Computing for Science (CFS) Ltd.,
CCLRC Daresbury Laboratory.

Generalised Atomic and Molecular Electronic Structure System

GAMESS-UK

USER'S GUIDE and REFERENCE MANUAL

Version 8.0 June 2008

PART 4. DATA INPUT - CLASS 2 Directives

M.F. Guest, H.J.J. van Dam, J. Kendrick, J.H. van Lenthe, P. Sherwood and G. D. Fletcher

Contents

1 Introduction

The *Class 2* directives define the nature of the computation in hand, and are used to input the details necessary to enable this computation to proceed. In this chapter we concentrate on those directives required to characterise;

1. SCF, Møller Plesset, MCSCF, MASSCF and CASSCF calculations;
2. Density Functional Theory (DFT) calculations;
3. Geometry optimisation, transition state calculations and force constant calculations.

Data input required in performing Direct-CI and Table-CI calculations, and in using the various wavefunction analysis tools are considered below in Parts 5,6 and 7 respectively.

2 RUNTYPE

The RUNTYPE directive is used to define the type of computation to be performed in the present run of the program, and consists of a single a line with the first two data fields read to variables TEXT, TYPE using format (2A).

- TEXT should be set to the character string RUNTYPE
- TYPE should be set to a character string defining the type of computation. Valid strings are shown in the table below.

Subsequent data fields are a function of the TYPE setting, and may be used to further characterise the nature of the computation. We consider the format for each TYPE specification in the notes below.

- In the absence of a RUNTYPE directive, the default of a single point SCF calculation is assumed.
- When restoring a Hessian Matrix for use in OPTIMISE and SADDLE computations, the source of this Hessian must be specified on the RUNTYPE directive. This may take two forms;
  1. When the initial Hessian has been computed under control of the RUNTYPE HESSIAN directive, the additional keyword FCM should be specified on the OPTIMISE or SADDLE data line, thus

     ```
     RUNTYPE SADDLE FCM
     ```

  2. When the Hessian is to be restored from some smaller basis set calculation, i.e. from a Dumpfile separate to that of the present calculation, this 'foreign Dumpfile' must be specified on the RUNTYPE directive, so the data line

```
RUNTYPE SADDLE ED4 150
```

requests the initial Hessian to be restored from the Dumpfile commencing at block 150 on the data set assigned with the LFN ED4.

RUNTYPE INTEGRAL	single point 1e- and 2e-Integral evaluation
RUNTYPE SCF	single point Integrals + SCF calculation
RUNTYPE OPTIMISE	search for a local minimum on the potential energy surface using an internal-coordinate quasi-Newton rank-2 update method.
RUNTYPE OPTXYZ	search for a local minimum on the potential energy surface using a cartesian-based quasi-Newton update method.
RUNTYPE SADDLE	search for a saddle point on the potential energy surface, using in default the 'trust region' method [5].
RUNTYPE FORCE	numerical force constant calculation at an equilibrium geometry.
RUNTYPE HESSIAN	analytical force constant calculation at an equilibrium geometry.
RUNTYPE POLARISABILITY	Polarisability calculation
RUNTYPE HYPER	Hyperpolarisability calculation
RUNTYPE MAGNET	Magnetisability calculation
RUNTYPE RAMAN	Calculation of Raman Intensities
RUNTYPE INFRARED	Calculation of IR intensities
RUNTYPE TRANSFORM	single point Integrals, SCF and 4-index transformation
RUNTYPE CI	single point Integrals, SCF, transformation and CI calculation
RUNTYPE GF	single point Integrals, SCF, transformation and Green's function OVGF calculation
RUNTYPE TDA	single point Integrals, SCF, transformation and Green's function 2ph-TDA calculation
RUNTYPE RESPONSE	single point Integrals, SCF, transformation and Response function (RPA, TDA or MCLR) calculation
RUNTYPE ANALYSE	analyse a nominated set of eigenvectors, either by computing 1-electron properties, localised orbitals, or by performing DMA, Mulliken or graphical analysis.

2.1 Notes on RUNTYPE Specification

The simplest RUNTYPE is either that requesting just an SCF calculation (TYPE=SCF) or some analysis of a pre-computed wavefunction (TYPE=ANALYSE). All other TYPEs comprise multiple tasks, each of which could itself be controlled by its own TYPE specification. Consider, for example, Direct-CI calculations performed under TYPE=CI specification; this of course involves three separate tasks, namely

- the SCF computation
- the integral transformation
- the Direct-CI calculation

The TYPE=CI specification implies the execution of all *three* steps. Performing the calculation under such control would involve the straightforward specification

```
RESTART CI
..
..
RUNTYPE CI
..
```

in any restart jobs. Consider now performing the same calculation in a sequence of steps. The SCF computation might, say, be completed as the first step, under control of RUNTYPE SCF specification. The second step, the integral transformation, might involve the directive sequence

```
RESTART NEW
..
BYPASS SCF
..
RUNTYPE TRANSFORM
..
```

assuming of course that the two-electron integral file in the SCF step had been generated in the appropriate format. Note that since the TRANSFORM specification implies both SCF and integral transformation, we need BYPASS the SCF step. Finally, having generated the transformed integrals. the third step would involve the data specification

```
RESTART NEW
..
BYPASS TRANSFORM
..
RUNTYPE CI
..
```

This type of breakdown is typical of that employed when

- performing many different CI calculations based on the same set of transformed integrals
- having to analyse the SCF calculation to provide the necessary data specification for the subsequent CI

Strictly speaking, the TYPE=SCF specification itself implies of course the execution of two separate steps, but it is unusual to invoke this functionality. The SCF job above could be split into two jobs, the first involving just integral evaluation, through the specification

```
RUNTYPE INTEGRAL
```

and the second by the directive sequence

```
RESTART NEW
..
BYPASS
..
RUNTYPE SCF
..
```

where the BYPASS directive avoids regeneration of the integral list.

A similar sequence to the CI example above is often required when carrying out both OVGF and TDA Green's function calculations of ionization energies. This would normally involve performing an initial SCF calculation, under control of RUNTYPE SCF, followed by the Green's function calculation, thus

```
RESTART NEW
..
BYPASS SCF
..
RUNTYPE GF
..
```

where the integral transformation and OVGF computation of the ionization energies are performed in the same job.

Finally consider RUNTYPE specification when performing geometry or transition state calculations. It is of course possible in the startup job to present a data sequence of the form

```
..
..
RUNTYPE OPTIMIZE
ENTER
```

If however the SCF converges to, say, an excited state on the first point, any subsequent computation will be wasted. It is normally better practice to perform the initial SCF under RUNTYPE SCF control, then initiate the optimization with a sequence such as

```
RESTART NEW
..
BYPASS SCF
..
RUNTYPE OPTIMIZE
ENTER
..
```

Subsequent restarts would be carried out with the sequence

```
RESTART OPTIMIZE
..
..
RUNTYPE OPTIMIZE
ENTER
..
```

Although perhaps an obvious point, note the removal of the BYPASS directive – failure to remove this will almost certainly lead to an erroneous energy and/or gradients in the optimisation, with predictable consequences on the convergence of the geometry optimization.

3 SCFTYPE

The SCFTYPE directive specifies the category of self-consistent field wavefunction to be used in conducting the task nominated by the RUNTYPE directive. Valid data lines include the following

SCFTYPE RHF	perform a restricted Hartree-Fock calculation
SCFTYPE DIRECT RHF	perform a restricted Hartree-Fock Direct-SCF calculation
SCFTYPE UHF	perform an unrestricted Hartree-Fock calculation
SCFTYPE DIRECT UHF	perform an unrestricted Direct-SCF calculation
SCFTYPE GVB *n*	perform a GVB-n/PP calculation i.e. involving *n* GVB pairs.
SCFTYPE DIRECT GVB *n*	perform a direct-GVB-n/PP calculation
SCFTYPE MP2	perform a second-order M oller Plesset MP2 calculation
SCFTYPE DIRECT MP2	perform a M oller Plesset Direct-MP2 calculation
SCFTYPE MP3	perform a third-order M oller Plesset MP3 calculation
SCFTYPE CASSCF	perform a complete active space SCF calculation
SCFTYPE MCSCF	perform a second-order MCSCF calculation
SCFTYPE MASSCF	perform an ORMAS-MCSCF calculation

The following points should be noted;

- in the absence of a SCFTYPE directive, a restricted Hartree-Fock calculation is assumed for both closed- and open-shell systems.
- performing RHF calculations on open-shell systems will assume the appropriate high-spin configuration in default. If this is not the required configuration, then the OPEN directive must also be specified to define the required orbital occupancies.
- when performing CASSCF calculations, the CONFIG directive must be specified.
- when performing MCSCF calculations, the MCSCF and ORBITAL directives must be specified.
- when performing MASSCF calculations, a MASSCF section must be present in the input that specifies values for at least the following directives: NCORE, NACT and NELS.

- the significance of the sections nominated under control of the VECTORS and ENTER directives is a function of SCFTYPE.
- when restarting a task involving GVB wavefunctions, the RESTORE parameter must be specified on the SCFTYPE directive. Thus the data line

  ```
  SCFTYPE GVB 1 RESTORE
  ```

  signifies a GVB-1/PP calculation, with pair coefficients restored from the Dumpfile.
- the format of the LEVEL directive is a function of SCFTYPE.

4 Directives defining the wavefunction

4.1 SCF Wavefunctions: The OPEN Directive

The default electronic configuration in open-shell RHF calculations corresponds to the high-spin configuration; in such cases the OPEN directive is not required. If this default does not apply then the OPEN directive must be used to define the electronic configuration and hence the energy expression in both open-shell RHF and open-shell GVB calculations. In performing such calculations, the User must define the shell structure, where orbitals which can have the same Fock operator are said to belong to the same shell [2]. The OPEN directive is used to define

- the number of orbitals in each open shell
- the number of electrons in each open shell

The general syntax of the directive is

```
OPEN  NO1 NE1  NO2 NE2  .......  STATE
```

where (NO1,NE1) correspond to the number of orbitals and electrons respectively in the first open shell, (NO2,NE2) to the number of orbitals and electrons in the second open shell, and so on until all open shells have been specified. As outlined in Section 3.II, certain assumptions are made concerning the ordering of the input orbitals within RHF and GVB modules, and this ordering ties in with the OPEN definition. Specifically, it is assumed that the orbitals are ordered thus:

- all doubly occupied orbitals occur first in the list, the complete set defining Shell 1, the closed-shell.
- following the closed-shell manifold comes the open-shells, with the number of orbitals and electrons in each open-shell defined by the OPEN directive.
- the GVB pair orbitals.

In many cases the specification of the number of component orbitals and electrons associated with each open shell provides, together with the MULT directive, a unique definition of the associated electronic state. When this is not the case, and ambiguity remains in the definition, the character string STATE may be used to nominate the required state.

A full list of the possible STATEs available is shown in Table 1. below. Note they include all those possible under the SERHF and GRHF options of ATMOL3 [8]. The STATE parameter is not required when no ambiguity in the required energy expression occurs.

Example

When performing an open shell SERHF calculation on the $^4\Sigma^-$ state of NH$^+$, characterised by the configuration

$$1\sigma^2 2\sigma^2 3\sigma^1 1\pi_x 1\pi_y \tag{1}$$

The data line

```
OPEN 1 1 2 2
```

would be required, together with a MULT 4 specification, with the input orbitals ordered thus:

M.O.	1	2	3	4	5
Symmetry	1σ	2σ	3σ	1πx	1πy

where shell 1 comprises the closed-shell orbitals (1σ , 2σ), shell 2 the singly occupied 3σ orbital, and shell 3 the degenerate 1π orbital. Calculations on the corresponding $^2\Delta$ state would require the directives

```
MULT 2
```

and

```
OPEN 1 1 2 2 DELTA
```

Table 1: Energy Expressions Available and the OPEN Directive

Configuration	State	Data Specification		
		OPEN	STATE	MULT
δ^1	$^2\Delta$	2 1		2
π^1	$^2\Pi$			
e^1	^2E			
δ^2	$^3\Sigma^-$	2 2		3
π^2	$^3\Sigma^-$			
e^2	^3A			
δ^2	$^1\Gamma$	2 2	GAMMA	1
π^2	$^1\Delta$	2 2	DELTA	1
δ^2	$^1\Sigma^+$	2 2	SIGP	1
π^2	$^1\Sigma^+$		SIGP	
e^2	^1A		A	
δ^3	$^2\Delta$	2 3		2
π^3	$^2\Pi$			
e^3	^2E			
p^1	^2P	3 1		2
t^1	^2T			
p^2	^3P	3 2		3
t^2	^3T			
p^2	^1D	3 2	D	1
p^2	^1S	3 2	S	1
t^2	^1A	3 2	A	1
p^3	^4S	3 3		4
t^3	^4A	3 3		4
p^3	^2D	3 3	D	2
t^3	^2P	3 3	P	2
p^4	^3P	3 4		3
t^4	^3T	3 4		3

Configuration	State	Data Specification					
		OPEN				STATE	MULT
p^4	^1D	3	4			D	1
p^4	^1S	3	4			S	1
t^4	^1A	3	4			A	1
p^5	^2P	3	5				2
t^5	^2T	3	5				2
$\sigma^1\pi^1$	$^1\Pi$	1	1	2	1		1
$\sigma^1\pi^1$	$^3\Pi$	1	1	2	1		3
$\sigma^1\pi^2$	$^4\Sigma^-$	1	1	2	2		4
$\sigma^1\pi^2$	$^2\Sigma^+$	1	1	2	2	SIGP	2
$\sigma^1\pi^2$	$^2\Sigma^-$	1	1	2	2	SIGM	2
$\sigma^1\pi^2$	$^2\Delta$	1	1	2	2	DELTA	2
$\sigma^1\pi^3$	$^1\Pi$	1	1	2	3		1
$\sigma^1\pi^3$	$^3\Pi$	1	1	2	3		3
$\pi^1\pi^1$	$^3\Sigma^+$	2	1	2	1	SIGP	3
$\pi^1\pi^1$	$^3\Sigma^-$	2	1	2	1	SIGM	3
$\pi^1\pi^1$	$^3\Delta$	2	1	2	1	DELTA	3
$\pi^1\pi^1$	$^1\Sigma^+$	2	1	2	1	SIGP	1
$\pi^1\pi^1$	$^1\Sigma^-$	2	1	2	1	SIGM	1
$\pi^1\pi^1$	$^1\Delta$	2	1	2	1	DELTA	1
$\pi^3\pi^1$	$^3\Sigma^+$	2	3	2	1	SIGP	3
$\pi^3\pi^1$	$^3\Sigma^-$	2	3	2	1	SIGM	3
$\pi^3\pi^1$	$^3\Delta$	2	3	2	1	DELTA	3
$\pi^3\pi^1$	$^1\Sigma^+$	2	3	2	1	SIGP	1
$\pi^3\pi^1$	$^1\Sigma^-$	2	3	2	1	SIGM	1
$\pi^3\pi^1$	$^1\Delta$	2	3	2	1	DELTA	1
$\pi^3\pi^3$	$^3\Sigma^+$	2	3	2	3	SIGP	3
$\pi^3\pi^3$	$^3\Sigma^-$	2	3	2	3	SIGM	3
$\pi^3\pi^3$	$^3\Delta$	2	3	2	3	DELTA	3
$\pi^3\pi^3$	$^1\Sigma^+$	2	3	2	3	SIGP	1
$\pi^3\pi^3$	$^1\Sigma^-$	2	3	2	3	SIGM	1
$\pi^3\pi^3$	$^1\Delta$	2	3	2	3	DELTA	1

4.2 GVB Wavefunctions

While there are no additional directives required in characterising a GVB wavefunction, we elaborate below on certain features of such calculations, presenting a somewhat more complex example than that given previously (section 2.6). In particular, we consider the impact of using a set of localised orbitals (see Part 8) to initiate the GVB calculation.

