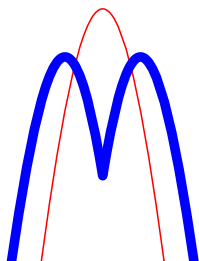


MOLPRO



Getting Started with Molpro *Version 2010.1*

H.-J. Werner

*Institut für Theoretische Chemie
Universität Stuttgart
Pfaffenwaldring 55
D-70569 Stuttgart
Federal Republic of Germany*

P. J. Knowles

*School of Chemistry
Cardiff University
Main Building, Park Place, Cardiff CF10 3AT
United Kingdom*

SHA1 833379180250ebe11052d872bd9ff53639690c67

(Copyright ©2008 University College Cardiff Consultants Limited)

References

All publications resulting from use of this program must acknowledge the following.

MOLPRO, version 2010.1, a package of *ab initio* programs, H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, P. Celani, T. Korona, R. Lindh, A. Mitrushenkov, G. Rauhut, K. R. Shamasundar, T. B. Adler, R. D. Amos, A. Bernhardsson, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, E. Goll, C. Hampel, A. Hesselmann, G. Hetzer, T. Hrenar, G. Jansen, C. Köppl, Y. Liu, A. W. Lloyd, R. A. Mata, A. J. May, S. J. McNicholas, W. Meyer, M. E. Mura, A. Nicklass, D. P. O'Neill, P. Palmieri, K. Pflüger, R. Pitzer, M. Reiher, T. Shiozaki, H. Stoll, A. J. Stone, R. Tarroni, T. Thorsteinsson, M. Wang, and A. Wolf, see <http://www.molpro.net>.

Some journals insist on a shorter list of authors; in such a case, the following should be used instead.

MOLPRO, version 2010.1, a package of *ab initio* programs, H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, and others, see <http://www.molpro.net>.

Depending on which programs are used, the following references should be cited.

Integral evaluation (SEWARD)

R. Lindh, U. Ryu, and B. Liu, *J. Chem. Phys.* **95**, 5889 (1991).

Integral-direct Implementation

M. Schütz, R. Lindh, and H.-J. Werner, *Mol. Phys.* **96**, 719 (1999).

MCSCF/CASSCF :

H.-J. Werner and P. J. Knowles, *J. Chem. Phys.* **82**, 5053 (1985);

P. J. Knowles and H.-J. Werner, *Chem. Phys. Lett.* **115**, 259 (1985).

See also:

H.-J. Werner and W. Meyer, *J. Chem. Phys.* **73**, 2342 (1980);

H.-J. Werner and W. Meyer, *J. Chem. Phys.* **74**, 5794 (1981);

H.-J. Werner, *Adv. Chem. Phys.* **LXIX**, 1 (1987).

Internally contracted MRCI:

H.-J. Werner and P. J. Knowles, *J. Chem. Phys.* **89**, 5803 (1988);

P. J. Knowles and H.-J. Werner, *Chem. Phys. Lett.* **145**, 514 (1988).

See also:

H.-J. Werner and E.A. Reinsch, *J. Chem. Phys.* **76**, 3144 (1982);

H.-J. Werner, *Adv. Chem. Phys.* **LXIX**, 1 (1987).

Excited states with internally contracted MRCI:

P. J. Knowles and H.-J. Werner, *Theor. Chim. Acta* **84**, 95 (1992).

Internally contracted MR-ACPF, QDVPT, etc:

H.-J. Werner and P. J. Knowles, *Theor. Chim Acta* **78**, 175 (1990).

The original reference to uncontracted MR-ACPF, QDVPT, MR-AQCC are:

R. J. Gdanitz and R. Ahlrichs, *Chem. Phys. Lett.* **143**, 413 (1988);

R. J. Cave and E. R. Davidson, *J. Chem. Phys.* **89**, 6798 (1988);

P. G. Szalay and R. J. Bartlett, *Chem. Phys. Lett.* **214**, 481 (1993).

Multireference perturbation theory (CASPT2/CASPT3):

H.-J. Werner, *Mol. Phys.* **89**, 645 (1996);

P. Celani and H.-J. Werner, *J. Chem. Phys.* **112**, 5546 (2000).

Coupling of multi-reference configuration interaction and multi-reference perturbation theory, P. Celani, H. Stoll, H.-J. Werner and P. J. Knowles, *Mol. Phys.* **102**, 2369 (2004).

Analytical energy gradients and geometry optimization

Gradient integral evaluation (*ALASKA*): R. Lindh, *Theor. Chim. Acta* **85**, 423 (1993);
 MCSCF gradients: T. Busch, A. Degli Esposti, and H.-J. Werner, *J. Chem. Phys.* **94**, 6708 (1991);
 MP2 and LMP2 gradients: A. El Azhary, G. Rauhut, P. Pulay, and H.-J. Werner, *J. Chem. Phys.* **108**, 5185 (1998);
 DF-LMP2 gradients: M. Schütz, H.-J. Werner, R. Lindh and F. R. Manby, *J. Chem. Phys.* **121**, 737 (2004).
 QCISD and LQCISD gradients: G. Rauhut and H.-J. Werner, *Phys. Chem. Chem. Phys.* **3**, 4853 (2001);
 CASPT2 gradients: P. Celani and H.-J. Werner, *J. Chem. Phys.* **119**, 5044 (2003).
 Geometry optimization: F. Eckert, P. Pulay and H.-J. Werner, *J. Comp. Chemistry* **18**, 1473 (1997);
 Reaction path following: F. Eckert and H.-J. Werner, *Theor. Chem. Acc.* **100**, 21, 1998.

Harmonic frequencies

G. Rauhut, A. El Azhary, F. Eckert, U. Schumann, and H.-J. Werner, *Spectrochimica Acta* **55**, 651 (1999).
 T. Hrenar, G. Rauhut, H.-J. Werner, *J. Phys. Chem. A* **110**, 2060 (2006).

Anharmonic frequencies (SURF, VSCEF, VCI):

G. Rauhut, *J. Chem. Phys.* **121**, 9313 (2004);
 T. Hrenar, H.-J. Werner, G. Rauhut, *J. Chem. Phys.* **126**, 134108 (2007);
 G. Rauhut, T. Hrenar, *Chem. Phys.* **346**, 160 (2008);
 M. Neff, G. Rauhut, *J. Chem. Phys.* **131**, 124129 (2009).

Møller-Plesset Perturbation theory (MP2, MP3, MP4):

Closed-shell Møller-Plesset Perturbation theory up to fourth order [MP4(SDTQ)] is part of the coupled cluster code, see CCSD.

Open-shell Møller-Plesset Perturbation theory (RMP2):

R. D. Amos, J. S. Andrews, N. C. Handy, and P. J. Knowles, *Chem. Phys. Lett.* **185**, 256 (1991).

Coupled-Cluster treatments (QCI, CCSD, BCCD):

C. Hampel, K. Peterson, and H.-J. Werner, *Chem. Phys. Lett.* **190**, 1 (1992) and references therein. The program to compute the perturbative triples corrections has been developed by M. J. O. Deegan and P. J. Knowles, *Chem. Phys. Lett.* **227**, 321 (1994).

Equation-of-Motion Coupled Cluster Singles and Doubles (EOM-CCSD):

T. Korona and H.-J. Werner, *J. Chem. Phys.* **118**, 3006 (2003).

Open-shell coupled-cluster (RCCSD, UCCSD):

P. J. Knowles, C. Hampel and H.-J. Werner, *J. Chem. Phys.* **99**, 5219 (1993); Erratum: *J. Chem. Phys.* **112**, 3106 (2000).

Local MP2 (LMP2):

G. Hetzer, P. Pulay, and H.-J. Werner, *Chem. Phys. Lett.* **290**, 143 (1998)
 M. Schütz, G. Hetzer, and H.-J. Werner, *J. Chem. Phys.* **111**, 5691 (1999)
 G. Hetzer, M. Schütz, H. Stoll, and H.-J. Werner, *J. Chem. Phys.* **113**, 9443 (2000)
 See also references on energy gradients and density fitting.

Local Coupled Cluster methods (LCCSD, LQCISD, LMP4):

C. Hampel and H.-J. Werner, *J. Chem. Phys.* **104** 6286 (1996)
 M. Schütz and H.-J. Werner, *J. Chem. Phys.* **114**, 661 (2001)

M. Schütz, Phys. Chem. Chem. Phys. 4, 3941 (2002)
See also references on energy gradients and density fitting.

Local triple excitations:

M. Schütz and H.-J. Werner, Chem. Phys. Lett. **318**, 370 (2000);
M. Schütz, J. Chem. Phys. **113**, 9986 (2000).
M. Schütz, J. Chem. Phys. **116**, 8772 (2002).

Density fitting methods:

DFT, Poisson fitting: F. R. Manby, P. J. Knowles, and A. W. Lloyd,
J. Chem. Phys. **115**, 9144 (2001).
DF-MP2, DF-LMP2: H.-J. Werner, F. R. Manby, and P. J. Knowles,
J. Chem. Phys. **118**, 8149 (2003).
DF-LCCSD: M. Schütz and F. R. Manby,
Phys. Chem. Chem. Phys. **5**, 3349 (2003)
DF-HF: R. Polly, H.-J. Werner, F. R. Manby, and Peter J. Knowles,
Mol. Phys. **102**, 2311 (2004).
DF-LMP2 gradients: M. Schütz, H.-J. Werner, R. Lindh and F. R. Manby,
J. Chem. Phys. **121**, 737 (2004).
DF-LCCSD(T): H.-J. Werner and M. Schütz,
in preparation.

Explicitly correlated methods with density fitting:

DF-MP2-R12: F. R. Manby, J. Chem. Phys. **119**, 4807 (2003).
DF-MP2-F12/2A: A. J. May and F. R. Manby, J. Chem. Phys. **121**, 4479 (2004).
DF-MP2-F12: H.-J. Werner, T. B. Adler, and F. R. Manby, J. Chem. Phys. **126**,
164102 (2007).
DF-RMP2-F12: G. Knizia and H.-J. Werner, J. Chem. Phys. **128**, 154103 (2008).
DF-LMP2-F12: H.-J. Werner and F. R. Manby, J. Chem. Phys., **124**, 054114 (2006);
F. R. Manby H.-J. Werner, T. B. Adler, and A. J. May, J. Chem. Phys.
124, 094103 (2006); H.-J. Werner, T. B. Adler, and F. R. Manby, J.
Chem. Phys. **126**, 164102 (2007); T. B. Adler, H.-J. Werner, and F.
R. Manby, J. Chem. Phys. **130**, 054106 (2009).
CCSD(T)-F12: T. B. Adler, G. Knizia, and H.-J. Werner, J. Chem. Phys. **127**, 221106
(2007).
UCCSD(T)-F12: G. Knizia, T. B. Adler, and H.-J. Werner, J. Chem. Phys. **130**, 054104
(2009).

Rangehybrid methods:

T. Leininger, H. Stoll, H.-J. Werner, and A. Savin, Chem. Phys. Lett. **275**, 151 (1997).
J.G. Ángyán, I.C. Gerber, A. Savin, and J. Toulouse, Phys. Rev. A **72**, 012510 (2005).
E. Goll, H.-J. Werner, and H. Stoll, Phys. Chem. Chem. Phys. **7**, 3917 (2005).

Full CI (FCD):

P. J. Knowles and N. C. Handy, Chem. Phys. Letters **111**, 315 (1984);
P. J. Knowles and N. C. Handy, Comp. Phys. Commun. **54**, 75 (1989).

Distributed Multipole Analysis (DMA):

A. J. Stone, Chem. Phys. Letters **83**, 233 (1981).

Valence bond:

D. L. Cooper, T. Thorsteinsson, and J. Gerratt, Int. J. Quant. Chem. **65**, 439 (1997);

D. L. Cooper, T. Thorsteinsson, and J. Gerratt, Adv. Quant. Chem. **32**, 51-67 (1998).

See also "An overview of the CASVB approach to modern valence bond calculations",

T. Thorsteinsson and D. L. Cooper, in *Quantum Systems in Chemistry and Physics. Volume 1: Basic problems and models systems*, eds. A. Hernandez-Laguna, J. Maruani, R. McWeeny, and S. Wilson (Kluwer, Dordrecht, 2000); pp 303-26.

