User's Guide

DIESEL-MR-CI

by

Michael Hanrath and Bernd Engels

V 1.11

for DIESEL-MR-CI V 1.11

April 2000

User's Guide written by Michael Hanrath

Institute of Physical and Theoretical Chemistry
University of Bonn

Wegeler Straße 12
53115 Bonn
Germany

Contents

List of Tables

List of Figures

List of Listings

1 The program package

The following section discusses aspects common to all programs of the program package.

1.1 Program parts

The DIESEL-CI consists out of the main programs selector (”`sel`”), diagonalisator (”`diag`”) and Multi-Reference Perturbation Theory (”`mrpt`”). Table 1 shows a short description of these programs. In addition there are several tools to calculate density matrices or properties and the like. They are summarized in table 2.
The program ”`diesel`” is the central point of control and automization within the package. It improves user friendlyness and simplifies the task of an individually selected MR-CI calculation.

purpose	name of program	former[a] name	description
excitation and selection	`sel`	parkwa, parkeu	performs excitation from reference space and selects configurations with respect to several criterions
diagonalisation	`diag`	adler, condox	solving the eigenvalue problem (generation of the Hamilton Matrix and diagonalisation by a multi root Davidson algorithm)
MR-MP perturbation theory	`mrpt`	—	calculation of multi referenz perturbation theory (generation of the Hamilton Matrix and the inhomogenity, solving the linear equation system)

[a] These are the names within the old MRDCI package.

Table 1: Main programs of DIESEL-CI

1.2 Data flow

Figure 1 shows the flow of data in a DIESEL-CI calculation. For reasons of completeness the relevant MOLCAS programs and files required to generate the MO based integrals are also shown. The ”`form31`”-program transforms the ”`TRAONE`” and ”`TRAINT`” files into the STONEY-format[1]. This file contains symmetry information, one- and two-electron intergrals with double and single precision respectively.
Figure 1 is especially useful if you are carring out a property or natural orbital calculation.

[1] It was orginally used by the Hondo-package.

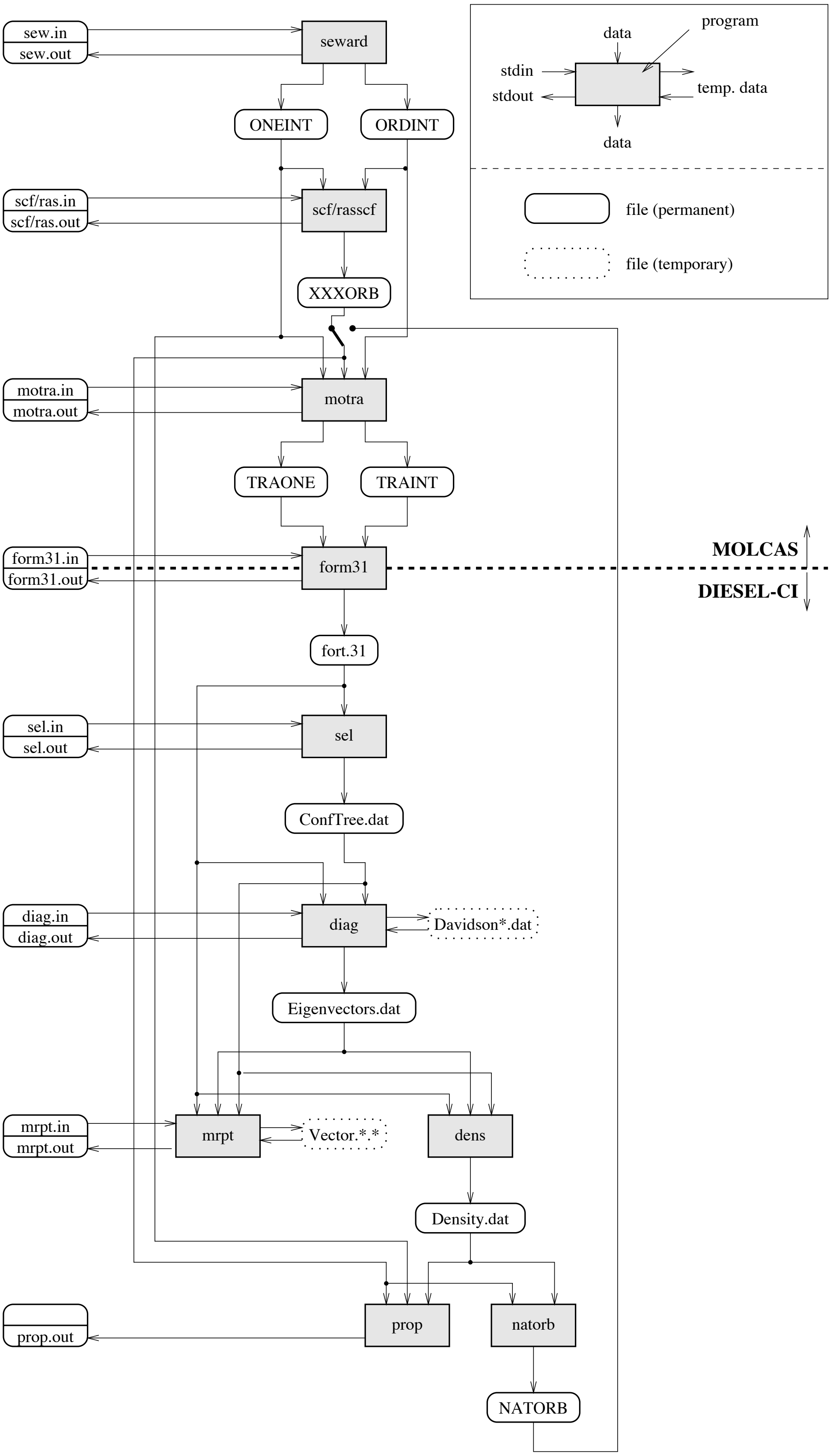

Figure 1: Programs and flow of data

purpose	name of program	former[a] name	description
control programm	`diesel`	—	automization of a MR-CI Rechnung: generation of space, perform selection of several thresholds, calculation of density matrices and properties, use of natural orbitals
density matrix calculation	`dens`	`jackal`	calculation of one-particle density matrices
natural orbitals	`natorb`	—	calculation of natural orbitals from one-particle density matrices, output to MOLCAS-format
properties	`prop`	`wolf`	calculation of properties from one-particle density matrices and one-electron integrals
set operations	`setops`	—	calculation of a set intersection, union, and difference
symmetrization	`symTree`	—	symmetrization of selected configurations with respect to MO equivalences
exciatation statistics	`confStat`	—	calculation of excitation levels in selected configuration with respect to a set of reference configurations
format conversion	`f31endian`	—	conversion of STONEY MO integral file between little- and big-endian architectures

[a] These are the names within the old MRDCI package.

Table 2: Tool programs of DIESEL-CI

1.3 User Input

There are two possible ways to supply the programs with input:

1. command line arguments
2. separate input files

Depending of the degree of complexity either method is chosen.

1.3.1 Philosophy

The input is designed on a "keyword = ..." philosophy.
Keywords are not case sensitive. The input is freely formatted. For a more precise input syntax rule specification see appendix A.
In order to improve readability the keywords described in the following sections are chosen to be quite long. As users tend to copy input files and successively

change parameters in them there should be no real drawback from the lack of abbreviation of the keywords.
The input in its completed form is repeated in the output. So one can check the "reasonability" of certain default parameters.

