the integrals in atomic basis, **SCF** computes the DFT energy, and the program **SLAPAF** controls the geometry optimization and uses the module **ALASKA** to compute the gradients of the energy with respect to the degrees of freedom. **SLAPAF** generates the new geometry to continue the iterative structure optimization process and checks to determine convergence parameters are satisfied notifying MOLCAS and stopping the loop.

```
&GATEWAY
 coord=water.xyz
```

```
  basis=ANO-S-MB
>> Do While
  &SEWARD
  &SCF
    ksdft=b3lyp
  &SLAPAF
>>> EndDo
```

The above example illustrates the default situation of optimizing to a minimum geometry without any further constraint. If other options are required such as determining a transition state, obtaining a states crossing, or imposing a geometry constraint, specific input should be added to program `SLAPAF`.

One of the most powerful aspects of MOLCAS is the possibility of computing excited states with multiconfigurational approaches. The next example demonstrates a calculation of the five lowest singlet roots in a State-Average (SA) CASSCF calculation using the `RASSCF` program. It also illustrates the addition of the `CASPT2` program to determine dynamical correlation which provides accurate electronic energies at the CASPT2 level. The resulting wave functions are used in the `RASSI` module to calculate state-interaction properties such as oscillator strengths and other properties.

```
&gateway
Coord
 8
Acrolein coordinates in Angstrom
 O     -1.808864   -0.137998    0.000000
 C      1.769114    0.136549    0.000000
 C      0.588145   -0.434423    0.000000
 C     -0.695203    0.361447    0.000000
 H     -0.548852    1.455362    0.000000
 H      0.477859   -1.512556    0.000000
 H      2.688665   -0.434186    0.000000
 H      1.880903    1.213924    0.000000
Basis=ANO-S-MB
Group=Nosym
&SEWARD
&RASSCF
  nactel  = 6 0 0
  inactive= 12
  ras2    = 5
  ciroot  = 5 5 1
&CASPT2
  multistate=5 1 2 3 4 5
&RASSI
  Nr of Job=1 5; 1 2 3 4 5
  EJob
```

Notice that the GROUP with the option NOSYM has been used to prevent `GATEWAY` from identifying the symmetry of the molecule (C_s in this case). Otherwise, the input of the `RASSCF` program will have to change to incorporate the classification of the active space into the corresponding symmetry species. Working with symmetry will be skipped at this stage, although its use is very convenient in many cases. A good strategy is to run only `GATEWAY` and let the program guide you.

The `RASSCF` input describes the active space employed, composed by six active electrons distributed in five active orbitals. By indicating twelve inactive orbitals (always doubly occupied), information about the total number of electrons and the distribution of the orbitals is then complete. Five roots will be obtained in the SA-CASSCF procedurei, and all them will be computed at the CASPT2 level to obtain the transition energies at the higher level of theory. Further, the `RASSI` will compute the transition properties, in particular, transition dipole moments and oscillator strengths.

2.6.3 GASSCF method

In certain cases it is useful/necessary to enforce restrictions on electronic excitations within the active space beyond the ones accessible by RASSCF. These restrictions are meant to remove configurations that contribute only marginally to the total wave function. In MOLCAS this is obtained by the GASSCF approach [10]. In GASSCF an arbitrary number of active spaces may be chosen. All intra-space excitations are allowed (Full-CI in subspaces). Constraints are imposed by user choice on inter-space excitations. This method, like RASSCF, allows restrictions on the active space, but they are more flexible than in RASSCF. These restrictions are particularly useful when the cost of using the full CI expansion of the active space is beyond reach. These restrictions allow GASSCF to be applied to larger and more complex systems at affordable cost. Instead of a maximum number of holes in RAS1 and particles in RAS3, accumulated minimum and maximum numbers of electrons are specified for GAS1, GAS1+GAS2, GAS1+GAS2+GAS3, etc. in order to define the desired CI expansion. The GAS scheme reduces to CAS or RAS when one or three spaces are chosen and restrictions on electron excitations are adequately imposed. When and how to use the GAS approach? We consider three examples: (1) an organometallic material with separated metal centers and orbitals not delocalized across the metal centers. One can include the near degenerate orbitals of each center in its own GAS space. This implies that one may choose as many GAS spaces as the number of multiconfigurational centers. (2) Lanthanide or actinide metal compounds where the f-electrons require a MC treatment but they do not participate in bonding neither mix with d orbitals. In this case one can put the f orbitals and their electrons into one or more separated GAS spaces and not allow excitations from and/or to other GAS spaces. (3) Molecules where each bond and its correlating anti-bonding orbital could form a separate GAS space as in GVB approach. Finally, if a wave function with a fixed number of holes in one or more orbitals is desired, without interference of configurations where those orbitals are fully occupied the GAS approach is the method of choice instead of the RAS approach. There is no rigorous scheme to choose a GAS partitioning. The right GAS strategy is system-specific. This makes the method versatile but at the same time it is not a black box method. An input example follows:

```
&RASSCF
nActEl
 6 0 0
FROZen
0 0 0 0 0 0 0 0
INACTIVE
2 0 0 0 2 0 0 0
GASScf
3
 1 0 0 0 1 0 0 0
2 2
 0 1 0 0 0 1 0 0
4 4
 0 0 1 0 0 0 1 0
6 6
DELEted
0 0 0 0 0 0 0 0
```

In this example the entire active space counts six active electrons and six active orbitals.

These latter are partitioned in three GAS spaces according to symmetry consideration and in the spirit of the GVB strategy. Each subspace has a fixed number of electrons, TWO, and no interspace excitations are allowed. This input shows clearly the difference with the RAS approach.

2.6.4 Solvation Effects

MOLCAS incorporates the effects of the solvent using several models. The most common is the cavity-based reaction-field Polarizable Continuum Model (PCM) which is invoked by adding the keyword RF-INPUT to the `SEWARD` code and is needed to compute the proper integrals.

```
&GATEWAY
  coord=CH4.xyz
  Basis=ANO-S-MB
&SEWARD
  RF-Input
   PCM-Model
   Solvent=Water
  End of RF-Input
&RASSCF
  Nactel=8 0 0
  Inactive=1
  Ras2=8
&CASPT2
  rfpert
```

The reaction field is computed in a self-consistent manner by the `SCF` or `RASSCF` codes and added as a perturbation to the Hamiltonian in the `CASPT2` method with the keyword RFPERT.

2.7 Analyzing Results: Output Files and the `LUSCUS` Program

MOLCAS provides a great deal of printed information in output files, and the printing level is controlled by the environmental variable MOLCAS˙PRINT. By default this value is set to TWO, but can be modified by environmental variable MOLCAS˙PRINT Typical MOLCAS output contains the program header and input information, conditions of the calculation, the number of steps to achieve convergence, the energies and wave functions, and final results, including in many cases the molecular orbital coefficients as well as an analysis of the properties for the computed states.

2.7.1 LUSCUS: Grid and Geometry Visualization

MOLCAS developers have developed a graphical interface that can be used both to create input for the MOLCAS program and to analyze the results in a graphical manner by visualizing molecular orbitals, density plots, and other output properties.

The first version of the code has the name GV (stands for Grid Viewer, or Geometry Visualization. By an accident, the name also matches the nicknames of the main developers). GV program uses a very limited set of graphic libraries, and thus has very primitive user interface.

The next generation of GV program has the name LUSCUS. Luscus re-uses the code of GV, and so GV users can use the same key combinations to operate with LUSCUS. At the same time, LUSCUS provides a user-friendly interface, and contains many new options, compared to GV.

LUSCUS can be obtained from https://sourceforge.net/projects/luscus/.

LUSCUS can read the files only in one format: Luscus internal format (*.lus*). This format contains two sections: XYZ cartesian coordinates, and XML formated data. It means that a standard XYZ file is a valid file in LUSCUS format.

Files with different formats, e.g. molden files, can be understood by LUSCUS since they can be converted to LUSCUS format by a corresponding plug-in. For instance, opening a file with the extension *.molden*, LUSCUS automatically runs a plug-in to convert a file from molden format to LUSCUS format. Saving a LUSCUS file as a Molcas orbital file will automatically run a converter from LUSCUS format to Orbital format.

- luscus xyz˙file: reads coordinates from a cartesian coordinate file.

  A molecule can be visualized and modified with the use of the left-button of the mouse and the keyboard. Below are some of the most useful commands.

Left mouse click:	Select atoms (if two, a bond is selected, if three a bond angle, if four a dihedral angle
Left mouse + Shift click:	Mark/unmark atoms to/from the group
Middle mouse/Space:	Remove selection, or marking
Insert key:	Insert atom
PageUp, PageDown:	Alter type of selected atom or bond
Delete/Supress key:	Delete a selected atom
+/-:	Change a value of selected bond/angle in steps
Backspace:	Undo last action
Home:	Set selected atom to center of coordinates
F8 key:	Find or apply symmetry

- luscus molden˙file: reads (check the comment about plug-in) from MOLDEN files such as *wavefunction.molden*, *freq.molden*, and *geo.molden*.

  Note that MOLCAS produces molden files with several extensions, so it is recommended to visualize these files by using `LUSCUS`.

- luscus grid˙file: reads coordinates and densities and molecular orbitals from a binary *grid_file*.

  This file is generated by `GRID_IT` and, by default, placed in the *$WorkDir* directory with the name *$Project.lus*. The program allows displaying total densities, molecular orbitals, and charge density differences.

  If MOLCAS and Luscus are installed locally, `LUSCUS` can also be called from user input as shown in the following example:

```
&GATEWAY
     coord = acrolein.xyz
     basis = ANO-L-MB
&SEWARD
&SCF
```

```
&GRID_IT
ALL
* running external GUI program luscus
! luscus $Project.lus
* User has to select active space and save GvOrb file!
&RASSCF
Fileorb=$CurrDir/$Project.GvOrb
```

Note, that in the example above, the `GRID_IT` program will generate a *`$Project.lus`* file which `LUSCUS` then uses, eliminating the need for defining *`$Project.lus`* and allowing this file to be overwritten. `RASSCF` will read starting orbitals from the *`$Project.GvOrb`* file.

Section 3

Problem Based Tutorials

3.1 Electronic Energy at Fixed Nuclear Geometry

The MOLCAS 8.2 suite of Quantum Chemical programs is modular in design, and a desired calculation is achieved by executing a list of MOLCAS program modules in succession, occasionally manipulating the program information files. If the information files from a previous calculation are saved, then a subsequent calculation need not recompute them. This is dependent on the correct information being preserved in the information files for the subsequent calculations. Each module has keywords to specify the functions to be carried out, and many modules rely on the specification of keywords in previous modules.

In the present examples the calculations will be designed by preparing a single file in which the input for the different programs is presented sequentially. The initial problem will be to compute an electronic energy at a fixed geometry of the nuclei, and this will be performed using different methods and thus requiring different MOLCAS program modules.

First, the proper MOLCAS environment has to be set up which requires that following variables must be properly defined, for instance:

```
export MOLCAS=/home/molcas/molcas.8.2
export Project=CH4
export WorkDir=/home/user/tmp
```

If not defined, MOLCAS provides default values for the above environment variables:

- The MOLCAS variable will be set to the latest implemented version of the code.

  This variable is set directly in the MOLCAS home directory

- Project and WorkDir have the default values None and $PWD, respectively.

  It is very important that the molcas driver, called by command molcas, and built during the installation of the code, is included in the $PATH.

The first run involves a calculation of the SCF energy of the methane (CH_4) molecule. Three programs should be used: `GATEWAY` to specify information about the system, `SEWARD` to compute and store the one- and two-electron integrals, and `SCF` to obtain the Hartree-Fock SCF wave function and energy.

The three MOLCAS programs to be used leads to three major entries in the input file: `GATEWAY`, `SEWARD`, and `SCF`. The `GATEWAY` program contains the nuclear geometry in cartesian coordinates and the label for the one-electron basis set. The keyword COORD allows automatic insertion of `GATEWAY` input from a standard file containing the cartesian coordinates in Angstrom which can be generated by programs like `LUSCUS` or `MOLDEN`). No symmetry is being considered so the keyword GROUP=C1 is used to force the program not to look for symmetry in the CH_4 molecule, and ,thus, input for `SEWARD` is not required. In closed-shell cases, like CH_4, input for `SCF` is not required. All the input files discussed here can be found at *$MOLCAS/doc/samples/problem_based_tutorials*, including the file ***SCF.energy.CH4*** described below.

```
*SCF energy for CH4 at a fixed nuclear geometry.
*File: SCF.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord  = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
```

where the content of the ***CH4.xyz*** file is:

```
5
distorted CH4 coordinates in Angstroms
C    0.000000     0.000000     0.000000
H    0.000000     0.000000     1.050000
H    1.037090     0.000000    -0.366667
H   -0.542115    -0.938971    -0.383333
H   -0.565685     0.979796    -0.400000
```

To run MOLCAS , simply execute the command

```
molcas SCF.energy.CH4.input > SCF.energy.CH4.log 2 > SCF.energy.CH4.err
```

where the main output is stored in file ***SCF.energy.CH4.log***

or

```
molcas -f SCF.energy.CH4.input
```

where the main output is stored in ***SCF.energy.CH4.log***, and the default error file in ***SCF.energy.CH4.err***.

The most relevant information is contained in the output file, where the `GATEWAY` program information describing the nuclear geometry, molecular symmetry, and the data regarding the one-electron basis sets and the calculation of one- and two-electron integrals, as described in section 4.4. Next, comes the output of program `SCF` with information of the electronic energy, wave function, and the Hartree-Fock (HF) molecular orbitals (see section 4.5).

Files containing intermediate information, integrals, orbitals, etc, will be kept in the $WorkDir directory for further use. For instance, files ***$Project.OneInt*** and ***$Project.OrdInt*** contain the one- and two-electron integrals stored in binary format. File ***$Project.ScfOrb*** stores the HF molecular orbitals in ASCII format, and ***$Project.RunFile*** is a communication file between programs. All these files can be used later for more advanced calculations avoiding a repeat of certain calculations.

There are graphical utilities that can be used for the analysis of the results. By default, MOLCAS generates files which can be read with the `MOLDEN` program and are found in the $WorkDir including the fileCH4.scf.molden. This file contains information about molecular geometry and molecular orbitals, and requires the use if *Density Mode* in `MOLDEN`. However, MOLCAS has its own graphical tool, program `LUSCUS`, which is a viewer based on openGL and allows the visualization of molecular geometries, orbitals, densities, and density differences. For example, a graphical display of the CH_4 molecule can be obtained from a standard coordinate file by the following command:

```
luscus CH4.xyz
```

In order to obtain the information for displaying molecular orbitals and densities, it is necessary to run the MOLCAS program called `GRID_IT`:

```
*SCF energy for CH4 at a fixed nuclear geometry plus a grid for visualization.
*File: SCF.energy_grid.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 Group = C1
&SEWARD; &SCF
&GRID_IT
 All
```

Now, execcute the MOLCAS program:

```
molcas SCF.energy_grid.CH4.input -f
```

In the $WorkDir and $PWD directories a new file is generated, *`CH4.lus`* which contains the information required by the `GRID_IT` input. The file can be visualized by `LUSCUS` (Open source program, which can be downloaded and installed to your Linux, Windows, or MacOS workstation or laptop). By typing the command:

```
luscus CH4.lus
```

a window will be opened displaying the molecule and its charge density. By proper selection of options with the mouse buttons, the shape and size of several molecular orbitals can be visualized.

`GRID_IT` can also be run separately, if an orbital file is specified in the input, and the $WorkDir directory is available.

More information can be found in section (See online manual).

As an alternative to running a specific project, the short script provided below can be placed in the directory $MOLCAS/doc/samples/problem_based_tutorials with the name *`project.sh`*. Simply execute the shell script, project.sh $Project, where $Project is the MOLCAS input, and output files, error files, and a $WorkDir directory called $Project.work will be obtained.

```
#!/bin/bash
export MOLCAS_MEM=1Gb
export MOLCAS_PRINT=3
molcas $1 -f
exit
```

In order to run a Kohn-Sham density functional calculation, MOLCAS uses the same `SCF` module, and, therefore, the only change needed are the specification of the DFT option and required functional (e.g. B3LYP) in the `SCF` input:

```
*DFT energy for CH4 at a fixed nuclear geometry plus a grid for visualization.
*File: DFT.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 KSDFT = B3LYP
&GRID_IT
 All
```

Similar graphical files can be found in $WorkDir and $PWD.