Relativistic corrections using the Douglas-Kroll Hamiltonian:

M. Reiher, A. Wolf, JCP **121**, 2037–2047 (2004);

M. Reiher, A. Wolf, JCP **121**, 10945–10956 (2004);

A. Wolf, M. Reiher, B. A. Hess, JCP **117**, 9215–9226 (2002).

Spin-orbit coupling:

A. Berning, M. Schweizer, H.-J. Werner, P. J. Knowles, and P. Palmieri, Mol. Phys., **98**, 1823 (2000).

Diabatization procedures:

H.-J. Werner and W. Meyer, J. Chem. Phys. **74**, 5802 (1981);

H.-J. Werner, B. Follmeg, and M. H. Alexander, J. Chem. Phys. **89**, 3139 (1988);

D. Simah, B. Hartke, and H.-J. Werner, J. Chem. Phys. **111**, 4523 (1999).

DF-DFT-SAPT:

A. Heßelmann, G. Jansen and M. Schütz, J. Chem. Phys. **122**, 014103 (2005).

NMR shielding tensors:

S. Loibl, F.R. Manby, M. Schütz, Mol. Phys. **108**, 1362 (2010).

Contents

1	Preface	1
2	How an <i>ab initio</i> calculation works	1
3	How to run MOLPRO	3
4	How to prepare an input file	4
4.1	Geometry specification	4
4.2	Basis set specification	5
4.3	Method and wavefunction specification	6
4.3.1	Closed-shell Hartree-Fock calculations	6
4.3.2	Open-shell Hartree-Fock calculations	7
5	Single-reference electron correlation treatments	9
5.1	Closed-shell correlation methods	9
5.2	Open-shell correlation methods	10
6	MCSCF and CASSCF calculations	11
6.1	Complete Active Space Self-Consistent Field, CASSCF	11
6.2	Restricted Active Space self-consistent field, RASSCF	13
6.3	State-averaged MCSCF	15
6.4	MCSCF with selected configurations	15
7	Multireference electron correlation methods	16
7.1	Multireference configuration interaction (MRCI)	16
7.2	Multireference perturbation theory, CASPT2, CASPT3	17
8	Density functional calculations	18
9	Geometry optimization	19
10	Frequency calculations	20
11	Relativistic effects and pseudopotentials	20
12	Integral-direct calculations	22
13	Density fitting approximations	22

14 Local correlation treatments	23
14.1 Introduction	23
14.2 Invoking local correlation methods	24
14.3 Orbital localization	24
14.4 Choice of domains	25
14.4.1 Standard domains	25
14.4.2 Extended domains	25
14.5 Choice of pair classes	25
14.5.1 Options for selection of pair classes	25
14.6 Doing it right	26
14.6.1 Basis sets	26
14.6.2 Symmetry and Orientation	26
14.6.3 Localization	27
14.6.4 Orbital domains	27
15 Explicitly correlated methods	27
15.1 Options	28
15.2 Reference functions	29
15.3 Wave function Ansätze	29
15.3.1 The <i>general ansatz</i> (ansatz=3C (FULL))	29
15.3.2 The <i>diagonal ansatz</i> (ansatz=3C (D))	30
15.3.3 The <i>fixed amplitude ansatz</i> (ansatz=3C (FIX))	30
15.4 The correlation factor \hat{F}_{12}	30
15.5 Basis sets	30
15.6 Symmetry	31
15.7 CABS Singles correction	31
15.8 CCSD(T)-F12	32
16 Advanced features of MOLPRO	33
16.1 Memory control	33
16.2 Restarting calculations	33
16.3 Print options	34
16.4 Convergence thresholds	35
16.5 Variables	35
16.6 Program control using <code>do</code> loops, <code>if</code> blocks and <code>goto</code> commands	36
16.7 Tables and Plotting	38

16.8 Interface to MOLDEN	39
16.9 molproView	39

1 Preface

The intention of this *quickstart* manual is to get you started quickly. Most input is explained using simple examples. Default values are used as much as possible to make the inputs very simple. Naturally, this is by no means an exhaustive description of MOLPRO. A more systematic and complete explanation of all features of MOLPRO can be found in the reference manual. We strongly recommend that you read the introductory chapters of the reference manual once you have gained a basic feeling how things are done with MOLPRO. In particular, the first chapters of the reference manual explain in more detail the use of symmetry, and the data structures (files and records) in MOLPRO.

2 How an *ab initio* calculation works

The electronic structure of molecules can be treated only by quantum mechanics, since the electrons are very quickly moving particles. Of course, this manual cannot teach you the underlying theory, and it is assumed that you are familiar with it. We just want to remind you of some basic approximations, which are made in any *ab initio* calculation, independent of which program is used.

Firstly, the Born-Oppenheimer approximation is applied, which means that the nuclear and electronic motions are decoupled and treated separately (in some cases, non-adiabatic couplings are taken into account at a later stage). Thus, each electronic structure calculation is performed for a fixed nuclear configuration, and therefore the positions of all atoms must be specified in an input file. The *ab initio* program like MOLPRO then computes the electronic energy by solving the electronic Schrödinger equation for this fixed nuclear configuration. The electronic energy as function of the $3N-6$ internal nuclear degrees of freedom defines the potential energy surface (PES). The PES is in general very complicated and can have many minima and saddle points. The minima correspond to equilibrium structures of different isomers or molecules, and saddle points to transition states between them. The aim of most calculations is to find these structures and to characterize the potential and the molecular properties in the vicinity of the stationary points of the PES.

Secondly, the electronic Schrödinger equation cannot be solved exactly, except for very simple systems like the hydrogen atom. Therefore, the electronic wavefunction is represented in certain finite basis sets, and the Schrödinger equation is transformed into an algebraic equation which can be solved using numerical methods. There are two classes of approximations: one concerning the choice of basis functions to represent the one-electron functions called *molecular orbitals*, and one concerning the choice of N -electron functions to represent the many-electron electronic wavefunction.

In most programs, and also in MOLPRO, Gaussian basis functions are used to approximate the molecular orbitals, since the required integrals can be computed very quickly in this basis. Many optimized basis sets are available in the MOLPRO *basis set library*, and in most cases the basis set can be selected using a simple keyword in the input.

The many-electron wavefunction for the molecule is represented as a linear combination of antisymmetrized products (Slater determinants) of the molecular orbitals. In a *full configuration interaction* calculation (FCI) all possible Slater determinants for a given orbital basis are used, and this gives the best possible result for the chosen one-electron basis. However, the number of Slater determinants which can be constructed is enormous, and very quickly increases with the number of electrons and orbitals. Therefore, approximations have to be made, in which

the wavefunction is expanded in only a subset of all possible of Slater determinants (or configuration state functions (CSFs), which are symmetry adapted linear combinations of Slater determinants).

Once such approximations are introduced, it matters how the orbitals are determined. The simplest choice is to use a single Slater determinant and to optimize the orbitals variationally. This is the *Hartree-Fock* (HF) *self consistent field* (SCF) method, and it is usually the first step in any *ab initio* calculation.

In the Hartree-Fock approximation each electron moves in an average potential of the remaining electrons, but has no knowledge of the positions of these. Thus, even though the Coulomb interaction between the electrons is taken into account in an averaged way, the electrons with opposite spin are unable to avoid each other when they come close, and therefore the electron repulsion is overestimated in Hartree-Fock. The purpose of post-Hartree-Fock *electron correlation* methods is to correct for this by taking the instantaneous correlation of the electrons into account. The corresponding energy lowering is called *electron correlation energy*. There are many different methods available and implemented in MOLPRO to approximate and optimize the wavefunction, for instance Møller-Plesset (MP) perturbation theory, configuration interaction (CI), or coupled cluster (CC) methods. Also density functional (DFT) methods take into account electron correlation, even though in a less systematic and less well defined way than *ab initio* methods.

One point of warning should be noticed here: electron correlation treatments require much larger one-electron basis sets than Hartree-Fock or DFT to yield converged results. Such calculations can therefore be expensive. For a fixed basis set, a correlation calculation is usually much more expensive than a HF calculation, and therefore many unexperienced people are tempted to use small basis sets for a correlation calculation. However, this is not reasonable at all, and for meaningful calculations one should at least use a triple-zeta basis with several polarization functions (e.g. cc-pVTZ). Using explicitly-correlated methods discussed later, the basis set problem is much alleviated.

Finally, it should also be noted that the HF approximation, and all single reference methods which use the HF determinant as zeroth order approximation, are usually only appropriate near the equilibrium structures. In most cases they are not able to dissociate molecular bonds correctly, or to describe electronically excited or (nearly) degenerate states. In such cases multireference methods, which use a multiconfiguration SCF wavefunctions (MCSCF) as zeroth order approximation, offer a reasonable alternative. Complete active space SCF (CASSCF) is a special variant of MCSCF. In MOLPRO various multireference electron correlation methods are implemented, e.g., multireference perturbation theory (MRPT, CASPT2) and multireference configuration interaction (MRCI), and variants of these such as multireference coupled-pair functional (MR-ACPF).

As you will see, it is quite easy to run an electronic structure calculation using MOLPRO, and probably you will have done your first successful run within the next 10 minutes. However, the art is to know which basis set and method to use for a particular problem in order to obtain an accurate result for a minimum possible cost. This is something which needs a lot of experience and which we cannot teach you here. We can only encourage you not to use MOLPRO or any other popular electronic structure program simply as a black box without any understanding or critical assessment of the methods and results!

Literature: *Introduction to Computational Chemistry*, F. Jensen, Wiley, 2006

3 How to run MOLPRO

In order to run MOLPRO you have first to write an input file. This can be done in any directory of your choice. Input files can have any name, but it is best to avoid using a file name with extension `.out`. It is often convenient to name the input file as `h2o.inp`, `benzene.inp`, depending on the molecule, or whatever you like.

Simple examples for input files will be given in the following sections. Once the input file is ready, a MOLPRO calculation can be started using the `molpro` command. Assume the input file is called `h2o.inp`. The command to run MOLPRO is then simply

```
molpro h2o.inp &
```

This will create an output file `h2o.out`, i.e. the extension `.inp` is replaced by `.out`. The same would happen with any other extension, e.g., `h2o.com`, `h2o.input`, `h2o.test` would all produce the output in `h2o.out`. If `h2o.out` already exists, the old output will be moved to a backup directory that is, by default, created with the name `h2o.d`. In this way old outputs are not lost. This mechanism can be disabled using the `-s` option:

```
molpro -s h2o.inp &
```

would not save an existing output file but simply overwrite it. If you want to give your output a different name than the default one, you can use the `-o` option, for instance

```
molpro -o water.output h2o.inp &
```

As well as producing the `.out` file, a structured XML file (which is actually a parent of the `.out` file) is created with suffix `.xml`. This file can be useful for automated post-processing of results, for example graphical rendering by the MolproView program.

There are many other options for the `molpro` command, most of which, however, do not often need to be specified. You can find a full description in the MOLPRO reference manual. Some of the more important ones are

1. The amount of memory MOLPRO is allowed to use can be specified using the `-m` option, e.g.,

```
molpro -m 4M h2o.inp &
```

This means that 4 megawords (MW) of memory are allocated by `molpro` (see also `memory` input card, section 16.1).

2. For running in parallel (assuming that the program has been suitably compiled),

```
molpro -n 8 h2o.inp &
```

will run 8 cooperative processes. Depending on how the program has been built, this will result in 7- or 8-way parallel execution.

3. Optionally, but not by default, it is possible so specify one or more named files in the input, and these are used by the program to store intermediate parameters and results, that can be used to restart (see section 16.2). If nothing is given in the input, these files are still present, but they are temporary, and disappear at the end of the job. For performance reasons, the program runs in scratch directory, for example a high-performance file system. By default, the location of this directory is `/tmp/$USER`, or taken from the environment variable `TMPDIR` (if set), but it can be controlled using the `-d` option, for example

```
molpro -d /scratch/$USER h2o.inp &
```

At the end of the job, the named files are copied to a permanent location. The most important file for a restart is file 2, which contains the wave function information (cf.

section 16.2). By default, this file is copied into the directory \$HOME/wfu, but this directory can be changed with the -W option. At the beginning of the job, if appropriate files exist in the permanent location, they are copied in to the scratch directory.

4 How to prepare an input file

As explained in the introduction, the input file should contain the following minimum information.

- Geometry specification.
- Specification of one-electron basis set (although there is a default).
- Specification of the method to compute the many-electron wavefunction. Several calculations of different type can follow each other.

We will consider these three in turn.

All input can be given in upper or lower case and in free format. Most input lines start with a keyword, which is followed by numbers or other specifications (options). The different entries on each line should be separated by commas (see next section for more explanation about this).

4.1 Geometry specification

The simplest way to specify the coordinates of the atoms is to use cartesian coordinates and the XYZ input format, which is standard in many programs. In this case the geometry input looks as follows (example for formaldehyde).

```
geometry={
  4
  FORMALDEHYDE
  C      0.0000000000      0.0000000000      -0.5265526741
  O      0.0000000000      0.0000000000      0.6555124750
  H      0.0000000000     -0.9325664988     -1.1133424527
  H      0.0000000000      0.9325664988     -1.1133424527
}
```

As seen in the example, the geometry is specified within the *geometry block*, enclosed by `geometry={` and `}`. The first line of the *xyz* geometry block holds the number of atoms (free format). The second line is an arbitrary title. Finally there is one line for each atom specifying the cartesian coordinates (x,y,z) in Ångström. For simplicity, the first two lines can also be omitted.

Alternatively, the geometry can be specified in the *Z-matrix* form, which is also used in many other programs. In this case, the geometry is defined by distances and angles. This is a little more involved, and here we give only a simple example, again for formaldehyde.

```
angstrom
geometry={
  C
  O  C  1.182
  H1 C  1.102  O  122.1789
  H2 C  1.102  O  122.1789  H1  180
}
```

Here, 1.182 and 1.102 are the C-O and the C-H bond distances, respectively. 122.1789 is the H-O-C angle, and 180 is the angle between the H1-C-O and H2-C-O planes (dihedral angle). Note that by default the bond distances are in atomic units (bohr), but one can give the `angstrom` keyword to use Ångström.

As an alternative to the `xyz` input explained above, it is also possible to specify cartesian coordinates in a `Z`-matrix. In this case the form is

```
angstrom
geometry={
C,,      0.0000000000 ,      0.0000000000 ,      -0.5265526741
O,,      0.0000000000 ,      0.0000000000 ,      0.6555124750
H,,      0.0000000000 ,     -0.9325664988 ,     -1.1133424527
H,,      0.0000000000 ,      0.9325664988 ,     -1.1133424527
}
```

Again, atomic units are the default, other than for the `xyz` input, where the coordinates need to be given in Ångström. In order to use Ångström, the `angstrom` command must be given before the geometry block.