1.3.2 Structure of input description

The following sections describe the user input for selection, diagonalisation and MRPT programs. These sections are organized in the following way:

1. **invocation:** the name of program and how to call it
2. **command line arguments:** optional arguments to be given on the command line
3. **files:** input and output files with meaning and type (binary/ASCII)
4. **keywords:**
   The keywords are listed in a table consisting of 4 columns: keyword, status, argument type and description. The status and argument type need to be explained in more detail:

   (a) The status column:

   - required:
     A miss of this statement will cause an error.
   - optional/checked:
     If this statement is missing the appropriate value will be taken from the remaining input. If the statement is given the input will be checked if it complains. In other words the specification of this statement produces some kind of redundancy.
   - optional:
     The program will set some reasonable default values if this statement is missing.

   (b) The argument type:

   - num: natural number
   - floatnum: floating point number
   - numSet: set of natural numbers, written: $\{i_1 \dots i_n\}$
   - confSet: set of configurations, written: $\{\text{conf}_1 \dots \text{conf}_n\}$
     A configuration is a sequence of natural numbers representing the MO numbers. This sequence consists of three parts and is structured as follows:

     $$\underbrace{n_{\text{open}}}_{\substack{\#\\ \text{open}\\ \text{shells}}}\ \underbrace{s_1 \dots s_{n_{\text{open}}}}_{\substack{\text{open}\\ \text{shells}}}\underbrace{d_1 \dots d_{n_{\text{closed}}}}_{\substack{\text{closed}\\ \text{shells}}}$$

     example: 1 4 1 2 3 15 16
     (meaning: one open shell "4" and five closed shells "1 2 3 15 16")
     The number of closed shells n_{closed} is not explicitly given. Therefore to make this way of specification unique a configuration must be terminated by a separator.

- bool: "yes" or "no"

5. **input example**

1.4 Levels of verbosity

Table 3 shows the available levels of verbosity. All information is written to the standard output.

1.5 Directory Structure

In order not to end up in chaos the `diesel` driver program creates a specific directory structure. Before presenting it there should be some general thoughts about organization.

1.5.1 Some thoughts concerning organization

There are lots of data involved with carrying out some quantum chemical project. The natural organization form is some kind of a matrix with several dimensions (e.g. geometry, basis, multiplicity, irreducible representation). Now there is a contradiction between this structure and the tools given by most operating systems which offer a hierarchical structure made up of directories. As a consequence there are many different opinions and personal preferences of how to manage and organize such a problem.

As the hierarchical filesystem does not fit the problem structure one possibility is simply to ignore it and create a naming scheme like the following: "`1_2`" or somewhat more explicit "`Mult=1_IrRep=2`". The first one has the drawback of being not quite intuitive and the second one repeats some redundant information with every file. Both tend to become rather complex and ugly for problems described by matrices of higher dimensions.

A second possibility is to try to map the matrix problem to the hierarchical tree of directories. In such a scheme each level in the tree corresponds to one dimension in the matrix problem and every directory contains information on that dimension only. As a consequence without tricks you do not see any information from the higher levels of the tree. You may print the whole working directory path but in this case you would see again only "`1_2...`" or somewhat confusing and long "`Mult=1_IrRep=2...`". To get rid of these drawbacks one may put a tag into each directory describing the name of the dimension contained and use a shell script to evaluate the path. This method is described in the following section. As a partial drawback there is again a repetition of the names of a dimension in the distinct branches of the tree.

1.5.2 The `diesel` directory structure

The `diesel` directory structure consists mainly of the levels multiplicity and irreducible representation. Figure 2 shows an example. The dashed parts are optional and are only created if a calculation involving natural orbitals[2] or density matrices is carried out. Listing 1.1 shows the directory contents on the level of irreducible representations of some example.

[2] For details on how to do a calculation with natural orbitals see section **??**

	de-	available with			
keyword	fault	`sel`	`diag`	`mrpt`	description
Input	•	•	•	•	The input in its with default values completed form is echoed to the standard output.
Integrals	–	•	•	•	Information on the number of one- and two-electron integrals is printed to the standard output.
MOs	–	•	•	•	The following information is printed to the output: • MO-mapping to continous space within each irreducible representation • number of total MOs • MOs per irreducible representation (total/internal/external) • product table • list of internal/external MOs • share of internal MOs in percent
RefGuess	•	•	–	–	Details on the reference configuration first guess are printed.
SGA	–	•	•	•	Output from the SGA table initialisation is printed.
RefMat	–	•	•	•	The reference matrix is printed. Attention: probably quite large!
RefMatEigenValues	•	•	–	–	The eigenvalues of the reference matrix are printed.
RefMatEigenVectors	–	•	–	–	The first $1.5 * n_{\mathrm{roots}}$ reference vectors are written to the standard output.
IterationBlocks	•	–	•	–	The progress of one hamilton matrix generation and multiplication including
WaveFunction	•	–	•	–	The dominant configurations of the resultant wavefunction are printed out.
SelectionPerRoot	–	•	–	–	
CacheStatistics	–	•	•	•	The cache statistics are printed.
DegenGuess	•	•	–	–	Details on the MO degeneration guess are printed.
DiagHist	–	•	–	–	A histogram of the diagonal elements is printed.

Table 3: Verbosity levels, default values and availability

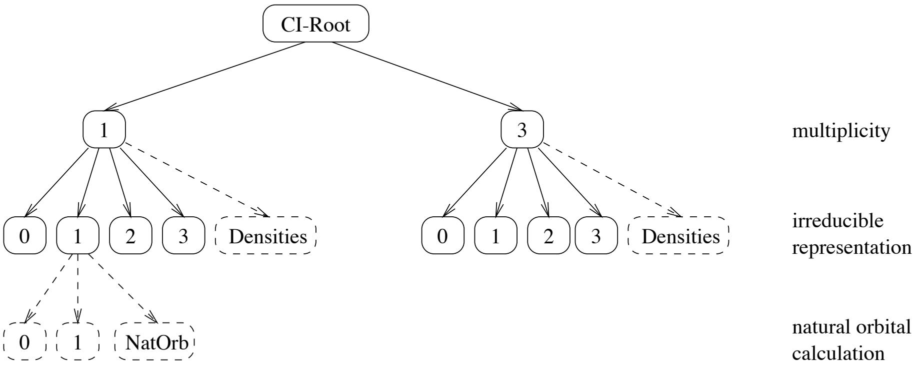

Figure 2: Directory structure

Listing 1.1 Listing of a directory on the level of irreducible representations

```
user@machine>ls
0/            5/            dens.out.1e-3  fort.31        prop.out.1e-7
1/            6/            dens.out.1e-4  prop.out.1e-3
2/            7/            dens.out.1e-5  prop.out.1e-4
3/            =Irrep        dens.out.1e-6  prop.out.1e-5
4/            Densities/    dens.out.1e-7  prop.out.1e-6
```

Listing 1.2 shows the output of the shell script `getDirInfo` in a directory of some imaginary project:

Listing 1.2 Project info by getDirInfo

```
user@machine>getDirInfo
            Project = c8
           Geometry = linear
              Basis = C.DUN...5S3P1D
      ReferenceType = CI.RefActive
       Multiplicity = 3
              Irrep = 3
```

2 Main Programs

2.1 Common Input Keywords

keyword	status	default	argument type	description
`Roots`	opt.	selector input	*numSet*	set of roots to be used in the diagonalisation procedure
`MOIntegral-Filename`	opt.	"fort.31" / "ftn31"	*name*	name of MO integral file
`MOIntegral-FileFormat`	opt.	"auto"	*MOFile-Format*	file format of MO integral file
`Verbosity`	opt.	app. dependent	*verbosity-Set*	

2.2 Driver

The diesel-driver was designed to simplify the application of an individually selecting MR-CI calculation. It handles the following tasks:

1. management of calculation of several multiplicities and irreps within a well defined directory structure
2. iterative generation of the MR-CI space (selection of References)
3. run selector and diagonalisator automatically on given thresholds
4. calculation of density matrices and properties
5. prepare results in a condensed form.