The next step is to obtain the second-order Møller–Plesset perturbation (MP2) energy for methane at the same molecular geometry using the same one-electron basis set. Program `MBPT2` is now used, and it is possible to take advantage of having previously computed the proper integrals with `SEWARD` and the reference closed-shell HF wave function with the `SCF` program. In such cases, it is possible to keep the same definitions as before and simply prepare a file containing the `MBPT2` input and run it using the `molcas` command.

The proper intermediate file will be already in $WorkDir. On the other hand, one has to start from scratch, all required inputs should be placed sequentially in the *`MP2.energy.CH4`* file. If the decision is to start the project from the beginning, it is probably a good idea to remove the entire $WorkDir directory, unless it is known for certain the exact nature of the files contained in this directory.

```
*MP2 energy for CH4 at a fixed nuclear geometry.
*File: MP2.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
&MBPT2
 Frozen = 1
```

In addition to the calculation of a HF wave function, an MP2 calculation has been performed with a frozen deepest orbital, the carbon 1s, of CH_4. Information about the output of the `MBPT2` program can be found on section 4.6.

The `SCF` program works by default with closed-shell systems with an even number of electrons at the Restricted Hartee-Fock (RHF) level. If, instead there is a need to use the Unrestricted

Hartree Fock (UHF) method, this can be schieved by invoking the keyword UHF. This is possible for both even and odd electron systems. For instance, in a system with an odd number of electrons such as the CH_3 radical, with the following Cartesian coordinates

```
4
CH3 coordinates in Angstrom
C    0.000000     0.000000     0.000000
H    0.000000     0.000000     1.050000
H    1.037090     0.000000    -0.366667
H   -0.542115    -0.938971    -0.383333
```

the input to run an open-shell UHF calculation is easily obtained:

```
*SCF/UHF energy for CH3 at a fixed nuclear geometry
*File: SCF.energy_UHF.CH3
*
&GATEWAY
 Title = CH3 molecule
 coord = CH3.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 UHF
```

If the system is charged, this must be indicated in the `SCF` input, for example, by computing the cation of the CH_4 molecule at the UHF level:

```
*SCF/UHF energy for CH4+ at a fixed nuclear geometry
*File: SCF.energy_UHF.CH4plus
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz
 basis = STO-3G
 group = c1
&SEWARD
&SCF
 UHF
 Charge = +1
```

The Kohn-Sham DFT calculation can be also run using the UHF algorithm:

```
*DFT/UHF energy for CH4+ at a fixed nuclear geometry
*File: DFT.energy.CH4plus
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 KSDFT = B3LYP
 UHF
 Charge = +1
```

For the UHF and UHF/DFT methods it is also possible to specify α and β orbital occupations in two ways.

1. First, the keyword ZSPIN can be invoked in the `SCF` program, which represents the difference between the number of α and β electrons.

   For example, setting the keyword to 2 forces the program to converge to a result with two more α than β electrons.

   ```
   *DFT/UHF energy for different electronic occupation in CH4 at a fixed nuclear geometry
   *File: DFT.energy_zspin.CH4
   *
   &GATEWAY
    Title = CH4 molecule
    coord = CH4.xyz
    basis = STO-3G
    group = c1
   &SEWARD
   &SCF
    Title = CH4 molecule zspin 2
    UHF; ZSPIN =  2
    KSDFT =  B3LYP
   ```

   The final occupations in the output will show six α and four β orbitals.

2. Alternatively, instead of ZSPIN, it is possible to specify occupation numbers with keyword OCCUPATION at the beginning of the SCF calculation.

   This requires an additional input line containing the occupied α orbitals (e.g. 6 in this case), and a second line with the β orbitals (e.g. 4 in this case). Sometimes, SCF convergence may be improved by using this option.

Different sets of methods use other MOLCAS modules. For example, to perform a Complete Active Space (CAS) SCF calculation, the `RASSCF` program has to be used. This module requires starting trial orbitals, which can be obtained from a previous SCF calculation or, automatically, from the `SEWARD` program which provides trial orbitals by using a model Fock operator.

Recommended keywords are

- NACTEL defines the total number of active electrons, holes in Ras1, and particles in Ras3, respectively. The last two values are only for RASSCF-type calculations.
- INACTIVE indicates the number of inactive orbitals where the occupation is always 2 in the CASSCF reference, and
- RAS2 defines the number of active orbitals.

  By default, the wave function for the lowest state corresponds to the symmetry with spin multiplicity of 1. Most of the input may not be necessary, if one has prepared and linked an INPORB file with the different orbital types defined by a program like `LUSCUS`.

```
*CASSCF energy for CH4 at a fixed nuclear geometry
*File: CASSCF.energy.CH4
*
&GATEWAY
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
```

```
&RASSCF
 Title = CH4 molecule
 Spin = 1; Nactel = 8 0 0; Inactive = 1; Ras2 = 8
&GRID_IT
 All
```

In this case, the lowest singlet state (i.e. the ground dstate) is computed, since this is a closed-shell situation with an active space of eight electrons in eight orbitals and with an inactive C 1s orbital, the lowest orbital of the CH_4 molecule. This is a CASSCF example in which all the valence orbitals and electrons (C 2s, C 2p and 4 x H 1s) are included in the active space and allows complete dissociation into atoms. If this is not the goal, then the three almost degenerate highest energy occupied orbitals and the corresponding antibonding unoccupied orbitalsmust be active, leading to a 6 in 6 active space.

Using the CASSCF wave function as a reference, it is possible to perform a second-order perturbative, CASPT2, correction to the electronic energy by employing the `CASPT2` program. If all previously calculated files are retained in the $WorkDir directory, in particular, integral files (*`CH4.OneInt`*,*`CH4.OrdInt`*), the CASSCF wave function information file (*`CH4.JobIph`*), and communication file *`CH4.RunFile`*), it will not be necessary to re-run programs `SEWARD`, and `RASSCF`. In this case case, it is enough to prepare a file containing input only for the `CASPT2` program followed be execution. Here, however, for the sake of completness, input to all MOLCAS moddules is provided:

```
*CASPT2 energy for CH4 at a fixed nuclear geometry
*File: CASPT2.energy.CH4
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   LumOrb
   Spin = 1; Nactel = 8 0 0; Inactive = 1; Ras2 = 8
&CASPT2
 Multistate = 1 1
```

In most of casesi, the Hartree-Fock orbitals will be a better choice as starting orbitals. In that case, the `RASSCF` input has to include keyword LUMORB to read from any external source of orbitals other than those generated by the `SEWARD` program. By modifying input to the `SCF` program, it is possible to generate alternative trial orbitals for the `RASSCF` program. Since a new set of trial orbitals is used, the input to the `RASSCF` program is also changed. Now, the number of active orbitals, as well as the number of active electrons, are 6.

The two lowest orbitals (INACTIVE 2) are excluded from the active space and one other orbital is placed in the secondary space. If the previous (8,8) full valence space was used, the `CASPT2` program would not be able to include more electronic correlation energy, considering that the calculation involves a minimal basis set. The input for the `CASPT2` program includes a frozen C 1s orbital, the lowest orbital in the CH_4 molecule.

The charge and multiplicity of our wave function can be changed by computing the CH_4^+ cation with the same methods. The `RASSCF` program defines the character of the problem by specifying the number of electrons, the spin multiplicity, and the spatial symmetry. In the example below, there is one less electron giving rise to doublet multiplicity:

```
*CASSCF energy for CH4+ at a fixed nuclear geometry
*File: CASSCF.energy.CH4plus
```

```
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz; basis = STO-3G; Group = C1
&SEWARD
&RASSCF
   Spin = 2; Nactel = 7 0 0; Inactive = 1; Ras2 = 8
```

No further modification is needed to the `CASPT2` input:

```
*CASPT2 energy for CH4+ at a fixed nuclear geometry
*File: CASPT2.energy.CH4plus
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   Title = CH4+ molecule
   Spin = 2; Nactel = 1 0 0; Inactive = 4; Ras2 = 1
&CASPT2
```

A somewhat more sophisticated calculation can be performed at the Restricted Active Space (RAS) SCF level. In such a situation, the level of excitation in the CI expansion can be controlled by restricting the number of holes and particles present in certain orbitals.

```
*RASSCF energy for CH4 at a fixed nuclear geometry
*File: RASSCF.energy.CH4
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   Title = CH4 molecule
   Spin = 1; Nactel = 8 1 1
   Inactive = 1; Ras1 = 1; Ras2 = 6; Ras3 = 1
```

In particular, the previous calculation includes one orbital within the Ras1 space and one orbital within the Ras3 space. One hole (single excitation) at maximum is allowed from Ras1 to Ras2 or Ras3, while a maximum of one particle is allowed in Ras3, derived from either Ras1 or Ras2. Within Ras2, all types of orbital occupations are allowed. The RASSCF wave functions can be used as reference for multiconfigurational perturbation theory (RASPT2), but this approach has not been as extensively tested as CASPT2, and, so experience is still somewhat limited.

MOLCAS also has the possibility of computing electronic energies at different CI levels by using the `MRCI` program. The input provided below involves a Singles and Doubles Configuration Interaction (SDCI) calculation on the CH_4 molecule. To set up the calculations, program `MOTRA` which transforms the integrals to molecular basis, and program `GUGA` which computes the coupling coefficients, must be run before the `MRCI` program. In `MOTRA` the reference orbitals are specifiedi, and those employed here are from an HF `SCF` calculation including frozen orbitals. In `GUGA` the reference for the CI calculation is described by the number of correlated electrons, the spatial and spin symmetry, the inactive orbitals always occupation 2 in the reference space, and the type of CI expansion.

```
*SDCI energy for CH4 at a fixed nuclear geometry
*File: SDCI.energy.CH4
*
&GATEWAY
```

```
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons = 8
 Spin = 1
 Inactive= 4
 Active= 0
 Ciall= 1
&MRCI
 SDCI
```

To use reference orbitals from a previous CASSCF calculation, the `RASSCF` program will have to be run before the `MOTRA` module. Also, if the spatial or spin symmetry are changed for the CI calculation, the modifications will be introduced in the input to `GUGA` program. Many alternatives are possible for performing an MRCI calculation as shown in the next example below, in which the reference space to perform the CI is multiconfigurational:

```
*MRCI energy for CH4 at a fixed nuclear geometry
*File: MRCI.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 2; Ras2= 6
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons= 8
 Spin= 1
 Inactive= 2
 Active= 3
 Ciall= 1
&MRCI
 SDCI
```

The `MRCI` program also allows the calculation of electronic energies using the ACPF method. Another MOLCAS program, `CPF`, offers the possibility to use the CPF, MCPF, and ACPF methods with a single reference function. The required input is quite similar to that for the `MRCI` program:

```
*CPF energy for CH4 at a fixed nuclear geometry
*File: CPF.energy.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons= 8
```

```
  Spin = 1
  Inactive = 4
  Active = 0
  Ciall= 1
&CPF
  CPF
End Of Input
```

Finally, MOLCAS can also perform closed- and open-shell coupled cluster calculations at the CCSD and CCSD(T) levels. These calculations are controlled by the `CCSDT` program, whose main requirement is that the reference function has to be generated with the `RASSCF` program. The following input is required to obtain a CCSD(T) energy for the CH_4 molecule:

```
*CCSDT energy for CH4 at a fixed nuclear geometry
*File: CCSDT.energy.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&RASSCF
 Spin= 1; Nactel= 0 0 0; Inactive= 5; Ras2= 0
 OutOrbitals
 Canonical
&MOTRA
 JobIph
 Frozen= 1
&CCSDT
 CCT
```

Since this is a closed-shell calculation, the `RASSCF` input computes a simple RHF wave function with zero active electrons and orbitals using keywords OUTORBITALS and CANONICAL. The `MOTRA` must include the keyword JOBIPH to extract the wave function information from file *JOBIPH* which is automatically generated by `RASSCF`. Finally, the keywork CCT in program `CCSDT` leads to the calculation of the CCSD(T) energy using the default algorithms.

The `CCSDT` program in MOLCAS is especially suited to compute open-shell cases. The input required to obtain the electronic energy of the CH_4^+ cation with the CCSD(T) method is:

```
*CCSDT energy for CH4+ at a fixed nuclear geometry
*File: CCSDT.energy.CH4plus
*
&GATEWAY
 Title= CH4+ molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&RASSCF
 Spin= 2; Nactel= 1 0 0; Inactive= 4; Ras2= 1
 OutOrbitals
 Canonical
&MOTRA
 JobIph
 Frozen= 1
&CCSDT
 CCT
```

where the `RASSCF` program generated the proper Restricted Open-Shell Hartree-Fock (ROHF) reference. Different levels of spin adaptation are also available.

If solvent effects are desired, MOLCAS includes two models: Kirkwood and PCM. Adding a solvent effect to a ground state at HF, DFT, or CASSCF levels, simply requires the inclusion of the keyword RF-INPUT within the input for the `SEWARD` which calculates a self-consistend reaction field.

```
*DFT energy for CH4 in water at a fixed nuclear geometry
*File: DFT.energy_solvent.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
 RF-input
   PCM-model; solvent= water
 End of RF-input
&SEWARD
&SCF
KSDFT= B3LYP
```

Other programs such as `CASPT2`, `RASSI`, and `MOTRA` require that the reaction field is included as a perturbation with keyword RFPERTURBATION. In the next example the correction is added at both the CASSCF and CASPT2 levels.

```
*CASPT2 energy for CH4 in acetone at a fixed nuclear geometry
*File: CASPT2.energy_solvent.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
  RF-input
   PCM-model; solvent= acetone; AAre= 0.2
  End of RF-input
&SEWARD
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 2; Ras2= 6
&CASPT2
 Frozen= 1
 Multistate= 1 1
 RFPert
```

Notice that the tesserae of the average area in the PCM model (keyword has been changed to the value required for acetone by the keyword AARE, while the default is 0.4 Å^2 for water

3.2 Optimizing geometries

It is now useful to explore potential energy surfaces (PES) and optimize the molecular geometry for specific points along the PES. Different cases are discussed including a way to obtain the optimal geometry in a minimum energy search, to obtain a transition-state structure connecting different regions of the PES, to find the crossing between two PES where the energy becomes degenerate, or to map the minimum steepest-descent energy path (MEP) from an initial point to the final a minimum energy geometry as the PES progresses in a downward manner.

All these types of searches can be performed either by fully optimizing all degrees of freedom of the system or by introducing certain restrictions. MOLCAS 8.2 can perform geometry optimizations at the SCF (RHF and UHF), DFT (RHF and UHF based), CASSCF (CASSCF and RASSCF) levels of theory, where efficient analytical gradients are available and at the CASPT2 and other correlated levels where numerical gradients are used.

Geometry optimizations require many cycles, in which the electronic energy is estimated at a specific level of calculation followed by calculation of the gradient of the energy with respect to the geometric degrees of freedom (DOF). With this information at hand, the program must decide if the molecule is already at the final required geometry (i.e. gradient ∼ 0 for all DOF) indicating a minimum in the PES or if the geometry must be modified and continue the cycle. The input file should, therefore, be built in a way that allows a loop over the different programs.

The general input commands Do while and Enddo control the loop and program input is inserted within these commands. Instructions for the number of maximum iterations allowed and the type of output required can also be added.

All examples previously discussed, use COORD keyword, but it also possible to use *native format*, where symmetry unique atoms are specified (SYMMETRY) and provide generators to construct all atoms in the molecule.

The selected example describes geometry optimization of the water molecule at the SCF RHF level of calculation:

```
3
 coordinates for water molecule NOT in equilibrium
O 0.000000  0.000000  0.000000
H 0.758602  0.000000  0.504284
H 0.758602  0.000000 -0.504284
```

```
*SCF minimum energy optimization for H2O
*File: SCF.minimum_optimization.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Do while
 &SEWARD ;&SCF; &SLAPAF
>>> EndDo
```

The sequence of programs employed includes `GATEWAY` which is external to the loop, followed by `SEWARD`, `SCF`, and `SLAPAF`. `SEWARD` computes the integrals, `SCF` program computes the RHF energy and wave function, and `SLAPAF` will control the calculation of gradients and estimate if the calculation has already finished or needs to proceed to a new nuclear geometry for the next iteration. Automatically, a file named *`$Project.geo.molden`* will be generated in $WorkDir containing all the geometric steps contained in the optimization process. `MOLDEN` or `LUSCUS` can then read this file to display the individual molecular geometries which form the optimization cycle.