Instead of constant numbers it is also possible to use variables in the `Z`-matrix input. For instance, the input for formaldehyde can be written as

```
geometry={
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree
```

The values of the variables can be changed in the course of the calculations, so that calculations can be performed for different geometries in one run. This will be explained in more detail later.

By default, the program repositions and rotates the molecule so that the coordinate axes have the centre of mass of the molecule as origin, and are aligned with the principal inertial axes. This behaviour is usually what is required, in order that the program can recognize and exploit point-group symmetry, but can be changed through parameters specified on the `ORIENT` command (see manual). By default, the program tries to find and use the maximum abelian point-group symmetry, but this behaviour can be controlled using the `SYMMETRY` command (see manual).

4.2 Basis set specification

In many cases the one-electron basis set can be specified in the simple form

```
basis=name
```

where *name* refers to the basis set name in the library. You can find all possible basis sets on the MOLPRO Web page. Popular and often used for high-level calculations are the correlation-consistent polarized basis sets of Dunning and coworkers, denoted `cc-pVDZ` (double zeta), `cc-pVTZ` (triple zeta), `cc-pVQZ` (quadruple zeta), `cc-pV5Z` (quintuple zeta). These names can be used directly or abbreviated by `VDZ`, `VTZ`, `VQZ` and so on. For computing certain properties like electron affinities, polarizabilities, or intermolecular potentials additional diffuse basis

functions are needed, and such functions are included in the augmented correlation-consistent basis sets, denoted `aug-cc-pVDZ`, `aug-cc-pVTZ` etc. These names can be abbreviated by `AVDZ`, `AVTZ`, `AVQZ` etc.

Another popular choice are the Gaussian basis sets of Pople and coworkers, e.g. `6-31G**` (double zeta), `6-311G(2DF,2PD)` (triple zeta), `6-311G++(2DF,2PD)` (augmented triple zeta) etc. It should be noted here that by default MOLPRO uses spherical harmonics (*5d*, *7f* etc), while Gaussian uses cartesian functions (*6d*, *10f* etc). This yields slightly different energies. Cartesian functions can also be used in MOLPRO, but then the keyword

```
cartesian
```

has to be given before the basis set specification.

It is also possible to use different basis sets for different atoms. In this case the simplest input reads for example

```
basis,c=vtz,o=avtz,h=vdz
```

Basis sets can also be specified manually (exponents and contraction coefficients). Refer to the MOLPRO reference manual for details.

If no basis set is specified at all, MOLPRO uses `cc-pVDZ`, but this does not mean that this is a sensible choice!

4.3 Method and wavefunction specification

After defining the geometry and basis set (in any order) one has to specify the methods to be used. This is simply done by keywords, which are normally the same as the usual abbreviations for the methods. In most cases, the first step is a Hartree-Fock calculation, in which the molecular orbitals to be used in subsequent electron correlation treatments are optimized.

4.3.1 Closed-shell Hartree-Fock calculations

The Hartree-Fock program is invoked by the command

```
hf
```

The program will then first compute the one- and two-electron integrals and store these on disk. Once this step is completed, the self-consistent field calculation, which is iterative, is carried out, using the precomputed integrals (for integral-direct calculations, in which the integrals are not stored but recomputed whenever needed, see section 12).

Now you are ready to try your first MOLPRO calculation. The complete input for a HF calculation for formaldehyde is

```
***,formaldehyde
print,basis,orbitals ! this is optional: print the basis set and
                    ! the occupied orbitals
geometry={          ! define the nuclear coordinates
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
```

```

rch=1.102 Ang
hco=122.1789 Degree

basis=vdz          ! Select basis set
hf                 ! Invoke Hartree-Fock program
---
```

The first line, starting with `***` is optional and is used to specify a title. The last line, `---` is also optional. This indicates the end of the input; any further input after `---` is ignored. Any text written after an exclamation mark is treated as comment and also ignored. Blanks and blank lines have no effect.

If you inspect the output, you will find that MOLPRO detected that the molecule has C_{2v} symmetry, and that there are 5 orbitals belonging to the irreducible representation a_1 , one belonging to b_1 , and two to b_2 . In MOLPRO the irreducible representations are numbered, and in the C_{2v} group the numbering is 1–4 for a_1 , b_1 , b_2 , and a_2 , respectively. The third orbital in symmetry a_1 is denoted 3.1, the second orbital in symmetry b_2 is denoted 2.3. Please take a little time to study the output in order to understand these conventions.

In the current case, MOLPRO has worked out the number of occupied orbitals in each symmetry automatically according to the Aufbau principle. This works in most but not all cases. You can specify the number of occupied orbitals in each symmetry using the `occ` directive. In the current case, it would read

```
occ, 5, 1, 2
```

and the meaning of this should be quite obvious from the above.

If special directives are given for a command, like above `occ` is a directive for command `hf`, the command and the associated directives have to be enclosed by curly brackets, e.g.

```
{hf                 ! Invoke Hartree-Fock program
occ, 5, 1, 2}      ! Specify number of occupied orbitals in each irrep
```

This is called a command block. The curly brackets are needed to avoid ambiguities, since some directives can also be used outside command blocks. Please refer to the manual for more details.

Each command or directive can either begin on a new line, or be separated by semicolons. For instance

```
{hf;occ, 5, 1, 2}
```

is equivalent to the previous example.

4.3.2 Open-shell Hartree-Fock calculations

In the above example all orbitals are doubly occupied, and therefore the total symmetry of the ground state wavefunction, which is the direct product of the spin-orbital symmetries, is A_1 . Since all electrons are paired, it is a singlet wavefunction, 1A_1 .

Even though most stable molecules in the electronic ground states are closed-shell singlet states, this is not always the case. Open-shell treatments are necessary for radicals or ions. As a first simple example we consider the positive ion of formaldehyde, H_2CO^+ . In this case it turns out that the lowest cation state is 2B_2 , i.e., one electron is removed from the highest occupied orbital in symmetry b_2 (denoted 2.3 in MOLPRO, see above). By default MOLPRO assumes that the

number of electrons equals the total nuclear charge. Therefore, in order to compute an ion, one has to specify the number of electrons. Alternatively, the total charge of the molecule can be given (see below). Furthermore, one has to specify the symmetry and spin of the wavefunction. This is done using the `wf` directive (`wf` stands for *wavefunction*):

```
wf, 15, 3, 1
```

The first entry on a `wf` card is the number of electrons. The second entry is the total symmetry of the wavefunction. For a doublet this equals the symmetry of the singly occupied molecular orbital. Finally, the third entry specifies the number of singly occupied orbitals, or, more generally the total spin. Zero means singlet, 1 doublet, 2 triplet and so on. Alternatively, the `WF` card can be written in the form

```
wf, charge=1, symmetry=3, spin=1
```

where now the total charge of the molecule instead of the number of electrons is given. In this case the number of electrons is computed automatically from the nuclear and total charges.

In summary, the input for H_2CO^+ is

```
{geometry specification}
{basis specification}

{hf          !invoke spin restricted Hartree-Fock program (rhf can also be used)
occ,5,1,2    !number of occupied orbitals in the irreps a1, b1, b2, respectively
wf,15,3,1}  !define wavefunction: number of electrons, symmetry and spin
---
```

Note the curly brackets, which are required and enclose the command block `hf`.

As a second example we consider the ground state of O_2 , which is ${}^3\Sigma_g^-$. The geometry specification is simply

```
geometry={      !geometry specification, using z-matrix
o1
o2, o1, r
}

r=2.2 bohr    !bond distance
```

MOLPRO is unable to use non-abelian point groups, and can therefore only use D_{2h} in the present case. The axis of a linear molecule is placed on the z -axis of the coordinate system. Then the symmetries of the σ_g , $\pi_{u,x}$, $\pi_{u,y}$, σ_u , $\pi_{g,x}$, $\pi_{g,y}$ orbitals are 1,2,3,5,6,7, respectively. It is easier to remember that the irreducible representations in D_{2h} are carried by the functions (molpro symmetry numbers in parenthesis) $s(1)$, $x(2)$, $y(3)$, $xy(4)$, $z(5)$, $xz(6)$, $yz(7)$, $xyz(8)$. The order in C_{2v} is the same, but then there are only the first four irreducible representations.

The electron configuration of the electronic ground state of O_2 is

$$1\sigma_g^2 1\sigma_u^2 2\sigma_g^2 2\sigma_u^2 3\sigma_g^2 1\pi_{u,x}^2 1\pi_{u,y}^2 1\pi_{g,x}^1 1\pi_{g,y}^1.$$

Thus, the number of occupied orbitals in the 8 different irreducible representations of the D_{2h} point group are specified as

```
occ, 3, 1, 1, 0, 2, 1, 1, 0
```

The product symmetry of the singly occupied orbitals 1.6 (xz) and 1.7 (yz) is 4 (xy), and therefore the symmetry of the total wavefunction is 4 (please refer to the MOLPRO reference manual for a more complete account of symmetry groups and the numbering of irreducible representations). Thus, the `wf` card reads

```
wf,16,4,2      !16 electrons, symmetry 4, triplet (2 singly occupied orbitals)
```

This is still not unambiguous, since the product symmetry of $\pi_{u,x}$ (2) and $\pi_{u,y}$ (3) is also 4, and therefore the program might not be able to decide if it should singly occupy the π_u or π_g orbitals. Therefore, the singly occupied orbitals can be specified using the open directive:

```
open,1.6,1.7
```

This now defines the wavefunction uniquely. In summary, the input for O₂ reads

```
***,O2
print,basis,orbitals
geometry={                !geometry specification, using z-matrix
o1
o2,o1,r
}

r=2.2 bohr                !bond distance

basis=vtz                 !triple-zeta basis set

{hf                        !invoke RHF program
wf,16,4,2                 !define wavefunction: 16 electrons, symmetry 4, triplet
occ,3,1,1,,2,1,1         !number of occupied orbitals in each symmetry
open,1.6,1.7}            !define open shell orbitals
```

In fact, the last 2 lines are not necessary in the present case, since the correct configuration can be automatically determined using the Aufbau principle, but this might not always be true.

5 Single-reference electron correlation treatments

Once the Hartree-Fock calculation is done, electron correlation effects can be taken into account using various methods. Each available method in MOLPRO is specified by a different keyword.

5.1 Closed-shell correlation methods

For closed-shell calculations the following methods/keys are available:

```
mp2      !second-order Moeller-Plesset perturbation theory
lmp2     !second-order local Moeller-Plesset perturbation theory
mp3      !third-order Moeller-Plesset perturbation theory
mp4      !fourth-order Moeller-Plesset perturbation theory
ci       !singles and doubles configuration interaction (using MRCI program)
cisd     !singles and doubles configuration interaction (using CCSD program)
cepa(1)  !Coupled electron pair approximation, version 1
cepa(2)  !Coupled electron pair approximation, version 2
cepa(3)  !Coupled electron pair approximation, version 3
acpf     !Averaged couplec pair functional
ccsd     !Coupled cluster with singles and doubles
ccsd(t)  !Coupled cluster with singles and doubles and perturbative
          !treatment of triples
bccd     !Brueckner coupled cluster with doubles
bccd(t)  !Brueckner coupled cluster with doubles and perturbative
          !treatment of triples
qcisd    !Quadratic configuration interaction with singles and doubles
qcisd(t) !Quadratic configuration interaction with perturbative
          !treatment of triples
```

One or more of these commands should be given after the Hartree-Fock input. Note that some methods include others as a by-product; for example, it is wasteful to ask for `mp2` and `mp4`, since the MP4 calculation returns all of the second, third and fourth order energies. CCSD also returns the MP2 energy.

By default, only the valence electrons are correlated. To modify the space of uncorrelated inner-shell (core) orbitals, the `core` directive can be used, on which the numbers of core orbitals in each symmetry are specified in the same way as the occupied orbitals on the `occ` card. In order to correlate all electrons in a molecule, use

```
core
```

without any further arguments (all entries zero). This card must follow the directive for the method, e.g.,

```
{MP2      !second-order Moeller-Plesset perturbation theory
core}     !correlate all electrons
```

Note, however, that special basis sets are needed for correlating inner shells, and it does not make much sense to do such calculations with the standard basis sets described above. The correlation-consistent core-valence basis sets (cc-pCV x Z where x is D, T, Q, 5, ...) are available for this purpose.

```
{MP2      !second-order Moeller-Plesset perturbation theory
core,2}   !don't correlate electrons in the orbitals 1.1 and 2.1.
```

The number of occupied orbitals is automatically remembered from the preceding HF calculation. If necessary for special purposes, it can be specified using `occ` cards as in HF. The orbitals are taken from the most recent HF calculation in the input. See the MOLPRO reference manual for other choices of orbitals.

Example for a complete CCSD(T) calculation for formaldehyde:

```
***,formaldehyde
print,basis,orbitals !this is optional: print the basis set and the occupied orbitals
angstrom
geometry={           !define the nuclear coordinates
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

basis=vdz           !Select basis set
hf                  !Perform HF calculation
ccsd(t)            !Perform CCSD(T) calculation
```

5.2 Open-shell correlation methods

For open shell the following single-reference methods are implemented in MOLPRO:

```

ump2      !second-order Moeller-Plesset perturbation theory with UHF reference
rmp2      !second-order Moeller-Plesset perturbation theory with RHF reference.
ci        !singles and doubles configuration interaction (using the MRCI program)
rcisd     !spin-restricted singles and doubles configuration interaction
ucisd     !spin-unrestricted singles and doubles configuration interaction
rccsd     !partially spin adapted coupled cluster with singles and doubles
rccsd(t) !partially spin adapted coupled cluster with perturbative
          !treatment of triples
uccsd     !spin-unrestricted coupled cluster with singles and doubles
uccsd(t) !spin-unrestricted coupled cluster with perturbative
          !treatment of triples

```

Note that all methods except `ump2` use spin-restricted Hartree-Fock (RHF) reference functions. This is also the case for the `ucisd` and `uccsd` methods, in which the correlated wavefunction may be slightly spin contaminated, even though the RHF reference function is spin adapted. `ump2` is not generally recommended, since spin contamination can lead to large errors.