The following points on performing GVB pair calculations should again be noted.

1. Remember that in the general case of a GVB [2] calculation on an open-shell system, comprising $\mathbf{m}$ doubly-occupied orbitals, $\mathbf{n}$ open shell orbitals and $\mathbf{2p}$ GVB orbitals (that is p GVB-pairs) the program expects the trial vectors to be organised thus:

```
orbitals   1 ->  m          doubly occupied

        m+1 ->  m+n        open shell orbitals

      m+n+1 ->  m+n+2      the first GVB pair, with the strongly
                       occupied MO preceding the weakly
                       occupied MO
             .
             .
             .

   m+n+2p-1 -> m+n+2p      the component orbitals of the
                      p-th GVB pair
```

   It is the users responsibly to ensure, through use of the SWAP directive, that the input orbitals are so arranged.

2. In default GAMESS–UK will automatically, based on the z-matrix geometry specification, deduce the molecular point group and hence generate and retain only the unique integrals required in the process of constructing a 'skeletonised' Fock matrix [9]. Such a symmetry-truncated integral list is, however, *NOT* usable in pair-GVB calculations, and *considerable caution* should be exercised when considering use of an integral file generated in an earlier SCF run directly in a subsequent pair-GVB calculation under control of the BYPASS directive. The way to proceed in such cases has been outlined in section 2.6.

Assume we wish to perform a 4-pair GVB/PP calculation on H_2CO, treating the C-H bonds and two C-O orbitals within the perfect pairing approximation. In such cases little progress is possible using the set of SCF MOs directly in the GVB; it will be necessary to generate a trial set of GVB vectors based on a localised orbital set, using the VECTORS option NOGEN to generate the set of secondary GVB pair orbitals based on the set of LMO. Thus the sequence of calculations required is

- perform the closed shell SCF calculation;
- localise the set of valence SCF orbitals;

- perform the GVB calculation using the set of LMO input under control of the VECTORS option NOGEN.

First, we show the data sequences for performing the initial closed shell calculation, and the subsequent Boys localisation in which the LOCAL directive is used to exclude the inner shell and virtual orbitals from the unitary transformation. Note the SUPER OFF NOSYM specification in the closed-shell calculation, enabling subsequent use of the BYPASS directive in the GVB calculation itself.

Closed-shell SCF

```
TITLE
H2CO - 3-21G  CLOSED SHELL SCF  -
SUPER OFF NOSYM
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
ENTER
```

Localised Orbital Analysis

```
RESTART NEW
TITLE
H2CO - 3-21G DEFAULT BASIS - VALENCE LMOS
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
RUNTYPE ANALYSE
LOCAL
3 TO 8 END
ENTER 2
```

An examination of the closed-shell SCF-MOs and localised MOs reveals the ordering shown below:

MO Sequence Number	Symmetry	LMO Sequence Number	Type
1	1a$_1$	1	1a$_1$
2	2a$_1$	2	2a$_1$
3	3a$_1$	3	C-O$_1$
4	4a$_1$	4	C-H$_1$
5	1b$_2$	5	C-H$_2$
6	5a$_1$	6	lp$_1$O
7	1b$_1$	7	C-O$_1$
8	2b$_2$	8	lp$_2$O

The following points should be noted regarding the data sequence shown below for performing the GVB/PP calculation:

- Note the form of the SCFTYPE directive - the integer specified after the GVB keyword indicates the number of GVB pairs - in the present case, just 4.
- Taking the localised orbitals as the starting point, the trial vectors for the GVB calculation are obtained through use of the NOGEN directive, where the specified integer identifies the Dumpfile section containing the input set of localised orbitals.
- NOGEN assumes that the orbitals to be correlated are positioned at the *top* of the doubly occupied orbital input set (see section 4.8.1). As only a subset of four of the LMOs are to be correlated, we must ensure that the corresponding orbitals occupy the top four positions in the occupied set (with sequence numbers 5-8), using the SWAP directive to reposition the two oxygen lone pair LMOs below the two C-O and C-H bonds (i.e., with sequence numbers 3 and 4).
- Note the presence of the ADAPT OFF data line. This is now required when using the set of orbitals from the LMO analysis, where the process of symmetry adaptation is automatically suppressed.

GVB/PP Data

```
RESTART NEW
TITLE
H2CO - 3-21G  4PAIR GVB
BYPASS
SUPER OFF NOSYM
ADAPT OFF
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
SCFTYPE GVB 4
VECTORS NOGEN 2
SWAP
3 6\4 8\END
ENTER 3 4
```

4.3 CASSCF wavefunctions: The CONFIG Directive

CONFIG

The CONFIG directive must be presented in a CASSCF calculation, and acts

1. to define the active space in the calculation, by partitioning the orbitals into a primary and secondary set [3].

2. to specify an initial reference configuration (for example the associated SCF configuration) that will be employed in generating the complete CI space and the associated loop-formulae tape.

CONFIG data comprises a sequence of orbital classification lines in which each orbital in the primary space is classified by type, with the following orbital TAGs used in this classification:

- FZC - frozen core orbital i.e. an orbital which will remain doubly occupied in all configurations.
- DOC - doubly occupied i.e. an orbital which is doubly occupied in the reference configuration, and which will be permitted variable occupancy in the CASSCF treatment.
- ALP - an unpaired orbital i.e. an orbital which is singly occupied in the reference configuration, and which will be permitted variable occupancy in the MCSCF treatment.
- AOS, BOS - those singly occupied orbitals in the reference configuration belonging to non-identical singlet-coupled pairs. Again such orbitals will be permitted variable occupancy.
- UOC - formally unoccupied orbitals, corresponding to SCF virtual MOs, which will be permitted variable occupancy in the MCSCF.

Each orbital definition line comprises an orbital TAG in the first data field, followed by the sequence numbers of the input orbitals (as restored under control of the VECTORS directive) of the nominated type.

Example 1

Let us consider initially various calculations on the water molecule to illustrate CONFIG specification, and assume the following set of input MOs, derived from a closed-shell SCF calculation.

MO Sequence number	Symmetry	SCF Occupation	MO Sequence number	Symmetry	SCF Occupation
1	$1a_1$	2.0	6	$4a_1$	0.0
2	$2a_1$	2.0	7	$2b_2$	0.0
3	$1b_2$	2.0	8	$5a_1$	0.0
4	$3a_1$	2.0	9	$2b_1$	0.0
5	$1b_1$	2.0			

To perform a full-valence space calculation, comprising 7 primary orbitals, with the $4a_1$ and $2b_2$ virtual MOs included in the active space, would require the following data lines:

```
CONFIG
DOC 1 TO 5
UOC 6 7
END
```

To freeze the O1s orbital, we would present the sequence

```
CONFIG
FZC 1
DOC 2 TO 5
UOC 6 7
END
```

The following sequence would be used to extend the space to include the 5a$_1$ and 2b$_1$ orbitals:

```
CONFIG
FZC 1
DOC 2 TO 5
UOC 6 TO 9
END
```

Note that all orbitals in the active space must appear first in the input orbital set. Thus if we wished to include the 2b$_1$ MO, but not the 5a$_1$, we would present the sequence:

```
CONFIG
FZC 1
DOC 2 TO 5
UOC 6 TO 8
END
```

in conjunction with the SWAP data lines

```
SWAP
8 9
END
```

Assuming the same orbital ordering had been derived from an RHF calculation on the X$^2B_1$ state of the H_2O^+ ion, then a full-valence calculation, with frozen O1s, would be performed under control of the following CONFIG data:

```
CONFIG
FZC 1
DOC 2 TO 4
ALP 5
UOC 6 7
END
```

Example 2

Assume the following set of input MOs from an RHF calculation on the X$^3B_1$ state of methylene:

MO Sequence number	Symmetry	SCF Occupation	MO Sequence number	Symmetry	SCF Occupation
1	1a$_1$	2.0	6	4a$_1$	0.0
2	2a$_1$	2.0	7	2b$_2$	0.0
3	1b$_2$	2.0			
4	3a$_1$	1.0			
5	1b$_1$	1.0			

A full-valence space calculation on the X$^3B_1$ state, with frozen C1s, would be controlled thus:

```
CONFIG
FZC 1
DOC 2 3
ALP 4 5
UOC 6 7
END
```

The corresponding calculation on the $^1B_1$ state, featuring singlet coupling of the 3a$_1$ and 1b$_1$ MOs, would require use of the AOS and BOS tags, thus

```
CONFIG
FZC 1
DOC 2 3
AOS 4
BOS 5
UOC 6 7
END
```

remembering to change (or remove) the MULT 3 specification!

Note

Additional keywords may be specified on the directive initiator line, as follows:

- **PRINT**: to obtain a complete list of the CASSCF configurations, characterised by occupation pattern.
- **BYPASS**: at the outset of a CASSCF calculation the loop-formulae tape (written in default to ED9) must be generated. The BYPASS keyword allows this step to be bypassed in a subsequent restart job, assuming of course that the data set had been retained between jobs (note that the PRINT option is not effective in BYPASS mode).
- **NOSORT**: if the 1-step Newton Raphson optimisation technique is to be used during CASSCF iteration (under control of the SIMUL directive, see 5.4.2), then the loop formulae tape is reordered in the interests of efficiency. This reordered file is written in default to ED10, with the reordering process carried out automatically unless suppressed by presenting the NOSORT keyword on the CONFIG data line.

Example

Assuming that simultaneous optimisation is not to be performed, and that a list of configurations is required, then the following data line

```
CONFIG PRINT NOSORT
```

should be presented in the startup job, and the line

```
CONFIG BYPASS NOSORT
```

in a subsequent restart, assuming that ED9 had been 'kept'. The user is recommended to take advantage of the BYPASS and NOSORT options where applicable, since the space requirements and generation time of the loop formulae tape may often prove costly.

5 Directives Controlling Wavefunction Convergence

5.1 MAXCYC

This directive consists of a single data line, read to variables TEXT,MAXC using format (A,I).

- TEXT should be set to the character string MAXCYC.
- MAXC is an integer used to specify the maximum number of iteration cycles required.

The directive may be omitted when MAXC will be set to the default value of 50. The following conditions cause termination of iteration:

- When the desired accuracy is reached, iteration of the SCF or MCSCF process stops.
- If the job time remaining is insufficient to complete another iterative cycle, or the maximum number of cycles as set by the MAXCYC directive (or default) has completed, iteration will cease.

Example

```
MAXCYC 100
```

Note that the maximum number of allowed cycles in CASSCF calculations is limited to 20.

5.2 THRESHOLD

This directive may be used to define a convergence threshold for SCF and MCSCF iterations, and comprises a single data line read to the variables TEXT, ISET using format (A,I);

- TEXT should be set to the character string THRESH
- ISET is an integer parameter used in defining the threshold.

At SCF convergence, the elements of the density matrix will be converged to within an absolute error 10^{-ISET}. The directive may be omitted, when the default value 10^{-5} will be used. For CASSCF iterations values of the maximum Brillouin element (super-CI) or maximum first derivative (Newton Raphson) control the monitoring of convergence. Note that in most CASSCF and MCSCF calculations setting ISET=4 will prove quite adequate.

6 SCF Convergence - Default Driver

In default the RHF, UHF and GVB modules of GAMESS–UK iterate under control of a hybrid scheme of level shifting [10] and DIIS (Direct inversion in the Iterated Subspace [1]). A set of built-in level shifters will, when used in conjunction with DIIS, lead in most cases to adequate convergence, and the user need only consider providing data in troublesome cases.

A typical SCF calculation will, when far from convergence, proceed under control of level shifting alone, and it is at this point of the calculation that overriding the default shifters (under control of the LEVEL directive) may be necessary. Once convergence has 'set-in', the DIIS procedure is initiated (corresponding to a TESTER of ca. 0.1); experience to date suggests that rapid convergence proceeds once this point has been reached.

6.1 LEVEL

In its most general form the LEVEL directive may be used to nominate two sets of level shifters to apply during an SCF calculation, the first set to be used up to and including some user-nominated iteration, the second to apply after this point. Note that the number and role of the level shifters within each set is a function of SCFTYPE. Note also that the primary role of level shifting is to ensure the calculation 'arrives' in the quadratic region of convergence, from which point DIIS will effectively control the SCF iterations. This on occasions requires setting higher values than those used, say, in a level-shifting only environment, where larger values would act to slow down convergence in the latter stages of the computation. We describe below the format of the directive for each SCFTYPE.

6.1.1 Closed-shell RHF Calculations

In the case of a closed-shell RHF calculation the LEVEL directive consists of a single data line read to variables TEXT, E1, IBRK, E2 using format (A,F,I,F).

- TEXT should be set to the character string LEVEL.
- E1 is the level shifter up to iterative cycle specified by IBRK.
- IBRK is an integer used to a specify the cycle number.

- E2 is the level shifter after the iterative cycle given by IBRK.

An alternative form of LEVEL is permitted, consisting of only two parameters, read to variables TEXT, E1 using format (A,F). If used, this sets the IBRK parameter to the default 999, E2 will be given the value of 0.0, while E1 has its usual meaning. The LEVEL directive may be omitted, when the program will assign the following default settings for most molecular systems:

```
E1=1.0 , E2=0.3 and IBRK=5
```

SCF calculations on transition metal complexes comprising first row metals are typically found to require higher values of level shifter for satisfactory convergence. In such cases the above defaults are now modified by the code to the following:

```
E1=2.0 , E2=2.0 and IBRK=999
```

The following points should be noted on the specification of level shifters:

- For the first three cycles, a value of at least unity should be chosen, to stabilise what are usually the most erratic cycles.
- If divergence is experienced, with the DIIS procedure never instigated, increasing the level shifter will force convergence to the onset of DIIS. Increasing the level shifters through a data line of the form data line

  ```
  LEVEL 2.0
  ```

  is usually sufficient to force convergence to this onset.