2.2.1 Invocation

```
diesel
```

2.2.2 Command Line Arguments

- none

2.2.3 Files

status	name	type	content
in	stdin	ASCII	user input
in	"$MOLCASRootDir-/fort.31"	binary	integrals in MO basis (AO→MO transformation performed), point symmetry, number of orbitals in certain irreducible representation
out	stderr	ASCII	protocol output, progress indication
out	stdout	ASCII	results (wave function, energies, ...)

2.2.4 User Input and Keywords

keyword	status	default	argument type	description
`Multiplicities`	req.	–	*numSet*	multiplicities to be calculated
`IrReps`	req.	–	*numSet*	irreducible representations to be calculated
`fullMRCI-Extrapolation`	req.		*ExtrapolationSet*	methods to be used for extrapolation
`useNatural-Orbitals`	req.	no	*boolean*	
`NaturalOrbital-Selection-Threshold`	opt.	–	*floatNum*	selection threshold to be used in natural orbital calculation
`averagedNatural-Orbitals`	opt.	no	*boolean*	use state averaged natural orbitals
`property-Thresholds`	opt.	{}	*floatSet*	thresholds to be used in property calculation
`orbitalFile`	req.		*name*	use state averaged natural orbitals

Additionally the following keywords are passed to the called subprograms. They are grouped by the subprograms.

- **Selector:**
  SelectionThresholds, NumberOfElectrons, ExcitationLevel, selectInternal, selectNthExcitation, MORestrictions, MOEquivalence, Roots, SelectionEstimationMode
- **Diagonalisator:**
  ReferenceThreshold, PTReferenceThreshold, ConvergenceEnergyChange, ConvergenceEigenvectorChange, MaxIters, Roots, MaxHamiltonStorageMem
- **MR-PT:**
  MRPTInhomogenityThreshold, MRPTSelectionThresholds
- **Reference Selection:**
  MOEquivalence, ReferenceThreshold, RefSelMode Roots, Irreps

For further information on these keywords please see sections 2.4.4, 2.5.4 and 2.6.3 respectively.

2.2.5 Input Example

```
# MO integral file
MOIntegralFilename              = fort.31
MOIntegralFileFormat            = New
MOLCASRootDir                   = /XYZ
# MOs
MORestrictions                  = none
MOEquivalence                   = auto

# electrons / state
NumberOfElectrons               = 24
Multiplicities                  = { 1 3 }
IrReps                          = { 0 1 2 3 4 5 6 7 }
Roots                           = { 1 2 3 4 5 6 7 8 }

# selection
SelectionThresholds             = { 1e-3 1e-4 1e-5 1e-6 1e-7 }
SelectionEstimationMode         = EpsteinNesbet
selectInternal                  = no
selectNthExcitation             = { }
ExcitationLevel                 = 2

# reference generation
ReferenceThreshold              = 0.004
maxRefGenIters                  = 6


# diagonalization / convergence
MaxDavidsonIters                = 40
ConvergenceEnergyChange         = 1
ConvergenceEigenvectorChange    = 1

# natural orbitals
#useNaturalOrbitals              = yes
#NaturalOrbitalSelectionThreshold = 1e-6
#averagedNaturalOrbitals         = no

# extrapolation
fullMRCIExtrapolation           = { EpsteinNesbet }
#MRPTInhomogenityThreshold       = 1e-4
MRPTSelectionThresholds         = { 1e-3 1e-4 1e-5 1e-6 }
```

```
# properties
propertyThresholds              = { 1e-3 1e-4 1e-5 1e-6 }
orbitalFile                     = RASORB

MaxHamiltonStorageMem           = 300MB
```

2.3 Reference Selection

2.3.1 Invocation

`refsel` *arguments*

2.3.2 Command Line Arguments

none.

2.3.3 Files

status	name	type	content
in	stdin	ASCII	user input (roots, reference selection threshold, etc.), see 2.5.4
in	”ConfTree.dat.ref”	ASCII	tree of selected configurations
in	”fort.31”	binary	integrals in MO basis (AO→MO transformation performed), point symmetry, number of orbitals in certain irreducible representation
out	stdout	ASCII	program output (containing protocol, recommended references, etc., depending on verbosity level)
out	*irrep*/”refs.out”	ASCII	recommended references

2.3.4 User Input and Keywords

keyword	status	default	argument type	description
`MOEquivalence`	opt.	none	*MOEquivalence*	equivalent (degenerated) MO list; if a selected configuration contains an equivalent MO the same configuration with equivalent MOs substituted is selected also

`Reference-Threshold`	opt.	0.004	*floatnum*	threshold for automatic reference space generation
`Roots`	req.	–	*numSet*	set of roots
`Irreps`	req.	–	*numSet*	set of irreps
`RefSelMode`	opt.	ConfThresh	*RefSelMode*	`ConfThresh`: all symmetry blocks of references with c^2 above `ReferenceThreshold` are selected, typical values: 0.001–0.01 `SumThresh`: all symmetry blocks of references ordered by c^2 and summing up to `ReferenceThreshold` are selected, typical values: 0.8–0.9

2.4 Selector

2.4.1 Invocation

`sel` *arguments*

2.4.2 Command Line Arguments

- `-r` : recalculation mode: perform the perturbation estimation on a given MR-CI tree (useful after an applied merge operation).