Core orbitals can be specified as for the closed-shell methods. The number of electrons and occupations as well as the orbitals are taken from the most recent RHF calculation. It is possible to modify these defaults using `occ`, `closed`, `wf`, or `orbital` directives; see the MOLPRO reference manual for details.

The following is an example of a complete CCSD(T) calculation for O₂.

```

***,O2
print,basis,orbitals
geometry={                !geometry specification, using z-matrix
o1
o2,o1,r
}

r=2.2 bohr                !bond distance

basis=vtz                 !triple-zeta basis set

{hf                       !invoke RHF program
wf,16,4,2                 !define wavefunction: 16 electrons, symmetry 4, triplet
occ,3,1,1,,2,1,1         !number of occupied orbitals in each symmetry
open,1.6,1.7}            !define open shell orbitals

rccsd(t)                  !perform partially spin adapted CCSD(T)

```

6 MCSCF and CASSCF calculations

In the MCSCF method a multiconfiguration wavefunction is variationally optimized with respect to simultaneous variations of the orbitals and configuration coefficients. As explained in section 4.3, the Hartree-Fock wavefunction is uniquely defined by specifying the number of occupied orbitals in each symmetry and possibly the singly occupied orbitals. In the MCSCF case some more information is necessary. We will begin to explain the input for CASSCF (complete active space SCF), which is the simplest case.

6.1 Complete Active Space Self-Consistent Field, CASSCF

In a CASSCF wavefunction the occupied orbital space is divided into a set of *inactive* or *closed-shell* orbitals and a set of *active* orbitals. All inactive orbitals are doubly occupied in each Slater determinant. On the other hand, the active orbitals have varying occupations, and all possible

Slater determinants (or CSFs) are taken into account which can be generated by distributing the $N_{act} = N_{el} - 2m_{closed}$ electrons in all possible ways among the active orbitals, where m_{closed} is the number of closed-shell (inactive) orbitals, and N_{el} is the total number of electrons. Thus, it corresponds to a full CI in the active space.

The CASSCF program is invoked using the

```
casscf
```

Aliases for this command are `mcscf` or `multi`. This command is optionally followed by further input defining the wavefunction. The inactive orbital space is defined using the `closed` directive.

```
closed,n1,n2,...,nd
```

where n_i is the number of doubly occupied (inactive) orbitals in irreducible representation i . The total number of occupied orbitals is specified using the `occ` directive, as in Hartree-Fock.

```
occ,m1,m2,...,md
```

where $m_i \geq n_i$. The number of active orbitals in irreducible representations i is then $m_i - n_i$. Note that the inactive orbitals are always the first in each symmetry, i.e., inactive and active spaces cannot be mixed. The number of electrons, as well as the symmetry of the wavefunction and the total spin is specified using the `wf` directive, as explained already for open-shell HF:

```
wf, Nel, isym, ms2
```

where `isym` is the symmetry of the total wavefunction (the direct product of the symmetries of all occupied spin orbitals), and `ms2` = $2S$ defines the spin (0=singlet, 1=doublet, 2=triplet etc).

From the above it follows that the number of active electrons is

$$N_{act} = N_{el} - 2 \sum_i^{m_{closed}} n_i \quad (1)$$

By default, the inactive space consists of all inner-shell orbitals, and the active space of all valence orbitals which are obtained from the atomic valence orbitals (full valence active space). The default number of electrons equals the sum of nuclear charges, the default wavefunction symmetry is 1 and singlet. The default starting guess for the orbitals is taken from the most recent orbital optimization, e.g., Hartree-Fock. The simplest input for a CASSCF calculation for formaldehyde is therefore

```
***,formaldehyde
print,orbitals,civector      !this is optional: print the occupied orbitals
                              !and the CI vector
                              !by default, only coefficients larger than 0.05
                              !are printed.

angstrom
geometry={                   !define the nuclear coordinates
C
O  , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

basis=vdz                    !Select basis set
```

```
hf                !Perform HF calculation
casscf           !Perform CASSCF calculation,
                !using the HF orbitals as starting guess.
```

In this case, the carbon and oxygen $1s$ orbitals are inactive, and the carbon and oxygen $2s$, $2p$ as well as the hydrogen $1s$ orbitals are active. This corresponds to the following input, which could be given after the `casscf` directive:

```
{casscf
closed,2          !2 inactive orbitals in Symmetry 1 (a1)
occ,7,2,3        !7a1, 2b1, 3b2 occupied orbitals
wf,16,1,0}      !16 electrons, Symmetry 1 (A1), singlet
```

Thus, there are five a_1 , two b_1 , and three b_2 active orbitals. This yields 3644 CSFs or 11148 Slater determinants. *Note that the `wf` directive must be given after the `occ` and `closed` ones.* A shorter expansion results if the $2s$ orbital of oxygen is made inactive. In this case the input would be

```
{casscf
closed,3          !3 inactive orbitals in Symmetry 1 (a1)
occ,7,2,3        !7a1, 2b1, 3b2 occupied orbitals
wf,16,1,0}      !16 electrons, Symmetry 1 (A1), singlet
```

and now only 1408 CSFs or 4036 Slater determinants are generated.

6.2 Restricted Active Space self-consistent field, RASSCF

Since the number of CSFs or Slater determinants and thus the computational cost quickly increases with the number of active orbitals, it may be desirable to use a smaller set of CSFs. One way to make a selection is to restrict the number of electrons in certain subspaces. One could for instance allow only single and double excitations from some strongly-occupied subset of active orbitals, or restrict the number of electrons to at most 2 in another subset of active orbitals. In general, such restrictions can be defined using the `restrict` directive:

```
restrict,min, max, orbital list
```

where *min* and *max* are the minimum and maximum number of electrons in the given orbital subspace, as specified in the *orbital list*. Each orbital is given in the form *number.symmetry*, e.g. 3.2 for the third orbital in symmetry 2. The `restrict` directives (several can follow each other) must be given after the `wf` card. As an example, consider the formaldehyde example again, and assume that only single and double excitations are allowed into the orbitals 6.1, 7.1, 2.2, 3.3, which are unoccupied in the HF wavefunction. Then the input would be

```
{casscf
closed,2          !2 inactive orbitals in Symmetry 1 (a1)
occ,7,2,3        !7a1, 2b1, 3b2 occupied orbitals
wf,16,1,0        !16 electrons, Symmetry 1 (A1), singlet
restrict,0,2, 6.1,7.1,2.2,3.3} !max 2 electrons in the given orbital list
```

One could further allow only double excitations from the orbitals 3.1, 4.1, but in this case this has no effect since no other excitations are possible anyway. In order to demonstrate such a case, we increase the number of occupied orbitals in symmetry 1 to 8, and remove the restriction for orbital 6.1.

```
{casscf
closed,2                !2 inactive orbitals in Symmetry 1 (a1)
occ,8,2,3              !7a1, 2b1, 3b2 occupied orbitals
wf,16,1,0              !16 electrons, Symmetry 1 (A1), singlet
restrict,0,2, 7.1,8.1,2.2,3.3 !max 2 electrons in the given orbital list
restrict,2,4, 3.1,4.1} !at least 2 and max 4 electrons in the
                       !given orbitals
```

It is found that this calculation is not convergent. The reason is that some orbital rotations are almost redundant with single excitations, i.e., the effect of an orbital transformation between the strongly and weakly occupied spaces can be expressed to second order by the single and double excitations. This makes the optimization problem very ill conditioned. This problem can be removed by eliminating the single excitations from / into the restricted orbital space as follows:

```
{casscf
closed,2                !2 inactive orbitals in Symmetry 1 (a1)
occ,8,2,3              !7a1, 2b1, 3b2 occupied orbitals
wf,16,1,0              !16 electrons, Symmetry 1 (A1), singlet
restrict,0,2, 7.1,8.1,2.2,3.3 !max 2 electrons in the given orbital list
restrict,-1,0 7.1,8.1,2.2,3.3 !1 electron in given orbital space not
                               !allowed (no singles)
restrict,2,4, 3.1,4.1    !at least 2 and max 4 electrons in the
                               !given orbitals
restrict,-3,0,3.1,4.1} !3 electrons in given orbital space not
                               !allowed (no singles)
```

and now the calculation converges smoothly.

Converging MCSCF calculations can sometimes be tricky and difficult. Generally, CASSCF calculations are easier to converge than restricted calculations, but even in CASSCF calculations problems can occur.

The reasons for slow or no convergence could be one or more of the following.

- Near redundancies between orbital and CI coefficient changes as shown above.
Remedy: eliminate single excitations
- Two or more weakly occupied orbitals have almost the same effect on the energy, but the active space allows the inclusion of only one of them.
Remedy: increase or reduce active space (occ card).
- An active orbital has an occupation number very close to two. The program may have difficulties to decide which orbital is inactive.
Remedy: increase inactive space
- Correlation of an active orbital gives a smaller energy lowering than would be obtained by correlating an inactive orbital. The program tries to swap active and inactive orbitals.
Remedy: reduce (or possibly increase) active space.
- Another state of the same symmetry is energetically very close (nearly degenerate). The program might oscillate between the states (root flipping).
Remedy: include all nearly degenerate states into the calculation, and optimize their average energy (see below).

As a rule of thumb, it can be said that if a CASSCF calculation does not converge or converges very slowly, the active or inactive space is not chosen well.

6.3 State-averaged MCSCF

In order to compute excited states it is usually best to optimize the energy average for all states under consideration. This avoids root-flipping problems during the optimization process and yields a single set of compromise orbitals for all states.

The number of states to be optimized in a given symmetry is specified on a `state` directive, which must follow directly after the `wf` directive, e.g.,

```
wf,16,1,0;state,2 !optimize two states of symmetry 1
```

It is also possible to optimize states of different symmetries together. In this case several `wf / state` directives can follow each other, e.g.,

```
wf,16,1,0;state,2 !optimize two states of symmetry 1
wf,16,2,0;state,1 !optimize one states of symmetry 2
```

etc. Optionally also the weights for each state can be specified, e.g.

```
wf,16,1,0;state,2;weight,0.2,0.8 !optimize two states of symmetry 1
                                !first state has weight 0.2,
                                !second state weight 0.8
```

By default, the weights of all states are identical, which is normally the most sensible choice. The following example shows a state-averaged calculation for O₂, in which the valence states ($^3\Sigma_g^-$, $^1\Delta_g$) are treated together.

```
***,O2
print,basis,orbitals
geometry={                      !geometry specification, using z-matrix
o1
o2,o1,r
}

r=2.2 bohr                      !bond distance

basis=vtz                       !triple-zeta basis set

{hf                              !invoke RHF program
wf,16,4,2                       !define wavefunction: 16 electrons, symmetry 4, triplet
occ,3,1,1,,2,1,1               !number of occupied orbitals in each symmetry
open,1.6,1.7}                  !define open shell orbitals

{casscf                          !invoke CASSCF program
wf,16,4,2                       !triplet Sigma-
wf,16,4,0                       !singlet delta (xy)
wf,16,1,0}                      !singlet delta (xx - yy)
```

Note that averaging of states with different spin-multiplicity, as in the present examples, is possible only for CASSCF, but not for more restricted RASSCF or MCSCF, wavefunctions.

6.4 MCSCF with selected configurations

MCSCF with arbitrary selected configuration spaces can also be performed. The only restriction is that always all CSFs with different spin couplings arising from the same orbital occupancy are included. There are two ways to select configurations. Either, they are selected from a previous

calculation with a threshold, or they are defined explicitly in the input. Here we only describe the latter case; for more details please refer to the reference manual.

The configuration input starts with the `select` directive, and subsequently each orbital configuration is given by one line starting with the keyword `con`, followed by the occupation numbers of the active orbitals. The following example shows an input for formaldehyde.

```

***,formaldehyde
angstrom
geometry={                !define the nuclear coordinates
C
O  , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

basis=vdz                !Select basis set
hf                        !Perform HF calculation
{mcscf                   !invoke MCSCF program
closed,4,,1              !4a1 and 1b2 inactive orbitals
occ,6,2,3                 !6a1, 2b1, 3b2 occupied orbitals
wf,16,1,0                !16 electrons, Symmetry 1 (A1), singlet
select                   !start configuration selection
con, 3.1,-3.1,1.2,-1.2,1.3,-1.3      !occupied orbitals in 1st configuration
con, 3.1,-3.1,2.2,-2.2,1.3,-1.3
con, 3.1,-3.1,1.2,-1.2,2.3,-2.3
con, 3.1,4.1,1.2,2.2,1.3,-1.3
con, 3.1,-3.1,1.2,2.2,1.3,2.3
con, 4.1,-4.1,1.2,-1.2,1.3,-1.3}

```

This calculation includes all CSFs of the CASSCF with coefficients larger than 0.04.