Using another reference wave function can be simply performed by changing the sequence of programs. For instance, we can perform an UHF calculation of the H_2O^+ cation:

```
*UHF minimum energy optimization for H2O+
*File: UHF.minimum_optimization.H2Oplus
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
```

```
>> Do while
 &SEWARD
 &SCF; Title="H2O minimum optimization"; UHF; Charge=1
 &SLAPAF
>> EndDo
```

The same procedure can be followed if we pretend to perform a DFT geometry optimization:

```
*DFT minimum energy optimization for H2O
*File: DFT.minimum_optimization.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Export MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &SCF ; Title="H2O minimum optimization"; KSDFT=B3LYP
 &SLAPAF &END
>>> EndDo
```

Once an energy minimum is found based on the calculation of gradients, it is necessary to ensure that the geometry really is a minimum energy point. This can be only accomplished by computing second derivatives of the energy (i.e. the Hessian). MOLCAS can compute analytical Hessians for SCF and single state CASSCF wave functions. For other methods, numerical procedures can be used to compute the Hessian. Once the Hessian is computed, vibrational frequencies are calculated, and Statistical Mechanics is used to obtain thermodynamic properties. At a true energy minimum, there will be 3N-6 real frequencies Program `MCKINLEY` computes second derivatives of a predefined (SCF or CASSCF) wave function, while `MCLR` performs the vibrational and statistical analyses. MOLCAS simply requires input for the `MCKINLEY` program to perform the entire calculation by using keywords PERTURBATION and HESSIAN, while program `MCLR` will be called automatically but requires no input. The full set of calculationsi is included below first a geometry optimization followed by the calculation of a Hessian.

```
*SCF minimum energy optimization plus hessian of the water molecule
*File: SCF.minimization_plus_hessian.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Export MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &SCF; Title="H2O minimum optimization"
 &SLAPAF
>>> EndDo
&MCKINLEY
```

Note that `MCKINLEY` input above is placed after EndDo, and, therefore, is external to the looping scheme. Once the geometry optimization at the desired level of theory has finished, the Hessian will be computed at the final geometry. In general, any calculation performed using a $WorkDir directory where a previous geometry optimization has taken place will use the last geomtry calculated from that optimization as the input geometry even if `SEWARD` input is present. To avoid that, the only solution is to remove the communication file ***RUNFILE*** where the geometry is stored. Note also, that the frequencies are computed in a cartesian basis, and that three translational and three rotational frequencies which should be very close to zero are included in the output file. This is not the case when numerical gradients and Hessians are used. In particular, for water at its minimum energy structure three (3N-6) real vibrational frequencies. By default, in $WorkDir a file ***$Project.freq.molden*** is generated containing the vibrational frequencies and modes, which can be visualized by `MOLDEN`.

A new level of theory, CASSCF, is introduced here which is especially suited for geometry optimizations of excited states discussed in the next chapter. A geometry optimization is performed to illustrate a broader range of possibilities including the imposition of a geometric restrain that the HOH angle in water should be constrained to 120^o during the optimization. This means that only the O-H bond distances be optimized in this partial minimization. The restriction is indicated in in `GATEWAY` by invoking the keyword CONSTRAINTS and ending with the keyword END OF CONSTRAINTS. The names of variables corresponding to geometrical variables in either internal or Cartesian coordinates that are to be constrained are placed between these two keywords. In the case of H_2O, the H1-O-H2 angle is fixed at 120^o, so a variable, a, is first defined with the keywork (Angle), which relates it to the H1-O1-H2 angle, followed by the second keyword, VALUE, where the variable a is specified as 120^o. It is not required that the initial geometry is 120^o, only that the final result for the calculation will become 120^o.

Note that the `RASSCF` program requires initial trial orbitals, and those which are automatically generated by `SEWARD` are used. The resulting CASSCF wave function includes all valence orbitals and electrons.

```
*CASSCF minimum energy optimization of the water molecule with geometrical restrictions
*File: CASSCF.minimum_optimization_restricted.H2O
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
Constraint
   a = Angle H2 O1 H3
  Value
   a = 90. degree
End of Constraints
>>> Do while
 &SEWARD
 &RASSCF; nActEl=8 0 0; Inactive=1; Ras2=6
 &SLAPAF
>>> EndDo
```

Other more flexible ways to impose geometric restrictions involve the specification of which internal coordinates should remain fixed and which should change. In the next example, the bond lengths are forced to remain fixed at their initial distance (here 0.91 Å), while the bond angle, having an initial of 81°, is optimized.

```
*DFT minimum energy optimization of the angle in the water molecule at fixed bond lengths
*File: DFT.minimum_optimization_restricted.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
```

```
>>> EXPORT MOLCAS_MAXITER=100
>>> Do while
 &SEWARD; &SCF; Title="H2O restricted minimum"; KSDFT=B3LYP
 &SLAPAF
  Internal Coordinates
     b1 = Bond O1 H2
     b2 = Bond O1 H3
     a1 = Angle H2 O1 H3
  Vary
     a1
  Fix
     b1
     b2
  End of Internal
>>> EndDo
```

In the final output, the two O-H bond lengths remain at the initia values, while the H1-O1=H2 angle is optimized to a final angle of 112^o.

The next step entails the computation of a transition state, a structure connecting different regions of the potential energy hypersurface, and is a maximum for only one degree of freedom. The most common saddle points have order one, that is, they are maxima for one of one displacement and minima for the others. The simplest way to search for a transition state in MOLCAS is to add the keyword TS to the `SLAPAF` input. Keyword PRFC is suggested in order to verify the nature of the transition structure. Searching for transition states is, however, not an easy task. An illustration of the input required for transition state optimization for water at the DFT level is given below:

```
3
water in Transition state in bohr
O1             0.750000        0.000000         0.000000
H2             1.350000        0.000000         1.550000
H3             1.350000        0.000000        -1.550000


*DFT transition state optimization of the water molecule
*File: DFT.transition_state.H2O
*
&Gateway
 Coord=Water_TS.xyz
 Basis=ANO-S-VDZ
 Group=C1
>>> Do while
 &SEWARD
 &SCF; Title="H2O TS optimization"; KSDFT=B3LYP
 &SLAPAF ; ITER=20 ; TS
>>> EndDo
```

The initial coordinates were chosen in units of Bohr, to illustrare that this is the default case. The optimal geometry for ground state of water is a structure with C_{2v} symmetry. A transition state has been found with a linear HOH angle of 180^o. In many cases, there may be a clue along the energy pathway for a chemical reaction about the nature of the transition state structure, which typically represents an intermediate conformation between reactants and products. If this turns out to be the case, it is possible to help the optimization process proceed toward an informed guess, by invoking the keyword FINDTS in `SLAPAF`. FINDTS must to be accompanied with a definition of constrained geometric definitions. `SLAPAF` will guide the optimization of the transition state towards a region in which the restriction is fulfilled. Once there, the restriction will be released and a free search of the transition state will be performed. This technique is frequently quite effective and makes it possible to find difficult transition states or reduce the number of required iterations. Here, an example is provided, in which the initial geometry of water is clearly bent, and a trial restraint is imposed such that the angle for the transition state should be near 180^o. The final transition state will, however, be obtained without any type of geometrical restriction.

```
*DFT transition state optimization of the water molecule with geometrical restrictions
*File: DFT.transition_state_restricted.H2O
*
&Gateway
 Coord=Water_TS.xyz
 Basis=ANO-S-VDZ
 Group=C1
 Constraints
   a = Angle H2 O1 H3
 Value
   a = 180.0 degree
 End of Constraints
>>> Do while
 &SEWARD
 &SCF; Title="H2O TS optimization"; KSDFT=B3LYP
 &SLAPAF ;FindTS
>>> EndDo
```

The `CASPT2` geometry optimizations are somewhat different because `ALASKA` is not suited to compute `CASPT2` analytical gradients. Therefore the `ALASKA` program is automatically substituted by program `NUMERICAL_GRADIENT`, which will take care of performing numerical gradients. From the user point of view the only requirement is to place the `CASPT2` input after the `RASSCF` input. The CASSCF wave function has of course to be generated in each step before performing CASPT2. To compute a numerical gradient can be quite time consuming, although it is a task that can be nicely parallelized. In a double-sided gradient algorithm like here a total of 6N-12+1 CASPT2 calculations are performed each pass of the optimization, where N is the number of atoms.

```
*CASPT2 minimum energy optimization for water
*File: CASPT2.minimum_optimization.H2O
*
&GATEWAY
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Do while
 &SEWARD
 &RASSCF; Title="H2O restricted minimum"; nActEl=8 0 0; Inactive=1; Ras2=6
 &CASPT2; Frozen=1
 &SLAPAF
>>> EndDo
```

The use of spatial symmetry makes the calculations more efficient, although they may again complicate the preparation of input files. We can repeat the previous `CASPT2` optimization by restricting the molecule to work in the C_{2v} point group, which, by the way, is the proper symmetry for water in the ground state. The `GATEWAY` program (as no symmetry has been specified) will identify and work with the highest available point group, C_{2v}. Here the molecule is placed with YZ as the molecular plane. By adding keyword SYMMETRY containing as elements of symmetry the YZ (symbol X) and YX (symbol Z), the point group is totally defined and the molecule properly generated. From that point the calculations will be restricted to use symmetry restrictions. For instance, the molecular orbitals will be classified in the four elements of symmetry of the group, a_1, b_1, b_2, and a_2, and most of the programs will require to define the selection of the orbitals in the proper order. The order of the symmetry labels is determined by `SEWARD` and must be checked before proceeding, because from that point the elements of symmetry will be known by their order in `SEWARD`: a_1, b_1, b_2, and a_2, for instance, will be symmetries 1, 2, 3, and 4, respectively. `SCF` does not require to specify the class of orbitals and it can be used as a learning tool.

```
*CASPT2 minimum energy optimization for water in C2v
*File: CASPT2.minimum_optimization_C2v.H2O
*
 &GATEWAY
Title= H2O caspt2 minimum optimization
Symmetry= X Z
Basis set
O.ANO-S...2s1p.
O        0.000000  0.000000  0.000000 Angstrom
End of basis
Basis set
H.ANO-S...1s.
H1       0.000000  0.758602  0.504284 Angstrom
End of basis
>>> EXPORT MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &RASSCF; nActEl=8 0 0; Inactive=1 0 0 0; Ras2=3 1 2 0
 &CASPT2; Frozen=1 0 0 0
 &SLAPAF &END
>>> EndDo
```

Thanks to symmetry restrictions the number of iterations within `NUMERICAL_GRADIENT` has been reduced to five instead of seven, because many of the deformations are redundant within the C_{2v} symmetry. Also, symmetry considerations are important when defining geometrical restrictions (see online manual).

3.3 Computing excited states

The calculation of electronic excited states is typically a multiconfigurational problem, and therefore it should preferably be treated with multiconfigurational methods such as CASSCF and CASPT2. We can start this section by computing the low-lying electronic states of the acrolein molecule at the CASSCF level and using a minimal basis set. The standard file with cartesian coordinates is:

```
 8
Angstrom
 O      -1.808864   -0.137998    0.000000
 C       1.769114    0.136549    0.000000
 C       0.588145   -0.434423    0.000000
 C      -0.695203    0.361447    0.000000
 H      -0.548852    1.455362    0.000000
 H       0.477859   -1.512556    0.000000
 H       2.688665   -0.434186    0.000000
 H       1.880903    1.213924    0.000000
```

We shall carry out State-Averaged (SA) CASSCF calculations, in which one single set of molecular orbitals is used to compute all the states of a given spatial and spin symmetry. The obtained density matrix is the average for all states included, although each state will have its own set of optimized CI coefficients. Different weights can be considered for each of the states, but this should not be used except in very special cases by experts. It is better to let the CASPT2 method to handle that. The use of a SA-CASSCF procedure has an great advantage. For example, all states in a SA-CASSCF calculation are orthogonal to each other, which is not necessarily true for state specific calculations. Here, we shall include five states of singlet character the calculation. As no symmetry is invoked all the states belong by default to the first symmetry, including the ground state.

```
*CASSCF SA calculation on five singlet excited states in acrolein
*File: CASSCF.excited.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord = acrolein.xyz; basis = STO-3G; group = c1
&SEWARD; &SCF
&RASSCF
  LumOrb
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 5 5 1
&GRID_IT
  All
```

We have used as active all the π and π^* orbitals, two bonding and two antibonding π orbitals with four electrons and in addition the oxygen lone pair (n). Keyword CiRoot informs the program that we want to compute a total of five states, the ground state and the lowest four excited states at the CASSCF level and that all of them should have the same weight in the average procedure. Once analyzed we find that the calculation has provided, in this order, the ground state, two $n \rightarrow \pi^*$ states, and two $\pi \rightarrow \pi^*$ states. It is convenient to add the `GRID_IT` input in order to be able to use the `LUSCUS` interface for the analysis of the orbitals and the occupations in the different electronic states. Such an analysis should always be made in order to understand the nature of the different excited states. In order to get a more detailed analysis of the nature of the obtained states it is also possible to obtain in a graphical way the charge density differences between to states, typically the difference between the ground and an excited state. The following example creates five different density files:

```
*CASSCF SA calculation on five singlet excited states in acrolein
*File: CASSCF.excited_grid.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord= acrolein.xyz; basis= STO-3G; group= c1
&SEWARD; &SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
 OutOrbital
 Natural= 5
&GRID_IT
 FILEORB = $Project.RasOrb.1
 NAME = 1; All
&GRID_IT
 FILEORB = $Project.RasOrb.2
 NAME = 2; All
&GRID_IT
 FILEORB = $Project.RasOrb.3
 NAME = 3; All
&GRID_IT
 FILEORB = $Project.RasOrb.4
 NAME = 4; All
&GRID_IT
```

```
  FILEORB = $Project.RasOrb.5
  NAME = 5; All
```

In `GRID_IT` input we have included all orbitals. It is, however, possible and in general recommended to restrict the calculation to certain sets of orbitals. How to do this is described in the input manual for `GRID_IT`.

Simple math operations can be performed with grids of the same size, for example, `LUSCUS` can be used to display the difference between two densities.

CASSCF wave functions are typically good enough, but this is not the case for electronic energies, and the dynamic correlation effects have to be included, in particular here with the CASPT2 method. The proper input is prepared, again including `SEWARD` and `RASSCF` (unnecessary if they were computed previously), adding a `CASPT2` input with the keyword MULTISTATE set to 5 1 2 3 4 5. The `CASPT2` will perform four consecutive single-state (SS) CASPT2 calculations using the SA-CASSCF roots computed by the `RASSCF` module. At the end, a multi-state CASPT2 calculation will be added in which the five SS-CASPT2 roots will be allowed to interact. The final MS-CASPT2 solutions, unlike the previous SS-CASPT2 states, will be orthogonal. The FROZEN keyword is put here as a reminder. By default the program leaves the core orbitals frozen.

```
*CASPT2 calculation on five singlet excited states in acrolein
*File: CASPT2.excited.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group= c1
&SEWARD; &SCF
&RASSCF
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&GRID_IT
 All
&CASPT2
 Multistate= 5 1 2 3 4 5
 Frozen= 4
```

Apart from energies and state properties it is quite often necessary to compute state interaction properties such as transition dipole moments, Einstein coefficients, and many other. This can be achieved with the `RASSI` module, a powerful program which can be used for many purposes (see online manual) . We can start by simply computing the basic interaction properties

```
*RASSI calculation on five singlet excited states in acrolein
*File: RASSI.excited.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group = c1
&SEWARD; &SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&CASPT2
 Frozen = 4
 MultiState= 5 1 2 3 4 5
>>COPY $Project.JobMix JOB001
&RASSI
```

```
Nr of JobIph
1 5
1 2 3 4 5
EJob
```

Oscillator strengths for the computed transitions and Einstein coefficients are compiled at the end of the `RASSI` output file. To obtain these values, however, energy differences have been used which are obtained from the previous CASSCF calculation. Those energies are not accurate because they do not include dynamic correlation energy and it is better to substitute them by properly computed values, such those at the CASPT2 level. This is achieved with the keyword EJOB.