2.4.3 Files

status	name	type	content
in	stdin	ASCII	user input (reference configurations, selection thresholds, etc.), see 2.4.4
in	”fort.31”	binary	integrals in MO basis (AO→MO transformation performed), point symmetry, number of orbitals in certain irreducible representation
out	stdout	ASCII	program output (containing protocol, eigenvalues, statistics, etc., depending on verbosity level)
out	”ConfTree.dat”	ASCII	tree of selected configurations

2.4.4 User Input and Keywords

keyword	status	default	argument type	description
`Multiplicity`	req.	–	*num*	multiplicity of state (= $2S + 1$, S: spin quantum number)
`Selection-Thresholds`	req.	–	*floatSet*	a set of thresholds in Hartree for selection procedure
`RefConfs`	req.	–	*confSet*	set of configurations spanning the multi reference space (used as basis for excitations)
`NumberOf-Electrons`	opt.	conf. input	*num*	
`ExcitationLevel`	opt.	2	*num*	maximum excitation level used in MR-CI
`selectInternal`	opt.	no	*bool*	completely select internal space
`selectNth-Excitation`	opt.	{}	*numSet*	set of excitation levels to be completely selected
`AnnihilatorSpace`	opt.	inactive	*numSet*	orbitals from which electrons may be excited

`CreatorSpace`	opt.	inactive	*numSet*	orbitals to which electrons may be excited
`ActiveSpace-ExcitationLevel`	opt.	1	*num*	excitation level to generate references
`maxRefOpenShells`	opt.	4	*num*	maximum number of open shells for references
`FirstGuessConfs`	opt.	3	*num*	number of configurations increased by the number of ordered roots to be used from the first guess
`MORestrictions`	opt.	none	*MO-Restrict*	restrictions on the MO occupation pattern to be applied on the generated configuration space
`MOEquivalence`	opt.	none	*MOEquivalence*	equivalent (degenerated) MO list; if a selected configuration contains an equivalent MO the same configuration with equivalent MOs substituted is selected also

`MOStatistics`	opt.	no	*boolean*	print out MO statistics
`EstimationMode`	opt.	Epstein-Nesbet	*EstimationMode*	determines the method to estimate the energy contribution of a certain configuration
`StorePTEnergy`	opt.	no	*bool*	flag if perturbational energies are stored in configuration tree
`StorePTCoef`	opt.	no	*bool*	flag if perturbational CI coefficients are stored in configuration tree
`PTRefConfs`	opt.	`RefConfs`	*confSet*	set of configurations to be used as zero order wave function by perturbation theory

If the reference configurations are given explicitly they are checked for consistency among each other and the `IrRep`-keyword becomes optional. Any specified number of electrons or irreducible representation is checked against the given references.

2.4.5 Automatic Initial Reference Space Guess

Figure 3 shows the procedure that is used to generate a first reference configuration guess. It works pretty well for different irreducible representations and several roots. This feature is very useful as it saves one from the error intensive handling with the MO numbers. This procedure is quite reliable. There may be problems if many roots are ordered and the wave function is "diffuse". In that case the selection due to the diagonal element may be not really optimal.

2.4.6 Input Examples

first example:

```
# this is a comment
VerbosityLevel          = { RefGuess RefMatEigenValues IterationBlocks WaveFunction }
MORestrictions          = { 21-24>0 30,31=1 25<2 50,51,52<=2 }
MOIntegralFilename      = fort.31
MOIntegralFileFormat    = auto
NumberOfElectrons       = 19
Multiplicity            = 2
IrRep                   = 2
ExcitationLevel         = 2
SelectionThresholds     = { 0.5 }
Roots                   = { 1 2 5 }
selectInternal          = no
selectNthExcitation     = { }
RefConfs                = {
```

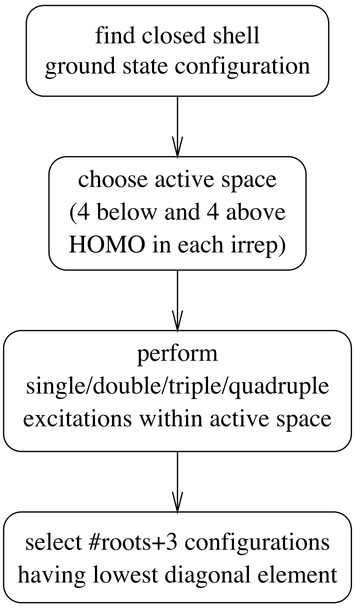
find closed shell ground state configuration
choose active space (4 below and 4 above HOMO in each irrep)
perform single/double/triple/quadruple excitations within active space
select #roots+3 configurations having lowest diagonal element

Figure 3: steps to generate first reference space guess

```
1 20 1 2 3 4 5 6 7 8 9
# another comment
1 21 1 2 3 4 5 6 7 8 9
3 10 15 25 1 2 3 4 5 6 7 8
}
```

second example:

```
#this works too!
MULTIPLICITY = 2;excitationlevel=2
NumberOfElectrons       = 19
 SelectionThresholds= { 0.5 }
RefConfs                = auto
```

Note that the starting configurations are generated automatically so there is no need to worry about the sophisticated process of MO-renumbering.

The reference configuration guess is performed by the following scheme:

1. The closed shell ground state occupation pattern is calculated. This is done by calculating the diagonal elements of the Hamilton Matrix whith any possible occupation pattern matching the irreducible representations and the given number of electrons. The configuration lowest in energy is selected as a base for the following step. If the number of given electrons is odd the calculation is done with the positive ion.
2. A certain set of MOs is chosen to be active. This set consists of the HOMO. . .HOMO−3 and the LUMO. . .LUMO+5 in every irreducible representation.

3. Within the active space single, double, triple, and quadruple excitations are performed and the diagonal hamilton matrix elements are calculated. The 3+*number of roots* configurations lowest in energy are chosen to be the reference configuration set.

This scheme is rather reliable especially for ground state occupation patterns. There may be problems if there are many roots ordered which can be described properly by several configurations only. Usually the following iterative space generation process is able to handle this problem.

2.5 Diagonalisator

2.5.1 Invocation

`diag` *arguments*

2.5.2 Command Line Arguments

- `-p procs`: number of parallel processes to be used in Davidson diagonalisation
- `-i`: read input from standard input (otherwise no user input is read)
- `-s` *thresh*: read start vectors from a previous smaller calculation at threshold *thresh*. This depends on the existence of the files "ConfTree.dat.*thresh*" and "Eigenvectors.dat.*thresh*"
- `-r`: restart a previously aborted calculation. This depends on the existence of the files "Davidson_b.dat" and "Davidson_Ab.dat".
- `-w`: write Hamiltonmatrix to file "MatrixStorageWrite.dat". No Davidson iteration is performed.

2.5.3 Files

status	name	type	content
in	stdin	ASCII	user input (roots, reference selection threshold, etc.), see 2.5.4
in	"ConfTree.dat"	ASCII	tree of selected configurations
in	"fort.31"	binary	integrals in MO basis (AO→MO transformation performed), point symmetry, number of orbitals in certain irreducible representation
out	stdout	ASCII	program output (containing protocol, eigenvalues, etc., depending on verbosity level)
out	"Eigenvectors.dat"	binary	eigenvectors of ci-matrix
temporary	"Davidson_b.dat"	binary	basis vectors in Davidson diagonalisation
temporary	"Davidson_Ab.dat"	binary	mapped basis vectors

2.5.4 User Input and Keywords

keyword	status	default	argument type	description
`Reference-Threshold`	opt.	0.004	*floatnum*	threshold for automatic reference space generation
`PTReference-Threshold`	opt.	0.004	*floatnum*	threshold for automatic generation of 0th-order wave function used in perturbation theory
`Convergence-EnergyChange`	opt.	1e-5	*floatnum*	energy change that is sufficient for convergence of a certain root
`Convergence-EigenvectorChange`	opt.	1	*floatnum*	eigenvector change that is sufficient for convergence of a certain root
`MaxIters`	opt.	20	*intnum*	maximum number of iterations
`Roots`	opt.	selector output	*numSet*	set of roots to be used in the diagonalisation procedure
`RootHoming`	opt.	no	*boolean*	freeze roots to characters given by the reference space
`StorePTEnergy`	opt.	no	*bool*	flag if perturbational energies are stored in configuration tree

`StorePTCoef`	opt.	no	*bool*	flag if perturbational CI coefficients are stored in configuration tree
`MaxHamilton-StorageMem`	opt.	0	*intnum* {GB, MB, KB}	maximum amount of memory available to store Hamilton Matrix; if there is not enough memory calculation falls back to direct mode
`Precision`	opt.	double	”float”, ”double”	precision of matrix elements and vectors (float: 32 bit[3], double: 64 bit[4]), using 32 bit-precision will result in less memory consumption (especially when using stored Hamilton Matrices), and slightly faster execution, but may cause convergence problems.
`IterationMode`	opt.	”CI”	”CI”, ”ACPF”, ”AQCC”	iteration mode

Since all input keywords are optional, the input may be missing completely. Therefore by default no input is read. To make the program read any user input from stdin an explicit flag (`-i`) must be specified.