6.1.2 Open-shell RHF and GVB Calculations

In the case of open-shell RHF and GVB calculations the LEVEL directive consists of a single data line read to variables TEXT, OCC1, V1, IBRK, OCC2, V2 using format (A,2F,I,2F).

- TEXT should be set to the character string LEVEL.
- OCC1, V1 are the doubly-occupied–partially-occupied and occupied-virtual level shifters up to the iterative cycle specified by IBRK.
- IBRK is an integer used to a specify the cycle number.
- OCC2, V2 are the doubly-occupied–partially-occupied and occupied-virtual level shifters after the iterative cycle specified by IBRK.

An alternative form of LEVEL is permitted, consisting of only three parameters, read to variables TEXT, OCC1, V1 using format (A,2F). In this form TEXT, OCC1 and V1 have their usual meanings, whilst the program will set IBRK=999 and OCC2, V2 to zero. The LEVEL directive may be omitted, when the program will assign the following default settings:

```
OCC1=0.05 , V1=1.0 ,  IBRK=5
OCC2=0.01 , V2=0.5
```

The following points should be noted on using level shifters to proceed to the onset of DIIS:

- Note that the value of the doubly-occupied–partially-occupied level shifter is typically far smaller than that involving the virtual interaction;
- It can be shown that convergence to a stationary point on the energy surface can be guaranteed if 'sufficiently' large and positive level shifters are used. Thus if divergence is experienced the user may repeat the job but with increased level shifters. The data line
- As with closed shell systems, the default shifters above are doubled in value for systems containing first row transition metal atom(s).

```
LEVEL 0.3 1.5
```

is usually sufficient to force convergence to the onset of DIIS.

6.1.3 Open-shell UHF Calculations

In the case of an open-shell UHF calculation the LEVEL directive consists of a single data line read to variables TEXT, EA1, EB1, IBRK, EA2, EB2 using format (A,2F,I,2F).

- TEXT should be set to the character string LEVEL.
- EA1, EB1 are the α-spin orbital and β-spin orbital level shifters up to the iterative cycle specified by IBRK.
- IBRK is an integer used to specify the cycle number.
- EA2, EB2 are the α-spin and β-spin level shifters after the iterative cycle specified by IBRK.

An alternative form of LEVEL is permitted, consisting of only three parameters, read to variables TEXT, EA1, EB1 using format (A,2F). In this form TEXT, EA1 and EB1 have their usual meanings, whilst the program will set IBRK=999 and EA2, EB2 to zero. The LEVEL directive may be omitted, when the program will assign the following default settings:

```
EA1=1.0, EB1=1.0 ,  IBRK=5
EA2=0.3, EB2=0.3
```

These defaults are doubled in value for systems containing first-row transition metals atoms, thus:

```
EA1=2.0, EB1=2.0 ,  IBRK=999
EA2=2.0, EB2=2.0
```

6.2 Core-Hole States

The present implementation of LEVEL within the open-shell RHF program is such that core-hole states may be converged with an appropriate setting of the parameters of the LEVEL directive. Specifically, such states may be studied by presenting the data line

```
LEVEL 0.0 1.0
```

where the doubly-occupied–partially occupied level shifter is set to zero. It is assumed in such studies that the open-shell orbital comprises the core orbital involved in the ionization process.

Example

The following two data files may be used to optimise the geometry of the 1s core-hole state of the water molecule. The first file generates the closed-shell MOs, the second performs the hole-state calculation, with the SWAP directive placing the oxygen 1s MO as the singly occupied orbital.

The Closed-shell SCF

```
TITLE
H2O .. 3/21G
ZMAT ANGSTROM
O
H 1 ROH
H 1 ROH 2 THETA
VARIABLES
ROH 0.956   HESSIAN 0.7
THETA 104.5  HESSIAN 0.2
END
ENTER
```

The Core-Hole state open-shell SCF

```
RESTORE NEW
TITLE
H2O +  1S-CORE HOLE STATE
CHARGE 1
MULT 2
ZMAT ANGSTROM
O
H 1 ROH
H 1 ROH 2 THETA
VARIABLES
ROH 0.956   HESSIAN 0.7
THETA 104.5  HESSIAN 0.2
END
SWAP
1 5
END
```

```
RUNTYPE OPTIMIZE
LEVEL 0.0 1.0
XTOL 0.003
ENTER
```

6.3 DIIS

The DIIS directive consists of a single data line containing the character string DIIS in the first data field. Subsequent data fields may be used

- to suppress DIIS, by presenting a data line of the form

  ```
  DIIS OFF
  ```

  when in default only level shifting will be in effect;

- to request printing of the DIIS equations by specifying the character string PRINT;

- to modify the onset of DIIS by specifying the value of TESTER: this is achieved by typing the character string ONSET, immediately followed by a floating point variable defining the required value of TESTER. Thus the data line

  ```
  DIIS ONSET 0.2
  ```

  would override the default onset of 0.1.

- to route information necessary for SCF cycling to proceed in uninterrupted fashion across restart jobs. Such a process is important, for example, in large direct-SCF calculations when perhaps only a single SCF cycle may be possible in a given step. The successful functioning of DIIS relies on the process controlling the SCF over multiple iterations, hence the need arises to save DIIS information. This is achieved by presenting the LFNAME of a direct-access file, and starting block number to which DIIS information will be written. Thus presenting the data line

  ```
  DIIS ED4 1
  ```

  will result in the DIIS information being written to ED4 commencing at block 1.

Example

The following two data files illustrate this saving of DIIS information between separate jobs. Assuming the file allocated to ED4 is saved in the Startup job below, which terminated during SCF processing, and is subsequently allocated to the Restart job, the SCF cycling will be identical to that observed if the first job had been run to completion.

Startup Job

```
TITLE
C6H5.NO2  TZVP DIIS INFORMATION TO ED4
NOPRINT
ZMAT ANGSTROM
C
N 1 RCN
X 2 1.0 1 90.0
C 1 RCC1 2 T1 3 P1
C 1 RCC1 2 T1 3 -P1
C 4 RCC2 1 T2 2 P2
C 5 RCC2 1 T2 2 P2
C 7 RCC3 5 T3 1 P3
O 2 RNO1 1 T5 3 -90.0
O 2 RNO1 1 T5 3  90.0
H 4 RCH1 1 T6 2 P5
H 5 RCH1 1 T6 2 P5
H 6 RCH2 4 T7 11 P6
H 7 RCH2 5 T7 12 P6
H 8 RCH3 7 T8 14 P7
VARIABLES
RCN 1.49\RCC1 1.37\RCC2 1.43\RCC3 1.37
RNO1 1.21\RCH1 1.084\RCH2 1.084\RCH3 1.084
T1 120.0\T2 120.0\ T3 120.0\T5 120.0\T6 120.0
T7 120.0\T8 120.0\P1 90.0\P2 180.0\P3 0.0
P5 0.0\P6 0.0\P7 0.0\END
BASIS TZVP
SCFTYPE DIRECT RHF
DIIS ED4 1
ENTER
```

Restart Job

```
RESTART SCF
TITLE
C6H5.NO2  TZVP DIIS INFORMATION TO ED4
NOPRINT
ZMAT ANGSTROM
C
N 1 RCN
X 2 1.0 1 90.0
C 1 RCC1 2 T1 3 P1
C 1 RCC1 2 T1 3 -P1
C 4 RCC2 1 T2 2 P2
C 5 RCC2 1 T2 2 P2
C 7 RCC3 5 T3 1 P3
O 2 RNO1 1 T5 3 -90.0
O 2 RNO1 1 T5 3  90.0
H 4 RCH1 1 T6 2 P5
H 5 RCH1 1 T6 2 P5
H 6 RCH2 4 T7 11 P6
H 7 RCH2 5 T7 12 P6
H 8 RCH3 7 T8 14 P7
VARIABLES
RCN 1.49\RCC1 1.37\RCC2 1.43\RCC3 1.37
RNO1 1.21\RCH1 1.084\RCH2 1.084\RCH3 1.084
```

```
T1 120.0\T2 120.0\ T3 120.0\T5 120.0\T6 120.0
T7 120.0\T8 120.0\P1 90.0\P2 180.0\P3 0.0
P5 0.0\P6 0.0\P7 0.0\END
BASIS TZVP
SCFTYPE DIRECT RHF
DIIS ED4 1
ENTER
```

6.4 CONV

This directive may be used to override the default convergence techniques of level shifting and DIIS, and consists of a single data line read to the variables TEXT,INDEX using format(A,I).

- TEXT should be set to the character string CONV
- INDEX is an integer used to specify the particular technique(s) required. The following options are available:

```
 0........use Pople's extrapolation
 1........do NOT use damping, extrapolation or level shifting
 2........use damping and Pople's extrapolation
 3........use Davidson's damping
 4........use level shifting and extrapolation
 5........use level shifting (default)
 6........use damping, extrapolation and level shifting
 7........use level shifting and damping
10........use damping, extrapolation and restrict orbital
          mixing in GVB or open shell calculations.
11........use damping and restrict orbital mixing
14........use damping, extrapolation, level shifting and
           restrict orbital mixing
15........use level shifting, damping and restrict orbital
          mixing
```

6.5 AVERAGE

AVERAGE [ON | OFF | <tolerance>]

In SCF calculations convergence problems may arise if the molecule has a partly occupied set of degenerate orbitals. A way of dealing with this situation is to occupy all degenerate orbitals equally, using fractional occupations if neseccary. This equals building the density from an average of a number of states. The AVERAGE directive controls this procedure which is "on" by default. The options "on" and "off" do the obvious, alternatively the tolerance for detecting degenerate orbitals can be specified. All orbitals in the energy range of the HOMO energy plus/minus the tolerance will be included in the degenerate set. By default the tolerance is 10^{-4}.

6.6 SMEAR

The SMEAR directive implements Fermi smearing [11] for filling up the molecular orbitals. Normally, orbitals are either fully occupied or empty. Fermi smearing allows orbitals to be fractionally filled, according to a step function, that depends on the "Fermi temperature" employed.

Fermi smearing can be useful in certain problematic convergence cases where degenerecies in the orbital energies mean that it is uncertain which state the SCF should converge on. Smearing may alleviate the problem by using an average state thereby removing the need to make a discrete choice.

The format of the directive is:

```
SMEAR <START_TEMP> [<FINAL_TEMP>] [<UNITS>] [SCALE <SCALE_VALUE>]
```

where:

- START_TEMP (F) is the starting Fermi Temperature
- FINAL_TEMP (F) is the final Fermi Temperature
- UNITS (A) specifies the units to be used for the final and starting tempratures. By default the units are Hartrees, but setting UNITS to EV changes the units to Electron Volts.
- SCALE (A) is the keyword SCALE followed by the scale factor SCALE_VALUE (F).

The scale factor is used in the temperature updates going from the start to the final temperature. To ensure that the final temperature has been reached at convergence the current temperature is updated as a linear function of the difference between the SCF tester and the SCF convergence criterion

$$T_0 = T_{start} \quad (2)$$

$$T_{i+1} = \max(T_{final}, \min(T_i, SCALE\_VALUE * (tester - convergence)) \quad (3)$$

A couple of points should be noted about the use of Fermi-smearing:

1. The Fermi-Dirac smearing enforces strict Aufbau ordering of the orbitals through the occupations. Thus it cannot be used together with options that may break the Aufbau ordering such as locking.
2. The Fermi-Dirac smearing breaks the strict distinction between occupied and virtual orbitals. In practice there will be three categories occupied, partially occupied and virtual orbitals. This requires a modified definition of the tester. The tester now becomes the absolute maximum off-diagonal value of the Fock matrix excluding the occupied-occupied and virtual-virtual blocks.

Finally with the "IPRINT SMEAR" directive more detailed information about the Fermi smearing can be obtained, such as the actual orbital occupations used in every iteration, and the chemical potential in Hartree.

6.7 CASSCF Convergence

Four directives are available for specifying the optimisation techniques to be used in the course of a CASSCF calculation. The user nominates which techniques are to be employed on each iterative cycle of the computation through use of the SUPERCI, NEWTON, HESSIAN and SIMUL directives. Thus the data sequence

```
SUPERCI 1 TO 4
NEWTON  5 TO 20
HESSIAN 5 9 13 17
```

would specify super-CI optimisation for the first four cycles, followed by subsequent 2-step Newton-Raphson (NR) for the remainder of the computation, with explicit construction of the orbital Hessian on cycles 5, 9, 13 and 17. The above sequence corresponds in fact to the default specification, and will be used in the absence of controlling directives. The following points should be noted:

1. The optimum technique on a given cycle is very much dependent on the current degree of convergence. At the outset of the calculation starting from, say, an SCF orbital set, the user is strongly recommended to use the first-order super-CI option, and to continue with this until the degree of convergence (monitored by the maximum Brillouin element) suggests switching to the 'pseudo second-order' NR technique. Instigating NR too rapidly however will lead to divergence, with the maximum first derivative increasing, leading finally to the error message 'Hessian diagonalisation has failed to converge'. Experience suggests that a maximum Brillouin element of 0.05-0.01 represents the optimum point at which to instigate NR control, with between 5 - 10 cycles of super-CI normally required to reach this point.

2. The sequence of directives specified is dynamic, and is not remembered between separate runs of the program. Thus while the data sequence

   ```
   SUPERCI 1 TO 7
   NEWTON  8 TO 20
   HESSIAN 8 to 20
   ```

   may be presented in a startup job, the user should modify this sequence in any restart job to reflect the current degree of convergence. Thus if the startup job dumped, say, on cycle 11 with the maximum first derivative suggesting satisfactory convergence of the NR process, the data sequence

   ```
   NEWTON 1 TO 20
   HESSIAN 1 TO 20
   ```

   should be presented in the restart job.

3. Within a given run of the program the sequence is remembered, and will be applied, for example, in each separate CASSCF calculation of a geometry optimisation. Some caution should be exercised at the outset of such an optimisation, with the onset of the NR method specified to allow for possible large changes in wavefunction associated with large steps in the geometry optimisation.

4. Hessian construction must be performed on the first NR cycle.

5. The program currently requires 'in-core' treatment of the orbital hessian, and as such NR usage is limited to cases with a rather modest number of orbital rotation parameters. If memory requirements preclude such a treatment, the user should present the data sequence

   ```
   SUPERCI 1 TO 20
   THRESH 4
   ```

   which will lead to satisfactory convergence in many cases.