Now a more complex case. We want to compute vertical singlet-triplet gaps from the singlet ground state of acrolein to different, up to five, triplet excited states. Also, interaction properties are requested. Considering that the spin multiplicity differs from the ground to the excited states, the spin Hamiltonian has to be added to our calculations and the `RASSI` program takes charge of that. It is required first, to add in the `SEWARD` input the keyword AMFI, which introduces the proper integrals required, and to the `RASSI` input the keyword SPINORBIT. Additionally, as we want to perform the calculation sequentially and `RASSI` will read from two different wave function calculations, we need to perform specific links to save the information. The link to the first `CASPT2` calculation will saved in file *`$Project.JobMix.S`* the data from the `CASPT2` result of the ground state, while the second link before the second `CASPT2` run will do the same for the triplet states. Later, we link these files as *`JOB001`* and *`JOB002`* to become input files for `RASSI`. In the `RASSI` input NROFJOBIPH will be set to two, meaning two *`JobIph`* or *`JobMix`* files, the first containing one root (the ground state) and the second five roots (the triplet states). Finally, we have added EJOB, which will read the CASPT2 (or MS-CASPT2) energies from the *`JobMix`* files to be incorporated to the `RASSI` results. The magnitude of properties computed with spin-orbit coupling (SOC) depends strongly on the energy gap, and this has to be computed at the highest possible level, such as CASPT2.

```
*CASPT2/RASSI calculation on singlet-triplet gaps in acrolein
*File: CASPT2.S-T_gap.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group= c1
 AMFI
&SEWARD
&SCF
&RASSCF
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 1 1 1
&CASPT2
 Frozen= 4
 MultiState= 1 1
>>COPY $Project.JobMix JOB001
&RASSCF
 LumOrb
 Spin= 3; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&CASPT2
 Frozen= 4
 MultiState= 5 1 2 3 4 5
>>COPY $Project.JobMix JOB002
&RASSI
 Nr of JobIph= 2 1 5; 1; 1 2 3 4 5
 Spin
```

```
 EJob
```

As here with keyword AMFI, when using command Coord to build a `SEWARD` input and we want to introduce other keywords, it is enough if we place them after the line corresponding to Coord. Observe that the nature of the triplet states obtained is in sequence one $n\pi^*$, two $\pi\pi^*$, and two $n\pi^*$. The `RASSI` output is somewhat complex to analyze, but it makes tables summarizing oscillator strengths and Einstein coefficients, if those are the magnitudes of interest. Notice that a table is first done with the spin-free states, while the final table include the spin-orbit coupled eigenstates (in the CASPT2 energy order here), in which each former triplet state has three components.

In many cases working with symmetry will help us to perform calculations in quantum chemistry. As it is a more complex and delicate problem we direct the reader to the examples section in this manual. However, we include here two inputs that can help the beginners. They are based on trans-1,3-butadiene, a molecule with a C_{2h} ground state. If we run the next input, the `SEWARD` and `SCF` outputs will help us to understand how orbitals are classified by symmetry, whereas reading the `RASSCF` output the structure of the active space and states will be clarified.

```
*CASSCF SA calculation on 1Ag excited states in tButadiene
*File: CASSCF.excited.tButadiene.1Ag
*
&SEWARD
  Title= t-Butadiene molecule
  Symmetry= Z XYZ
Basis set
C.STO-3G...
C1   -3.2886930 -1.1650250 0.0000000  Bohr
C2   -0.7508076 -1.1650250 0.0000000  Bohr
End of basis
Basis set
H.STO-3G...
H1   -4.3067080  0.6343050 0.0000000  Bohr
H2   -4.3067080 -2.9643550 0.0000000  Bohr
H3    0.2672040 -2.9643550 0.0000000  Bohr
End of basis
&SCF
&RASSCF
 LumOrb
 Title= tButadiene molecule (1Ag states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 1; Nactel= 4 0 0; Inactive= 7 0 6 0; Ras2= 0 2 0 2
 CiRoot= 4 4 1
&GRID_IT
 All
```

Using the next input will give information about states of a different symmetry. Just run it as a simple exercise.

```
*CASSCF SA calculation on 1Bu excited states in tButadiene
*File: CASSCF.excited.tButadiene.1Bu
*
&SEWARD
 Title= t-Butadiene molecule
 Symmetry= Z XYZ
Basis set
C.STO-3G...
C1   -3.2886930 -1.1650250 0.0000000  Bohr
C2   -0.7508076 -1.1650250 0.0000000  Bohr
End of basis
```

```
Basis set
H.STO-3G...
H1   -4.3067080  0.6343050 0.0000000  Bohr
H2   -4.3067080 -2.9643550 0.0000000  Bohr
H3    0.2672040 -2.9643550 0.0000000  Bohr
End of basis
&SCF
&RASSCF
 FileOrb= $Project.ScfOrb
 Title= tButadiene molecule (1Bu states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 1; Nactel= 4 0 0; Inactive= 7 0 6 0
 Ras2= 0 2 0 2
 CiRoot= 4 4 1
>COPY $Project.RasOrb $Project.1Ag.RasOrb
>COPY $Project.JobIph JOB001
&GRID_IT
 Name= $Project.1Ag.lus
 All
&RASSCF
 FileOrb= $Project.ScfOrb
 Title= tButadiene molecule (1Bu states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 3; Nactel= 4 0 0; Inactive= 7 0 6 0; Ras2= 0 2 0 2
 CiRoot= 2 2 1
>COPY $Project.RasOrb $Project.1Bu.RasOrb
>COPY $Project.JobIph JOB002
&GRID_IT
 Name= $Project.1Bu.lus
 All
&RASSI
 NrofJobIph= 2 4 2; 1 2 3 4; 1 2
```

Structure optimizations can be also performed at the CASSCF, RASSCF or CASPT2 levels. Here we shall optimize the second singlet state in the first (here the only) symmetry for acrolein at the SA-CASSCF level. It is strongly recommended to use the State-Average option and avoid single state CASSCF calculations for excited states. Those states are non-orthogonal with the ground state and are typically heavily contaminated. The usual set of input commands will be prepared, with few changes. In the `RASSCF` input two states will be simultaneously computed with equal weight (CiRoot 2 2 1), but, in order to get accurate gradients for a specific root (not an averaged one), we have to add Rlxroot and set it to two, which is, among the computed roots, that we want to optimize. The proper density matrix will be stored. The `MCLR` program optimizes, using a perturbative approach, the orbitals for the specific root (instead of using averaged orbitals), but the program is called automatically and no input is needed.

```
*CASSCF excited state optimization in acrolein
*File: CASSCF.excited_state_optimization.acrolein
*
 &GATEWAY
Title= acrolein minimum optimization in excited state 2
Basis set
O.STO-3G...2s1p.
O1       1.608542      -0.142162       3.240198 Angstrom
End of basis
Basis set
C.STO-3G...2s1p.
C1      -0.207776       0.181327      -0.039908 Angstrom
C2       0.089162       0.020199       1.386933 Angstrom
C3       1.314188       0.048017       1.889302 Angstrom
End of basis
Basis set
H.STO-3G...1s.
H1       2.208371       0.215888       1.291927 Angstrom
```

```
H2      -0.746966      -0.173522       2.046958 Angstrom
H3      -1.234947       0.213968      -0.371097 Angstrom
H4       0.557285       0.525450      -0.720314 Angstrom
End of basis
>>> Do while
 &SEWARD
>>> If ( Iter = 1 ) <<<
 &SCF
Title= acrolein minimum optimization
>>> EndIf <<<
 &RASSCF
LumOrb
Title= acrolein
Spin= 1; nActEl= 4 0 0; Inactive= 13; Ras2= 4
CiRoot= 2 2 1
Rlxroot= 2
 &SLAPAF
>>> EndDo
```

In case of performing a `CASPT2` optimization for an excited state, still the SA-CASSCF approach can be used to generate the reference wave function, but keyword RLXROOT and the use of the `MCLR` program are not necessary, because `CASPT2` takes care of selecting the proper root (the last one).

A very useful tool recently included in MOLCAS is the possibility to compute minimum energy paths (MEP), representing steepest descendant minimum energy reaction paths which are built through a series of geometry optimizations, each requiring the minimization of the potential energy on a hyperspherical cross section of the PES centered on a given reference geometry and characterized by a predefined radius. One usually starts the calculation from a high energy reference geometry, which may correspond to the Franck-Condon (FC) structure on an excited-state PES or to a transition structure (TS). Once the first lower energy optimized structure is converged, this is taken as the new hypersphere center, and the procedure is iterated until the bottom of the energy surface is reached. Notice that in the TS case a pair of steepest descent paths, connecting the TS to the reactant and product structures (following the forward and reverse orientation of the direction defined by the transition vector) provides the minimum energy path (MEP) for the reaction. As mass-weighted coordinates are used by default, the MEP coordinate corresponds to the so-called Intrinsic Reaction Coordinates (IRC). We shall compute here the MEP from the FC structure of acrolein along the PES of the second root in energy at the CASSCF level. It is important to remember that the CASSCF order may not be accurate and the states may reverse orders at higher levels such as CASPT2.

```
*CASSCF excited state mep points in acrolein
*File: CASSCF.mep_excited_state.acrolein
*
 &GATEWAY
Title = acrolein mep calculation root 2
Basis set
O.STO-3G...2s1p.
 O1    1.367073     0.000000     3.083333 Angstrom
End of basis
Basis set
C.STO-3G...2s1p.
 C1    0.000000     0.000000     0.000000 Angstrom
 C2    0.000000     0.000000     1.350000 Angstrom
 C3    1.367073     0.000000     1.833333 Angstrom
End of basis
Basis set
H.STO-3G...1s.
```

```
  H1    2.051552     0.000000     0.986333 Angstrom
  H2   -0.684479     0.000000     2.197000 Angstrom
  H3   -1.026719     0.000000    -0.363000 Angstrom
  H4    0.513360     0.889165    -0.363000 Angstrom
 End of basis
 >>> EXPORT MOLCAS_MAXITER=300
 >>> Do while
  &SEWARD
 >>> If ( Iter = 1 ) <<<
  &SCF
 >>> EndIf <<<
  &RASSCF
    Title="acrolein mep calculation root 2"; Spin=1
    nActEl=4 0 0; Inactive=13; Ras2=4; CiRoot=2 2 1; Rlxroot=2
  &SLAPAF
    MEP-search
    MEPStep=0.1
 >>> EndDo
```

As observed, to prepare the input for the MEP is simple, just add the keyword MEP-SEARCH and specify a step size with MEPSTEP, and the remaining structure equals that of a geometry optimization. The calculations are time consuming, because each point of the MEP (four plus the initial one obtained here) is computed through a specific optimization. A file named ***$Project.mep.molden*** (read by `MOLDEN`) will be generated in $WorkDir containing only those points belonging to the MEP.

We shall now show how to perform geometry optimizations under nongeometrical restrictions, in particular, how to compute hypersurface crossings, which are key structures in the photophysics of molecules. We shall get those points as minimum energy crossing points in which the energy of the highest of the two states considered is minimized under the restriction that the energy difference with the lowest state should equal certain value (typically zero). Such point can be named a minimum energy crossing point (MECP). If a further restriction is imposed, like the distance to a specific geometry, and several MECP as computed at varying distances, it is possible to obtain a crossing seam of points where the energy between the two states is degenerated. Those degeneracy points are funnels with the highest probability for the energy to hop between the surfaces in internal conversion or intersystem crossing photophysical processes. There are different possibilities. A crossing between states of the same spin multiplicity and spatial symmetry is named a conical intersection. Elements like the nonadiabatic coupling terms are required to obtain them strictly, and they are not computed presently by MOLCAS. If the crossing occurs between states of the same spin multiplicity and different spatial symmetry or between states of different spin multiplicity, the crossing is an hyperplane and its only requirement is the energetic degeneracy and the proper energy minimization.

Here we include an example with the crossing between the lowest singlet (ground) and triplet states of acrolein. Notice that two different states are computed, first by using `RASSCF` to get the wave function and then `ALASKA` to get the gradients of the energy. Nothing new on that, just the information needed in any geometry optimizations. The `GATEWAY` input requires to add as constraint an energy difference between both states equal to zero. A specific instruction is required after calculating the first state. We have to copy the communication file ***RUNFILE*** (at that point contains the information about the first state) to ***RUNFILE2*** to provide later `SLAPAF` with proper information about both states:

```
 *CASSCF singlet-triplet crossing in acrolein
 *File: CASSCF.S-T_crossing.acrolein
```

```
*
 &GATEWAY
Title= Acrolein molecule
Basis set
O.sto-3g....
 O1            1.5686705444       -0.1354553340        3.1977912036  Angstrom
End of basis
Basis set
C.sto-3g....
 C1           -0.1641585340        0.2420235062       -0.0459895824  Angstrom
 C2            0.1137722023       -0.1389623714        1.3481527296  Angstrom
 C3            1.3218729238        0.1965728073        1.9959513294  Angstrom
End of basis
Basis set
H.sto-3g....
 H1            2.0526602523        0.7568282320        1.4351034056  Angstrom
 H2           -0.6138178851       -0.6941171027        1.9113821810  Angstrom
 H3           -0.8171509745        1.0643342316       -0.2648232855  Angstrom
 H4            0.1260134708       -0.4020589690       -0.8535699812  Angstrom
End of basis
Constraints
   a = Ediff
  Value
   a = 0.000
End of Constraints
>>> Do while
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF
>>> ENDIF <<<
 &RASSCF
   LumOrb
   Spin= 1; Nactel= 4 0 0; Inactive= 13; Ras2= 4
   CiRoot= 1 1; 1
 &ALASKA
>>COPY $WorkDir/$Project.RunFile $WorkDir/RUNFILE2
 &RASSCF
   LumOrb
   Spin= 3; Nactel= 4 0 0; Inactive= 13; Ras2= 4
   CiRoot= 1 1; 1
 &ALASKA
 &SLAPAF
>>> EndDo
```

Solvent effects can be also applied to excited states, but first the reaction field in the ground (initial) state has to be computed. This is because solvation in electronic excited states is a non equilibrium situation in with the electronic polarization effects (fast part of the reaction field) have to treated apart (they supposedly change during the excitation process) from the orientational (slow part) effects. The slow fraction of the reaction field is maintained from the initial state and therefore a previous calculation is required. From the practical point of view the input is simple as illustrated in the next example. First, the proper reaction-field input is included in `SEWARD`, then a `RASSCF` and `CASPT2` run of the ground state, with keyword RFPERT in `CASPT2`, and after that another SA-CASSCF calculation of five roots to get the wave function of the excited states. Keyword NONEEQUILIBRIUM tells the program to extract the slow part of the reaction field from the previous calculation of the ground state (specifically from the *`JOBOLD`* file, which may be stored for other calculations) while the fast part is freshly computed. Also, as it is a SA-CASSCF calculation (if not, this is not required) keyword RFROOT is introduced to specify for which of the computed roots the reaction field is generated. We have selected here the fifth root because it has a very large dipole moment, which is also very different from the ground state dipole moment. If you compare the excitation energy obtained for the isolated and the solvated system, a the large red shift is obtained in the later.

```
*CASPT2 excited state in water for acrolein
*File: CASPT2.excited_solvent.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord = acrolein.xyz; basis = STO-3G; group= c1
  RF-input
   PCM-model; solvent= water
  End of RF-input
&SEWARD
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 1 1 1
&CASPT2
  Multistate= 1 1
  RFPert
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 5 5 1
  RFRoot= 5
  NONEquilibrium
&CASPT2
  Multistate= 1 5
  RFPert
```

A number of simple examples as how to proceed with the most frequent quantum chemical problems computed with MOLCAS have been given above. Certainly there are many more possibilities in MOLCAS 8.2 such as calculation of 3D band systems in solids at a semiempirical level, obtaining valence-bond structures, the use of QM/MM methods in combination with a external MM code, the introduction of external homogeneous or non homogeneous perturbations, generation of atomic basis sets, application of different localization schemes, analysis of first order polarizabilities, calculation of vibrational intensities, analysis, generation, and fitting of potentials, computation of vibro-rotational spectra for diatomic molecules, introduction of relativistic effects, etc. All those aspects are explained in the manual and are much more specific. Next section 4 details the basic structure of the inputs, program by program, while easy examples can also be found. Later, another chapter includes a number of extremely detailed examples with more elaborated quantum chemical examples, in which also scientific comments are included. Examples include calculations on high symmetry molecules, geometry optimizations and Hessians, computing reaction paths, high quality wave functions, excited states, solvent models, and computation of relativistic effects.