2.6 MR Perturbation Theory

2.6.1 Invocation

`mrpt`

2.6.2 Files

status	name	type	content
in	stdin	ASCII	user input (selection thresholds, roots, etc.), see 2.6.3
in	"fort.31"	binary	integrals in MO basis (AO→MO transformation performed), point symmetry, number of orbitals in certain irreducible representation
in	"ConfTree.dat.*thresh*"	ASCII	tree of selected configurations
in	"Eigenvectors.dat.*thresh*"	binary	state eigenvectors
out	stdout	ASCII	program output (containing protocol, perturbation energies, etc., depending on verbosity level)

2.6.3 User Input and Keywords

keyword	status	default	argument type	description
`Selection-Thresholds`	req.	–	*floatSet*	a set of thresholds in Hartree for selection procedure
`Roots`	opt.	{ 1 }	*numSet*	set of roots to be used in the selection
`ProjectionMode`	opt.	"no0"	"no0", "0Complement"	
`Inhomogenity`	opt.	"Psi0" (|0⟩)	*floatnum*, "Psi0"	inhomogenity in system of linear equations
`calcMP3`	opt.	no	*boolean*	flag if MP3 energies should be calculated

2.6.4 Input Example

```
Roots                 = { 1 2 3 4 }
SelectionThresholds   = { 0.001 0.0001 1e-05 1e-06 }
ProjectionMode        = no0
Inhomogenity          = Psi0
calcMP3               = no
```

This input may result in the following output:

.
.

```
.

MR-MP2 Results:

      threshold/mH      sel. conf.       sel. CSFs     energy sum/mH     energy sum/mH     energy sum/mH     energy sum/mH
                                                            root # 1          root # 2          root # 3          root # 4
============================================================================================================================
      1.000000e+00             108             145      6.704442e+01      1.689585e+02      1.971939e+02      2.060199e+02
      1.000000e-01             671            1130      1.981235e+01      8.135658e+01      6.852698e+01      7.491947e+01
      1.000000e-02            2772            5563      4.856928e+00      2.998879e+01      2.144414e+01      2.325910e+01
      1.000000e-03            7913           18094      1.303418e+00      8.178762e+00      5.088133e+00      6.088736e+00
```

3 Examples

3.1 C_6 with automatic reference generation and property calculation

Listing 3.1 Listing of a simple diesel driver job file for C_6 with 20 core electrons using automatic reference generation

```
#!/bin/bash

export DIESEL_EXE_DIR=somepath
export MOLCAS_EXE_DIR=somepath
cd whereEver

cat <<! >diesel.in
# MO integral file
MOIntegralFileFormat             = New
MOLCASRootDir                    = ‘pwd‘/..

# MOs
MOEquivalence                    = auto

# electrons / state
NumberOfElectrons                = 16
Multiplicities                   = { 1 3 }
IrReps                           = { 0 1 3 }
Roots                            = { 1 2 3 4 }

# references
RefConfs                         = auto

#selection
SelectionThresholds              = { 1e-3 1e-4 1e-5 }

#properties
propertyThresholds               = { 1e-3 1e-4 1e-5 }
orbitalFile                      = INPORB

MaxHamiltonStorageMem            = 500MB
!

$DIESEL_EXE_DIR/diesel <diesel.in 1>diesel.out 2>diesel.prot.out
```

Listing 3.2 Protocol output (stderr)

```
*******************************************************************************
*                                                                             *
*                                diesel protocol                              *
*                                                                             *
*******************************************************************************


multiplicity=1
    irrep=0
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
            iteration #4
            iteration #5
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    irrep=1
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    irrep=3
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    calculating one particle density matrices
        threshold=1e-3
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
        threshold=1e-4
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
        threshold=1e-5
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
    one particle density matrices calculation finished
```

Listing 3.3 Protocol output (stderr, continued)

```
multiplicity=3
    irrep=0
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    irrep=1
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
            iteration #4
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    irrep=3
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished

    calculating one particle density matrices
        threshold=1e-3
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
        threshold=1e-4
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
        threshold=1e-5
            iirrep=0
                jirrep=1
                jirrep=3
            iirrep=1
                jirrep=3
            iirrep=3
            calculating properties
            property calculation finished
    one particle density matrices calculation finished
```

Listing 3.4 Recursive directory contents

```
user@machine>ls -R
1                =Multiplicity    diesel.in        diesel.prot.out
3                CI.job           diesel.out

1:
0              3              Densities      dens.out.1e-4  fort.31        prop.out.1e-4
1              =Irrep         dens.out.1e-3  dens.out.1e-5  prop.out.1e-3  prop.out.1e-5

1/0:
ConfTree.dat           Eigenvectors.dat.1e-5  genspace.0             sel.in.2
ConfTree.dat.0.001     diag.in                genspace.1             sel.in.3
ConfTree.dat.1e-3      diag.in.RefGen         genspace.2             sel.in.4
ConfTree.dat.1e-4      diag.out.1e-3          genspace.3             sel.in.all
ConfTree.dat.1e-5      diag.out.1e-4          genspace.4             sel.out.all
Eigenvectors.dat.1e-3  diag.out.1e-5          sel.in.0
Eigenvectors.dat.1e-4  fort.31                sel.in.1

1/1:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

1/3:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

1/Densities:
Density.dat.I0R1_I0R1.1e-3  Density.dat.I0R3_I1R2.1e-3  Density.dat.I1R2_I1R4.1e-3
.
.
.

3:
0              3              Densities      dens.out.1e-4  fort.31        prop.out.1e-4
1              =Irrep         dens.out.1e-3  dens.out.1e-5  prop.out.1e-3  prop.out.1e-5

3/0:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

3/1:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.0
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.1
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.2
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.in.3
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2             sel.in.all
Eigenvectors.dat.1e-3  diag.out.1e-4          genspace.3             sel.out.all

3/3:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

3/Densities:
Density.dat.I0R1_I0R1.1e-3  Density.dat.I0R3_I1R2.1e-3  Density.dat.I1R2_I1R4.1e-3
.
.
.
```

The property output is written to the files `prop.out.`*thresh*.