6. In many instances the most rapid convergence is realised using the 1-step NR method, featuring simultaneous optimisation of both CI coefficients and orbital rotation parameters. This technique may be requested under control of the SIMUL directive, with a typical data sequence shown below:

   ```
   SUPERCI 1 TO 5
   NEWTON 6 TO 20
   HESSIAN 6 TO 20
   SIMUL 8 TO 20
   ```

   The memory requirements are, of course, aggravated by the inclusion of the CI terms in the Hessian. When applicable this technique is highly effective in stable geometry-optimisations.

7 SCF Convergence - Alternate Driver

A new SCF driver has has been developed within GAMESS-UK that provides an alternative way of controlling SCF calculations. The driver is orthogonal to the default driver and is currently still in the "development" stage, and is therefore not included in the default build of GAMESS-UK. Users who have access to the source code and wish to include the driver in their build, should add the:

```
newscf_f90
```

keyword as an option when configuring the code.

There were a number of motives for developing the new driver. These included:

- to allow more flexible control of convergence schemes for cases that were proving difficult to converge with the default driver.
- to reduce IO activity to the dumpfile/scratchfile by holding more structures in memory, thereby reducing a bottleneck for parallel calculations.

The consequence of more structures being held in memory has enabled the development of a parallel version of the driver in which these data structures can be distributed across all the nodes of a parallel machine, allowing the code to take advantage of the large aggregate memory on these machines (see the parallel ScaLAPACK version described in chapter 14). However, a consequence of this is that the code is rather memory-hungry when run in serial or on small processor counts.

As the driver is still relatively new, it does not have support for all of the features within GAMESS-UK (such as ZORA, DRF, etc.). As of the writing of the manual, the driver supports:

- RHF and UHF (but not ROHF) SCF Calculations (both direct and conventional)
- RHF and UHF (but not ROHF) DFT Calculations (both direct and conventional)

7.1 Input Control

Use of the module is requested by including a block of directives of the following form:

```
newscf
<control directives>
end
```

The <control directives> block can be empty (in which case a default convergence scheme will be used), or it can contain the control directives described in the following sections.

An input file with a set of example control directives is provided at the end of this section for readers who wish to gain an overview of how things are structured before before becoming bogged down in the details of each directive.

7.2 Overall Control Flags

7.2.1 Controlling Printed Output - PRINT

This directive controls which data from the calculation will be included in the output. The format of the directive is a single line containing the keyword PRINT (A), followed by any combination of the following parameters to the PRINT command (all in A format).

FOCK	requests printing of the Fock matrix at SCF convergence
GUESS	Print out the trail vectors
DENSITY	Print out the density matrix
VECTORS	Print out the eigenvectors
FULL	Print everything (WARNING: this may produce a lot of output!)
FRONTIER	Print the frontier orbitals
DIIS	Monitoring of the solution of the DIIS equations
TIME	Print the cycle and total wallclock and cpu times each cycle

So, for example, the line:

```
PRINT FOCK
```

would cause the entire Fock matrix to be printed on SCF convergence.

7.2.2 SCF exit status - SOFTFAIL

This directive consists of the the single character string SOFTFAIL (A), and will prevent the code from generating an error if MAXCYC SCF cycles are reached without convergence.

7.3 Controlling the convergence of a calculation

Convergence is controlled by providing parameters for a series of "phases", each becoming active depending on particular criteria and employing various convergence controls.

If convergence control directives are omitted, a default scheme is adopted, otherwise the user will need to provide the control information for a number of phases, including the criteria that are used to switch from one phase to another.

The control information is provided in a "phase block", which is started by a line with two data fields PHASE (A) and IPHASE (I).

- PHASE is set to the character string PHASE
- IPHASE is an integer (greater than zero) identifying the phase

A phase block is terminated by a subsequent "PHASE" directive, or an "END" directive, indicating the end of the newscf control directives.

Within a phase block, the criteria for when to jump to a subsequent phase is determined by the "NEXT" directive.

An individual phase block is arranged as shown below:

```
PHASE <N>
<convergence controls>
NEXT <M>
<convergence criteria>
```

where <N> is an integer denoting the phase in questions and <M> an integer denoting the next phase to jump to.

Within a phase block, the user can determine which convergence controls are to be used, under what conditions to switch to another phase block and when the SCF will be deemed to have converged.

7.3.1 CONVERGENCE CONTROLS

The convergence controls available (together with the default values) are:

7.3.2 Level Shifters - LEVEL

The format of the LEVEL directive is the same as that for the Default SCF driver described in section 6.1.

The default value for the NEWSCF driver is 0.5

7.3.3 DIIS

The single keyword DIIS in format (A) indicates that DIIS should be active for this phase.

7.3.4 NEWDIIS

This directive consists of the single keyword NEWDIIS in format (A), and resets the DIIS space for the phase, as explained below.

DIIS works by taking a linear combination of the previous Fock matricies to determine the subsequent set of vectors. DIIS therefore has a "memory" of the previous Fock matrices that it uses to generate the next solution. The NEWDIIS keyword causes the memory from subsequent phases to be wiped and for DIIS to start afresh.

NB: DIIS requires a "memory" of at least 3 cycles to be able to function, so resetting DIIS means that it will only become active again on the 4th cycle of the phase.

7.3.5 EXTRAPOLATION

The format of this directive is a line with three data fields, the first being the character EXTRAP (A) followed by the two values TEST (F) and COEF (F). The TEST and COEF variables are used as explained below.

When two successive SCF steps satisfy the colinearity criteria

$$\frac{d1.d2}{|d1| \times |d2|} > TEST \tag{4}$$

where

$$d1 = f_1 - f_{-2} \tag{5}$$

and

$$d2 = f - f_{-1} \tag{6}$$

and f, f_{-1} and f_{-2} are the current, previous and next previous fock matrices, the extrapolation.

$$f += COEF \times d2 \tag{7}$$

will be applied.

Typically TEST should be around 0.95 and COEF around 1. In addition, there is also a test on $|d1|/|d2|$, so that if successive steps are in the same direction but of very different lengths extrapolation is suppressed.

The line below demonstrates the use of the directive with the recomended values:

```
EXTRAP 0.95  1
```

7.3.6 NEWEXT

This directive consists of the single keyword NEWEXT in format (A), and resets the extrapolation counter to zero so that extrapolation will be inactive for the first cycle.

7.3.7 RESTORE

This directive consists of the single keyword RESTORE in format (A).

The restore directive recovers the best set of vectors attained during the calculation thus far at the start of the phase.

7.3.8 LOCK

This directive consists of the single keyword LOCK in format (A) and applies configurational locking to the phase.

Normally, the electrons populate the orbitals according to Hund's rules, so that they are filled up sequentially starting from the lowest energy orbital and leaving no gaps until all the electrons are placed in an orbital.

When configurational locking is applied, the overlap between the occupied orbitals in the previous SCF cycle and the current is calculated and the electrons remain in the orbital that overlaps most closely with the one they were in previously - regardless of whether there is an unoccupied orbital lower in energy.

Locking can therefore be used to maintain an excited state configuration during a calculation.

7.3.9 SMEAR

The SMEAR directive implements Fermi smearing [11] for filling up the molecular orbitals. Normally, orbitals are either fully occupied or empty. Fermi smearing allows orbitals to be fractionally filled, according to a step function, that depends on the "Fermi temperature" employed.

Fermi smearing can be useful in certain problematic convergence cases where degenerecies in the orbital energies mean that it is uncertain which state the SCF should converge on. Smearing may alleviate the problem by using an average state thereby removing the need to make a discrete choice.

The format of the directive is:

```
SMEAR <START_TEMP> [<FINAL_TEMP>] [<UNITS>] [SCALE <SCALE_VALUE>]
```

where:

- START_TEMP (F) is the starting Fermi Temperature
- FINAL_TEMP (F) is the final Fermi Temperature
- UNITS (A) specifies the units to be used for the final and starting tempratures. By default the units are Hartrees, but setting UNITS to EV changes the units to Electron Volts.
- SCALE (A) is the keyword SCALE followed by the scale factor SCALE_VALUE (F).

The scale factor is used in the temperature updates going from the start to the final temperature. To ensure that the final temperature has been reached at convergence the current temperature is updated as a linear function of the difference between the SCF tester and the SCF convergence criterion

$$
\begin{aligned}
T_0 &= T_{start} \qquad (8)\\
T_{i+1} &= \max(T_{final}, \min(T_i, SCALE\_VALUE * (tester - convergence)) \qquad (9)
\end{aligned}
$$

A couple of points should be noted about the use of Fermi-smearing:

1. The Fermi-Dirac smearing enforces strict Aufbau ordering of the orbitals through the occupations. Thus it cannot be used together with options that may break the Aufbau ordering such as locking.
2. The Fermi-Dirac smearing breaks the strict distinction between occupied and virtual orbitals. In practice there will be three categories occupied, partially occupied and virtual orbitals. This requires a modified definition of the tester. The tester now becomes the absolute maximum off-diagonal value of the Fock matrix excluding the occupied-occupied and virtual-virtual blocks.

7.4 Changing Phase

The change from one phase to another is controlled by a number blocks of commands, each starting with a line with two data fields NEXT (A) and PHASE (I).

- NEXT is set to the character string NEXT
- PHASE is an integer identifying the phase to jump to once the relevant criteria (described below) have been met.

The block is terminated either by another NEXT directive, a PHASE directive, or an END directive.

Within a NEXT block, the criteria for when to change to phase 0 (i.e. NEXT 0) is particularly important as this specifies when the calculation is determined to have converged. If no "NEXT 0" directive is specified for a phase then the calculation cannot converge from this phase and can only jump to other phases when it meets their jump criteria; converging from them if it meets their "NEXT 0" criteria. If no "NEXT 0" is specified in any phase, the calculation will run until it runs out of cycles (as specified by MAXCYC).

Multiple criteria can be specified within a "NEXT" block, and the jump will only then occur when *all* of the criteria have been met (i.e. a logical *and* test is used).

Users wishing to use an *or* test, can specify multiple next blocks for the same phase.

The criteria for changing phases are as follows:

7.4.1 TESTER

This directive consists of the keyword TESTER (A) followed by the parameters DIRECTION (A) and CRITERIA (F).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the tester is greater than or less than the specified value.
- CRITERIA should be set to the desired value of the TESTER.

The TESTER is defined as the maximum Fock matrix element in the MO basis for the occupied-virtual block. This value will tend to zero when the calculation has converged.

7.4.2 Change in TESTER - DTESTER

This directive consists of the keyword DTESTER (A) followed by the parameters DIRECTION (A) and CRITERIA (F).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the change in the tester is greater than or less than the specified value.
- CRITERIA should be set to the change in the tester within this phase.

7.4.3 Change in energy - DE

This directive consists of the keyword DE (A) followed by the parameters DIRECTION (A) and CRITERIA (F).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the change in energy is greater than or less than the specified value.
- CRITERIA should be set to the change in energy in Hartrees within this phase.

7.4.4 Absolute change in energy - DEABS

This directive consists of the keyword DEABS (A) followed by the parameters DIRECTION (A) and CRITERIA (F).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the change in energy is greater than or less than the specified value.
- CRITERIA should be set to the absolute change in energy (i.e. from the starting "Guess" energy) in Hartrees.

7.4.5 Number of cycles in this phase - NCYC

This directive consists of the keyword NCYC (A) followed by the parameters DIRECTION (A) and CRITERIA (I).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the number of cycles is greater than or less than the specified value.
- CRITERIA should be set to the number of SCF cycles within this phase.

7.4.6 Total number of SCF cycles - TOTCYC

This directive consists of the keyword TOTCYC (A) followed by the parameters DIRECTION (A) and CRITERIA (I).

- DIRECTION should be either set to the character string "ABOVE" or "BELOW" to indicate the the change should happen when the number of cycles is greater than or less than the specified value.
- CRITERIA should be set to the total number of SCF cycles within this energy calculation.

This directive is most useful when used to trigger a phase change / indicate convergence when used in conjunction with a slacker TESTER than the usual.

Often, when performing geometry optimisations, the wavefunction will not converge to the desired value of the tester at a paticular geometry, regardless of the number of SCF cycles undertaken. However, the quality of the wavefunction may still be good enough to generate an energy profile that allows the optimiser to take a step towards a more favourable geometry where the wavefunction will converge with no problems.

The TOTCYC directive therefore allows a user to make this step after the SCF has been grinding away unsuccessfully for a large number of cycles without compromising on the quality of the wavefunction for less problematic steps.

An example of its usage is below:

```
next 0
tester below 1.E-06
info Converged normally as tester < 1.E-06
next 0
totcyc above 50
tester below  1.E-05
info converged with slacker tester of 1.E-05 as totcyc > 50
```

7.4.7 Information on a phase change - INFO

This directive consists of the keyword INFO (format A) followed by a string of text, that may extend up to the end of the logical line (80 characters by default)

This directive can be used to print a string to the output file indicating why a particular phase change has been carried out, e.g.

```
info Jumping to phase 3 as tester  < 0.001
```

7.5 Example newscf input file

The example below demonstrates a newscf convergence scheme that uses the following phases and criteria for shifting between them:

Phase 1

- Level shifters of 2.0 and 2.0 (alpha and beta)
- Switch to phase 2 when tester < 0.01

Phase 2

- Level shifters of 0.5 and 0.5, DIIS and configurational locking
- Switch to phase 3 when tester < 0.002

Phase 3

- DIIS and configurational locking
- convergence when tester $<$ 5.0d-6

The input is as follows:

```
time 1000
core 3000000
#restart new
punch coor conn
punch basi vect 1 occu scfe eige
mult 2
symt 6
title
Cu-NO, DZ Cation  - new geom, new code
geometry
      1.931427      -0.080801      -0.022645  29  Cu
     -0.924018      -1.179405       2.034581  8    O
     -3.444799       0.046586       0.020112  13  Al
      7.478471       0.371579       0.192578  8    O
     -5.043143       2.660967       1.071269  8    O
     -0.902242       1.249695      -1.985263  8    O
     -5.027333      -2.570596      -1.047347  8    O
     -6.519617      -3.214017      -1.846467  1    H
     -0.954846      -2.572273       3.214984  1    H
      5.298164      -0.243155      -0.154613  7    N
     -0.874905       2.707837      -3.083889  1    H
     -6.599492       3.284123       1.755577  1    H
end
basis dz
runt scf
# Start newscf directives
newscf
print full diis frontier
#
phase 1
level 2.0 2.0
next 2
info Changing to phase 2 as tester < 0.01
tester below  0.01
#
phase 2
level 0.5 0.5
diis
lock
```

```
next 3
info Changing to phase 3 as tester < 0.002
tester below 0.002
#
phase 3
lock
diis
next 0
info Converged form phase 3 as tester < 5.0d-6
tester below 5.0d-6
end
# End newscf directives
runtype optx
scftype uhf
cdft quad high
cdft b3lyp screen
#vectors 3 4
enter 1 2
```

8 Specification of Dispersion Corrections

A problem frequently encountered with effective one-electron models is that long range correlation effects such as dispersion or Van der Waals interactions are not properly described. This leads to problems where these relatively weak dispersion forces are important such as in DNA base pair stacking or the interaction between aromatic molecules. So far attempts to rigorously address these problems have had limited success. In response to this empirical approaches to correct for the lack of dispersion have been suggested. GAMESS-UK supports two versions of one of these approaches [47, 48, 49]. Both these approaches were originally designed to be used in the context of DFT calculations but they may be useful in a wider context.