Section 4

Program Based Tutorials

The MOLCAS 8.2 suite of Quantum Chemical programs is modular in design. The desired calculation is achieved by executing a list of MOLCAS program modules in succession, while potentially manipulating the program information files. If the information files from a previous calculation are saved, then a subsequent calculation need not recompute them. This is dependent on the correct information being preserved in the information files for subsequent calculations. Each module has keywords to specify the functions to be carried out, and many modules rely on the specification of keywords in previous modules.

The following sections describe the use of the MOLCAS modules and their inter-relationships. Each module is introduced in the approximate order for performing a typical calculation. A complete flowchart for the MOLCAS 8.2 suite of programs follows.

4.1 8.2 Flowchart

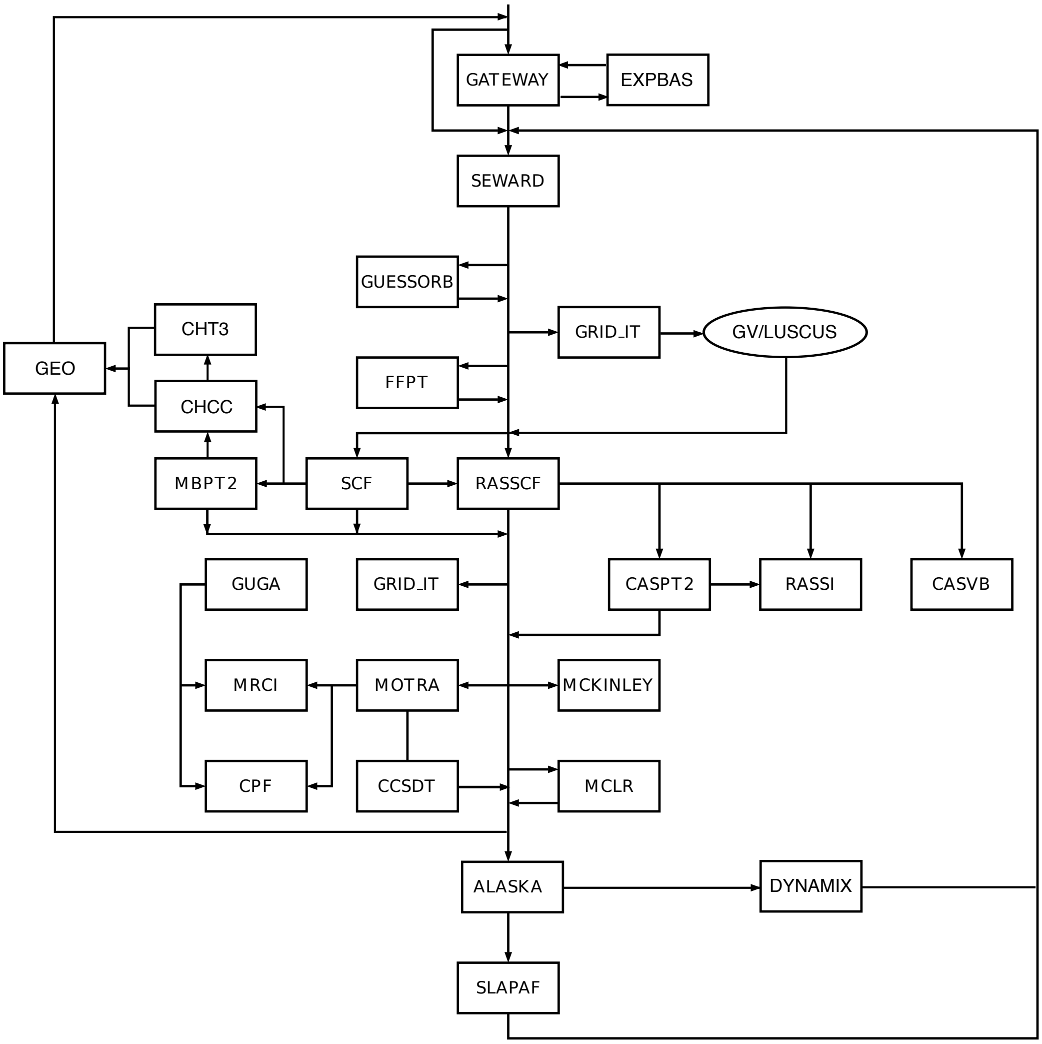

Figure 4.1: Flowchart for Module Dependencies in MOLCAS

4.2 Environment and EMIL Commands

The following are basic and most common commands for the MOLCAS environment variables and input language (EMIL):

Variable	*Purpose*
MOLCAS	MOLCAS home directory.
MOLCAS_MEM	Memory definition in Mb. Default is 2048Mb.
MOLCAS_PRINT	Set printing level: 2 for Normal, 3 for Verbose

MOLCAS_PROJECT	Name used for the project/files.
MOLCAS_WORKDIR	Scratch directory for intermediate files.
MOLCAS_NPROCS	Use NPROCS parallel processes

Special commands in the input always start from symbol > (same as several repeated signs >>). Comment lines can be written using C/C++ syntax (//, or / ∗ ... ∗ /).

Command	*Purpose*
>>Do While	Start of a loop in an input file for geometry optimization with conditional termination.
>>Foreach	Start of a loop in an input file over a number of items.
>>EndDo	End of a loop in an input file.
>>If (condition)	Start of If block.
>>EndIf	End of If block.
>>Label Mark	Setting the label "Mark" in the input.
>>Goto Mark	Forward jump to the label "Mark" skipping that part of the input.
>>SHELL command	Execute command once
>>EXEC command	Execute command for all processes
>>echo string	make an echo

Some simple examples of EMIL commands:

Say “Hello World” 5 times

```
>>export MOLCAS_MAXITER=5
>> Do While
>> echo 'Hello World'
&LOOP  // it needed to terminate loop
>> End Do
```

Do some arithmetics:

```
>>foreach XX in ( 1..10 )  // Note: All variables: XX, YY must be uppercased
>>eval YY=1/$XX
>>echo $YY
>>enddo
```

Print working directory:

```
>>shell pwd
```

Print working directory for each process:

```
>>export MOLCAS_NPROCS=2
>>exec pwd
```

4.3 GATEWAY - Definition of geometry, basis sets, and symmetry

The program `GATEWAY` handles the basic molecular parameters in the calculation. It generates data that are used in all subsequent calculations. These data are stored in the ***`RUNFILE`***. `GATEWAY` is the first program to be executed, if the $WorkDir directory and the ***`RUNFILE`*** file has not already been generated by a previous calculation.

This tutorial is describes how to set up the basic Molcas input for the water molecule. For a more general description of the input options for `GATEWAY`, please refer to the Users Guide. The first line of the input is the program identifier &GATEWAY. Then follows the keyword used is TITLe which will also get printed in the `GATEWAY` section of the calculation output. The title line is also saved in the integral file and will appear in subsequent programs.

The GROUp keyword is followed by the generators for the C_{2v} point group, since the example deals with the water molecule. The specification of the C_{2v} point group given in Table 4.1 is not unique, but, in this tutorial, the generators have been input in an order that reproduces the ordering in the character tables. A complete list of symmetry generator input syntax is given in Table 4.1. The symmetry groups available are listed with the symmetry generators defining the group. The Molcas keywords required to specify the symmetry groups are also listed. The last column contains the symmetry elements generated by the symmetry generators.

```
 &GATEWAY
Title= Water in C2v symmetry - A Tutorial
Coord = water.xyz
Group =  XY Y
Basis Set = O.ANO-S-MB,H.ANO-S-MB
```

Table 4.1: Symmetries available in MOLCAS including generators, MOLCAS keywords and symmetry elements.

Group	Generators			Molcas			Elements							
	g_1	g_2	g_3	g_1	g_2	g_3	E	g_1	g_2	g_1g_2	g_3	g_1g_3	g_2g_3	$g_1g_2g_3$
C_1							E							
C_2	C_2			XY			E	C_2						
C_s	σ			X			E	σ						
C_i	i			XYZ			E	i						
C_{2v}	C_2	σ_v		XY	Y		E	C_2	σ_v	σ'_v				
C_{2h}	C_2	i		XY	XYZ		E	C_2	i	σ_h				
D_2	C_2^z	C_2^y		XY	XZ		E	C_2^z	C_2^y	C_2^x				
D_{2h}	C_2^z	C_2^y	i	XY	XZ	XYZ	E	C_2^z	C_2^y	C_2^x	i	σ^{xy}	σ^{xz}	σ^{yz}

To reduce the input, the unity operator E is always assumed. The twofold rotation about the z-axis, $C_2(z)$, and the reflection in the xz-plane, $\sigma_v(xz)$, are input as XY and Y respectively. The Molcas input can be viewed as symmetry operators that operate on the Cartesian elements specified. For example, the reflection in the xz-plane is specified by the input keyword Y which is the Cartesian element operated upon by the reflection.

The input produces the character table in the `GATEWAY` section of the output shown in Figure 4.3. Note that $\sigma_v(yz)$ was produced from the other two generators. The last column contains the basis functions of each irreducible symmetry representation. The totally symmetric a_1 irreducible representation has the z basis function listed which is unchanged by any of the symmetry operations.

	E	C2(z)	s(xz)	s(yz)	
a1	1	1	1	1	z
b1	1	-1	1	-1	x, xz, Ry
a2	1	1	-1	-1	xy, Rz, I
b2	1	-1	-1	1	y, yz, Rx

The geometry of the molecule is defined using the keyword COORD. On the next line, the name of the xyz file that defines the geometrical parameters of the molecule (*`water.xyz`*) is given.

1. The first line of the *`water.xyz`* file contains the number of atoms.
2. The second line is used to indicate the units: Ångström or atomic units.

   The default is to use Ångström.
3. Then follows one line for each atom containing the name of each atom and its coordinates.

Basis sets are defined after the keyword BASIs SETS. The oxygen and hydrogen basis set chosen, for this example, are the small Atomic Natural Orbitals (ANO) sets. There are three contractions of the basis included in the input, which can be toggled in or excluded with an asterisk, according to the desired calculation: minimal basis, double zeta basis with polarization, or triple zeta basis with polarization.

4.3.1 `GATEWAY` Output

The `GATEWAY` output contains the symmetry character table, basis set information and input atomic centers. The basis set information lists the exponents and contraction coefficients as well as the type of Gaussian functions (Cartesian, spherical or contaminated) used.

The internuclear distances and valence bond angles (including dihedral angles) are displayed after the basis set information. Inertia and rigid-rotor analysis is also included in the output along with the timing information.

A section of the output that is useful for determining the input to the MOLCAS module `SCF` is the symmetry adapted basis functions which appears near the end of the `GATEWAY` portion of the output. This is covered in more detail in the `SCF` tutorial.

The most important file produced by the `GATEWAY` module is the *`RUNFILE`* which in our case is linked to *`water.RunFile`*. This is the general MOLCAS communications file for transferring data between the various MOLCAS program modules. Many of the program modules add data to the *`RUNFILE`* which can be used in still other modules. A new *`RUNFILE`* is produced every time `GATEWAY` is run. It should finally be mentioned that for backwards compatibility one can run `MOLCAS` without invoking `GATEWAY`. The corresponding input and output will then be handled by the program `SEWARD`.

4.3.2 `GATEWAY` Basic and Most Common Keywords

Keyword	*Meaning*
COORD	File name or inline number of atoms and XYZ coordinates
BASIS SET	Atom_label.Basis_label (for example ANO-L-VTZP)
GROUP	Full (find maximum), NoSym, or symmetry generators
SYMMETRY	Symmetry generators: X, Y, Z, XY, XZ, YZ, XYZ (in native format)
RICD	On-the-fly auxiliary basis sets.

4.4 SEWARD — An Integral Generation Program

An *ab initio* calculation always requires integrals. In the MOLCAS suite of programs, this function is supplied by the `SEWARD` module. `SEWARD` computes the one- and two-electron integrals for the molecule and basis set specified in the input to the program `GATEWAY`, which should be run before `SEWARD`. `SEWARD` can also be used to perform some property expectation calculations on the isolated molecule. The module is also used as an input parser for the reaction field and numerical quadrature parameters.

We commence our tutorial by calculating the integrals for a water molecule. The input is given in Figure 4.4. Each MOLCAS module identifies input from a file by the name of the module. In the case of `SEWARD`, the program starts with the label &SEWARD, which is the first statement in the file shown below.

In normal cases no input is required for `SEWARD`, so the following input is optional. The first keyword used is TITLE. Only the first line of the title is printed in the output. The first title line is also saved in the integral file and appears in any subsequent programs that use the integrals calculated by `SEWARD`.

```
 &SEWARD
Title
 Water - A Tutorial. The integrals of water are calculated using C2v symmetry
```

In more complicated cases more input may be needed, to specify certain types of integrals, that use of Cholesky decomposition techniques (CHOLESKY keyword), etc. We refer to the specific sections of the Users-Guide for more information. The output from a `SEWARD` calculation is small and contains in principle only a list of the different types of integrals that are computed.

The integrals produced by the `SEWARD` module are stored in two files in the working directory. They are ascribed the `FORTRAN` names *`ONEINT`* and *`ORDINT`* which are automatically symbolically linked by the MOLCAS script to the file names $PROJECT.*`OneInt`* and $PROJECT.*`OrdInt`*, respectively or more specifically, in our case, *`water.OneInt`* and *`water.OrdInt`*, respectively. The default name for each symbolical name is contained in the corresponding program files of the directory $MOLCAS/shell. The *`ONEINT`* file contains the one-electron integrals. The *`ORDINT`* contains the ordered and packed two-electron integrals. Both files are used by later Molcas program modules.

4.5 SCF — A Self-Consistent Field program and Kohn Sham DFT

The simplest *ab initio* calculations possible use the Hartree-Fock (HF) Self-Consistent Field (SCF) method with the program name `SCF` in the Molcas suite. It is possible to calculate the HF energy once we have calculated the integrals using the `SEWARD` module, although Molcas can perform a direct SCF calculation in which the two-electron integrals are not stored on disk. The Molcas implementation performs a closed-shell (all electrons are paired in orbitals) and open-shell (Unrestricted Hartree-Fock) calculation. It is not possible to perform an Restricted Open-shell Hartree-Fock (ROHF) calculation with the `SCF`. This is instead done using the program `RASSCF`. The `SCF` program can also be used to perform calculations using Kohn Sham Density Functional Theory (DFT).

The `SCF` input for a Hartree-Fock calculation of a water molecule is given in figure 4.2 which continues our calculations on the water molecule.

There are no compulsory keywords following the program name, &SCF. If no input is given the program will compute the SCF energy for a neutral molecule with the orbital occupations giving the lowest energy. Here, we have used the following input: the first is TITLe. As with the `SEWARD` program, the first line following the keyword is printed in the output.

No other keyword is required for a closed-shell calculation. The program will find the lowest-energy electron configuration compatible with the symmetry of the system and will distribute the orbitals accordingly. In complex cases the procedure may fail and produce a higher-lying configuration. It is possible to use the keyword OCCUpied which specifies the number of occupied orbitals in each symmetry grouping listed in the `GATEWAY` output and given in Figure 4.3, forcing the method to converge to the specified configuration. The basis label and type give an impression of the possible molecular orbitals that will be obtained in the SCF calculation. For example, the first basis function in the a_1 irreducible representation is an s type on the oxygen indicating the oxygen 1s orbital. Note, also, that the fourth basis function is centered on the hydrogens, has an s type and is symmetric on both hydrogens as indicated by both hydrogens having a phase of 1, unlike the sixth basis function which has a phase of 1 on center 2 (input H1) and -1 on center 3 (generated H1). As an alternative you can use the keyword Charge with parameters 0 and 1 to indicate a neutral molecule and optimization procedure 1 that searches for the optimal occupation.