7 Multireference electron correlation methods

MCSCF or CASSCF calculations usually account only for a small percentage of the dynamical correlation energy. Therefore, in order to obtain accurate results, subsequent correlation treatments are possible. This can either be done by multireference configuration interaction (MRCI) or by multireference perturbation theory (MRPT). The latter is simpler and much cheaper, but also less reliable than MRCI.

7.1 Multireference configuration interaction (MRCI)

MRCI calculations are invoked using the

```
mrci
```

directive. By default, the same occupied and closed-shell spaces as in the preceding MCSCF (CASSCF) calculation are used, and the inner-shell *core* orbitals are not correlated (i.e., the *1s* orbitals of carbon or oxygen, or the *1s*, *2s*, and *2p* orbitals in chlorine). The number of uncorrelated core orbitals can be modified using the `core` directive (see section 5.1). It is not necessary that the reference wavefunction is exactly the same as in the MCSCF, and the `occ`,

closed, restrict, and select directives can be used exactly in the same way as explained for MCSCF and CASSCF.

By default, the orbitals are taken from the most recent orbital optimization calculation (HF or MCSCF/CASSCF). Other orbitals can be specified using the `orbital` directive. Please refer to the reference manual for further details.

The following is an example of a CASSCF/MRCI calculations for O₂.

```

***,O2
print,orbitals,civector      !print orbitals and ci-coefficients
geometry={                  !geometry specification, using z-matrix
o1
o2,o1,r
}

r=2.2 bohr                  !bond distance

basis=vtz                   !triple-zeta basis set

{hf                          !invoke RHF program
wf,16,4,2                   !define wavefunction: 16 electrons,
symmetry 4,                 !triplet
occ,3,1,1,,2,1,1           !number of occupied orbitals in each symmetry
open,1.6,1.7}              !define open shell orbitals

casscf                       !casscf using full valence active space
mrci                         !mrci using full valence casscf reference function
{mrci                        !mrci with only 2p orbitals active, 2s closed
                             !in reference
closed,2,,,,2}             !inactive orbitals in the reference function.

```

7.2 Multireference perturbation theory, CASPT2, CASPT3

The input for MRPT/CASPT2 is similar to MRCI, but the following commands are used.

```

rs2          !second-order multireference perturbation theory
rs3          !third-order multireference perturbation theory
rs2c        !second-order multireference perturbation theory
             !with a more contracted configuration space.

```

In case of `rs2` and `rs3`, exactly the same configuration spaces as in the MRCI are used. In this case the excitations with two electrons in the external orbital space are *internally contracted*. The total number of correlated orbitals is restricted to 16 for machines with 32-bit integers and to 32 for machines with 64-bit integers.

In the `rs2c` case certain additional configuration classes involving internal and semi-internal excitations are also internally contracted (see J. Chem. Phys. **112**, 5546 (2000)). This method is much more efficient than `rs2` and more suitable for large cases. In particular, in this case only the number of *active* orbitals is restricted to 16 or 32 on 32 and 64 bit machines, respectively, and any number of closed-shell (inactive) orbitals can be used (up to a maximum as defined by a program parameter).

Note that the `RS2` and `RS2c` methods yield slightly different results. In both cases the results also slightly differ from those obtained with the method of of Roos et al. (J. Chem. Phys. **96**, 1218 (1992)), as implemented in MOLCAS, since in the latter case *all* configuration spaces are internally contracted. This introduces some bottlenecks that are not present in MOLPRO.

Restricted active space (RASPT2) or general MRPT2 calculations can be performed using the `restrict` and/or `select` directives as explained in section for MCSCF and CASSCF.

MRPT2 and CASPT2 calculations often suffer from so-called intruder state problems, leading to a blow-up of the wavefunction and no convergence. This problem can often be avoided by using *level shifts*. These shifts can be specified on the `rs2` and `rs2c` cards:

```
rs2, shift=value
rs2c, shift=value
```

The energy is approximately corrected for the shift as proposed by Roos and Andersson. (Chem. Phys. Lett. **245**, 215 (1995)).

It is also possible to use modified zeroth-order Hamiltonians; see reference manual for details.

Energy gradients are available, with some restrictions in the case of multistate calculations, for RS2; see the manual for details.

8 Density functional calculations

MOLPRO is able to carry out standard Kohn-Sham DFT calculations using either the spin-restricted (`ks` or `rks`) or spin-unrestricted (`uks`) formalisms. Here is a simple example, which does a local-density approximation (LDA) calculation for formaldehyde.

```
***, formaldehyde
geometry={
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

rks
```

The exchange-correlation functional, and associated potential, are integrated numerically, and in principle it is necessary to supply parameters to allow the construction of an appropriate grid. However, the grid is built using a model adaptive scheme, and it is normally necessary to specify only the required energy accuracy. This threshold is in turn taken by default from the overall energy convergence threshold, so in most calculations it is not normally necessary to specify anything at all.

Many different exchange and correlation functionals are contained in the program. They are implemented in a modular fashion, with both computer code and documentation being built directly from their mathematical definition. This means that you can always find the precise definition of a functional in the user manual. Each functional has a keyword which is used to identify it in the input file. The functionals to be used are given one after each other as options to the `ks` command; for example, the following does a Kohn-Sham calculation using Becke's exchange functional, and the Lee–Yang–Parr correlation functional.

```
ks, b, lyp
```

Another commonly used combination (in fact, the default) is `ks, s, vwn` which gives LDA (Slater–Dirac exchange, Vosko–Wilk–Nusair correlation). Finally, the program is aware of some

combinations of functionals, for example `ks,b3lyp` which is the hybrid B3LYP functional consisting of weighted combinations of various density functionals together with a fraction of exact (Hartree–Fock) exchange.

Other options, such as the closed- and open-shell orbitals and the wavefunction symmetry can be defined as explained in section 4.3.

9 Geometry optimization

Geometry optimizations are invoked by the `optg` input directive, which must follow the input of the method for which the optimization is to be performed. For instance, optimization of the geometry for formaldehyde at the MP2 level is performed with the following input

```

***,formaldehyde
print,basis,orbitals      !this is optional: print the basis set and the
                          !occupied orbitals

angstrom
geometry={                !define the nuclear coordinates
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

basis=vdz                 !Select basis set
hf                        !Perform HF calculation
mp2                       !Perform MP2 calculation
optg                      !Optimize geometry for MP2

```

Optimizations use analytical energy gradients if available; otherwise the gradients are computed by finite differences. Presently, analytical energy gradients are available for the following methods.

<code>hf</code>	Closed and open-shell spin restricted Hartree-Fock (RHF)
<code>uhf</code>	Spin-unrestricted Hartree-Fock (UHF).
<code>ks</code>	Spin-restricted closed and open-shell Kohn-Sham calculations.
<code>uks</code>	Spin-unrestricted Kohn-Sham calculations.
<code>mcscf</code>	MCSCF and CASSCF, including state-averaged calculations
<code>mp2</code>	closed-shell MP2
<code>lmp2</code>	closed-shell local MP2
<code>qcisd(t)</code>	closed-shell quadratic configuration interaction, including perturbative triple-excitation contribution
<code>rs2</code>	Second-order multireference perturbation theory

Various options are available for modifying the convergence thresholds and the optimization method. For details see the reference manual.

10 Frequency calculations

Harmonic vibrational frequencies can be computed automatically using the `frequency` directive. Normally, this should be preceded by a geometry optimization. A typical input reads as follows.

```
hf                !Perform HF calculation
mp2              !Perform MP2 calculation
optg            !Optimize geometry for MP2
frequencies,sym=auto !Perform frequency calculation for MP2; use symmetry in hessian calculation
```

The second energy derivatives are computed by finite differences using analytical energy gradients when available (see section 9), otherwise from energies (which may take long time and is less accurate!). Note that during frequency calculations the symmetry of the molecule may be lowered, and then the calculation may fail. It is therefore advisable to use the `nosym` directive as first line in the geometry input.

In the following cases the second derivatives can be computed analytically:

```
hf                Closed shell Hartree-Fock (RHF)
mcscf            Single state MCSCF and CASSCF without symmetry (i.e. using the
                 nosym directive in the geometry input).
```

In these cases the input is

```
frequencies,analytical !Perform frequency calculation
```

11 Relativistic effects and pseudopotentials

Scalar-relativistic effects can be included explicitly, either by means of the Pauli or the Douglas-Kroll Hamiltonian. In the former case, the effects are evaluated to first order perturbation theory (including the mass-velocity and Darwin terms) by setting

```
gexpec,rel
```

at the beginning of the input. The relativistic contributions are stored by the program within the variable `erel`. An example is

```
***,Cu ground state
! Pauli Hamiltonian

gexpec,rel        !compute relativistic correction using perturbation theory
geometry={cu}    !geometry
basis=vtz        !basis set
hf               !Hartree-Fock calculation
e_rhf=energy+erel !store total relativistic energy in variable e_rhf
```

To use the Douglas-Kroll Hamiltonian, one has to specify

```
dkroll=1
```

at the beginning of the input. The relativistic contributions are then included as part of the total energies. In this case the example reads

```

***,Cu ground state

! Douglas-Kroll-Hess Hamiltonian

dkroll=1      !activate Douglas-Kroll treatment
geometry={cu} !geometry
basis=vtz-dk  !special DK basis set
hf           !Hartree-Fock
e_rhf=energy  !Total relativistic energy

```

An implicit treatment of scalar-relativistic effects is possible with pseudopotentials (ECPs). In that case, one has to specify ECP parameters and basis sets within basis blocks:

```

basis={...
ecp, atom,ecpname
spd... , atom,basisname
...}

```

Often, it is even sufficient to specify a ECP basis-set keyword like

```
basis=vtz-PP
```

and this will automatically select the given basis set and the associated pseudopotential. In this case the input for the copper atom calculation could be

```

***,Cu ground state

! ECP

geometry={cu} !geometry
basis=vtz-pp  !special pseudo potential basis set; this also selects the ECP
hf           !HF
e_rhf=energy  !Total energy. This does not include contributions from the core
              !orbitals that are included in the ECP.

```

For a list of available ECPs, ECP basis sets, and corresponding keywords, see <http://www.theochem.uni-stuttgart.de/pseudopotentials/index.en.html>

Spin-orbit splittings can be calculated by setting up (and diagonalizing) spin-orbit matrices between scalar-relativistic states. The latter have to be calculated and stored within CI calculations:

```

{ci;...;save,record1.file}
...
{ci;...;save,recordn.file}

```

In all-electron calculations, one proceeds with the calculation of SO integrals

```
lsint
```

(For ECP calculation, this is not needed.)

Generating and processing of SO matrices is done with

```
{ci;hlsmat,keyword3,record1.file,...,recordn.file}
```

where *keyword3* is *ls* or *ecp* for all-electron or ECP calculations, respectively.

An example input with ECPs is

```

***,Br
geometry={br}
basis=vtz-pp
{rhf;wf,sym=5}

```

```
{multi;wf,sym=2;wf,sym=3;wf,sym=5} !2P states, state averaged
{ci;wf,sym=2;save,5101.2} !2Px state
{ci;wf,sym=3;save,5102.2} !2Py state
{ci;wf,sym=5;save,5103.2} !2Pz state
{ci;hlsmat,ls,5101.2,5102.2,5103.2} !compute and diagonalize SO matrix
```

The corresponding input for an all-electron calculation is

```
***,Br
dkroll=1
geometry={br}
basis=vtz-dk
{rhf;wf,sym=5}
{multi;wf,sym=2;wf,sym=3;wf,sym=5} !2P states, state averaged
{lsint} !Compute spin-orbit 2-electron integrals
{ci;wf,sym=2;save,6101.2} !2Px state
{ci;wf,sym=3;save,6102.2} !2Py state
{ci;wf,sym=5;save,6103.2} !2Pz state
{ci;hlsmat,ls,6101.2,6102.2,6103.2} !compute and diagonalize SO matrix
```

12 Integral-direct calculations

With the exception of perturbative triples corrections, all methods implemented in MOLPRO can be performed in integral-direct mode, i.e., without storage of the two-electron integrals on disk. Naturally, this takes longer than conventional calculations, since the integrals are recomputed whenever needed. The integral-direct mode is activated by giving the `gdirect` directive before the first energy calculation, e.g.,

```
***,formaldehyde
print,basis,orbitals !this is optional: print the basis set and the
!occupied orbitals

angstrom
geometry={ !define the nuclear coordinates
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

gdirect !integral-direct calculation
basis=vdz !Select basis set
hf !Perform HF calculation
mp2 !Perform MP2 calculation
```

13 Density fitting approximations

Density fitting can be used to approximate the integrals in spin restricted Hartree-Fock (HF), density functional theory (KS), second-order Møller-Plesset perturbation theory (MP2 and RMP2), explicitly correlated MP2 (MP2-F12), and all levels of closed-shell local correlation methods (LCC2, LMP2-LMP4, LQCISD (T), LCCSD (T)). In the literature the corresponding methods

are often also denoted as RI-HF, RI-DFT, or RI-MP2 etc. We prefer the acronym DF since RI is used for another purpose in explicitly correlated methods (cf. section 15).

These methods are either direct or semi-direct, i.e. only relatively small sets of transformed four-index integrals are stored. They are very much faster than standard direct calculations and highly recommended, in particular for calculations on larger molecules. The errors caused by density fitting approximations are generally negligible.

Density fitting is invoked by adding the prefix DF- to the command name, e.g. DF-HF, DF-KS, DF-MP2 and so on. Gradients are available for DF-HF, DF-KS, and DF-LMP2. Symmetry is not implemented for density fitting programs. Therefore, symmetry is turned off automatically if DF- is found in the input.