programs	motra form31	
read files	motra form31	
written files	motra form31	

Table 4: MOLCAS-dependencies

Listing 3.5 Report output (stdout)

```
============================================
             Project = c6
            Geometry = linear
               Basis = C.DUN...5S3P1D
            Orbitals = 2_2.CAS
                Core = 20e
       ReferenceType = RefIter
        Multiplicity = 1
               Irrep = 0
============================================



reference configurations:
0 1-2 23-24 35-36 50 72        # 1
.
.
.
4 35-36 84-85 1-2 23-24 50 72   # 32


number of configurations: 32
dimension of ci matrix  : 42


selected roots within reference space:
root #1: -226.910979
root #2: -226.901841
root #3: -226.725320
root #4: -226.721146


total generated configurations/CSFs:

                confs.        CSFs
intern-0:          352         603
intern-1:        23423       70382
intern-2:       531645     2215655
          ------------------------
total   :       555420     2286640



++++++++
root #1:
reference energy: -226.91098
character (at threshold 1.00e-02 mH, ci^2>0.01):
        0.0126213162       -0.1123446312:   ref.   0 1-2 23-24 35-36 50 72  # 1
        0.0120907020        0.1099577283:   ref.   0 1-2 23-24 35-36 50 84  # 2
        0.4144999165        0.6438166793:   ref.   0 1-2 23-24 35 50 72 84  # 4
        0.3989897564       -0.6316563594:   ref.   0 1-2 23 35-36 50 72 84  # 10

   threshold  sel. confs   sel. CSFs   CI energy    ref ci^2     overlap      PT(EN)      PT(EN)
         /mH                                  /H                     max         /mH weighted/mH
----------------------------------------------------------------------------------------------------
           1          18          28   -226.9351  0.97464208  0.99998533    -434.763  -434.76363
         0.1         648        1014  -227.04893  0.90646824  0.99992052    -299.391  -299.39962
        0.01        8279       14719  -227.19895  0.87097902  0.99995333    -100.904  -100.90425


extrapolation to full MRCI (all Energies in H):

   threshold _______________ EN ________________
         /mH      E(l=1)           l        E(l)
------------------------------------------------
           1  -227.36987         ---         ---
         0.1  -227.34833  0.84093363  -227.30071
        0.01  -227.29985  0.75575772  -227.27521


extrapolation to full CI (all Energies in H):

         /mH  ____________________  EN  ____________________
   threshold   ____ Davidson 1 _____   ____ Davidson 2 _____
                  E(l=1)        E(l)      E(l=1)        E(l)
------------------------------------------------------------
           1   -227.3815         ---  -227.38181         ---
         0.1  -227.38924  -227.33716  -227.39346  -227.34092
        0.01  -227.35003   -227.3222  -227.35746  -227.32916
++++++++
.
.
.
```

3.2 C_6 with active space references and root homing

Listing 3.6 Listing of a simple diesel driver job file for C_6 with 20 core electrons using active space references

```
#!/bin/bash

export DIESEL_EXE_DIR=somepath
export MOLCAS_EXE_DIR=somepath
cd whereEver

cat <<! >diesel.in
# MO integral file
MOIntegralFileFormat             = New
MOLCASRootDir                    = ‘pwd‘/..

# MOs
MOEquivalence                    = auto

# electrons / state
NumberOfElectrons                = 16
Multiplicities                   = { 1 3 }
IrReps                           = { 0 1 3 }
Roots                            = { 1 2 3 4 }

# references
RefConfs                         = {
0 1-2 23-24 35 50 72 84
0 1-2 23 35-36 50 72 84
2 24 36 1-2 23 35 50 72 84
}
AnnihilatorSpace                 = { 2 23 24 35 36 50 72 84 }
CreatorSpace                     = { 23 24 35 36 72 73 84 85 }
activeSpaceExcitationLevel       = 2
maxRefOpenShells                 = 6

RootHoming                       = yes

#selection
SelectionThresholds              = { 1e-3 1e-4 1e-5 }

MaxHamiltonStorageMem            = 500MB
!

$DIESEL_EXE_DIR/diesel <diesel.in 1>diesel.out 2>diesel.prot.out
```

If you later decide you need one more threshold simply add it in the input (s. Listing 3.7) and restart.

Listing 3.7 Input for an additional threshold

```
SelectionThresholds              = { 1e-3 1e-4 5e-5 1e-5 }
```

As you see in Listing 3.8 only the necessary steps are executed.

Listing 3.8 Protocol output (stderr)

```
multiplicity=1
        irrep=0
                using active space references
                performing selection on given thresholds
                diagonalization steps
                        threshold 1e-3: already done
                        threshold 1e-4: already done
                        threshold 5e-5
                        threshold 1e-5: already done
                diagonalization finished
.
.
.
```

The same way you may append additional irreducible representations or multiplicities. The output of results will contain all information as if you originally started with the whole set of parameters (s. Listing 3.9). If you want to change something different from that described above (e. g. different fort.31-file, reference configurations, convergence criterions, ...) please delete the calculation tree before restarting the calculation in order to get what you want.

Listing 3.9 Report output (stdout)

```
============================================
             Project = c6
            Geometry = linear
               Basis = C.DUN...5S3P1D
            Orbitals = 2_2.CAS
                Core = 20e
       ReferenceType = RefActive
        Multiplicity = 1
               Irrep = 0
============================================



reference configurations:
0 1-2 23-24 35-36 50 72         # 1
.
.
.
4 72-73 84-85 1-2 23 35-36 50   # 72


number of configurations: 72
dimension of ci matrix  : 100


selected roots within reference space:
root #1: -226.926197
root #2: -226.917450
root #3: -226.743097
root #4: -226.738563


total generated configurations/CSFs:

                confs.        CSFs
intern-0:          412         751
intern-1:        41684      133864
intern-2:      1133336     5392979
          ------------------------
total   :      1175432     5527594



++++++++
root #1:
reference energy: -226.9262
character (at threshold 1.00e-02 mH, ci^2>0.01):
        0.0121657647       -0.1102985255:   ref.   0 1-2 23-24 35-36 50 72  # 1
        0.0122560845        0.1107072015:   ref.   0 1-2 23-24 35-36 50 84  # 2
        0.4050904682        0.6364671777:   ref.   0 1-2 23-24 35 50 72 84  # 5
        0.4078713680       -0.6386480784:   ref.   0 1-2 23 35-36 50 72 84  # 15

   threshold  sel. confs   sel. CSFs   CI energy    ref ci^2     overlap      PT(EN)      PT(EN)
         /mH                                  /H                     max         /mH weighted/mH
----------------------------------------------------------------------------------------------------
           1          10          14  -226.93881  0.98552683  0.99987399     -427.22  -427.22388
         0.1         603         953  -227.04791  0.91783744  0.99999991     -301.27  -301.27001
        0.05        1501        2383  -227.09915  0.90225781  0.99994973    -235.914  -235.92348
        0.01        8088       14344  -227.19816  0.88045173  0.99997603    -106.048  -106.05273


extrapolation to full MRCI (all Energies in H):

   threshold _______________ EN ________________
         /mH      E(l=1)           l        E(l)
------------------------------------------------
           1  -227.36603         ---         ---
         0.1  -227.34918  0.86624541  -227.30889
        0.05  -227.33508  0.78410956  -227.28414
        0.01  -227.30422  0.76237506  -227.27902


extrapolation to full CI (all Energies in H):

         /mH   ____________________ EN _____________________
   threshold   ____ Davidson 1 _____   ____ Davidson 2 _____
                  E(l=1)        E(l)      E(l=1)        E(l)
------------------------------------------------------------
           1   -227.3724         ---  -227.37249         ---
         0.1  -227.38394  -227.34033  -227.38705  -227.34315
        0.05  -227.37504  -227.31913  -227.37937  -227.32292
        0.01  -227.34941  -227.32119  -227.35554  -227.32692
++++++++
.
.
.
```