Figure 4.2: Sample input requesting the SCF module to calculate the ground Hartree-Fock energy for a neutral water molecule in C_{2v} symmetry.

```
 &SCF
Title= Water - A Tutorial. The SCF energy of water is calculated using C2v symmetry
End of Input
```

Figure 4.3: Symmetry adapted Basis Functions from a GATEWAY output.

```
          Irreducible representation : a1
          Basis function(s) of irrep: z

 Basis Label        Type   Center Phase
   1   O1           1s        1     1
   2   O1           2s        1     1
   3   O1           2p0       1     1
   4   H1           1s        2     1      3     1

          Irreducible representation : b1
          Basis function(s) of irrep: x, xz, Ry

 Basis Label        Type   Center Phase
   5   O1           2p1+      1     1
   6   H1           1s        2     1      3    -1

          Irreducible representation : b2
          Basis function(s) of irrep: y, yz, Rx

 Basis Label        Type   Center Phase
   7   O1           2p1-      1     1
```

We have ten electrons to ascribe to five orbitals to describe a neutral water molecule in the ground state. Several techniques exist for correct allocation of electrons. As a test of the electron allocation, the energy obtained should be the same with and without symmetry. Water is a simple case, more so when using the minimal basis set. In this case, the fourth irreducible representation is not listed in the `GATEWAY` output as there are no basis functions in that representation.

To do a UHF calculation, the keyword UHF must be specified. To force a specific occupation for alpha and beta orbitals In this keyword OCCNUMBERS has to be used with two entries, one for alpha and beta occupied orbital. It is possible to use UHF together with keyword CHARGE or AUFBAU, in this case you have to specify a keyword ZSPIN set to the difference between alpha and beta electrons.

If you want to do an UHF calculation for a closed shell system, for example, diatomic molecule with large interatomic distance, you have to specify keyword SCRAMBLE.

To do the Density Functional Theory calculations, keyword KSDFT followed in the next line by the name of the available functional as listed in the input section is compulsory. Presently following Functional Keywords are available: BLYP, B3LYP, B3LYP5, HFB, HFS, LDA, LDA5, LSDA, LSDA5, SVWN, SVWN5, TLYP, XPBE, MO6, MO6/HF, MO6/2X, MO6/L. The description of functional keywords and the functionals is defined in the online manual for the `SCF` program.

The input for KSDFT is given as,

```
KSDFT= B3LYP5
```

In the above example B3LYP5 functional will be used in KSDFT calculations.

4.5.1 Running `SCF`

Performing the Hartree-Fock calculation introduces some important aspects of the transfer of data between the MOLCAS program modules. The `SCF` module uses the integral files computed by `SEWARD`. It produces a orbital file with the symbolic name *`SCFORB`* which contains all the MO information. This is then available for use in subsequent MOLCAS modules. The `SCF` module also adds information to the *`RUNFILE`*. Recall that the `SEWARD` module produces two integral files symbolically linked to *`ONEINT`* and *`ORDINT`* and actually called, in our case, *`water.OneInt`* and *`water.OrdInt`*, respectively (this is for non-Cholesky-type calculations only). Because the two integral files are present in the working directory when the `SCF` module is performed, MOLCAS automatically links them to the symbolic names.

If the integral files were not deleted in a previous calculation the `SEWARD` calculation need not be repeated. Furthermore, integral files need not be in the working directory if they are linked by the user to their respective symbolic names. Integral files, however, are often very large making it desirable to remove them after the calculation is complete. The linking of files to their symbolic names is useful in other case, such as input orbitals.

If nothing else is stated, the `SCF` program will use the guess orbitals produced by `SEWARD` as input orbitals with the internal name *`GUESSORB`*. If one wants to use any other input orbitals for the `SCF` program the option LUMORB must be used. The corresponding file should be copied to the internal file *`INPORB`*. This could for example be an orbital file generated by an earlier SCF calculation, *`$Project.ScfOrb`*. Just copy or link the file as *`INPORB`*.

4.5.2 `SCF` Output

The `SCF` output includes the title from the input as well as the title from the `GATEWAY` input because we used the integrals generated by `SEWARD`. The output also contains the cartesian coordinates of the molecule and orbital specifications including the number of frozen, occupied and virtual (secondary) orbitals in each symmetry. This is followed by details regarding the `SCF` algorithm including convergence criteria and iteration limits. The energy convergence information includes the one-electron, two-electron, and total energies for each iteration. This is followed by the final results including the final energy and molecular orbitals for each symmetry.

The Density Functional Theory Program gives in addition to the above, details of grids used, convergence criteria, and name of the functional used. This is followed by integrated DFT energy which is the functional contribution to the total energy and the total energy including the correlation. This is followed results including the Kohn Sham orbitals for each symmetry.

The molecular orbital (MO) information lists the orbital energy, the electron occupation and the coefficients of the basis functions contributing to that MO. For a minimal basis set, the basis functions correspond directly to the atomic orbitals. Using larger basis sets means that a combination of the basis functions will be used for each atomic orbital and more so for the MOs. The MOs from the first symmetry species are given in Figure 4.4. The first MO has an energy of -20.5611 hartree and an occupation of 2.0. The major contribution is from the first basis function label '`O1 1s`' meaning an *s* type function centered on the oxygen atom. The orbital energy and the coefficient indicates that it is the MO based largely on the oxygen 1*s* atomic orbital.

Figure 4.4: Molecular orbitals from the first symmetry species of a calculation of water using C_{2v} symmetry and a minimal basis set.

```
          ORBITAL         1          2          3          4
```

```
        EneRGY    -20.5611   -1.3467    -.5957     .0000
        Occ. NO.    2.0000    2.0000    2.0000     .0000

      1 O1   1s      1.0000    -.0131    -.0264    -.0797
      2 O1   2s       .0011     .8608    -.4646    -.7760
      3 O1   2p0      .0017     .1392     .7809    -.7749
      4 H1   1s      -.0009     .2330     .4849    1.5386
```

The second MO has a major contribution from the second oxygen `1s` basis function indicating a mostly oxygen 2*s* construction. Note that it is the absolute value of the coefficient that determines it importance. The sign is important for determining the orthogonality of its orbitals and whether the atomic orbitals contributions with overlap constructively (bonding) or destructively (anti-bonding). The former occurs in this MO as indicated by the positive sign on the oxygen 2*s* and the hydrogen 1*s* orbitals, showing a bonding interaction between them. The latter occurs in the third MO, where the relative sign is reversed.

The third MO has an energy of -0.5957 hartree and major contributions from the second oxygen `1s` basis function, the oxygen `2p0` basis function and the hydrogen `1s` basis functions which are symmetrically situated on each hydrogen (see Figure 4.3). The mixing of the oxygen `2s` and `2p0` basis functions leads to a hybrid orbital that points away from the two hydrogens, to which it is weakly antibonding.

A similar analysis of the fourth orbital reveals that it is the strongly anti-bonding orbital partner to the third MO. The oxygen `2p0` basis function is negative which reverses the overlap characteristics.

The molecular orbital information is followed by a Mulliken charge analysis by input center and basis function. This provides a measure of the electronic charge of each atomic center.

Towards the end of the `SCF` section of the MOLCAS output various properties of the molecule are displayed. By default the first (dipole) and second cartesian moments and the quadrupoles are displayed.

4.5.3 SCF - Basic and Most Common Keywords

Keyword	*Meaning*
UHF	Unrestricted Hartee Fock or unrestricted DFT calculation
KSDFT	DFT calculations, with options: BLYP, B3LYP, B3LYP5, HFB, HFS, LDA, LDA5, LSDA, LSDA5, SVWN, SVWN5, TLYP, PBE, PBE0
CHARGE	Net charge of the system (default zero)
ZSPIN	Difference between α and β electrons
OCCUPIED	Specify the orbital occupations per irreps

4.6 MBPT2 — A Second-Order Many-Body PT RHF Program

The `MBPT2` program performs second-order Many Body Perturbation Theory calculations based on a RHF-type of wave function (MP2 method). The calculation is to some extent defined by the SCF calculation which must be performed before running the `MBPT2` program. Therefore, there is no difficulty related to the input file unless an analysis of the correlation energies of specific electron pairs or contribution from external orbitals wants to be performed. In this case keywords SFROzen and SDELeted have to be used as described in MBPT2 section of the user's guide.

To run the program the *`ORDINT`* integral file(s) generated by the `SEWARD` program and the *`RUNFILE`* file generated by the `SCF` program are needed. The program can be otherwise run in a direct manner. Therefore the `SEWARD` program can be run with the option DIREct included in its input. Only the *`ONEINT`* will then be generated and used by the `SCF` module. The input file used to run an `MBPT2` calculation on the ground state of the water molecule is displayed in figure 4.5. For large molecules it is also possible to use the Cholesky decomposition technique to speed up the calculations. This will be described in another section of the tutorials.

```
 &MBPT2
Title= MP2 of ground state of C2v Water
Frozen= 1 0 0 0
```

Figure 4.5: Sample input requested by the MBPT2 module to calculate the MP2 energy for the ground state of the water in C_{2v} symmetry.

The output of `MBPT2` is self-explanatory.

4.7 RASSCF — A Multi Configurational Self-Consistent Field Program

One of the central codes in Molcas is the `RASSCF` program, which performs multiconfigurational SCF calculations. Both Complete Active Space (CASSCF) and Restricted Active Space (RASSCF) SCF calculations can be performed with the `RASSCF` program module [11]. An open shell Hartree-Fock calculation is not possible with the `SCF` but it can be performed using the `RASSCF` module. An input listing for a CASSCF calculation of water appears in Figure 4.6. `RASSCF` requires orbital information of the system which can be obtained in two ways. The LUMOrb indicates that the orbitals should be taken from a user defined orbital file, which is copied to the internal file INPORB. If this keyword is not given, the program will look for orbitals on the runfile in the preference order: *`RASORB`*, *`SCFORB`* and *`GUESSORB`*

The TITLe performs the same function as in the previous Molcas modules. The keyword INACtive specifies the number of doubly occupied orbitals in each symmetry that will not be included in the electron excitations and thus remain doubly occupied throughout the calculation. A diagram of the complete orbital space available in the `RASSCF` module is given in Figure 4.7.

In our calculation, we have placed the oxygen 1s orbital in the inactive space using the INACtive keyword. The keyword FROZen can be used, for example, on heavy atoms to reduce the Basis Set Superposition Error (BSSE). The corresponding orbitals will then

Figure 4.6: Sample input requesting the RASSCF module to calculate the eight-electrons-in-six-orbitals CASSCF energy of the second excited triplet state in the second symmetry group of a water molecule in C_{2v} symmetry.

```
 &RASSCF
Title= The CASSCF energy of water is calculated using C2v symmetry. 2 3B2 state.
nActEl= 8 0 0
Inactive= 1 0 0 0; Ras2= 3 2 0 1
Symmetry= 2; Spin= 3
CIRoot= 1 2; 2
LumOrb
```

not be optimized. The RAS2 keyword specifies the number of orbitals in each symmetry to be included in the electron excitations with all possible occupations allowable. Because the RAS1 and RAS3 spaces are zero (not specified in the input in Figure 4.6) the `RASSCF` calculation will produce a CASSCF wave function. The RAS2 space is chosen to use all the orbitals available in each symmetry (except the oxygen 1s orbital). The keyword NACTEL specifies the number of active electrons (8), maximum number of holes in the Ras1 space (0) and the maximum number of electrons in the Ras3 space (0). Using the keywords RAS1 and/or RAS3 to specify orbitals and specifying none zero numbers of holes/electrons will produce a RASSCF wave function.We are, therefore, performing an 8in6 CASSCF calculation of water.

Table 4.2: Examples of types of wave functions obtainable using the RAS1 and RAS3 spaces in the RASSCF module.

Description	Number of holes in RAS1 orbitals	RAS2 orbitals	Number of electrons in RAS3 orbitals
SD-CI	2	0	2
SDT-CI	3	0	3
SDTQ-CI	4	0	4
Multi Reference SD-CI	2	n	2
Multi Reference SD(T)-CI	3	n	2

There are a number of wave function types that can be performed by manipulating the RAS1 and RAS3 spaces. Table 4.2 lists a number of types obtainable. The first three are Configuration Interaction (CI) wave functions of increasing magnitude culminating with a Single, Double, Triples and Quadruples (SDTQ) CI. These can become multi reference if the number of RAS2 orbitals is non-zero. The last type provides some inclusion of the triples excitation by allowing three holes in the RAS1 orbitals but save computation cost by only allowing double excitations in the RAS3 orbitals.

The symmetry of the wave function is specified using the SYMMETRY keyword. It specifies the number of the symmetry subgroup in the calculation. We have chosen the second symmetry species, b_2, for this calculation. We have also chosen the triplet state using the keyword SPIN. The keyword CIROOT has been used to instruct `RASSCF` to find the second excited state in the given symmetry and spin. This is achieved by specifying the number of roots, 1, the dimension of the small CI matrix which must be as large as the highest required root and the number of the required second root. Only for averaged calculations CIROOT needs an additional line containing the weight of the selected roots (unless equal weights are used for all states).

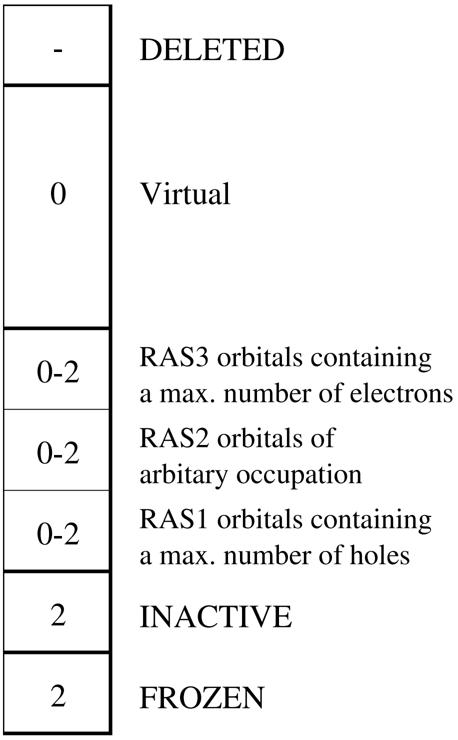

Figure 4.7: RASSCF orbital space including keywords and electron occupancy ranges.

As an alternative to giving inactive and active orbital input we can use the type index input on the *`INPORB`* and indicate there which type the different orbitals should belong to: frozen (f), inactive (i), RAS1 (1), RAS2 (2), RAS3 (3), secondary (s), or deleted (d). This approach is very useful when the input orbitals have been run through `LUSCUS`, which is used to select the different subspaces. `LUSCUS` will relabel to orbitals according to the users instructions and the corresponding orbital file ,*`GvOrb`* can be linked as the *`INPORB`* in the `RASSCF` program without any further input.

A level shift was included using the LEVShift keyword to improve convergence of the calculation. In this case, the calculation does not converge without the use of the level shift. It is advisable to perform new calculations with a non-zero LEVShift value (the default value is 0.5). Another possibility is to increase the maximum number of iterations for the macro and the super-CI Davidson procedures from the default values (200,100) using the keyword ITERations.

Sometimes convergence problems might appear when the wave function is close to fulfill all the convergence criteria. An infrequent but possible divergence might appear in a calculation starting from orbitals of an already converged wave function, or in cases where the convergence thresholds have been decreased below the default values. Option TIGHt may be useful in those cases. It contains the thresholds criteria for the Davidson diagonalization procedure. In situations such as those described above it is recommended to decrease the first parameter of TIGHt to a value lower than the default, for instance 1.0d-06.

4.7.1 `RASSCF` Output

The `RASSCF` section of the MOLCAS output contains similar information to the `SCF` output. Naturally, the fact that we have requested an excited state is indicated in the output. In fact, both the lowest triplet state and the first excited state or second root are documented including energies. For both of these states the CI configurations with a coefficient greater than 0.05 are printed along with the partial electron distribution in the active space. Figure 4.8 shows the relevant output for the second root calculated. There are three configurations with a CI-coefficient larger than 0.05 and two with very much larger values. The number of the configuration is given in the first column and the CI-coefficient and weight are given in the last two columns. The electron occupation of the orbitals of the first symmetry for each configuration is given under the '`111`' using '`2`' for a fully occupied orbital and '`u`' for a singly occupied orbital containing an electron with an up spin. The down spin electrons are represented with a '`d`'. The occupation numbers of the active space for each symmetry is given below the contributing configurations. It is important to remember that the active orbitals are not ordered by any type of criterion within the active space.