Both of the supported approaches are based on a simple model for the dispersion energy of two interacting atoms

$$D_{ij}(R_{ij}) = C_6^{ij}\frac{f_{\mathrm{damp}}(R_{ij})}{R_{ij}^6}. \tag{10}$$

The function f_{damp} is simply chosen such that it goes to zero when R_{ij} goes to zero to prevent that atoms collapse onto eachother. This damping function is of the form

$$f_{\mathrm{damp}}(R_{ij}) = \frac{1}{1+e^{-\alpha(R_{ij}/R_0-1)}} \tag{11}$$

where α is a universal exponent and R_0 is the sum of Van der Waals radii of atoms i and j.

The total dispersion energy may now be expressed as

$$E_{\mathrm{disp}} = -s_6 \sum_{i=1}^{N_{\mathrm{atom}}-1} \sum_{j=i+1}^{N_{\mathrm{atom}}} D_{ij} \tag{12}$$

where s_6 is a scale factor that depends on *ab initio* energy expression that is used in conjunction to the dispersion correction.

The dispersion energy expression 10 requires a C_6 coefficient for every pair of chemical elements. To reduce the number of these required parameters the pair C_6 coefficients are approximated and expressed in terms of C_6 coefficients of the elements. The current implementation supports two models for this:

The *Average* C_6 pair model In this model the pair C_6 coefficient is approximated simply as the average of the elemental C_6 coefficients [47]:

$$C_6^{ij} = 2\frac{C_6^i\, C_6^j}{C_6^i + C_6^j} \tag{13}$$

This is presently the default.

The *Geometric Mean* C_6 pair model In this model the pair C_6 coefficient is approximated as the geometric mean of the elemental C_6 coefficients [48, 49]:

$$C_6^{ij} = \sqrt{C_6^i\, C_6^j} \tag{14}$$

This is currently considered as the most accurate model. It also supports the most chemical elements through published atomic C_6 coefficients [48].

8.0.1 VDWAALS – Directives controlling the dispersion corrections

Detailed control of the dispersion corrections is available through a special input block (for use with DFT calculations a simplified input is available, see 11.5). This input block starts with the VDWAALS directive and terminates with the END directive. In between these two directives any number of directives to fine tune the dispersion corrections may be presented. So the input block is of the form:

Example

```
VDWAALS
   ...
END
```

Within the above structure the directives ON, OFF, C6MODEL, SCALE, ALPHA, RADIUS, and C6 may be used as described below. Note that the directives SCALE, ALPHA, RADIUS and C6 set values independently for the active C_6 pair model. I.e.

```
VDWAALS
   C6MODEL AVERAGE
   SCALE 1.3
   EXPONENT 30.0
   C6MODEL GEOMETRIC
END
```

results in using the geometric mean C_6 pair model with the default settings and NOT with the scale factor and exponent from the input. The reason is that changing the parameters for the average C_6 pair model does not affect those of the geometric mean model and *vice versa*.

8.0.1.1 The ON directive

The ON directive may be used to explicitly turn the dispersion corrections on. The directive is read to the variable TEXT using the format (A) where TEXT should be set to the character string ON.

8.0.1.2 The OFF directive

The OFF directive may be used to explicitly turn the dispersion corrections off. The directive is read to the variable TEXT using the format (A) where TEXT should be set to the character string OFF.

8.0.1.3 The C6MODEL directive

The C6MODEL directive may be used to choose the C_6 coefficient pair model to be used. The directive consists of two data fields read to the variables TEXT and TEXTA using the format (A,A)

- TEXT should be set to the character string C6MODEL
- TEXTA may be set to either
  - AVERAGE to select the *average* C_6 coefficient pair model according to equation 13
  - GEOMETRIC to select the *geometric mean* C_6 coefficient pair model according to equation 14

8.0.1.4 The SCALE directive

The SCALE directive may be used to change the overall scale factor s_6 for the dispersion correction. The directive consists of two data fields read to the variables TEXT and SCALE using the format (A,F)

- TEXT should be set to the character string SCALE.
- SCALE should be set to the scale factor for the current C_6 coefficient pair model which has to be a non-negative floating point value.

8.0.1.5 The ALPHA directive

The ALPHA directive may be used to change the exponent α for the damping function of equation 11. The directive consists of two data fields read to the variables TEXT and EXPONENT using the format (A,F)

- TEXT should be set to the character string ALPHA.
- EXPONENT should be set to the exponent of the damping function for the current C_6 coefficient pair model. It has to be a non-negative floating point value.

8.0.1.6 The RADIUS directive The RADIUS directive may be used to change the Van der Waals radius of a chemical element. The directive consists of three data fields read to the variables TEXT, ELEMENT and RAD0 using the format (A,A,F)

- TEXT should be set to the character string RADIUS.
- ELEMENT should be set to the chemical symbol of the element.
- RAD0 should be set to the Van der Waals radius of the element for the current C_6 coefficient pair model. The value should be specified in Bohr (atomic units).

8.0.1.7 The C6 directive The C6 directive may be used to change the C_6 coefficient of a chemical element. The directive consists of three data fields read to the variables TEXT, ELEMENT and COEFF using the format (A,A,F)

- TEXT should be set to the character string C6.
- ELEMENT should be set to the chemical symbol of the element.
- COEFF should be set to the C_6 coefficient of the element for the current C_6 coefficient pair model. The value should be specified in Hartree*Bohr^6 (atomic units).

Example

This example sets the parameters as recommended by Antony et al. [49] for a B3LYP calculation on formaldehyde.

```
VDWAALS
    C6MODEL GEOMETRIC
    SCALE     1.05
    ALPHA    20.0
    RADIUS H  1.89
    RADIUS C  2.74
    RADIUS O  2.54
    C6     H  2.43
    C6     C 30.35
    C6     O 12.14
END
```

9 Directives Controlling MCSCF Calculations

Data input characterising the MCSCF calculation commences with the MCSCF data line and is typically followed by a sequence of directives, terminated by presenting a valid *Class 2* directive, such as VECTORS or ENTER.

9.1 MCSCF

The MCSCF data initiator consists of a single data line with the character string MCSCF in the first data field. It acts to transfer control to those routines responsible for inputing all data relevant to the MCSCF calculation. Termination of this data is achieved by presenting a valid *Class 2* directive that is not recognised by the MCSCF input routines, for example ENTER.

9.2 ORBITAL

The ORBITAL directive must be presented in a MCSCF calculation, and acts to,

1. define the active space in the calculation, by partitioning the orbitals into a primary and secondary set [3];
2. specify an initial reference configuration that will be employed, for example, in generating the complete CI space in a CASSCF calculation.

The directive comprises a number of data lines, with the first line containing the character string ORBITAL in the first data field. Subsequent lines comprise a sequence of orbital TAGS whereby each orbital in the primary space is classified both by type and by symmetry, with this sequence terminated by the character string END. The following orbital types are used in this classification:

- FZC - frozen orbital i.e. an orbital which will remain frozen as input throughout the MCSCF iterations.
- COR - core orbital i.e. an orbital which will remain doubly occupied in all configurations.
- DOC - doubly occupied i.e. an orbital which is doubly occupied in the reference configuration, and which will be permitted variable occupancy in the MCSCF treatment.
- ALP - an unpaired orbital i.e. an orbital which is singly occupied with α–spin in the principle reference configuration, and which will be permitted variable occupancy in the MCSCF treatment.
- BET - an unpaired orbital i.e. an orbital which is singly occupied with β–spin in the principle reference configuration, and which will be permitted variable occupancy in the MCSCF treatment.
- UOC - formally unoccupied orbitals, corresponding to SCF virtual MOs, which will be permitted variable occupancy in the MCSCF.

Each of the core and active orbitals must be classified according to its type above, and in addition, its symmetry (IRrep) under the point group symmetry in use. The integer flag characterising the symmetry (as produced for example in the SCF output) is appended to the appropriate 3-character string above, so that for a C_{2v} molecule, a doubly occupied orbital of a_1 symmetry would be tagged DOC1, an unoccupied orbital of b_2 symmetry, UOC3.

9.2.1 ORBITAL - Example 1

Let us consider initially various calculations on the water molecule to illustrate ORBITAL specification. The example is based on a TZVP basis, with the following set of input MOs, derived from a closed-shell SCF calculation:

```
             TOTAL ENERGY       -76.0553958459
===============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
===============================================
  1      1    -20.55996277           2.0000000
  2      1     -1.35317135           2.0000000
  3      3     -0.71806974           2.0000000
  4      1     -0.58054044           2.0000000
  5      2     -0.50734492           2.0000000
  6      1      0.13628400           0.0000000
  7      3      0.19133694           0.0000000
  8      3      0.52235466           0.0000000
  9      1      0.52876578           0.0000000
 10      2      0.55149622           0.0000000
 11      1      0.62397742           0.0000000
 12      3      0.73116091           0.0000000
 13      1      1.04616882           0.0000000
 14      1      1.88402305           0.0000000
 15      4      1.92286966           0.0000000
 16      2      2.12532394           0.0000000
 17      3      2.18404917           0.0000000
 18      1      2.28261839           0.0000000
 19      3      2.37902858           0.0000000
 20      3      2.69431254           0.0000000
 21      1      2.70971740           0.0000000
 22      2      2.71654746           0.0000000
 23      1      3.05834268           0.0000000
 24      3      3.25415102           0.0000000
 25      2      3.54107441           0.0000000
 26      1      3.55600364           0.0000000
 27      4      3.59173828           0.0000000
 28      1      3.83258378           0.0000000
 29      1      4.79289351           0.0000000
 30      3      5.12382812           0.0000000
 31      1      7.71922625           0.0000000
 32      1     47.55358026           0.0000000
===============================================
```

To perform a full-valence space calculation, comprising 7 primary orbitals, with the 4a_1 and 2b_2 virtual MOs (with sequence numbers 6 and 7 respectively) included in the active space, would require the following ORBITAL data:

```
ORBITAL
DOC1 DOC1 DOC3 DOC1 DOC2 UOC1 UOC3
END
```

Note that it is possible to abbreviate the data specification when successive orbitals of identical symmetry and type are involved. This is achieved by preceding the orbital tag with an integer depicting the number of repeated orbitals. Thus the data above may be presented thus:

```
ORBITAL
2DOC1 DOC3 DOC1 DOC2 UOC1 UOC3
END
```

To freeze the O1s orbital as the SCF orbital, we would present the sequence

```
ORBITAL
FZC1 DOC1 DOC3 DOC1 DOC2 UOC1 UOC3
END
```

and to maintain the double-occupancy of the orbital, while enabling it to relax, would require the sequence:

```
ORBITAL
COR1 DOC1 DOC3 DOC1 DOC2 UOC1 UOC3
END
```

The following sequence would be used to extend the space to include the 5a$_1$ and 2b$_1$ orbitals:

```
ORBITAL
COR1 DOC1 DOC3 DOC1 DOC2 UOC1 UOC3 UOC1 UOC2
END
```

Note that it is not required that all orbitals in the active space appear first in the input orbital set, the specified sequence being derived automatically from the input MOs. Based on an RHF calculation on the X$^2B_1$ state of the H_2O^+ ion, then a full-valence calculation, with frozen O1s, would be performed under control of the following ORBITAL data:

```
ORBITAL
2DOC1 DOC3 DOC1 ALP2 UOC1 UOC3
END
```

9.2.2 ORBITAL - Example 2

Assume the following set of input MOs from a TZVP–RHF calculation on the X$^3B_1$ state of methylene:

```
            TOTAL  ENERGY        -38.9321296186
===============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
===============================================
  1      1    -11.24483142          2.0000000
  2      1     -0.85646872          2.0000000
  3      3     -0.59962511          2.0000000
  4      1     -0.47810023          1.0000000
  5      2     -0.40348167          1.0000000
```

```
   6      1      0.16128671          0.0000000
   7      3      0.24034249          0.0000000
   8      3      0.28796496          0.0000000
   9      1      0.30384049          0.0000000
  10      2      0.32139063          0.0000000
  11      1      0.45957080          0.0000000
  12      1      0.63630157          0.0000000
  13      3      0.70724008          0.0000000
  14      3      1.40064194          0.0000000
  15      1      1.44828705          0.0000000
  16      2      1.47979149          0.0000000
  17      4      1.54845894          0.0000000
  18      1      1.69050079          0.0000000
  19      2      1.73938256          0.0000000
  20      1      1.78688113          0.0000000
  21      1      1.94167219          0.0000000
  22      3      1.94783841          0.0000000
  23      3      2.30755067          0.0000000
  24      2      2.48250909          0.0000000
  25      1      2.48720842          0.0000000
  26      4      2.73866260          0.0000000
  27      3      2.93545830          0.0000000
  28      1      2.99931959          0.0000000
  29      3      3.66924826          0.0000000
  30      1      3.71124278          0.0000000
  31      1      4.85256025          0.0000000
  32      1     26.27807526          0.0000000
=================================================
```

A full-valence space calculation on the X$^3B_1$ state, with frozen C1s, would be controlled thus:

```
ORBITAL
2DOC1 DOC3 ALP1 ALP2 UOC1 UOC3
END
```

The corresponding calculation on the $^1B_1$ state, featuring singlet coupling of the 3a$_1$ and 1b$_1$ MOs, would require use of the ALP and BET tags, thus

```
ORBITAL
2DOC1 DOC3 ALP1 BET2 UOC1 UOC3
END
```

remembering to change (or remove) the MULT 3 specification !

9.3 CANONICAL

The CANONICAL directive may be used to control the routing of MCSCF natural orbitals to the Dumpfile, and to specify the canonicalisations in effect for the three categories of orbital – core, active and secondary. The directive consists of a single data line read to the variables TEXT, ISECNO, TEXTC, TEXTA, TEXTS using format (A,I,3A).