By default, a fitting basis set will be chosen automatically that corresponds to the current orbital basis set and is appropriate for the method. For instance, if the orbital basis set is VTZ, the default fitting basis is VTZ/JKFIT for DF-HF or DF-KS, and VTZ/MP2FIT for DF-MP2. Other fitting basis sets from the library can be chosen using the DF_BASIS option, e.g.

```
BASIS=VTZ           !use VTZ orbital basis
DF-HF,DF_BASIS=VQZ !use VQZ/JKFIT fitting basis
DF-MP2,DF_BASIS=VQZ !use VQZ/MP2FIT fitting basis
```

The program then chooses automatically the set which is appropriate for the method. Optionally, the basis type can be appended to the basis name and then this supercedes the default, e.g.

```
DF-HF,DF_BASIS=VQZ/JKFIT !use VQZ/JKFIT fitting basis
```

There are many options affecting, e.g. screening and other details, but the default values should normally be appropriate. A full description can be found in the MOLPRO users manual.

14 Local correlation treatments

14.1 Introduction

The purpose of local correlation methods is to reduce the scaling of the computational effort as function of the molecular size and to make it possible to perform accurate calculations for larger molecules. In MOLPRO, local correlation methods are based on the Ansatz by Pulay (Chem. Phys. Lett. **100**, 151 (1983)). The occupied valence orbitals are localized by one of the standard localization methods (by default Pipek-Mezey localization is used) and the virtual orbital space is represented by projected atomic orbitals (PAOs). The orbital pairs are classified according to distance criteria. By default only *strong pairs*, in which the two orbitals are close together and which account for most of the correlation energy are treated at the highest computational level (e.g., local coupled cluster, LCCSD), while weak pairs are treated at the local MP2 (LMP2) level. Very distant pairs can be neglected. For each of the correlated pairs, a different subspace (*domain*) of virtual orbitals is automatically chosen, and the excitations are restricted into these domains. The basic features of the LCCSD method are described in J. Chem. Phys. **104**, 6286 (1996); further references describing more recent developments can be found in the References section.

The local correlation program of MOLPRO can currently perform closed-shell LMP2, LMP3, LMP4(SDTQ), LCISD, LQCISD(T), and LCCSD(T) calculations. For large molecules, all methods scale linearly with molecular size, provided very distant pairs are neglected, and the integral-direct algorithms are used.

Much higher efficiency is achieved by using density fitting (DF) approximations (see section 13) to compute the integrals. Density fitting is available for all local methods up to LCCSD(T), as well as for analytical LMP2 gradients. The errors introduced by DF are negligible, and the use of the DF methods is highly recommended.

Energy gradients are available for LMP2, DF-LMP2, DF-SCS-LMP2, and LQCISD (in the latter case only for LOCAL=1, i.e. the local calculation is simulated using the canonical program, and savings only result from the reduced number of pairs).

Naturally, the local approximation can introduce some errors, and therefore the user has to be more careful than with standard black box methods. On the other hand, the low-order scaling makes it possible to treat much larger systems at high levels of theory than it was possible so far. Before using these methods, it is strongly recommended to read the literature in order to understand the basic concepts and approximations. A recent review [1] and Ref. [2] may be suitable for an introduction.

References:

- [1] H.-J. Werner and K. Pflüger, *On the selection of domains and orbital pairs in local correlation treatments*, Ann. Rev. Comp. Chem. **2**, 53 (2006).
 [2] C. Hampel and H.-J. Werner, *Local Treatment of electron correlation in coupled cluster (CCSD) theory*, J. Chem. Phys. **104**, 6286 (1996).

14.2 Invoking local correlation methods

The local correlation treatment is switched on by preceding the command name by an L, i.e., by using the LMP2, LMP3, LMP4, LQCISD, LCCSD, or LCISD commands.

The LQCISD and LCCSD commands can be appended by a specification for the perturbative treatment of triple excitations, e.g. LCCSD(T). Density fitting can be invoked by prepending the command name by DF-, e.g. DF-LMP2, DF-LCCSD(0) etc. In density fitting calculations an additional auxiliary basis set is needed. Details about choosing such basis sets and other options for density fitting are described in sections 13.

The general input for local coupled LMP2 or coupled cluster calculations is:

LMP2, <i>options</i>	Local MP2 calculation
LCCSD, <i>options</i>	Local CCSD calculation
LCCSD(T), <i>options</i>	Local CCSD(T) calculation

There are many options and directives to control the local approximations (e.g., domains and pair classes). Here we only described the most important ones. For a full description please read the MOLPRO users manual.

14.3 Orbital localization

By default, the orbitals are localized using the method of Pipek and Mezey (PM). Alternatively, natural localized molecular orbitals (NLMOs) can be used [see Mol. Phys. **105**, 2753 (2007)] by specifying the option LOC_METHOD=NLMO. Note that analytical energy gradients can be computed only with PM orbitals.

14.4 Choice of domains

14.4.1 Standard domains

Standard domains are always determined first. They are used to define strong, close, weak, and distant pairs. More accurate results can be obtained with extended domains (see section 14.4.2), but these extensions do not affect the pair classification.

The domains can be determined either by the method of Boughton and Pulay (BP) [J. Comp. Chem. **14**, 736 (1993)] or using natural population analysis (NPA) as described in Mol. Phys. **105**, 2753 (2007). By default, the BP method is used with PM orbitals, while NPA method is employed with NLMOs. This default behaviour can be superseded using `THRBP=value` (use the BP method with the given threshold) or `NPASEL=value` (use the NPA method with the given threshold).

This BP method is somewhat basis set dependent and therefore the default value of `THRBP` depends on the basis set: 0.980 for AVDZ, 0.985 for AVTZ, and 0.990 for AVQZ. The NPA method is much less basis set dependent and the default value is `NPASEL=0.03`.

14.4.2 Extended domains

In order to increase the accuracy various options can be used to extend the domains (for details see MOLPRO manual). But note that the computational effort increases with the fourth power of the domain sizes and can therefore increase quite dramatically when extending domains. This does not affect the linear scaling behaviour in the asymptotic limit.

14.5 Choice of pair classes

The *strong*, *close*, *weak* and *distant* pairs are selected using distance or connectivity criteria as described in more detail in section 14.5.1. *Strong* pairs are treated by CCSD, all other pairs by LMP2. In triples calculations, all orbital triples (ijk) are included for which (ij), (ik), and (jk) are *close pairs*. In addition, one of these pairs is restricted to be strong. The triples energy depends on the strong and close pair amplitudes. The close pair amplitudes are taken from the LMP2 calculation. Thus, increasing the distance or connectivity criteria for close and weak pairs will lead to more accurate triples energies. While for near equilibrium properties like geometries and harmonic vibrational frequencies the default values are normally appropriate, larger distance criteria are sometimes needed when computing energy differences, in particular barrier heights. In cases of doubt, `RWEAK` should first be increased until convergence is reached, and then `RCLOSE` can be varied as well. Such tests can be performed with small basis sets like cc-pVDZ, and the optimized values then be used in the final calculations with large basis sets.

14.5.1 Options for selection of pair classes

There are two alternative modes for defining the pair classes: either *distance criteria* `RCLOSE`, `RWEAK`, `RDIST`, `RVDIST` can be used. These are in Bohr and refer for a given orbital pair (ij) to the minimum distance $R^{(ij)}$ between any atom in the standard orbital domains $[i]$ and any atom in the standard orbital domains $[j]$. Alternatively, the *connectivity criteria* `ICLOSE`, `IWEAK`, `IDIST`, `IVDIST` can be used. These refer to the minimum number of bonds between any atom contained in the standard domain $[i]$ and any atom contained in the standard domain $[j]$. The advantage of using connectivity criteria is the independence of the bond lengths, while the advantage of distance criteria (default) is that they are also effective in non-bonding situations.

Only one of the two possibilities can be used, i.e., they are mutually exclusive. The use of distance criteria is the default. Using connectivity criteria for pair selection requires to set the option `USE_DIST=0`.

<code>RCLOSE</code>	(default 1) Strong pairs are defined by $0 \leq R^{(ij)} < \text{RCLOSE}$. Close pairs are defined by $\text{RCLOSE} \leq R^{(ij)} < \text{RWEAK}$.
<code>RWEAK</code>	(default 3) Weak pairs are defined by $\text{RWEAK} \leq R^{(ij)} < \text{RDIST}$.
<code>RDIST</code>	(default 8) Distant pairs are defined by $\text{RDIST} \leq R^{(ij)} < \text{RVDIST}$.
<code>RVDIST</code>	(default 15) Very distant pairs for which $R^{(ij)} \geq \text{RVDIST}$ are neglected.
<code>ICLOSE</code>	(default 1) Strong pairs are separated by less than <code>ICLOSE</code> bonds. Close orbital pairs are separated by at least <code>ICLOSE</code> bonds but less than <code>IWEAK</code> bonds.
<code>IWEAK</code>	(default 2) Weak orbital pairs are separated by at least <code>IWEAK</code> bonds but less than <code>IDIST</code> bonds.
<code>IDIST</code>	(default 5) Distant orbital pairs are separated by at least <code>IDIST</code> bonds but less than <code>IVDIST</code> bonds.
<code>IVDIST</code>	(default 8) Very distant orbital pairs (neglected) are separated by at least <code>IVDIST</code> bonds.

Setting a distance criterion to zero means that all pairs up to the corresponding class are treated as strong pairs. For instance, `RCLOSE=0` means that strong and close pairs are fully included in the LCCSD.

14.6 Doing it right

Here we summarize only a few important aspects. Further information and hints can be found in the MOLPRO users manual.

14.6.1 Basis sets

For local calculations we recommend the use of generally contracted basis sets, e.g., the correlation consistent cc-pVnZ sets of Dunning and coworkers. For these basis sets the core basis functions are uniquely defined, and will always be eliminated if the defaults are used.

The correlation consistent basis sets are also recommended for all density fitting calculations, since optimized fitting basis sets are available for each basis.

14.6.2 Symmetry and Orientation

In local calculation, the use of symmetry should be disabled (option `NOSYM`), since otherwise orbital localization is not possible. We also recommend to use the `NOORIENT` option to avoid unintended rotations of the molecule when the geometry changes. This is particularly important for geometry optimizations and for domain restarts in calculations of interaction energies. The `NOSYM` and `NOORIENT` option can be given on a `SYMMETRY` directive that must precede the geometry block, e.g.,

```

symmetry,nosym,noorient
geometry={
  O1
  H1,O1,roh
  H2,O1,roh,h1,hoh
}

```

14.6.3 Localization

By default, Pipek-Mezey (PM) localization is used and performed automatically in the beginning of a local correlation calculation. As mentioned in section 14.3, it is also possible to use natural localized orbitals (NLMOs).

Poor localization is sometimes an intrinsic problem, in particular for strongly conjugated systems or when diffuse basis sets are used. This is caused by localization tails due to the overlapping diffuse functions. The NLMO method is less susceptible to such problems than the PM method. With PM localization problems are particularly frequent in calculations of systems with short bonds, e.g., aromatic molecules. Often such problems can be avoided using the option `PMDEL=n` with $n = 1$ or 2 . This means that the contributions of the n most diffuse basis functions of each angular momentum type are ignored in the localization. This often yields much better localized orbitals when diffuse basis sets are used. For aug-cc-pVTZ, $n = 2$ has been found to work very well, while for aug-cc-pVDZ $n=1$

14.6.4 Orbital domains

In most cases, the domain selection (cf. section 14.4) is uncritical for saturated molecules. Nevertheless, in particular for delocalized systems, it is recommended always to check the orbital domains, which are printed in the beginning of each local calculation. The orbital domains consist of all basis functions for a subset of atoms. These atoms are selected so that the domain spans the corresponding localized orbital with a preset accuracy (alterable with option `THRBP`). A typical domain output, here for water, looks like this:

Orbital domains

Orb.	Atom	Charge	Crit.
2.1	1 O1	1.17	0.00
	3 H2	0.84	1.00
3.1	1 O1	2.02	1.00
4.1	1 O1	1.96	1.00
5.1	1 O1	1.17	0.00
	2 H1	0.84	1.00

This tells you that the domains for orbitals 2.1 and 5.1 comprise the basis functions of the oxygen atom and one hydrogen atom, while the domains for orbitals 3.1 and 4.1 consist of the basis function on oxygen only. The latter ones correspond to the oxygen lone pairs, the former to the two OH bonds, and so this is exactly what one would expect. For each domain of AOs, corresponding projected atomic orbitals (PAOs) are generated, which span subspaces of the virtual space and into which excitations are made. Options which affect the domain selection are described in section 14.4. Improper domains can result from poorly localized orbitals (see section 14.6.3 or a forgotten `NOSYM` directive). This does not only negatively affect performance and memory requirements, but can also lead to unexpected results.

The choice of domains usually has only a weak effect on near-equilibrium properties like equilibrium geometries and harmonic vibrational frequencies. More critical are energy differences like reaction energies or barrier heights. In cases where the electronic structure strongly changes, e.g., when the number of double bonds changes, it is recommended to compare DF-LMP2 and DF-MP2 results before performing expensive LCCSD(T) calculations.