3.3 Using natural orbitals

Listing 3.10 Natural orbital calculation

```
# MO integral file
#MOIntegralFilename               = fort.31
MOIntegralFileFormat             = New  #test
MOLCASRootDir                    = ‘pwd‘/..
# MOs
#MORestrictions                   = none
MOEquivalence                    = auto

# electrons / state
NumberOfElectrons                = 16
Multiplicities                   = { 3 }
IrReps                           = { 3 }
Roots                            = { 1 2 3 4 }

RefConfs                         = auto

RootHoming                       = yes

# selection
SelectionThresholds              = { 1e-3 1e-4 1e-5 }

# natural orbitals
useNaturalOrbitals               = yes
NaturalOrbitalSelectionThreshold = 1e-5
orbitalFile                      = RASORB
#averagedNaturalOrbitals          = no

MaxHamiltonStorageMem            = 500MB
```

Be careful to set the `orbitalFile = XXXORB` keyword. This has to be the orbital file you previously generated the STONEY file with.

Listing 3.11 Protocol output

```
*******************************************************************************
*                                                                             *
*                                diesel protocol                              *
*                                                                             *
*******************************************************************************


multiplicity=3
    irrep=3
        creating natural orbitals
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-5
        diagonalization finished
        calculating density matrices
        calculating natural orbitals
        performing MO transformation for root #1
        performing MO transformation for root #2
        performing MO transformation for root #3
        performing MO transformation for root #4
        performing MR-CI calculation with natural orbitals for root #1
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
            iteration #4
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished
        performing MR-CI calculation with natural orbitals for root #2
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished
        performing MR-CI calculation with natural orbitals for root #3
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished
        performing MR-CI calculation with natural orbitals for root #4
        creating reference space:
            iteration #1
            iteration #2
            iteration #3
            iteration #4
            iteration #5
            iteration #6
        reference space generation completed
        performing selection on given thresholds
        diagonalization steps
            threshold 1e-3
            threshold 1e-4
            threshold 1e-5
        diagonalization finished
```

programs	motra form31	
read files	ONEINT ORDINT STONEY	
written files	STONEY.NatOrb.n	

Table 5: MOLCAS-dependencies

Listing 3.12 Recursive directory contents for natural orbital calculation

```
user@machine>ls -R
3                CI.job           diesel.out
=Multiplicity    diesel.in        diesel.prot.out

3:
3       =Irrep

3/3:
1            2            3            4            =NatOrbRoot  NatOrb

3/3/1:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.0
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.1
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.2
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.in.3
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2             sel.in.all
Eigenvectors.dat.1e-3  diag.out.1e-4          genspace.3             sel.out.all

3/3/2:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

3/3/3:
ConfTree.dat           Eigenvectors.dat.1e-4  diag.out.1e-5          sel.in.1
ConfTree.dat.0.001     Eigenvectors.dat.1e-5  fort.31                sel.in.2
ConfTree.dat.1e-3      diag.in                genspace.0             sel.in.all
ConfTree.dat.1e-4      diag.in.RefGen         genspace.1             sel.out.all
ConfTree.dat.1e-5      diag.out.1e-3          genspace.2
Eigenvectors.dat.1e-3  diag.out.1e-4          sel.in.0

3/3/4:
ConfTree.dat           Eigenvectors.dat.1e-5  genspace.0             sel.in.1
ConfTree.dat.0.001     diag.in                genspace.1             sel.in.2
ConfTree.dat.1e-3      diag.in.RefGen         genspace.2             sel.in.3
ConfTree.dat.1e-4      diag.out.1e-3          genspace.3             sel.in.4
ConfTree.dat.1e-5      diag.out.1e-4          genspace.4             sel.in.5
Eigenvectors.dat.1e-3  diag.out.1e-5          genspace.5             sel.in.all
Eigenvectors.dat.1e-4  fort.31                sel.in.0               sel.out.all

3/3/NatOrb:
ConfTree.dat                Density.dat.I.R2_I.R4.1e-5  fort.31
ConfTree.dat.0.001          Density.dat.I.R3_I.R3.1e-5  genspace.0
ConfTree.dat.1e-5           Density.dat.I.R3_I.R4.1e-5  genspace.1
Density.dat.I.R1_I.R1.1e-5  Density.dat.I.R4_I.R4.1e-5  genspace.2
Density.dat.I.R1_I.R2.1e-5  Eigenvectors.dat.1e-5       sel.in.0
Density.dat.I.R1_I.R3.1e-5  dens.out                    sel.in.1
Density.dat.I.R1_I.R4.1e-5  diag.in                     sel.in.2
Density.dat.I.R2_I.R2.1e-5  diag.in.RefGen              sel.in.all
Density.dat.I.R2_I.R3.1e-5  diag.out.1e-5               sel.out.all
```

3.4 Specialities

3.4.1 Distinct number of roots in each irrep/multiplicity

If you want to calculate a different number of roots in each irrep or multiplicity you will have to calculate each irrep/multiplicity seperately. If want to get properties from a stepwise calculation you are on your own. You have to call the "dens" and the "prop" programs manually.

4 Tool Programs

Most of the following tool programs are called from within the `diesel` main driver program. So you usually do not need to interact with them directly.

4.1 Diesel Results (`dr`)

This program collects the energy results of the calculation within one irrep, performs several extrapolation schemes, and prints the results.

command line: `dr [-R=`*n*`] [-T=`*t*`] [-C=`*r*`,`*c*`] [-h] [-w]`

1. -R: restrict output to root *n*
2. -T: restrict output to threshold *t*
3. -C: print row *r* and column *c* for each root block only
4. -h: surpress table headers
5. -w: surpress wave function output

These options are especially useful to grep a certain number out of the output in order to generate a potential surface for example.

4.2 Getting Project Information (`getDirInfo`)

`getDirInfo` evaluates the directory structure of a calculation and prints information about it.
For example:

```
     Project = c6
    Geometry = linear
       Basis = C.DUN...5S3P1D
Multiplicity = 3
       IrRep = 3
```

4.3 Grepping the Dominating Configurations (`grepImp`)

command line: `grepImp` *threshold* `<diag.out.`*thresh*
Print the configurations in the wave function having a coefficient greater than *threshold*.

4.4 Calculation of excitation statistics (`confStat`)

command line: `confStat ConfTree.dat` *ref*
Print the excitation levels in tree ”`ConfTree.dat`” relative to configuration in file *ref*.