- TEXT should be set to the character string CANONICAL.
- ISECNO should be set to a section number on the Dumpfile for output of the MCSCF natural orbitals.
- TEXTC is a character string for controlling the canonicalisation of the core orbitals. In generating the optimum MOs for subsequent use in CI calculations, TEXTC should be set to the string FOCK.
- TEXTA is a character string for controlling the canonicalisation of the active orbitals. In generating the MCSCF natural orbitals, TEXTA should be set to the string DENSITY.
- TEXTS is a character string for controlling the canonicalisation of the secondary orbitals. In generating the optimum MOs for subsequent use in CI calculations, TEXTS should be set to the string FOCK.

Example

```
CANONICAL 10 FOCK DENSITY FOCK
```

would be used to route the natural orbitals to section 10 on the Dumpfile. Note that the above settings are now applied in default (Version 6.3 onwards), so the user need only present the CANONICAL directive to override these defaults e.g. a different section for the MCSCF natural orbitals.

9.4 PRINT

The PRINT directive may be used to increase the default MCSCF output, and consists of a single data line with the character string PRINT in the first data field. Subsequent data fields may comprise any combination of the following parameters:

NATORB	1-particle density matrix and natural orbitals
ORBITALS	Molecular orbital coefficient array
CIVECTOR	The CI vector
VIRTUALS	Print virtual MOs in addition to core and active

Note that the default PRINT settings (Version 6.3 onwards) correspond to presenting the data line

```
PRINT ORBITALS VIRTUALS NATORB
```

10 Directives Controlling MASSCF Calculations

This section describes the parallel Multiple Active Space SCF (MASSCF) code in GAMESS-UK. The MASSCF method draws upon a mature body of literature covering techniques for constructing and optimizing MCSCF wavefunctions [50]. MASSCF is shorthand for ORMAS-MCSCF using the full Newton-Raphson (full-NR) orbital optimization technique [51]. MASSCF proceeds via a two-step decoupled approach in which the CI and orbital update problems are solved separately.

ORMAS (Occupation Restricted Multiple Active Space) allows multiple active spaces to be defined governed by certain occupation rules or restrictions [52]. The machinary of ORMAS then generates all possible determinants satisfying the occupation restrictions. The usefulness of this approach is twofold. Firstly, a single compact notation generalizes to traditional CI wave function types such as full CI as well as more novel situations involving more than two orbital spaces. Secondly, the use of multiple small active spaces has been shown to accurately recover the quality of much larger full CI calculations at a fraction of the cost [53]. In this way ORMAS may be viewed as a powerful tool for generating N-particle wave functions.

A MASSCF calculation is invoked by the directive,

```
        SCFTYPE MASSCF
```

The above keywords by themselves are insufficient to specify a MASSCF calculation. MASSCF requires minimal information regarding the numbers of electrons and orbitals involved. In most cases, a keyword is to be followed by one or more integers, or a floating-point number in any standard representation.

10.1 Basic MASSCF input keywords and their parameters

The minimum requirements for defining a MASSCF calculation are as follows,

- MASSCF – Keyword identifying the MASSCF input clause (this is a flag, no numerical parameter needed).
- NCORE – Total number of orbitals doubly occupied in all determinants.
- NACT – Total number of active orbitals.
- NELS – Total number of active electrons.

Additional input parameters controlling convergence of the full Newton-Raphson solver are as follows,

- TOLENG – Convergence tolerance on the total energy (Default value is 1.0d-10).
- ACURCY – Newton-Raphson and Davidson convergence (Default value is 1.0d-5).

- DAMP – Newton-Raphson damping factor (Default value is 0.0).
- MAXMAS – Maximum allowed MASSCF iterations (Default is 30).
- NFRZ – Number of frozen orbitals (Default is 0, maximum is 20).
- MOFRZ – A list of NFRZ integers specifying the orbitals to be kept frozen throughout the MASSCF calculation.
- NOROT – Number of frozen rotations (Default is 0, maximum is 20).
- NROTAB – A list of NOROT integer-pairs specifying the orbital rotations to be kept frozen throughout the MASSCF calculation.
- MXPN – Maximum number of Davidson expansions (Default is 10).
- MXIT – Maximum number of Davidson iterations (Default is 30).
- FCORE – Option to freeze all core orbitals (this is a flag, no numerical parameter needed).
- SRSO – Perform single-reference second order (SRSO) calculation. SRSO is useful in cases where an RHF or ROHF wavefunction is difficult to converge and the more powerful Newton-Raphson solver in MASSCF could be successful. A single SCF iteration is executed to provide orbitals and the CI step executes trivially only to provide the 1- and 2-particle densities necessary to form the orbital hessian matrix (this is a flag, no numerical parameter needed).

By themselves, the above input specifications define the equivalent of a CASSCF calculation using the full-NR method on the ground state. In order to make full use of the capabilities of the ORMAS code, the following input is required.

10.2 ORMAS input keywords and their parameters

There are no sensible defaults for the next four inputs but in their absense the ORMAS code will perform a full CI calculation, as described above. That is, the defaults are NSPACE 1; NORB [NACT]; MINE [NELS]; MAXE [NELS]; meaning all active orbitals are in one partition.

- NSPACE – Number of orbital groups you wish to define.
- NORB – A list of NSPACE integers. These specify the number of orbitals in each active space or group.
- MINE – A list of NSPACE integers. These specify the minimum numbers of electrons that must always occupy the orbital groups. In other words, MINE(I) is the minimum number of electrons that can occupy space I in any of the determinants.

- MAXE – A list of NSPACE integers. These specify the maximum numbers of electrons that must always occupy the orbital groups. In other words, MAXE(I) is the maximum number of electrons that can occupy space I in any of the determinants.
- ROOT – Selects the root of the CI problem to solve for. This enables the selection of ground and (singly) excited states, the default is ROOT=0 for the ground state, ROOT=1 will select the first excited state and so on (Default is 0).
- NSTATE – Number of states to average over (Default is 1).
- WSTATE – A list of NSTATE integers specifying the weights of the states to average over (default entry is 1.0 for the first state).
- SYMSTATE – Symmetry of the target state (Default is the input symmetry).

Additional input parameters controlling convergence of the ORMAS CI solver are as follows,

- KEEPVEC – Number of CI vectors to keep for next CI step (Default is 1 so the current solution provides the guess for the next CI step).
- CIGUESS – Dimension of CI guess Hamiltonian (Default is 300).
- ITERCI – Maximum number of Davidson iterations to solve CI (Default is 100).
- CRIT – Energy convergence criterion (Default value is 1.0d-5).
- MAXP – Maximum number of Davidson expansions in CI (Default is 10).

The following examples serve to illustrate the construction of CI wave functions using ORMAS.

10.2.1 MASSCF - Example 1

```
TITLE
H2O  - MASSCF   - 3-21G BASIS
ZMATRIX ANGSTROM
O
H 1 OH
H 1 OH 2 HOH
VARIABLES
OH  0.956
HOH 104.5
END
SCFTYPE MASSCF
MASSCF
  NCORE  1
  NACT   8
  NELS   8
END
ENTER
```

This is entirely equivalent to-

10.2.2 MASSCF - Example 2

```
TITLE
H2O  - MASSCF   - 3-21G BASIS
ZMATRIX ANGSTROM
O
H 1 OH
H 1 OH 2 HOH
VARIABLES
OH  0.956
HOH 104.5
END
SCFTYPE MASSCF
MASSCF
  NCORE  1
  NACT   8
  NELS   8
  NSPACE 1
  NORB   8
  MINE   8
  MAXE   8
END
ENTER
```

10.2.3 MASSCF - Example 3

In this example, up to double excitations (CISD) are allowed into a space containing eight virtual orbitals.

This case is difficult to converge, so the maximum number of MASSCF iterations needs to be increased.

```
      TITLE
      HCN  6-31G** MASSCF ...RHF geometry
      GEOMETRY ANGSTROM
        0.0 0.0 -1.0589956  1.0  H
        0.0 0.0  0.0000000  6.0  C
        0.0 0.0  1.1327718  7.0  N
      END
      BASIS 6-31G**
      SCFTYPE MASSCF
      MASSCF
        NCORE  2
        NELS  10
        NACT  13
        NSPACE 2
        NORB   5  8
        MINE   8  0
        MAXE  10  2
MAXMAS 100
      END
      ENTER
```

10.2.4 MASSCF - Example 4

One need not be limited to conventional CI wave function types as the following example serves to illustrate.

```
      TITLE
      HCN  6-31G** MASSCF ...RHF geometry
      GEOMETRY ANGSTROM
        0.0 0.0 -1.0589956  1.0  H
        0.0 0.0  0.0000000  6.0  C
        0.0 0.0  1.1327718  7.0  N
      END
      BASIS 6-31G**
      SCFTYPE MASSCF
      MASSCF
        NCORE  2
        NACT  11
        NELS  10
        NSPACE 3
        NORB   5  4  2
        MINE   4  0  0
        MAXE  10  4  2
MAXMAS 1000
      END
      ENTER
```

In the above example, up to quadruple excitations are allowed into the first virtual space and up to double excitations in the second virtual space.

Again, this calculation is extremely slow to converge so the maximum number of MASSCF iterations must be increased to nearly 1000 to ensure convergence.

11 Directives Controlling DFT Calculations

11.1 Introductory remarks

Before describing the DFT-specific input, we briefly outline some background material that users should be aware of before attempting to use the Density Functional Theory module within GAMESS-UK.

1. **The functionals**: An essential result from the paper by Hohenberg and Kohn [45] was that DFT would yield the exact ground state energy and electron density if the exchange-correlation functional was known. In practice the exact functional is unknown but one may try some approximate form. This has led to a extensive search for functionals with new variations being published on a regular basis. Because the quality of the results depends critically on the functional selecting a suitable form will be a vital factor in using the module.

2. **The integration grids**: Another issue related to the functionals stems from their form; most functionals are such that they can not be integrated analytically over all space. Therefore, the exchange-correlation energy can be evaluated only through numerical integration.

   It was found that this numerical integration could only be successful if the integration grid is adapted to the particular features of the molecular density. These features are that the density is high and nearly spherically symmetric near the nuclei. Between the nuclei the density has less symmetry and is smaller. However because most of the chemistry depends on the density between the nuclei accurate integration in that region is essential.

   To devise integration grids adapted to these features the atoms of a molecule were taken as the central points. Each nucleus would be the center of a set of spherical grids with ever larger radii. The simplest way to obtain such a grid would be to take the Cartesian product of a radial grid with a spherical grid. But more advanced schemes can be engineered.

   Once the atomic grids have been constructed they have to be merged into a molecular grid. To avoid artifacts from the finite size of the atomic grids it is essential that grid points be faded out if they get to close to another atom. So called weighting functions were designed for this purpose.

   So for proper integration the selection of angular grids, radial grids and weighting functions have to be addressed.

3. **Improving integration efficiency**: Although the above approach properly defines the molecular integration grid the efficiency of applying this grid can be improved through 2 strategies:

   (a) Screening – This is based on removing grid points or functions at grid points that contribute little to the exchange-correlation energy from the calculation at the earliest opportunity;

   (b) Pruning – This is based on replacing 2 close grid points by 1 new grid point.

11.2 The DFT Directive and Default Settings

In common with most Density Functional programs, the DFT module within GAMESS–UK is implemented as a modified Hartree-Fock program with the exchange term in the Hartree-Fock equations replaced by the exchange-correlation term [45]. Thus input for a DFT calculation is essentially that for the closed-shell RHF or UHF module, with additional keywords that control the DFT specific features.

In the simplest case, the user need just introduce a single data with the character string CDFT or DFT in the first data field to request a DFT rather than HF calculation, thus input for a closed-shell DFT calculation would appear as follows:

```
TITLE
H2CO - 3-21G  CLOSED SHELL DFT (B-LYP DEFAULT QUADRATURE)
SUPER OFF NOSYM
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
DFT
ENTER
```

while the corresponding UHF data for performing an open-shell unrestricted UKS calculation would appear thus,

```
TITLE
H2CO+ - 2B2 - DEFAULT 3-21G BASIS - UKS CALCULATION
CHARGE 1
MULT 2
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
SCFTYPE UHF
DFT
ENTER
```

The directive DFT thus "switches on" the DFT specific modifications to the Hartree-Fock scheme. Leaving the directive out would yield the corresponding Hartree-Fock input.

If, as in the above, the DFT module is switched on without specifying any options then the following functional and quadrature settings will apply;

- the Becke (1988) exchange functional [32]
- the Lee, Yang and Parr (LYP) correlation functional [34]

- quadrature grids designed to obtain a relative error of less than 1.0e-6 in the number of electrons per atom. These grids are constructed from the logarithmic radial grid [25, 31] and Lebedev angular grid [43], using the MHL8SSF weighting scheme with screening and MHL angular grid pruning [31]. Note that this choice corresponds to the "QUADRATURE MEDIUM" setting described below.
- the gradient of the energy will be evaluated without considering the gradient of the quadrature weights and grid points, this corresponds to "GRADQUAD OFF".

11.3 DFT Directive Options

The role of the DFT directive is twofold, (i) to trigger a DFT rather than HF calculation, and (ii) to provide a mechanism for overriding the default DFT functional and quadrature settings. The latter is achieved by specifying the DFT options described below on one or more data lines, each containing the character string DFT in the first data field; the user may present as many data lines as desired in specifying these options, providing the mechanism for presenting long option lists over several lines. Note that the DFT data lines should be presented after both RUNTYPE and SCFTYPE directives (if present), and before the VECTORS directive (if present).

Thus the default DFT specifications invoked by the data input above may also be invoked by explicit specification, thus

```
TITLE
H2CO - 3-21G  CLOSED SHELL DFT (B-LYP DEFAULT QUADRATURE)
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
DFT B-LYP QUADRATURE MEDIUM
ENTER
```

or by specifying the functional and quadrature settings on separate DFT data lines, thus

```
TITLE
H2CO - 3-21G  CLOSED SHELL DFT (B-LYP DEFAULT QUADRATURE)
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
DFT BLYP
DFT QUADRATURE MEDIUM
ENTER
```

or even,

```
TITLE
H2CO - 3-21G  CLOSED SHELL DFT (B-LYP DEFAULT QUADRATURE)
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
DFT BECKE88
DFT LYP
DFT QUADRATURE MEDIUM
ENTER
```

11.4 Specification of Functionals

As described above, The default functional used in the current DFT implementation is the so-called B-LYP functional, employing the Becke88 exchange functional [32] and Lee, Yang and Parr correlation (LYP) correlation energy functional [34]. Over-riding this default may be achieved through the following DFT keywords:

- NULL_X; The keyword NULL_X selects the null exchange functional. Obviously this is useful only in very special cases.

- HF_X; The keyword HF_X selects the Hartree-Fock exchange term as the exchange functional.

- LDA_X; The keyword LDA selects the LDA exchange energy functional.

- B88_X; The keyword B88_X selects the default Becke'88 exchange functional This is a gradient-corrected exchange energy functional with correct *1/r* asymptotic behaviour of the exchange-energy density [32].