Figure 4.8: RASSCF portion of output relating to CI configurations and electron occupation of natural orbitals.

```
printout of CI-coefficients larger than   .05 for root   2
energy=    -75.443990
conf/sym  111 22 4     Coeff  Weight
       3  22u u0 2    .64031  .40999
       4  22u 0u 2    .07674  .00589
      13  2u0 2u 2   -.75133  .56450
      14  2u0 u2 2    .06193  .00384
      19  udu 2u 2    .06489  .00421

Natural orbitals and occupation numbers for root  2
sym 1:   1.986957   1.416217    .437262
sym 2:   1.567238    .594658
sym 4:   1.997668
```

The molecular orbitals are displayed in a similar fashion to the `SCF` section of the output except that the energies of the active orbitals are not defined and therefore are displayed as zero and the electron occupancies are those calculated by the `RASSCF` module. In a state average calculation (more than one root calculated), the MOs will be the natural orbitals corresponding to the state averaged density matrix (called pseudo-natural orbitals) and the occupation numbers will be the corresponding eigenvalues. Natural orbital occupation numbers for each state are printed as shown in Figure 4.8, but the MOs specific to a given state are not shown in the output. They are, however, available in the *`JOBIPH`* file. A number of molecular properties are also computed for the requested electronic state in a similar fashion to the `SCF` module.

4.7.2 Storing and Reading `RASSCF` Orbitals and Wave Functions

Part of the information stored in the `RASSCF` output file, *`JOBIPH`*, for instance the molecular orbitals and occupation numbers can be also found in an editable file named *`RASORB`*, which is automatically generated by `RASSCF`. In case more than one root is used the natural orbitals are also stored in files ***RASORB.1***, ***RASORB.2***, etc, up to ten. In such cases the file ***RASORB*** contains the averaged orbitals. If more roots are used the files can be generated using the OUTORBITALS keyword. The type of orbital produced can be either AVERAGED, NATURAL, CANONICAL or SPIN (keywords) orbitals. The OUTORBITALS keyword, combined with the ORBONLY keyword, can be used to read the ***JOBIPH*** file and produce an orbital file, ***RASORB***, which can be read by a subsequent `RASSCF` calculation using the same input section. The formatted ***RASORB*** file is useful to operate on the orbitals in order to obtain appropriate trial orbitals for a subsequent `RASSCF` calculation. In particular the type index can be changed directly in the file if the `RASSCF` program has converged to a solution with wrong orbitals in the active space. The `RASSCF` program will, however, automatically place the orbital files from the calculation in the user's home directory under the name ***$Project.RasOrb***, etc. In calculations with spin different from zero the program will also produce the spin orbital files ***$Project.SpdOrb1***, etc for each state. These orbitals can be used by the program `LUSCUS` to produce spin densities.

4.7.3 RASSCF - Basic and Most Common Keywords

Keyword	*Meaning*
SYMMETRY	Symmetry of the wave function (according to `GATEWAY`) (1 to 8)
SPIN	Spin multiplicity
CHARGE	Molecular charge
NACTEL	Three numbers: Total number of active electrons, holes in Ras1, particles in Ras3
INACTIVE	By symmetry: doubly occupied orbitals
RAS1	By symmetry: Orbitals in space Ras1 (RASSCF)
RAS2	By symmetry: Orbitals in space Ras1 (CASSCF and RASSCF)
RAS3	By symmetry: Orbitals in space Ras1 (RASSCF)
CIROOT	Three numbers: number of CI roots, dimension of the CI matrix, relative weights (typically 1)
LUMORB/FILEORB	use definition of active space from Orbital file

4.8 CASPT2 — A Many Body Perturbation Program

Dynamic correlation energy of a molecular system can be calculated using the `CASPT2` program module in MOLCAS. A `CASPT2` calculation gives a second order perturbation estimate of the full CI energy using the CASSCF wave function of the system. The program can also perform Multi-State CASPT2 calculations (MS-CASPT2) in which different CASPT2 states are coupled using an effective Hamiltonian computed to second order in perturbation theory. This is necessary in cases where different CASSCF wave functions are strongly dependent on dynamical correlation effects. The wave function have to be obtained in a previous State-Average CASSCF calculation.

A sample input is given in Figure 4.9. The FROZEN keyword specifies the number of orbitals of each symmetry which will not be included in the correlation. We have chosen the `RASSCF` INACTIVE orbitals to be frozen for this calculation (the default is to freeze all core orbitals, so the input is strictly not needed). The remaining two keywords, CONVERGENCE and MAXITER, are included with there default values. The MULTISTATE keyword is included for clarity even if not needed in this single state calculation. A single line follows indicating the number of simultaneously treated CASPT2 roots and the number of the roots in the previous SA-CASSCF calculation.

4.8.1 `CASPT2` Output

Figure 4.9: Sample input requesting the CASPT2 module to calculate the CASPT2 energy of a water molecule in C_{2v} symmetry with one frozen orbital.

```
 &CASPT2
Frozen= 1 0 0 0
Multistate= 1 1
MaxIter= 40
```

The output of the `CASPT2` program begins with the title from the input as well as the title from the `SEWARD` input. It also contains the cartesian coordinates of the molecule and the CASSCF wave function and orbital specifications. This is followed by details about the type of Fock and H_0 operator used and, eventually, the value of the level-shift parameter employed. It is possible then to obtain, by input specifications, the quasi-canonical orbitals in which the wave function will be represented. The following CI vector and occupation number analysis will be performed using the quasi-canonical orbitals.

Two important sections follow. First a detailed report on small energy denominators, large components, and large energy contributions which will inform about the reliability of the calculation and finally the `CASPT2` property section including the natural orbitals obtained as defined in the output and a number of approximated molecular properties.

If the MULTISTATE option is used, the program will perform one CASPT2 calculation for each one of the selected roots, and finally the complete effective Hamiltonian containing the selected states will be solved to obtain the final MS-CASPT2 energies and PM-CASSCF wave functions [12].

The `CASPT2` module needs the integral files in $WorkDir and the *`RUNFILE`* file from the and the *`JOBIPH`* file from the `RASSCF` module. The orbitals are saved in the *`PT2ORB`* file. The new PM-CASSCF wave functions generated in a MS-CASPT2 calculation is saved in the *`JOBMIX`* file.

4.8.2 CASPT2 - Basic and Most Common Keywords

Keyword	*Meaning*
MULTISTATE	Multi-State CASPT2 calculation: number of roots and roots (Ex. 3 1 2 3)
IMAG	Value for the imaginary shift for the zero order Hamiltonian

4.9 RASSI — A RAS State Interaction Program

Program `RASSI` (RAS State Interaction) computes matrix elements of the Hamiltonian and other operators in a wave function basis, which consists of individually optimized CI expansions from the `RASSCF` program. Also, it solves the Schrödinger equation within the space of these wave functions. There are many possible applications for such type of calculations. The first important consideration to have into account is that `RASSI` computes the interaction among RASSCF states expanding the same set of configurations, that is, having the same active space size and number of electrons.

The `RASSI` program is routinely used to compute electronic transition moments, as it is shown in the Advanced Examples in the calculation of transition dipole moments for the excited states of the thiophene molecule using CASSCF-type wave functions. By default the program will compute the matrix elements and expectation values of all the operators for which `SEWARD` has computed the integrals and has stored them in the *`ONEINT`* file.

RASSCF (or CASSCF) individually optimized states are interacting and non-orthogonal. It is imperative when the states involved have different symmetry to transform the states to a common eigenstate basis in such a way that the wave function remains unchanged. The State Interaction calculation gives an unambiguous set of non-interacting and orthonormal eigenstates to the projected Schrödinger equation and also the overlaps between the original RASSCF wave functions and the eigenstates. The analysis of the original states in terms of RASSI eigenstates is very useful to identify spurious local minima and also to inspect the wave functions obtained in different single-root RASSCF calculations, which can be mixed and be of no help to compare the states.

Finally, the `RASSI` program can be applied in situations when there are two strongly interacting states and there are two very different MCSCF solutions. This is a typical situation in transition metal chemistry when there are many close states associated each one to a configuration of the transition metal atom. It is also the case when there are two close quasi-equivalent localized and delocalized solutions. `RASSI` can provide with a single set of orbitals able to represent, for instance, avoided crossings. `RASSI` will produce a number of files containing the natural orbitals for each one of the desired eigenstates to be used in subsequent calculations.

`RASSI` requires as input files the *`ONEINT`* and *`ORDINT`* integral files and the *`JOBIPH`* files from the `RASSCF` program containing the states which are going to be computed. The *`JOBIPH`* files have to be named consecutively as *`JOB001`*, *`JOB002`*, etc. The input for the `RASSI` module has to contain at least the definition of the number of states available in each of the input *`JOBIPH`* files. Figure 4.10 lists the input file for the `RASSI` program in a calculation including two *JOBIPH* files (2 in the first line), the first one including three roots (3 in the first line) and the second five roots (5 in the first line). Each one of the following lines lists the number of these states within each *JOBIPH* file. Also in the input, keyword NATORB indicates that three files (named sequentially *NAT001*, *NAT002*, and *NAT003*) will be created for the three lowest eigenstates.

Figure 4.10: Sample input requesting the RASSI module to calculate the matrix elements and expectation values for eight interacting RASSCF states

```
 &RASSI
NROFjobiph= 2 3 5; 1 2 3; 1 2 3 4 5
NATOrb= 3
```

4.9.1 RASSI Output

The `RASSI` section of the MOLCAS output is basically divided in three parts. Initially, the program prints the information about the *JOBIPH* files and input file, optionally prints the wave functions, and checks that all the configuration spaces are the same in all the input states. In second place `RASSI` prints the expectation values of the one-electron operators, the Hamiltonian matrix, the overlap matrix, and the matrix elements of the one-electron operators, all for the basis of input RASSCF states. The third part starts with the eigenvectors and eigenvalues for the states computed in the new eigenbasis, as well as the overlap of the computed eigenstates with the input RASSCF states. After that, the expectation values and matrix elements of the one-electron operators are repeated on the basis of the new energy eigenstates. A final section informs about the occupation numbers of the natural orbitals computed by `RASSI`, if any.

In the Advanced Examples a detailed example of how to interpret the matrix elements output section for the thiophene molecule is displayed. The rest of the output is self-explanatory. It has to be remembered that to change the default origins for the one electron operators (the dipole moment operator uses the nuclear charge centroid and the higher order operators the center of the nuclear mass) keyword CENTER in `GATEWAY` must be used. Also, if multipoles higher than order two are required, the option MULTIPOLE has to be used in `GATEWAY`.

The program `RASSI` can also be used to compute a spin-orbit Hamiltonian for the input CASSCF wave functions as defined above. The keyword AMFI has to be used in `SEWARD` to ensure that the corresponding integrals are available.

Figure 4.11: Sample input requesting the RASSI module to calculate and diagonalize the spin-orbit Hamiltonian the ground and triplet excited state in water.

```
 &RASSI
NROFjobiph= 2 1 1; 1; 1
Spinorbit
Ejob
```

The first *JOBMIX* file contains the wave function for the ground state and the second file the $^3B_2$ state discussed above. The keyword EJOB makes the `RASSI` program use the CASPT2 energies which have been written on the *JOBMIX* files in the diagonal of the spin-orbit Hamiltonian. The output of this calculation will give four spin-orbit states and the corresponding transition properties, which can for example be used to compute the radiative lifetime of the triplet state.

4.9.2 RASSI - Basic and Most Common Keywords

Keyword	*Meaning*
NROFJOB	Number of input files, number of roots, and roots for each file
EJOB/HDIAG	Read energies from input file / inline
SPIN	Compute spin-orbit matrix elements for spin properties

4.10 Other Multiconfigurational and Multireference Methods

MOLCAS contains computational modules to perform

- Coupled Cluster calculations: CCSD, CCSD(T), etc. (`CCSDT` and `CHCC`)
- Multireference singles and doubles Configuration Interaction calculations (`MRCI`)
- SDCI - single and double Configuration Interaction calculations (`SDCI`)
- CASVB/MCSCF Valence Bond calculations (`CASVB`)

The Users Guide and Tutorials should be consulted for relevant keywords to be used in these modules.

4.11 ALASKA and SLAPAF: A Molecular Structure Optimization

One of the most powerful functions of *ab initio* calculations is geometry predictions. The minimum energy structure of a molecule for a given method and basis set is instructive especially when experiment is unable to determine the actual geometry. MOLCAS performs a geometry optimization with analytical gradients at the SCF or RASSCF level of calculation, and with numerical gradients at the CASPT2 level.

In order to perform geometry optimization an input file must contain a loop, which includes several calls: calculation of integrals (`SEWARD`), calculation of energy (`SCF`, `RASSCF`, `CASPT2`), calculation of gradients (`ALASKA`), and calculation of the new geometry (`SLAPAF`).

This is an example of such input

```
&GATEWAY
 coord= file.xyz
```

```
   basis= ANO-S-MB
>> EXPORT MOLCAS_MAXITER=25
>> Do While  <<
  &SEWARD
  &SCF
  &SLAPAF
>> EndDo <<
```

The initial coordinates will be taken from xyz file file.xyz, and the geometry will be optimized at the SCF level in this case. After the wave function calculation, calculation of gradients is required, although code `ALASKA` is automatically called by Molcas. `SLAPAF` in this case required the calculation of an energy minimum (no input). Other options are transition states (TS), minimum energy paths (MEP-search), etc The loop will be terminated if the geometry converges, or maximum number of iterations (MaxIter) will be reached (the default value is 50).

There are several EMIL commands , which can be used to control geometry optimization. For example, it is possible to execute some Molcas modules only once:

```
>> IF ( ITER = 1 )
* this part of the input will be executed only during the first iteration
>> ENDIF
```

Program `SLAPAF` is tailored to use analytical or numerical gradients produced by `ALASKA` to relax the geometry of a molecule towards an energy minimum (default, no input required then) or a transition state. The program is also used for finding inter state crossings (ISC), conical intersections (CI), to compute reaction paths, intrinsic reaction coordinate (IRC) paths, etc.

4.11.1 SLAPAF - Basic and Most Common Keywords

Keyword	*Meaning*
TS	Computing a transition state
FindTS	Computing a transition state with a constraint
MEP-search	Computing a steepest-descent minimum reaction path
ITER	Number of iterations
INTErnal	Definition of the internal coordinates

4.12 GRID_IT: A Program for Orbital Visualization

`GRID_IT` is an interface program for calculations of molecular orbitals and density in a set of cartesian grid points. Calculated grid can be visualized by separate program `LUSCUS` in the form of isosurfaces.

`GRID_IT` generates the regular grid and calculates amplitudes of molecular orbitals in this net. Keywords SPARSE,DENSE, NPOINTS specify the density of the grid. And keywords ORANGE (occupation range), ERANGE (energy range), SELECT allow to select some specific orbitals to draw.

As default `GRID_IT` will use grid net with intermediate quality, and choose orbitals near HOMO-LUMO region. Note, that using keyword ALL - to calculate grids for all orbitals or DENSE - to calculate grid with very high quality you can produce a very huge output file.

`GRID_IT` requires the communication file *`RUNFILE`*, processed by `GATEWAY` and any formated *`INPORB`* file: *`SCFORB`*, *`RASORB`*, *`PT2ORB`*, generated by program `SCF`, `RASSCF`, or `CASPT2`, respectively. The output file *`M2MSI`* contains the graphical information.

Normally you do not need to specify any keywords for `GRID_IT`: the selection of grid size, as well as the selection of orbitals done automatically.

An input example for `GRID_IT` is:

```
 &GRID_IT
Dense
* compute orbitals from 20 to 23 form symmetry 1 and orbital 4 from symmetry 2
SELECT
1:20-23,2:4
```

`GRID_IT` can be run in a sequence of other computational codes (if you need to run `GRID_IT` several times, you have to rename grid file by using EMIL command, or by using keyword NAME)

```
&GATEWAY
  ...
&SEWARD
&SCF
&GRID_IT
NAME=scf
&RASSCF
&GRID_IT
NAME=ras
```

or, you can run `GRID_IT` separately, when the calculation has finished.

```
&GATEWAY
&GRID_IT
FILEORB=/home/joe/project/water/water.ScfOrb
```

4.12.1 GRID_IT - Basic and Most Common Keywords

Keyword	*Meaning*
ASCII	Generate the *`grid`* file in ASCII (e.g. to transfer to another computer), can be only used in combitation with NOLUSCUS
ALL	Generate all orbitals
SELECT	Select orbitals to compute

4.13 Tools for selection of the active space

Selecting an active space is sometimes easy. For a small molecule, an active space for the ground and the lowest valence excited states is usually the valence orbitals, i.e. orbitals composed of atomic orbitals belonging to the usual 'valence shells' (there are some exceptions to this rule). Problems arise for medium or large molecules, for higher excited states, and for molecules including transition, lanthanide or actinide elements. A good wish list of orbitals will give a CASSCF/CASPT2 calculation that demand unrealistically large computer resources and time. Compromises must be made. Any smaller selection of active orbitals can in general affect your results, and the selection should be based on the specific calculations: see 4.14 for advise.