4.5 Symmetrization of selected configurations (symTree)

command line: `symTree "equivalence" ["fort.31"-file]`

4.6 Symmetrization of references (refsym) (obsolete)

command line: `refsym <equivalences irrep sel.in.all-files`
The file "equivalences" contains equivalent MOs per line. For example if 1 2 3 were σ- and (10,20), (11,21) were π-orbitals this file would look like:

```
1
2
3
10 20
11 21
```

The references in "sel.in.all" are transformed to the representation of the degenerated point group. By transforming this orbital representation back to the original non degenerated point group the program completes or discards whole classes of configurations with respect to the full point group.

4.7 Listing the Configuration Tree (lstconfs)

command line: `lstconfs ConfTree.dat`
For example:

```
lstconfs (Part of DIESEL-MR-CI), Version 1.08pre, 22. Jan 1999

intern-0: 125
intern-1: 10
intern-2: 3

number of reference configurations          : 112
number of CSFs from reference configurations: 216
total number of configurations              : 138
total number of CSFs                        : 254

  ref.   2 38 58 1-3 39-40 59-60 83-85 121-122 141-142
  ref.   2 38 58 1-3 39-40 59 83-85 121-123 141-142
  ref.   2 38 58 1-3 39-40 59 83-85 121-122 141-143
  ref.   2 38 58 1-3 39 59-60 83-85 121-123 141-142
  ref.   2 38 58 1-3 39 59-60 83-85 121-122 141-143
.
.
.
```

4.8 Set Operations (setops)

The program calculates merge, intersection or difference sets of given configurations and writes them to the standard output in tree format.

command line: `setops -c|s|t -m|i|d file1 file2 ...`

1. -c: configurations in plain format
2. -s: configurations to be read from selector input
3. -t: configurations in tree format (as produced by the selector)

1. -m: perform merge
2. -i: perform intersection
3. -d: perform difference

4.9 Fort.31 File Format Conversion (`f31endian`)

The program tt f31endian is capable of converting the "fort.31" integral file format between little (e. g. Intel, Transputer, VAX) and big endian (e. g. RISC, HP, IBM, Sun) notation.
command line: `f31endian l2b|b2l old|new|tradpt input-filename output-filename`

1. l2b: little → big endian conversion
2. b2l: big → little endian conversion

1. old: fort.31 format from the HONDO program suite
2. new: fort.31 format from the MOLCAS program suite
3. tradpt: fort.31 format from the TRADPT program

4.10 One Electron Density Matrices (`dens`)

The program `dens` calculates the (transition) one electron density matrices. It depends on the configuration tree and the eigenvector files from a previous MR-CI calculation.

command line:
`dens motra-input-file Thresh`
`#Lstate1[-#LstateN], all Ldir [#Rstate1[-#RstateM], all` [Rdir]]
state means the nth state calculated in an MR-CI calculation. Arguments in square brackets are optional. If missing they default to corresponding arguments given first. The output is written to a file "Density.dat".

4.11 Natural Orbitals (`natorb`)

The program `natorb` calculates natural orbitals from one electron density matrices.

command line:
`natorb DensityMatrix1 ... [-weight w1 ... ] <InputOrbitals >OutputOrbitals`
Several density matrices may be weighted by the option *weight*. The input and output orbitals are in MOLCAS format.

4.12 Properties (`prop`)

The program `prop` calculates one electron operator properties based on the one electron density matrices generated with the program described in 4.10 and on the one electron integral file (ONEINT) generated by the MOLCAS program package. The result is written to stdout.

command line:
`prop` *operator component IntegralPath OrbitalPath DensityPath* ...
with

operator:	{ "Mltpl1", "Kinetic", "OneHam" }
component:	{ 1, 2, ... }
IntegralPath:	path to MOLCAS ONEINT File
OrbitalPath:	path to MOLCAS orbitals file (e.g. "SCFORB", "RASORB")
DensityPath:	path to CI density matrix(ces) (generated with "dens")

4.12.1 Driver to calculate and nicely print several Properties (`prettyProp`)

The program `prettyProp` is a driver for the `prop`-program.
calculates one electron operator properties based on the one electron density matrices generated with the program described in 4.10 and on the one electron integral file (ONEINT) generated by the MOLCAS program package. The result is written to stdout.

command line:

A Sytax Rules

Keywords are not case sensitive.

A.1 Common

statement ::= *comment* | *assignment* | *separator*;

comment ::= # *text newline*;

eq ::= *ws* = *ws*;

confSet ::= { [*conf* ... *conf*] };

floatSet ::= { *floatnum* ... *floatnum* };

numSet ::= { *numEnum* };

verbositySet ::= { [*verbosityKey* ... *verbosityKey*] };

conf ::= *numEnum separator*;

numEnum ::= [*numRange* [(*ws numRange*) ... (*ws numRange*)]];

numRange ::= *num* [- *num*];

separator ::= ; | *newline*;

newline ::= \n;

text ::= *char* ... *char*;

name ::= *alpha* [*alphanum* ... *alphanum*];

char ::= any printable character;

floatnum ::= [+|-] [*digit* ... *digit*] . [*digit* ... *digit*] [e|E [-|+] *num*];

num ::= *digit* [... *digit*];

alphanum ::= *alpha* | *digit*;

alpha ::= A | ... | Z | a | ... | z;

digit ::= 0 | ... | 9;

bool ::= yes | no;

ws ::= *blank* [... *blank*];

blank ::= <space> | \t;

verbosityKey ::= Input | Integrals | MOs | RefGuess SGA | | IterationBlocks |
RefMat | RefMatEigenValues | RefMatEigenVectors |
IterationBlocks | WaveFunction | SelectionPerRoot |
CacheStatistics;

A.2 Selector

Selector-Input ::= *statement* ... *statement*;
assignment ::= *nameAssign* | *keyAssign* | *boolAssign*
| *numAssign* | *floatAssign* | *setAssign*;
setAssign ::= *numSetAssign* | *confSetAssign*;
nameAssign ::= *nameAssignKeywords eq name separator*;
keyAssign ::= *FileFormatAssign*;
boolAssign ::= *boolAssignKeywords eq bool separator*;
numAssign ::= *numAssignKeywords eq num separator*;
floatAssign ::= *floatAssignKeywords eq foatnum separator*;
FileFormatAssign ::= *MOIntegralFileFormatAssign eq*
Fort31RecordFormatNew | Fort31RecordFormatOld
separator;
numSetAssign ::= *numSetAssignKeywords eq numSet separator*;
ConfSetAssign ::= *numSetAssignKeywords eq confSet separator*;
boolAssignKeywords ::= selectInternal;
numAssignKeywords ::= NumberOfElectrons | ExcitationLevel | Multiplicity | IrRep;
floatAssignKeywords ::= SelectionThreshold;
nameAssignKeywords ::= MOIntegralFilename;
MOIntegralFileFormatAssign ::= MOIntegralFileFormat;
numSetAssignKeywords ::= Roots | selectNthExcitation;
confSetAssignKeywords ::= RefConfs | PTRefConfs;

A.3 Diagonalisator

diagVerb ::= Input | MOs | Spin | Integrals |
RefMat | RefEigs | RefVectors |
IterBlocks | Wave | NewConfs;

A.4 Reference Selection

RefSelMode ConfThresh | ConfSum

B Citation

Please cite this program package as:
M. Hanrath, B. Engels: ”New algorithms for an individually selecting MR-CI program”, Chemical Physics, **255** (1997), 192-202.