- B3_X; The keyword B3_X selects the Becke3 exchange functional This is the three parameter gradient-corrected hybrid exchange energy functional with the Becke'88 functional as one of its components [28].

- B97_X; The keyword B97_X selects the Becke97 exchange functional [23].

- EDF1_X; The keyword EDF1_X selects the exchange functional of Empirical Density Functional one [16].

- FT97A_X; The keyword FT97A_X selects the Filatov-Thiel exchange functional variant A [39].

- FT97B_X or FT97_X; The keywords FT97B_X or FT97_X select the recommended Filatov-Thiel exchange functional variant B [39].

- PBE_X; The keyword PBE_X selects the Perdew-Burke-Ernzerhof exchange functional [20].

- PW91_X; The keyword PW91_X selects the Perdew-Wang'91 exchange functional [21].
- NULL_C; The keyword NULL_C selects the NULL correlation functional.
- B95 or B95_C; The keyword B95 or B95_C select the Becke'95 meta correlation functional [15].
- EDF1_C; The keyword EDF1_C selects the correlation part of the Empirical Density Functional one [16].
- FT97_C; The keyword FT97_C selects the Filatov-Thiel correlation functional [38]
- LYP or LYP_C; The keywords LYP or LYP_C select the default Lee-Yang-Parr correlation energy functional [34].
- P86 or P86_C; The keywords P86 or P86_C select the Perdew'86 gradient corrected correlation functional [35].
- PBE_C; The keyword PBE_C selects the Perdew-Burke-Ernzerhof correlation functional [20].
- PZ81 or PZ81_C; The keywords PZ81 or PZ81_C select the Perdew-Zunger local density correlation functional [36].
- PW91_C; The keyword PW91_C selects the Perdew-Wang'91 correlation functional [21].
- PW92 or PW92_C; The keywords PW92 or PW92_C selects the Perdew-Wang'92 local density correlation functional [22].
- VWN, VWN5 or VWN_C; The keywords VWN, VWN5 or VWN_C select the recommended Vosko-Wilk-Nusair local density correlation functional [37].
- VWNRPA, VWN5RPA or VWNRPA_C; The keywords VWNRPA, VWN5RPA or VWNRPA_C select the Vosko-Wilk-Nusair local density correlation functional with the RPA parametrisation [37].
- NULL; The keyword NULL selects the null exchange-correlation energy. Obviously this is should be used only in very special cases.
- B3LYP; The keyword B3LYP selects the infamous hybrid exchange-correlation functional proposed by Stephens et al. [12]. As Stephens et al. did not use the recommended VWN functional as one of the components this functional has attracted much controversy. The current understanding is that the functional employs VWN3 [13].
- B1B95; The keyword B1B95 selects a meta hybrid exchange-correlation functional build from Hartree-Fock exchange (28%), the Becke'88 exchange functional (72%) and the Becke'95 meta correlation functional [15].
- B97; The keyword B97 selects the Becke'97 hybrid exchange-correlation functional [23].
- B97-1; The keyword B97-1 selects the hybrid exchange-correlation functional of the same form as B97 but reoptimised by Hamprecht et al. [17, 23].

- B97-2; The keyword B97-2 selects the hybrid exchange-correlation functional of the same form as B97 but reoptimised by Wilson et al. [24, 23].
- BB1K; The keyword BB1K selects a meta hybrid exchange-correlation functional build from Hartree-Fock exchange (42%), the Becke'88 exchange functional (58%) and the Becke'95 meta correlation functional [14].
- BB95; The keyword BB95 selects the Becke'88 exchange functional and the Becke'95 meta correlation functional [15].
- BLYP; The keyword BLYP selects the Becke88 exchange energy functional [32] and the Lee, Yang and Parr correlation energy functional [34].
- BP86; The keyword BP86 selects the Becke88 exchange energy functional [32] and the Perdew 1986 gradient corrected correlation functional [35].
- EDF1; The keyword EDF1 selects Empirical Density Functional One as proposed by Adamson et al. [16].
- FT97; The keyword FT97 selects the Filatov, Thiel gradient corrected exchange-correlation energy functional [38, 39]. This functional comprises the Filatov, Thiel correlation energy functional and the exchange energy functional variant B.
- HCTH or HCTH93; The keywords HCTH or HTCH93 select the Hamprecht-Cohen-Tozer-Handy exchange-correlation energy functional fitted against a training set of 93 molecules [17].
- HCTH120; The keyword HCTH120 selects the Hamprecht-Cohen-Tozer-Handy exchange-correlation energy functional fitted against a training set of 120 molecules [18].
- HCTH147; The keyword HCTH147 selects the Hamprecht-Cohen-Tozer-Handy exchange-correlation energy functional fitted against a training set of 147 molecules [18].
- HCTH407; The keyword HCTH407 selects the Hamprecht-Cohen-Tozer-Handy exchange-correlation energy functional fitted against a training set of 407 molecules [19].
- PBE; The keyword PBE selects the Perdew-Burke-Enrzerhof gradient corrected exchange-correlation functional [20].
- PW91; The keyword PW91 selects the Perdew-Wang'91 gradient corrected exchange-correlation functional [21].
- SVWN; The keyword SVWN selects the LDA exchange functional and the Vosko-Wilk-Nusair local density correlation functional [37].

11.5 Specification of Dispersion Corrections

A problem frequently encountered with DFT calculations is that long range correlation effects such as dispersion or Van der Waals interactions are not properly described. This leads to problems where these relatively weak dispersion forces are important such as in DNA base pair stacking or the interaction between aromatic molecules. So far attempts to rigorously address these problems have had limited success. In response to this empirical approaches to correct for the lack of dispersion have been suggested. GAMESS-UK supports two versions of one of these approaches named DFT-D [47, 48, 49]. Both these approaches are based on a simple model for the dispersion energy of two interacting atoms

$$D_{ij}(R_{ij}) = C_6^{ij}\frac{f_{\mathrm{damp}}(R_{ij})}{R_{ij}^6}. \tag{15}$$

The function f_{damp} is simply chosen such that it goes to zero when R_{ij} goes to zero to prevent that atoms collapse onto eachother. The above expression requires a C_6 coefficient for every pair of chemical elements. To reduce the number of these required parameters the pair C_6 coefficients are approximated and expressed in terms of C_6 coefficients of the elements. The current implementation supports two models for this:

The *Average* C_6 pair model In this model the pair C_6 coefficient is approximated simply as the average of the elemental C_6 coefficients [47]:

$$C_6^{ij} = 2\frac{C_6^i\,C_6^j}{C_6^i + C_6^j} \tag{16}$$

This is presently the default.

The *Geometric Mean* C_6 pair model In this model the pair C_6 coefficient is approximated as the geometric mean of the elemental C_6 coefficients [48, 49]:

$$C_6^{ij} = \sqrt{C_6^i\,C_6^j} \tag{17}$$

This is currently considered as the most accurate model. It also supports the most chemical elements through published atomic C_6 coefficients [48].

11.5.1 The DISPERSION directive

In the DFT input section the basic properties of the dispersion correction may be controlled throught the DISPERSION directive (for a much more detailed control over these correction see section 8). This directive may consist of one or two data fields read to the variables TEXT and TEXTOPT using the format (A,A)

- TEXT should be set to the character string DISPERSION;
- TEXTOPT is an optional field which may be set to either of

  - ON to switch the dispersion corrections on (although this is currently implied in specifying the DISPERSION correction);
  - OFF to explicitly swith the dispersion correction off;
  - AVERAGE to select the *average* C_6 coefficient pair model (presently the default);
  - GEOMETRIC to select the *geometric mean* C_6 coefficient pair model

The following three examples all result in using the dispersion corrections in accordance with the default settings.

Examples

```
DFT DISPERSION

DFT DISPERSION ON

DFT DISPERSION AVERAGE
```

11.6 Specification of Integration Grids

While a large number of options are available in specifying possible integration grids (see below), the inexperienced user is strongly advised to use just the QUADRATURE directive for this purpose.

11.6.1 The QUADRATURE Directive

This directive may be used to select a quadrature grid that is designed to achieve a specified accuracy. The resulting grids are constructed from the logarithmic radial grid [25] and Lebedev angular grids [43], using the SSF weighting scheme with screening [26] and MHL angular grid pruning [31]. The directive consists of two data fields, read to the variables TEXT, ACCU using format 2A;

- TEXT should be set to the character string QUADRATURE;
- ACCU is a keyword used to define the required grid accuracy. Valid keywords include;
  - LOW - The LOW accuracy grid should only be used for preliminary studies; it is designed to obtain the total number of electrons from the density integration with a relative error of 1.0e-4 per atom.
  - MEDIUM - The MEDIUM accuracy grid is designed to obtain a relative error of less than 1.0e-6 in the number of electrons per atom.
  - HIGH - The HIGH accuracy grid is designed to obtain a relative error of less than 1.0e-8 in the number of electrons per atom.
  - VERYHIGH - The VERYHIGH accuracy grid is meant only for benchmark calculations. It is designed to be significantly more accurate than the high accuracy grid.

The directive may be omitted when ACCU will be set to the default MEDIUM quadrature setting.

Example

```
TITLE
H2CO - 6-31G  CLOSED SHELL DFT (B3LYP HIGH QUADRATURE)
ZMATRIX ANGSTROM
C
O 1 1.203
H 1 1.099 2 121.8
H 1 1.099 2 121.8 3 180.0
END
BASIS 6-31G
DFT B3LYP
DFT QUADRATURE HIGH
ENTER
```

Note that the LABEL and ELEMENT keywords discussed in the next subsection may also be used with the QUADRATURE sub-directive.

11.7 Detailed Grid Specification

A number of sub-directives of DFT are available to control the grids to be used, although it is not expected that these would be routinely invoked when running the DFT module. These sub-directives include those for (i) specifying both the angular integration grid (LEBEDEV or GAUSS-LEGENDRE) and radial grid (EULER-MACLAURIN or LOGARITHMIC), (ii) the screening of grid points (SCREEN), (iii) an appropriate weighting scheme (WEIGHT) and (iv) activating angular grid pruning (ANGPRUNE).

11.7.1 Grid Specification on a Per Atom Basis – ELEMENT and LABEL

For optimal control over the integration it is required that the grid may be modified for each atom separately. For this purpose the keywords ELEMENT and LABEL were introduced. The keyword ELEMENT is followed by a list of one or more elements and the requested setting is applied to all atoms of that element. The keyword LABEL works similarly to ELEMENT but uses the atom labels as specified in the geometry. The various specifications are executed in order of appearance.

Example

```
DFT LEBEDEV 302 ELEMENT C H 194 LABEL C1 H2 H4 266
```

This directive sets the angular grid for all atoms to the 302 point Lebedev grid, then sets the grid for all carbon and hydrogen atoms to the 194 point grid, and finally overrides the grid for all atoms with names C1, H2 and H4 giving them the 266 point grid.

11.7.2 Angular Integration Grid – LEBEDEV

The LEBEDEV directive requests the grids of Lebedev for angular integration [43]. These grids have been designed to integrate polynomials on a sphere exactly up to a specific order. Grids with 6, 14, 26, 38, 50, 74, 86, 110, 146, 170, 194, 234, 266, 302, 350, 434, 590, 770, 974 and 1202 points are supported. In its simplest form, the directive consists of two data fields read to the variables TEXT, NPT using format (A,I)

- TEXT should be set to the character string LEBEDEV
- NPT is an integer specifying the required number of points

It has been noted that close to the nucleus the density is more spherically symmetric than at larger distances, so that a smaller angular grid can be used for smaller radii. This capability is provided by an extension to the directive whereby different angular grid may be specified for different radii. In this case the LEBEDEV directive comprises the following data fields;

- TEXT should be set to the character string LEBEDEV
- pairs of data fields are then presented, each pair characterising a specific radial zone and read to the variables (NPT$_i$, RZ$_i$) using format (I,F), where
  - NPT$_i$ specifies the grid size in the i-th radial zone.
  - The floating point values of RZ$_i$ subdivide the radial coordinate into different zones. The values RZ$_i$ are fractions of the Bragg-Slater radius [46] of the atom. Each zone runs from RZ$_{i-1}$ to RZ$_i$. The first zone starts at 0, while the last zone runs up to infinity.
- NPT is again an integer specifying the required number of points in the outer most zone running up to infinity.

Lebedev published the grids with 38, 50, 86, 110, 146, 194, 266, 302, 434, 590, 770, 974, and 1202 points to be exact for polynomials up to orders 9, 11, 15, 17, 19, 23, 27, 29, 35, 41, 47, 53, and 59 respectively.

Examples

```
LEBEDEV 302

LEBEDEV 194 0.1 302 0.5 434

LEBEDEV LABEL C1 C2 194 0.1 302 0.5 434 ELEMENT CL 590
```

11.7.3 Angular integration Grid – GAUSS-LEGENDRE

The GAUSS-LEGENDRE directive requests a Gauss-Legendre grid for the angular integration. This grid is based on separating functions on a sphere into 2 functions of angles, θ, $(0 \leq \theta \leq$

π) and ϕ, ($0 \leq \phi \leq 2\pi$) respectively. The total grid size is specified through the number of grid points, NTHETA in the θ coordinate. The number of points in the ϕ coordinate will be simply 2 × NTHETA, so that the total angular grid size will be 2 × NTHETA^2. In its simplest form, the directive consists of two data fields read to the variables TEXT, NTHETA using format (A,I)

- TEXT should be set to the character string GAUSS-LEGENDRE, or more simply, GAUSS;
- NTHETA is an integer specifying the required number of points

In the same way as described above for LEBEDEV grids, it is possible to specify different angular grids for different radii. In the present case the GAUSS-LEGENDRE directive comprises the following data fields;

- TEXT should be set to the character string GAUSS-LEGENDRE;
- pairs of data fields are then presented, each pair characterising a specific radial zone and read to the variables (NTHETA$_i$, RZ$_i$) using format (I,F), where
  - NTHETA$_i$ specifies the grid size in the i-th radial zone.
  - The floating point values of RZ$_i$ subdivide the radial coordinate into different zones. The values RZ$_i$ are fractions of the Bragg-Slater radius [46] of the atom. Each zone runs from RZ$_{i-1}$ to RZ$_i$. The first zone starts at 0, while the last zone runs up to infinity.
- NPT is again an integer specifying the required number of points in the outer most zone running up to infinity.