15 Explicitly correlated methods

Explicitly correlated calculations provide a dramatic improvement of the basis set convergence of MP2 and CCSD correlation energies. Such calculations can be performed using the com-

mands of the form

command, options

where *command* can be one of the following:

MP2-F12	Closed-shell canonical MP2-F12. The F12-corrections is computed using density fitting, and then added to the MP2 correlation energy obtained without density fitting. By default, ansatz 3C(FIX) is used. Other ansätze, as fully described in J. Chem. Phys. 126 , 164102 (2007) can also be used (cf. section 15.3).
DF-MP2-F12	As MP2-F12, but the DF-MP2 correlation energy is used. This is less expensive than MP2-F12 since the standard two-electron integrals and the non-density fitted MP2 energy need not to be computed.
DF-RMP2-F12	Spin-restricted open-shell DF-RMP2-F12 as described in J. Chem. Phys. 128 , 154103 (2008) By default, ansatz 3C(FIX) is used, but ansatz 3C(D) can also be used (cf. sections 15.3).
CCSD-F12	Closed-shell CCSD-F12 approximations as described in J. Chem. Phys. 127 , 221106 (2007) and J. Chem. Phys. 130 , 054104 (2009). By default, the fixed amplitude ansatz 3C(FIX) is used and the CCSD-F12A and CCSD-F12B energies are computed. Optionally, the command can be appended by A or B, and then only the corresponding energy is computed. For more details see section 15.8.
CCSD(T)-F12	Same as CCSD-F12, but perturbative triples are added.
UCCSD-F12	Open-shell unrestricted UCCSD-F12 approximations as described in J. Chem. Phys. 130 , 054104 (2009). Restricted open-shell Hartree-Fock (RHF) orbitals are used. Optionally, the command can be appended by A or B, and then only the corresponding energy is computed.
UCCSD(T)-F12	Same as UCCSD-F12, but perturbative triples are added.
DF-LMP2-F12	Closed-shell DF-MP2-F12/3C(D) with localized orbitals. By default no local approximations are used. Domain approximations can be introduced using the DOMSEL or NPASEL options, cf. section 14.4. In order to use the loc-order scaling LMP2-F12 method as described in J. Chem. Phys. 130 , 054106 (2009) special options are necessary, see user's manual.

15.1 Options

There are many options available for explicitly correlated methods, but sensible defaults are used for all of them. For details see the MOLPRO reference manual. The most important options are:

DF_BASIS= <i>basis</i>	Select the basis for density fitting (see section 13 for details). <i>basis</i> can either refer to a set name defined in the basis block, or to a default MP2 fitting basis (e.g., DF_BASIS=VTZ generates the VTZ/MP2FIT basis). By default, the MP2FIT basis that corresponds to the orbital basis is used.
------------------------	---

RI_BASIS= <i>basis</i>	Select the basis for the resolution of the identity (RI). By default the JKFIT basis that corresponds to the chosen orbital basis is used. In conjunction with the VnZ-F12 basis sets, it is recommended to use the VnZ-F12/OPTRI sets of Yousaf and Peterson, <i>J. Chem. Phys.</i> 129 , 18410 (2008).
ANSATZ= <i>ansatz</i>	Select the explicitly correlated ansatz <i>ansatz</i> methods. See section 15.3 for the defaults and the MOLPRO manual for further details and possibilities.
GEM_BETA= <i>value</i>	Exponent for Slater-type frozen geminal (default $1.0 a_0^{-1}$).
SINGLES=0	Disable CABS singles correction. The default is SINGLES=1.

In the following, we briefly summarize the meaning of these options and of the approximations that can be used. For more details and further references to related work of other authors see H.-J. Werner, T. B. Adler, and F. R. Manby, *General orbital invariant MP2-F12 theory*, *J. Chem. Phys.* **126**, 164102 (2007) and G. Knizia and T. B. Adler and H.-J. Werner, *Simplified CCSD(T)-F12 methods: Theory and benchmarks*, *JCP* **130**, 054104 (2009).

15.2 Reference functions

The MP2-F12, CCSD-F12, and UCCSD-F12 methods must use conventional (non-density fitted) spin-restricted Hartree-Fock reference functions (HF or RHF). DF-HF cannot be used for these methods. This restriction is necessary to ensure that the Fock matrix is diagonal and consistent with the integrals used in these methods. For DF-MP2-F12, DF-LMP2-F12, and DF-RMP2-F12 either HF or DF-HF reference functions can be used.

15.3 Wave function Ansätze

The so called "ansatz" determines the definition of the explicitly correlated wave function. This is to be distinguished from the various approximations that can be used to approximate the Hamiltonian matrix elements. For details see *J. Chem. Phys.* **126**, 164102 (2007). In MOLPRO the following wave function ansätze can be used (specified using option *ansatz*) on the command line:

15.3.1 The general ansatz (*ansatz*=3C(FULL))

The conventional external pair functions are augmented by terms of the form

$$|u_{ijp}^{F12}\rangle = \sum_{p=\pm 1} \sum_{kl} T_{kl}^{ijp} \hat{Q}_{12} \hat{F}_{12} |kl\rangle \quad (2)$$

This ansatz is orbital invariant (i.e., the same results are obtained with canonical or localized orbitals), but it often suffers from geminal basis set superposition errors. Furthermore, singularities may occur in the zeroth-order Hamiltonian, in particular for larger systems. Therefore, this ansatz is normally not recommended.

15.3.2 The diagonal ansatz (ansatz=3C(D))

The sum over kl in equation (2) is restricted to ij . This eliminates the geminal basis set superposition errors of the general ansatz. However, since orbital invariance is lost, localized orbitals should be used to obtain size consistent wavefunctions. In most cases, this ansatz yields the most accurate results for MP2-F12 or LMP2-F12.

15.3.3 The fixed amplitude ansatz (ansatz=3C(FIX))

The wave function has the same form as for the diagonal ansatz, but the amplitudes are determined from the cusp conditions and fixed, i.e., $T_{ij}^{ij,1} = 1/2$ for singlet pairs ($p = 1$) and $T_{ij}^{ij,-1} = 1/4$ for triplet pairs ($p = -1$). This restores the orbital invariance, and usually the results are almost as accurate as with the diagonal ansatz. This method is size consistent for any choice of orbitals. Since the (T) correction in CCSD(T)-F12 calculations requires the use of canonical orbitals, the FIX approximation is used by default in CCSD-F12 and CCSD(T)-F12 calculations.

15.4 The correlation factor \hat{F}_{12}

In the F12 methods, the correlation factor \hat{F}_{12} is approximated by a frozen expansion of Gaussian type geminals that are functions of the interelectronic distance r_{12} . In principle, this can be any function, but normally a Slater function

$$F_{12}(r_{12}) = -\beta^{-1} \exp(-\beta r_{12}) \quad (3)$$

is used. By default this function is approximated by an expansion of six Gaussian functions, and the exponents and coefficients are optimized to obtain the best least squares fit, using a suitable weight function. The exponent β can be chosen using the option `gem.beta=value` (default `gem.beta=1.0`).

In addition, also linear R12-methods ($F_{12} = r_{12}$) are available (DF-MP2-R12 and DF-LMP2-R12). However, these are no longer recommended since the non-linear correlation factor yields much better accuracy, numerical stability and convergence with respect to the AO, DF and RI basis sets.

15.5 Basis sets

In MOLPRO the F12 integrals can only be computed using density fitting (DF) approximations. The many electron integrals are approximated by resolutions of the identity (RI) expansions. Thus, F12 calculations require three different basis sets: the orbital (AO) basis, the DF basis, and the RI basis.

We recommend as AO basis sets the augmented correlation consistent basis sets (denoted AVnZ) or the specially optimized correlation consistent F12 basis sets (denoted VnZ-F12, cf. K.A. Peterson, T.B. Adler, and H.-J. Werner, J. Chem. Phys. **128**, 084102 (2008)). Normally, triples zeta basis sets (AVTZ or VTZ-F12) yield excellent results that are close to the basis set limit. Diffuse basis functions are rather essential both for the HF and MP2-F12 energies, and therefore the standard VTZ sets are not recommended. If the AVnZ or VnZ-F12 orbital basis sets are used, suitable density fitting (DF) basis and resolution of the identity (RI) basis sets are automatically chosen. By default, the AVnZ/MP2FIT and VnZ/JKFIT basis sets are used for the DF and RI,

respectively. Other basis sets can be chosen using the `DF_BASIS` and `RI_BASIS` options (cf. section 15.1). See section 13 for more details about density fitting.

The following example shows a `ccsd(t)-f12` calculation using the VTZ-F12 basis of Peterson et al., J. Chem. Phys. **128**, 084102 (2008), along with the associated RI basis set of Yousaf and Peterson, J. Chem. Phys. **129**, 18410 (2008):

```

***, formaldehyde
geometry={
C
O , C , rco
H1 , C , rch , O , hco
H2 , C , rch , O , hco , H1 , 180
}

rco=1.182 Ang
rch=1.102 Ang
hco=122.1789 Degree

basis=vtz-f12
hf
ccsd(t)-f12,ri_basis=vtz-f12/optri,df_basis=avtz
e_f12a=energy(1) !save f12a energy in variable e_f12a
e_f12b=energy(2) !save f12b energy in variable e_f12b

```

15.6 Symmetry

Symmetry cannot be used in DF-LMP2-F12 calculations. However, in MP2-F12, DF-MP2-F12, DF-RMP2-F12, CCSD(T)-F12 and UCCSD(T)-F12 calculations Abelian symmetry can be used as usual; in these cases the preceding DF-MP2-F12 calculations will be automatically performed without symmetry, and the integrals that are necessary for subsequent CCSD-F12 or UCCSD-F12 calculations will be transformed to the symmetry adapted basis. This is fully automatic and transparent to the user. Note, however, that the prefix DF- turns off symmetry automatically, and if you want to use symmetry in the HF calculations preceding the DF-MP2-F12 or DF-MP2-F12 calculations the symmetry elements (or `AUTO`) must be specified either in the geometry block, or on a `SYMMETRY` directive preceding the geometry block.

15.7 CABS Singles correction

By default, the perturbative CABS singles correction as described in J. Chem. Phys. **127**, 221106 (2007) and J. Chem. Phys. **128**, 154103 (2008) is included in the reference energy of all MP2-F12 and CCSD-F12 calculations (closed and open-shell, except for LMP2-F12/3*A(loc), which is done with a different program). The corrected reference energy is stored in variable `ENERGR`, so that `ENERGY-ENERGR` are the total correlation energies.

The singles correction can be turned off by option `SINGLES=0`, e.g.

```
MP2-F12, SINGLES=0
```

The contribution of core orbitals to the singles energy is not included by default, but can be turned on by option `CORE_SINGLES`, e.g.

```
MP2-F12, CORE_SINGLES=1
```

15.8 CCSD(T)-F12

The CCSD-F12 and UCCSD-F12 programs first do DF-MP2-F12/3C(FIX) (closed-shell) or DF-RMP2-F12/3C(FIX) (open-shell) calculations, and then perform the CCSD-F12 (UCCSD-F12) without density fitting. By default, the CCSD-F12A and CCSD-F12B energies are both computed. A specific method can be requested by appending A or B to the -F12 suffix. Furthermore, instead of the 3C(FIX) ansatz, different ansätze (e.g. 3C) can be used. In this case the amplitudes of the explicitly correlated terms are determined in the MP2-F12 calculation and kept fixed in the CCSD-F12.

It should be noted that these methods involve approximations and do not yield the exact CCSD-F12 energies. Preliminary experience has shown that the CCSD-F12A method slightly overestimates the correlation energies, while CCSD-F12B underestimates them. For AVDZ or AVTZ basis sets, CCSD-F12A usually gives very good results, but for larger basis sets it may overestimate the basis set limit and converge from below to the limit. Thus, convergence may not be monotonic, and extrapolation of the correlation energies should not be attempted. CCSD-F12B usually converges monotonically from below to the limit and gives best results for AVQZ and larger basis sets. Thus, we currently recommend CCSD-F12A for AVDZ and AVTZ basis sets, and CCSD-F12B for larger basis sets (rarely needed).

The perturbative triples correction can be invoked by using CCSD(T)-F12 or UCCSD(T)-F12. There is no direct F12 correction to the triples, and therefore the basis set error of the triples is not affected by the F12 (small changes of the triples energy arise from the fact that the doubles amplitudes are affected by the F12 terms). In many cases, a simple and pragmatic improvement of the triples energy can be obtained by scaling the triples energy contribution as

$$\Delta E_{(T^*)} = \Delta E_{(T)} * E_{corr}^{MP2-F12} / E_{corr}^{MP2} \quad (4)$$

This can be done automatically by setting option SCALE_TRIP=1, i.e.

CCSD (T) -F12, SCALE_TRIP=1

16 Advanced features of MOLPRO

In the following sections, examples for some more special capabilities of MOLPRO are given. This description is not exhaustive. For more complete information, please refer to the reference manual.

16.1 Memory control

MOLPRO stores all internal data in a single big working area, which is allocated dynamically in the beginning of the calculation. The default amount of memory is 8 MW (64 Mb). For big calculations more memory may be needed, and the default can then be modified using the `memory` directive. This should be the first command in the input (after the `***` title card, if present, as shown in the following example.

```
***,title
memory,16,m           !allocate 16 MW of memory
geometry={...}        !define the nuclear coordinates
basis=vdz             !Select basis set
hf                   !Perform HF calculation
ccsd(t)              !Perform ccsd(t) calculation
```

The amount of memory needed depends on the size of the molecule, the basis set, the symmetry of the molecule, and the methods used, and therefore it is difficult to predict it in advance. Most calculation run with 16 MW or less, but in big cases with low symmetry more may be required. A rather safe choice is to specify 64 MW, but of course this requires that your machine has sufficient memory (in this case more than 512 MB). The memory can also be given using the `-m` option on the `molpro` command line, see section 3. If a `memory` card is given, it has preference over the command line option.