The following three tools may be help in the process:

localisation is a program that can take a (subrange of) orbitals from an orbital file, and produce a new orbital file where these orbitals have been transformed to become localized, while spanning the same space as the original ones.

expbas can take an orbital file using a smaller basis set, and 'expand' it into a new orbital file using a larger basis.

LUSCUS (is of course also described elsewhere) is the orbital viewer.

It is of course best to have a good perception of the electronic structure of the molecule, including all states of interest for the calculation. If it is a larger system, where lots of ligands can be assumed not to partake in non-dynamic correlation, it is a good idea to run some simple exploratory calculations with a much smaller model system. Check the literature for calculations on similar systems or model systems.

First of all, you need to know how many orbitals (in each symmetry) that should be active. Their precise identity is also good to know, in order to have a good set of starting orbitals, but we come to that later. **Necessary** active orbitals are: Any shells that may be open in any of the states or structures studied. Breaking a bond generally produces a correlated bond orbital and a correlating antibonding orbital, that must both be active (Since it is the **number** of orbitals we are dealing with as yet, you may as well think of the two radical orbitals that are produced by completely breaking the bond). You probably want to include one orbital for each aromatic carbon. **Valuable correlated** active orbitals are: Oxygen lone pair, CC π bonds. **Valuable correlating** active orbitals are: the antibonding π^* CC orbitals, and one additional set of correlating d orbitals for most transition elements (sometimes called the 'double d-shell effect').

The valuable correlated orbitals can be used as Ras-1 orbitals, and correlating ones can be used as Ras-3 orbitals, if the active space becomes too large for a casscf calculation.

Assuming we can decide on the number of active orbitals, the next task is to prepare starting orbitals that enables CASSCF to converge, by energy optimization, to the actual starting orbitals for your calculation. Use a very small basis set to begin with: This will usually be one of the minimal bases, e.g. ANO-S-MB. This is not just to save time: the small basis and the large energy spacings make it much easier to get well-defined correlating orbitals.

Performing the actual casscf (or rasscf) calculation may give you the active space you want: Viewing the orbitals by `LUSCUS` may confirm this, but very often the orbitals are too mixed up (compared to ones mental picture of what constitutes the best orbitals). Using localisation program solves this problem. In order to localise without mixing up orbitals from different subspaces may require to produce the new orbital file through several runs of the program; however, for the present perpose, it may be best not to have so very strict restrictions, for example: Allow mixing among a few high inactive and the most occupied orbitals; and also among the weakly occupied and some virtual orbitals.

Running the localisation program, and viewing the localised orbitals, is a great help since directly in `LUSCUS` one can redefine orbitals as being inactive, or ras3 , or whatever, to produce a new orbital file. The resulting annotated localised orbitals can be used in a new run.

Once a plausible active space has been found, use the expbas tool to obtain starting orbitals using, e.g. ANO-VDZP basis, or whatever is to be used in the bulk of the production run.

It is also a good idea to, at this point, 'waste' a few resources on a single-point calculation for a few more states than you are really interested in, and maybe look at properties, etc. There may be experimental spectra to compare with.

And please have a look at the following section, 'Some practical HINTS' 4.14.

4.14 Some practical HINTS

This section contains a collection of practical hints and advices how to use Molcas in solving quantum chemistry problems.

4.14.1 GATEWAY/SEWARD program:

- Try the Cholesky approximation (or RI)! It saves disk space and possibly time.
- Think about basis set. ANO-like basis sets have many advantages, but they are "marginal".
- Try to avoid inline basis sets, use the library.
- Remember that the quality of basis set should match quality of computational method.
- Use ANO-RCC even for atoms in the 2nd row.
- Be extremely careful when computing anions. Remember that special situations requires special basis sets.
- Use minimal or small basis set for understanding chemical problem. You always can use expbas later..

4.14.2 SCF program

- HF orbitals are in many cases good starting orbitals, but quite often GuessOrb is a better choice.
- Very large basis set together with HF can lead to linear dependences.

- Remember! Hartree-Fock method allows multiple solutions (even for trivial systems)
- Be reasonable selecting convergence thresholds
- UHF convergence is much poor comparing to RHF.
- Sometime you have to use Scramble keyword to break the symmetry.

- DFT convergence can't be better that HF convergence. Think about starting orbitals for DFT.
- Remember that DFT is a powerful method but it is still single configurational method. Don't use it beyond it's limits.
- Choose your favorite functional, and stay with your decision
- Note that MOLCAS is not the best DFT code available

4.14.3 RASSCF program:

- MCSCF are multi-solution methods that heavily depend on the starting orbitals and level of calculation (roots).
- On convergence ALWAYS (ALWAYS, ALWAYS, etc) check the orbitals (LUSCUS, molden, CMOcorr, etc). MCSCF methods will lead to different solutions for active spaces of different nature. Use your chemical intuition and let the calculation guide you.
- Analyze carefully the CI coefficients and natural occupation numbers together with the orbitals (average orbitals are fine in general for that) in order to understand the nature of the states .

- You get average orbitals, and orbitals for individual roots, which you can visualized by `LUSCUS` or molden etc, contain the natural orbitals of the different roots.
- Try increasing the number of SA-CASSCF roots to locate more excited states. They can be low-lying solutions at the CASPT2 level. In high symmetry cases you may also need to consider roots that have high energy at the initial steps and can become lower roots in the converged calculation.
- It is NOT advisable to play games with weights for the different roots. Roots with equal weights make your calculation more clear and reproducible.

- MOLCAS can handle only D_2h subgroups. Molecules with other symmetry (C_{3v}, D_{4d}, T_d, O_h) have a problem. Especially if you use approximations, like CD.
- Work in a symmetry point group that allows degenerate states to belong to the same irreducible representation (e.g. C_2 for linear molecules). Try C_1 too.
- Working in a too high symmetry might prevent you from obtaining less symmetric lowest-lying localized solutions (e.g. Ni^{2+})

- Start with clean symmetric orbitals (GUESSORB). Sometime (for example for a radical), an orbital for positively charged system can be more symmetric.
- use if needed, CLEANUP and SUPSYM, or for linear molecules: the LINEAR keyword.

- use it! RASSCF is a simple way to increase an active space.
- Balance RAS1/3 and RAS2 subspaces. Try to change orbitals between these subspaces.
- Removing RAS2 space completely is not a good idea.
- Note that RAS calculations have a slower convergence, and demand more resources.

- Increase LEVShift parameter in cases of slow or difficult convergence.
- Sometimes RASSCF is very sensitive when is close to convergence. Try restarting the calculation from the previous JOBIPH file
- Try to restart from orbitals (or JOBIPH) instead of starting from scratch.

4.14.4 Selection of active spaces:

- Always compare calculations with the same active space size (and nature if possible).
- Ask yourself first which is your goal. The selection of the active space depends on that.
- If you made a selection once, try to reuse orbitals! Especially for a set of calculations with different geometries
- In ground state calculations many orbitals can have an occupation close to 2 and 0 and might rotate with others in the inactive (secondary) space. It might be wise to skip them
- For low-lying excited states and few roots you might leave inactive quite a number of orbitals. Check with RASSCF for instance.

- SCF orbital energies sometimes help to choose the orbitals by using the energy order criterion, but you must learn to see the problems (like lone pair orbitals having too low energies at the SCF level).
- You typically will need correlating orbitals, that is, if you have a π orbital you need a π^*, the same for σ,σ^*, but not for lone pairs.
- CASSCF/RASSCF geometry optimizations are the worst case. If you miss orbitals you might end up in a totally wrong geometry (e.g. breaking a bond usually requires the bonding and antibonding orbitals in the space).
- Organic (1st row atoms) molecules usually require open shell orbitals, π, π^*, and lone pairs. If 2nd row atoms are added (S, P, Si, etc) s orbitals enter in action (s bonds are longer). CH bonds can often be left be inactive.

- Rydberg states require additional diffuse basis sets and specific orbitals in the active space. Use basis sets centered in the charge center of the positive ion (consult the manual).
- Transition metal chemistry (1st row) sometimes requires a double *d* shell description in the active space.
- Lanthanides have a quite inert 4*f* shell that must be active together with 5d, 6s (6p). Actinides 5*f*, 6*d*, 7*s*.

- **use expbas!** start from minimal basis set, decide the active space, and expand the basis to "normal". With small basis set you can clearly identify orbitals.
- **use localization!** Especially for virtual orbitals.
- **expand active space by adding RAS1/3** - give the system a freedom, and see how it reacts.

4.14.5 CASPT2 program:

- The new IPEA = 0.25 zeroth Hamiltonian is the default. It particularly improves open shell cases. But there are some cases where IPEA=0 gives better correlation with experiment.
- Energy differences between different states or situations are only reliable between calculations with the same active space size and similar reference weights in CASPT2.
- An intruder state (low reference weight in the CASPT2 state) might be informing you that your active space lacks an important orbital. Check the list of large perturbative contributions (small denominators combined with large RHS values; check the output) and also the occupation number of the CASPT2 orbitals.
- For weakly interacting intruder states cases try the IMAGINARY level shift parameter. Don't use the level shift to reach agreement with experiment!

- For heavy valence-Rydberg mixing cases or for closely degenerated CASPT2 states, use MS-CASPT2.
- If the MS-CASPT2 description differs a lot from the CASPT2 one, try to check the calculation by increasing the active space (introducing angular correlation if possible) until the result is converged. The "true" solution is typically between both cases (CASPT2 and MS-CASPT2). If you are suspicious about the MS-CASPT2 result, better keep the CASPT2 one. It has worked out generally well so far.

4.14.6 RASSI program:

- Remember that the program shows first the interaction among the input states and later this description might change. ALWAYS check the changing order of states.

- For spin-orbit coupling calculations don't forget to include the CASPT2 energies as input (EJOB or HDIAG keywords) because the results depend on the energy gap. In other cases having the CASPT2 energies as input will help you to get the right oscillator strength and Einstein coefficient in the final table.
- If you have degenerate states be sure that the CASPT2 energies are degenerate. If they are not (which is common) average the energies for the degenerate set (the two components of E symmetry for example).
- Remember that the spin-orbit coupled results (e.g. TDM) depend on the number of interacting singlet and triplet states included in RASSI.

4.14.7 Geometry optimization

- Not all methods have analytical derivatives.
- Default thresholds in slapaf are typically too tight. Do not waste computer time!
- Use constrained optimization
- For minima on flat hypersurfaces, such in loosely bound fragments, or in slow convergence cases you might have to decrease the CUTOFF threshold in ALASKA
- Be careful with the bond angle definition if you are close to a linear bond. You may have to switch to the LAngle definition
- Don't forget that CASSCF does not include dynamical correlation. In some cases you better change to DFT or numerical CASPT2 optimizations or, if this is not feasible, may be preferable to run RASSCF optimizations

- Poor active spaces may lead you to symmetry broken wrong solutions (e.g. a C_s minimum for water below the true C_{2v} one)
- Poor geometry convergence might be reduced or at least controlled by reducing the initial trust radius with the MAXSTEP keyword or/and by doing the optimization in Cartesian coordinates (CARTESIAN)
- In order to obtain localized solutions it might be a good idea to feed the program with a slightly distorted geometry that helps the method to reach the non symmetric solutions. Other possibilities are to use an electric field, to add a charge far from the system or use a solvent cavity. In all cases you break symmetry and allow less symmetric situations.

- Linearly interpolated internal coordinates geometries may be a good starting point to locate a transition state. Use also the useful FindTS command. Sometimes can be wise to compute a MEP from the TS to prove that it is relevant for the studied reaction path. Try also the new Saddle approach!
- When locating a CASSCF surface crossing (MECP) ALWAYS compute CASPT2 energies at that point. The gap between the states can be large at that level. In severe cases you might have to make a scan with CASPT2 to find a better region for the crossing.

- Remember that (so far) MOLCAS does not search for true conical intersections (CIs) but minimum energy crossing points (MECP) because it lacks NACMEs. Note however that typically computed minimum energy CIs (MECIs) may not be photochemically relevant if they are not easily accessible. Barriers have to be computed. Use MEPs!!

- Numerical hessians and optimizations may lead you to bad solutions when different electronic states are too close. As you move your calculation from the equilibrium geometry some of the points may belong to other state and corrupt your result. This might be the case for numerical CASPT2 crossing search. Use then MS-CASPT2 search.
- Remember that SA-RASSCF analytical gradients and SA-CASSCF analytical hessians are not implemented.
- Be careful with the change of roots and nature along a geometry optimization or MEP. For example, you start with the state in root 3 (at the CASSCF level) and reach a region of crossing root 3 and root 2. You may need to change to root 2 for your state. Not an easy solution (so far).

4.14.8 Solvent effects

- Some effects of the solvent are very specific, such as hydrogen bonds, and require to include explicit solvent molecules. Try adding a first solvent shell (optimized with molecular mechanics for instance) and then a cavity, for instance with PCM.
- Too small cavity sizes can lead you to unphysical solutions, even if they seem to match experiment.
- Remember using NonEquilibrium (final state) and RFRoot (SA-CASSCF) when required
- QM/MM is a much powerful strategy, but it requires experience and knowledge of the field

Bibliography

[1] Björn O. Roos, Valera Veryazov, and Per-Olof Widmark. Relativistic atomic natural orbital type basis sets for the alkaline and alkaline-earth atoms applied to the ground-state potentials for the corresponding dimers. *Theor. Chem. Acc.*, 111:345–351, 2004.

[2] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. Main group atoms and dimers studied with a new relativistic ANO basis set. *J. Phys. Chem. A*, 108:2851–2858, 2004.

[3] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. New relativistic ANO basis sets for transition metal atoms. *J. Phys. Chem. A*, 109:6575–6579, 2005.

[4] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. New relativistic ANO basis sets for actinide atoms. *Chem. Phys. Letters*, 409:295–299, 2005.

[5] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, Per-Olof Widmark, and Antonio Carlos Borin. New relativistic atomic natural orbital basis sets for lanthanide atoms with applications to the ce diatom and luf$_3$. *J. Phys. Chem. A*, 112:11431–11435, 2008.

[6] Björn O. Roos, Per-Åke Malmqvist, and Laura Gagliardi. Heavy element quantum chemistry – the multiconfigurational approach. In Erkki J. Brändas and Eugene S. Kryachko, editors, *Fundamental World of Quantum Chemistry. Vol. II*, pages 425–442. Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.

[7] Francesco Aquilante, Roland Lindh, and Thomas Bondo Pedersen. Unbiased auxiliary basis sets for accurate two-electron integral approximations. *J. Chem. Phys.*, 127:114107, 2007.

[8] Francesco Aquilante, Per-Åke Malmqvist, Thomas Bondo Pedersen, Abhik Ghosh, and Björn O. Roos. Cholesky decomposition-based multiconfiguration second-order perturbation theory (CD-CASPT2): Application to the spin-state energetics of co$^{\text{iii}}$(diiminato)(nph). *J. Chem. Theory Comput.*, 4:694–702, 2008.

[9] Francesco Aquilante, Thomas Bondo Pedersen, Björn O. Roos, Alfredo Sánchez de Merás, and Henrik Koch. Accurate *ab initio* density fitting for multiconfigurational self-consistent field methods. *J. Chem. Phys.*, 129:024113, 2008.

[10] Dongxia Ma, Giovanni Li Manni, and Laura Gagliardi. The generalized active space concept in multiconfigurational self-consistent field methods. *J. Chem. Phys.*, 135:044128, 2011.

[11] Björn O. Roos. The multiconfigurational (MC) self-consistent field (SCF) theory. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 177–254. Springer-Verlag, Berlin, Germany, 1992.

[12] James Finley, Per-Åke Malmqvist, Björn O. Roos, and Luis Serrano-Andrés. The multi-state CASPT2 method. *Chem. Phys. Letters*, 288:299–306, 1998.

Index