16.2 Restarting calculations

By default, and in all examples shown so far, scratch files are used to store all intermediate data MOLPRO needs, and the user will normally not see these files at all. However, it is possible to save computed data as orbitals and energies in named (permanent) files and use these for restarting a calculation at a later stage. MOLPRO uses a number of different files, but only one or two of them are needed for a restart. File 1 holds the one- and two electron integrals and related information, while on file 2 the wavefunction information like orbitals, orbital energies, and optionally CI-vectors are stored. Thus, file 2 is essential for restarting a calculation, while the integrals on file 1 can either be restarted or recomputed.

The use of named files can be requested using the `file` directive, which should be given immediately after the `memory` command (if present), e.g.,

```
***,title
memory,16,m           !allocate 16 MW of memory
file,1,filename.int   !allocate permanent integral file
file,2,filename.wfu   !allocate permanent wave-function (dump) file
geometry={...}        !define the nuclear coordinates
basis=vdz             !Select basis set
hf                   !Perform HF calculation
ccsd(t)              !Perform ccsd(t) calculation
```

If the same files are used in a subsequent calculation, MOLPRO will automatically recover all data saved on the given files. For instance, if the following input is used after the previous example, the integrals and orbitals will be restarted and used for the subsequent casscf and MRCI calculations:

```

***,title
memory,16,m           !allocate 16 MW of memory
file,1,filename.int   !allocate permanent integral file
file,2,filename.wfu   !allocate permanent wave-function (dump) file
casscf                !casscf, using previous scf orbitals as starting guess
mrci                  !mrci using casscf reference wavefunction

```

If only file 2 is defined in the input, the integrals will automatically be recomputed, as in the following input. Note that in most cases, file 1 can be very large (it contains the two-electron integrals), and the cost of recomputing the integrals can be a small fraction of the overall time; it is therefore usually sensible in this way to avoid declaring the file as permanent.

```

***,title
memory,16,m           !allocate 16 MW of memory
file,2,filename.wfu   !allocate permanent wave-function (dump) file
casscf                !casscf, using previous hf orbitals as starting guess
mrci                  !mrci using casscf reference wavefunction

```

The automatic restart mechanism can be disabled by specifying `new` after the filename(s), e.g.,

```

***,title
memory,16,m           !allocate 16 MW of memory
file,2,filename.wfu,new !allocate a new permanent wave-function (dump) file.
                        !If the file exists, overwrite it.
hf                    !Perform hartree-Fock calculation
casscf                !casscf, using hf orbitals as starting guess

```

Note that if permanent files are used, it is important to take care that two simultaneous jobs do not attempt to use the same file!

16.3 Print options

MOLPRO has very many print options as described in detail in the reference manual for the various methods, but in practice only a few of them are needed. In general, there are two kinds of print options: either global ones, specified with the `gprint` directive, being used by all methods, or local ones, given on `print` directives, being used only in one job step. The local print commands are part of the input for a specific method and must therefore directly follow the corresponding command. On the other hand, global print options act on all subsequent methods. For example

```

gprint,orbitals       !global print option: hf and casscf orbitals
                        !are printed

hf
casscf

{hf
print,orbitals}      !local print option, hf orbitals are printed

casscf                !casscf orbitals are not printed

```

To avoid confusion and unexpected results, we recommend to use global print options only, except for special debugging purposes. The most important global print options are

```
gprint,basis           !print basis set information
gprint,orbitals        !print occupied orbitals
gprint,orbitals=2      !print occupied and the two lowest virtual orbitals
                      !in each symmetry
gprint,civector        !print configuration coefficients
```

Note that by default only CI coefficients larger than 0.05 are printed. See section 16.4 for modifying this threshold.

Several print options can be given on one `gprint` directive, e.g.,

```
gprint,basis,orbitals=1,civector
```

16.4 Convergence thresholds

As for the print options, there are global and local thresholds, specified using the `gthresh` and `thresh` directives, respectively. Here we only mention some useful global thresholds (default values are shown).

```
gthresh,energy=1.d-6   !convergence threshold for energy.
                      !This affects all iterative methods
gthresh,orbital=1.d-5  !convergence threshold for orbitals.
                      !This affects scf only
gthresh,step=1.d-3     !convergence threshold for orbital rotations
                      !This affects mscf only
gthresh,gradient=1.d-2 !convergence threshold for gradient with respect to
                      !orbital rotations. This affects mscf only
gthresh,civec=1.d-5    !convergence threshold for ci-coefficients in ci and ccscf.
gthresh,optgrad=3.d-4  !threshold for gradient in geometry optimizations
gthresh,optenerg=1.d-6 !threshold for energy in geometry optimizations
gthresh,twoint=1.d-11  !Threshold for neglect of small 2-electron integrals
gthresh,compress=1.d-11 !Accuracy of integral compression (depends on twoint)
gthresh,printci=0.05   !print all ci-coefficients larger than 0.05
gthresh,punchci=0.05   !write all ci-coefficients larger than 0.05 to the punch file
```

Note that the energy threshold also affects the defaults for some other thresholds such as `orbital`, `step`, and `civec`, and therefore it is normally sufficient to lower the energy threshold if high accuracy is needed.

Any number of thresholds can be specified on a single `gthresh` directive, e.g.,

```
gthresh,energy=1.d-8,printci=0.03
```

16.5 Variables

Results and other values can be stored in variables for use at a later stage of the calculations. Variables can simply be set in the input as

```
name=value
```

Variables can also be one-dimensional arrays, in which case the format is

```
name=[value1,value2,value3,...]
```

The current dimension of such an array is #name.

MOLPRO stores certain results in variables with predefined names. The most important ones are

ORBITAL	Record in which last computed orbitals are stored
ENERGR(<i>istate</i>)	Reference energy for state <i>istate</i> in MRCI and CCSD.
ENERGY(<i>istate</i>)	last computed total energy for state <i>istate</i> .
ENERGC	Total energy excluding perturbative triples correction (set only in the CCSD/QCI program).
ENERGD(<i>istate</i>)	Total energy for state <i>istate</i> including Davidson correction (set only in CI).
ENERGP(<i>istate</i>)	Total energy for state <i>istate</i> including Pople correction (set only in CI).
ENERGT(1)	Total energy including perturbative triples (T) correction (set only in CCSD(T), QCI(T)).
ENERGT(2)	Total energy including perturbative triples [T] correction (set only in CCSD(T), QCI(T)).
ENERGT(3)	Total energy including perturbative triples -t correction (set only in CCSD(T), QCI(T)).
EMP2	holds MP2 energy in MPn, CCSD, BCCD, or QCISD calculations, and RS2 energy in MRPT2 (CASPT2) calculations.
EMP3	holds MP3 energy in MP3 and MP4 calculations, and RS3 energy in MRPT3 (CASPT3) calculations.
EMP4	holds MP4(SDQ) energy in MP4 calculations. The total MP4(SDTQ) energy is stored in variable ENERGY.
EF12	holds F12 contribution to energy. The total energy is stored in variable ENERGY.
STATUS	status of last step (1=no error, -1=error or no convergence)

See reference manual for further variables.

16.6 Program control using `do` loops, `if` blocks and `goto` commands

MOLPRO also allows the writing of simple input programs, which check for conditions or perform loops over certain parts of the input. `IF` blocks and `DO` loops have a similar form as in Fortran.

One line `IF`:

```
IF (condition) command
```

If more than one command depends on the condition, `IF` blocks can be used:

```
IF (condition) THEN
  commands
END IF
```

or

```

IF (condition) THEN
  commands
ELSE
  commands
END IF

```

Also the structure of DO loops is as in Fortran:

```

DO ivar=istart,iend,[increment]
  commands
ENDDO

```

ivar is the loop index variable, *istart*, *iend*, *increment* are either numbers or variables. The default for *increment* is 1.

Examples:

Loop over several geometries (potential curve for HCl):

```

***,HCl
geometry={          !Z-matrix geometry input
h
cl,h,r
}

r=1.5              !start value for bond distance
hf                !Hartree-Fock for start geometry

do i=1,10         !loop over bond distances

casscf;          !perform casscf
ecas(i)=energy   !save casscf energy in array ecas
mrci             !perform mrci
rhcl(i)=r        !save distances in array rhcl
emrci(i)=energy  !save mrci energies in array emrci
emrda(i)=energd  !save Davidson corrected energies in array emrda

r=r+0.2          !increment r
end do

```

Alternatively, one could predefine a number of distances:

```

***,HCl

rhcl=[1.6,1.8,2.0,2.2,2.3,2.4,2.5,2.7,3.0,3.5,4.0,5.0,6.0,7.0]

geometry={          !Z-matrix geometry input
h
cl,h,r
}

do i=1,#rhcl      !loop over all distances
r=rhcl(i)         !set r to current bond distance

if(i.eq.1) then  !in first calculation, do Hartree-Fock
  hf             !Hartree-Fock for start geometry
end if

casscf;          !perform casscf
ecas(i)=energy   !save casscf energy in array ecas

```

```

mrci                !perform mrci
rhcl(i)=r           !save distances in array rhcl
emrci(i)=energy     !save mrci energies in array emrci
emrda(i)=energd    !save Davidson corrected energies in array emrda
end do

```

One can skip to some command later in the input using GOTO. For instance

```

if(orbital.ne.0) goto casscf !skip to casscf if an orbital record exists
hf                          !Hartree-Fock
casscf;                     !casscf

```

16.7 Tables and Plotting

The results of the previous calculation can be printed in form of a table and plotted using the freely available plotting program XMGRACE. Appending the previous input by

```

{table,rhcl,ecas,emrci,emrda !print a table
plot,file='hcl.plot'        !generate an xmgrace plot file
Title,Results for HCl}      !title of table

```

prints the following table

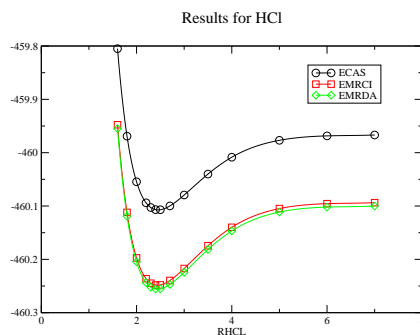
Results for HCl

RHCL	ECAS	EMRCI	EMRDA
1.6	-459.8049671	-459.9476516	-459.9549021
1.8	-459.9690821	-460.1118784	-460.1192275
2.0	-460.0545599	-460.1972494	-460.2046870
2.2	-460.0940081	-460.2362450	-460.2437423
2.3	-460.1028655	-460.2447475	-460.2522596
2.4	-460.1067696	-460.2482178	-460.2557336
2.5	-460.1069614	-460.2479059	-460.2554148
2.7	-460.0998440	-460.2396165	-460.2470838
3.0	-460.0793517	-460.2171250	-460.2244794
3.5	-460.0401054	-460.1744682	-460.1815696
4.0	-460.0085291	-460.1399223	-460.1467670
5.0	-459.9768057	-460.1047967	-460.1113263
6.0	-459.9685139	-460.0955779	-460.1020174
7.0	-459.9668469	-460.0937046	-460.1001220

Subsequent execution of

```
xmgrace hcl.plot
```

yields the following plot



16.8 Interface to MOLDEN

The geometries and orbitals can be saved for visualization using MOLDEN (see MOLDEN) using the PUT directive. In frequency calculations, also the normal modes are saved. An example is given below:

```

***, H2O
angstrom                !distances in angstrom
geometry={              !z-matrix input for h2o
o
h, o, roh
h, o, roh, h, theta
}

roh=1.0                 !bond distance
theta=104.0            !bond angle

hf                      !do hartree-fock
optg                   !optimize geometry at hf level
frequencies            !compute harmonic frequencies
put,molden,h2o.molden; !save results in file h2o.molden

```

The file h2o.molden can be used directly as input to *MOLDEN*.

16.9 molproView

molproView is a simple formatter for Molpro output files. It works by interpreting the XML markup in the output, and then translating it into an HTML formatted page. Molecular models are displayed using the Jmol toolkit.

molproView can be tried using either your own output files, or simple given examples, at <http://www.molpro.net/molproView> or it can be installed on your own machine.

Modes of use

Via a web browser: The URL should point to the place where the *molproView* has been installed. The resulting page prompts for the URL of the Molpro output file.

Via a shell script: If installed on your workstation,

```
molproView file | URL
```

URL should point to a Molpro output file produced with the `-X` or `--xml-output` option of molpro (usually having suffix `.xml`). If *molproView* has been installed in such a way that the configured web server has access to local files, a relative or absolute path pointing to the output file can be used instead of *URL*.

Direct production of html: Any Molpro output file with suffix `.xml` can be converted to html by copying it to the *molproView* source directory and typing `make`. If it is viewed in place, the Jmol models will work, but if it is copied elsewhere is likely that they will not. Features that require the Jmol applet to read auxiliary files (ie surface plotting) will not work, unless the html file is itself served through an http server.

Features The following, when found in the Molpro output, are recognized and marked up.

- All results (energies, properties).
- The input data for the job.
- Geometries, displayed using a configurable 3-dimensional model.
- 3-dimensional isosurface plots of molecular orbitals and electron densities, where these have been calculated by Molpro's CUBE facility.
- Intermediate points in geometry optimizations, either individually or as a complete trajectory. Graphical convergence of energy in geometry optimization.
- Generation of restart input files using the geometry at any chosen point in a geometry optimization within the job.
- Normal modes and frequencies, including graphical display of normal coordinates.
- Tables, including optional presentation as 2-dimensional plots using Google Chart.