Examples

```
GAUSSLEGENDRE 15

GAUSSLEGENDRE 11 0.1 15 0.5 17

GAUSSLEGENDRE LABEL C1 C2 11 0.1 15 0.5 17 ELEMENT CL 13
```

11.7.4 Radial Integration Grid - EULER-MACLAURIN

The EULER-MACLAURIN directive or shorter EULER selects the Euler-MacLaurin radial integration grid [31]. The grid size is specified through the number of grid points, NPT. The directive thus consists of two data field read to the variables TEXT, NPT using format (A,I)

- TEXT should be set to the character string EULER-MACLAURIN or more simply, EULER;
- NPT is an integer specifying the required number of points

The grid points will be located at

$$\begin{aligned} r_i &= a\frac{x_i^2}{1-x_i^2} \\ x_i &= \frac{i}{\mathrm{NPT}+1} \end{aligned}$$

where $1 \leq i \leq \mathrm{NPT}$. In this expression a is an element dependent scale factor. Note that all points with $i > (\mathrm{NPT}+1)/2$ will have $r_i > a$. Moreover the most distant point will be at $r_i = a \times \mathrm{NPT}^2$. In practice this means that relatively many points will be far from the nucleus.

Examples

```
EULER 45

EULER LABEL H1 H1 45

EULER ELEMENT C H 45 LABEL O1 20
```

11.7.5 Radial Integration Grid - LOG

The LOG directive selects the logarithmic radial integration grid [25]. The grid size is specified through the number of grid points, NPT. In addition a power M must be specified which is defined below. The directive thus consists of three data fields read to the variables TEXT, NPT, M using format (A,I,F)

- TEXT should be set to the character string LOG;
- NPT is an integer specifying the required number of points;
- M is a floating point number defined below

The grid points will be located at

$$\begin{aligned} r_i &= -a\log(1-x_i^{\mathrm{M}}) \\ x_i &= \frac{2i-1}{2\times\mathrm{NPT}} \end{aligned}$$

where $1 \leq i \leq \mathrm{NPT}$, and a is an element dependent scale factor. The recommended value for M is 3.0 [25]. Note that all points with $i > \mathrm{NPT}[1-\exp(-1)]^1/\mathrm{M}+1/2$ will have $r_i > a$. This means that with M $= 3.0$ about 85% of all grid point have $r_i < a$. In practice this means that this radial grid has a tendency to focus on the area close to the atom.

Examples

```
LOG 45 3.0

LOG LABEL H1 H1 45 1.0

LOG ELEMENT C H 45 3.0 LABEL O1 20 3.0
```

11.7.6 Scaling Radial Grids – SCALE

The SCALE directive provides the ability to scale the radial grids of all atoms by a uniform factor, FACTOR. This may prove helpful in moving points into a sensible range, especially with the Euler-MacLaurin radial integration grid [31]. The directive consists of two data field read to the variables TEXT, FACTOR using format (A,F)

- TEXT should be set to the character string SCALE;
- FACTOR specifies the required scaling factor.

Examples

```
SCALE 3.0

SCALE LABEL C1 H1 3.0 LABEL C2 O1 4.0

SCALE 4.0 LABEL C1 H1 3.0

SCALE 4.0 ELEMENT C1 H1 3.0
```

11.7.7 Weighting scheme – WEIGHT

The WEIGHT directive allows the user to select a weighting scheme to combine the atomic integration grids to a molecular integration grid. The directive consists of two data field read to the variables TEXT, SCHEME using format 2A.

- TEXT should be set to the character string WEIGHT
- SCHEME specifies the required weighting scheme, and should be set to one the following character strings;
  - BECKE – The original Becke weighting scheme [33]
  - BECKESCR – The Becke weighting scheme [33] with additional screening.
  - HML – The Murray, Handy and Laming weighting scheme [31]. This scheme differs from the Becke scheme in that it used a different cell function. It leads to more accurate integrals than the Becke scheme.
  - SSF – The Stratmann, Scuseria and Frisch weighting scheme [26]. For sufficiently large quadrature grids this scheme seems to be the most accurate.
  - SSFSCR – The Stratmann, Scuseria and Frisch weighting scheme [26] with screening.
  - MHL4SSF – The Stratmann, Scuseria and Frisch weighting scheme [26] with screening, but employing the cell function by Murray, Handy and Laming weighting scheme [31] with m_μ equals 4.
  - MHL8SSF – The Stratmann, Scuseria and Frisch weighting scheme [26] with screening, but employing the cell function by Murray, Handy and Laming weighting scheme [31] with m_μ equals 8.

The screening referred to reduces the cost of the normalisation of the molecular grid weights. This reduction becomes larger with increasing molecule size.

The weighting scheme is a global option and can not be set on a per atom basis, i.e. the ELEMENT and LABEL keywords can not be used with the WEIGHT directive.

11.7.8 Grid Point Screening – SCREEN

SCREEN [PSI PSITOL] [P DENTOL] [RHO RHOTOL] [CONV]

The SCREEN Directive allows the user to activate the screening of grid points. This may involve the optional specification of a number of tolerances and/or a request to change dynamically the quadrature grid size according to the degree of convergence of the calculation. Following the directive initiator, SCREEN, the following data fields may be presented;

- PSI PSITOL (format A,F) – This criterion is used in generating the radial grids for the atoms. Based on this criterion a radius is computed for every atom beyond which the most diffuse basis function is assumed to be zero. When building the radial grid all grid points that would end up outside this radius will be discarded.
- P DENTOL (format A,F) – The tolerance for the (spin) density matrix elements. If an element in the (spin) density matrix has a value smaller than DENTOL the matrix element will be discarded in the electron density evaluation.
- RHO RHOTOL (format A,F) – The tolerance for the (spin) density in a batch of grid points. If the maximal (spin) density in a batch of grid points is less than RHOTOL, the whole batch will excluded from the functional integration.
- CONV (format A) – This option switches on the dynamic adaption of the quadrature precision with the convergence of the calculation. The idea is that if the Kohn-Sham orbitals are not very precise than there is no reason to integrate the exchange-correlation energy very accurately. Through choosing a smaller quadrature computation can be saved in the early iterations. Along with the calculation converging the quadrature is improved. Near the convergence criterion the full quadrature grid as input will be applied.

Note that DENTOL and RHOTOL are global parameters for which LABEL and ELEMENT can not be used. However PSITOL can be set on a per atom basis.

Examples

```
SCREEN

SCREEN OFF

SCREEN P 1.0D-7 PSI 1.0D-5

SCREEN P 1.0D-7 PSI 1.0D-5 ELEMENT C 1.0D-6 LABEL H1 1.0D-5
```

The use of screening may significantly improve efficiency.

11.7.9 Angular Grid Pruning – ANGPRUNE

This directive activates angular grid pruning as function of radius, uses the scheme proposed by Murray, Handy and Laming [31]. This scheme chooses the number of angular grid points according to the equation

$$\mathrm{n}_{theta} = \min\,(\mathrm{K}_{theta}\ \mathrm{N}_{theta}\ \mathrm{r}/\mathrm{r}_{Bragg},\mathrm{N}_{theta})$$

where

- n_{theta} is the current number of grid points in the theta coordinate,
- N_{theta} is the maximum number of grid points in the theta coordinate,
- K_{theta} is some scaling factor which is set to 5 in as suggested by Murray et al.,
- r_{Bragg} is the Bragg-Slater radius of the atom [46]
- r is the radius of the current angular shell.

From the above equation it is clear that this pruning scheme is designed to be used with the Gauss-Legendre angular grid. When it is applied to the Lebedev grids the total number of grid points is set to $2\mathrm{n}_{theta}{}^{2}$ and then truncated to the first smaller sized Lebedev grid.

Examples

```
ANGPRUNE

ANGPRUNE OFF

ANGPRUNE AUTO

ANGPRUNE LABEL C1 C2 ON

ANGPRUNE LABEL C1 O1 AUTO LABEL C2 H2 OFF
```

11.7.10 Atom Radii – RADII

This directive sets the atomic radii to be used in the grid generation. The radii are to be specified in Bohr's. This is not the same as the SCALE directive which affects only the spacing between the radial grid points. Changing the atomic radii does the same thing as SCALE and affects the pruning of the angular grids and affects the atomic size adjustments in the weighting schemes. The envisaged use of this directive is mainly to change the default grid on a BQ center which matches that of a Carbon atom to one that matches the grid of some other element.

Examples

```
RADII 2.0

RADII LABEL BQ1 BQ2 3.0

RADII ELEMENT C O 1.5 LABEL C2 BQ2 2.0
```

11.7.11 Integration grids and BQ centers

Using the Becke approach for the numerical integration will result in each atom having an associated grid. However in various calculations BQ centers will be included in the geometry to specify point charges, additional basis functions or both. Whether or not a BQ center should have a grid and what the grid parameters should be is not entirely clear yet. The approach currently assumed is:

- A BQ center will be assigned a grid only if it has basis functions associated with it.
- The default grid parameters are chosen to equate those of Carbon atoms according to the current quadrature accuracy setting (low, medium, high veryhigh, or sg1).
- The grid parameters for BQ centers can be changed using the same directives as for atoms. In this context the name BQ can be thought of as a chemical element like C or H.

Although the ELEMENT and LABEL constructs work for BQ centers similar as they do for all normal atoms some care is required. In particular it is not allowed to try and change the grid of a BQ center that does not have one. Therefore the ELEMENT construct will not work if there is at least one BQ center without any basis functions.

11.8 Energy Gradient Evaluation – GRADQUAD

The GRADQUAD directive controls the form of the energy gradient expression in a DFT calculation. This directive consists of 2 data fields read to variables TEXT, TEXTOPT using format (A,A).

- TEXT should be set to the character string GRADQUAD
- TEXTOPT should be set to
  - ON or YES to include the gradients of the quadrature weights and grid points in the energy gradient evaluation,
  - OFF or NO to ignore the contributions from the gradient of the quadrature

For details see for example Johnson et al. [30].

11.9 Coulomb fitting

The cost of DFT calculations of medium sized molecules can be reduced significantly by avoiding the calculation of 4-center 2-electron integrals. This can be achieved by choosing a functional without Hartree-Fock exchange and evaluating the Coulomb energy with an auxiliary basis set. The basic idea behind this technology is described by Dunlap *et al.* [40] and is referred to as "Coulomb fitting". Currently Coulomb fitting can be used in energy and gradient evaluations. The four directives that control this functionality, JFIT, JFITG, JBAS and SCHWARZ, are described below.

11.9.1 JFIT and JFITG

The JFIT directive is used to switch on Coulomb fitting; it is read to the variables TEXT, TEXTA, and IMEM using format (A,A,I).

- TEXT should be set to the character string JFIT
- TEXTA may be set to the character string MEMORY. In this case the program will try to store as many of the required 3-center 2-electron integrals as possible in memory. These integrals will only be calculated once during a KS calculation, while those 3-center integrals that do not fit into memory will be recomputed whenever needed. The program will estimate how much memory is needed for the normal KS operation and set all other memory aside for the 3-center integral storage. Although the implementation aims to avoid user intervention, it may happen that the memory needed for the normal KS activities is underestimated. This results in the calculation aborting with an "out of memory" error. In such cases the User may either, (i) increase the memory available to the calculation (through use of the CORE pre-directive), or (ii) use the IMEM parameter to control memory allocation in more detailed fashion.
- IMEM may be set to a number of words. This should be used only with the MEMORY subdirective and only if the program runs out of memory. With the MEMORY subdirective the program estimates the amount needed for the normal KS operations and reserves all other memory for the 3-center integral storage. If the memory needed for the normal KS operations is underestimated IMEM may be set. IMEM words of memory will then be added to the amount estimated for the normal KS operations. i.e., the 3-center integral storage will be reduced by IMEM words.

To switch on Coulomb fitting for the energy gradient evaluation, use the JFITG directive. This comprises the single character string JFITG.

11.9.2 JBAS

To use Coulomb Fitting requires an auxiliary basis set to be specified for each atom of the molecule. The JBAS directive should be used for this purpose. It consists of 2 data fields, read to variables TEXT, TEXTOPT using format (A,A).

- TEXT should be set to the character string JBAS
- TEXTOPT should be set to one of the following;
  - GAMESS to initiate explicit basis set specification from standard input in GAMESS-UK format (see Part 3 and Example 2 below).
  - NWCHEM to initiate explicit basis set specification from standard input in NWChem format,

- Setting TEXTOPT to one of the strings A1-DGAUSS, A2-DGAUSS, DEMON or AHLRICHS signals that the fitting basis set is not to be defined in the input stream, but is to be loaded from the appropriate library of internal basis sets. A1-DGAUSS or A2-DGAUSS result in the A1 or A2 DGauss fitting sets [41], DEMON the DeMon fitting basis [41], and AHLRICHS the fitted basis sets tabulated by Ahlrichs and co-workers [42] (see Example 1 below).

TEXTOPT may be omitted in which case it is assumed that the basis set is to be specified in GAMESS-UK format.

11.9.3 SCHWARZ

Finally, to reduce the number of 3-center integrals, small terms may be eliminated using the Schwarz inequality. The SCHWARZ directive which is read to the variables TEXT, ISCHWARZ using the format (A,I) can be used to set the tolerance.

- TEXT should be set to the character string SCHWARZ.
- ISCHWARZ should be set to an integer value. The Schwarz tolerance will then be set to $10^{-ISCHWARZ}$.

Example 1: Coulomb Fitting Using the A1-DGAUSS Fitting Basis

```
TITLE
H2CO - 6-31G/BLYP-DFT WITH A1-DGAUSS COULOMB FITTING
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 121.8
H 1 CH 2 121.8 3 180.0
VARIABLES
CO 1.203
CH 1.099
END
BASIS 6-31G
RUNTYPE OPTIMISE
SCFTYPE DIRECT RHF
DFT BLYP
DFT JFIT MEMORY
DFT SCHWARZ 6
DFT JBAS A1-DGAUSS
ENTER
```

Example 2: Coulomb Fitting with Explicit Specification of the A1-DGAUSS Basis

```
TITLE
H2CO - 6-31G/BLYP-DFT WITH A1-DGAUSS COULOMB FITTING
ZMATRIX ANGSTROM
```

```
C
O 1 CO
H 1 CH 2 121.8
H 1 CH 2 121.8 3 180.0
VARIABLES
CO 1.203
CH 1.099
END
BASIS 6-31G
RUNTYPE OPTIMISE
SCFTYPE DIRECT RHF
DFT BLYP
DFT JFIT MEMORY
DFT SCHWARZ 6
DFT JBAS
#
# DGauss A1 Coulomb fitting basis (gamess basis set format)
#
S H
1.000000   45.000000000
S H
1.000000    7.500000000
S H
1.000000    1.500000000
S H
1.000000    0.300000000
S C
1.000000 1114.000000000
S C
1.000000  223.000000000
S C
1.000000   55.720000000
S C
1.000000   13.900000000
SP C
1.000000    4.400000000   1.00000000
SP C
1.000000    0.870000000   1.00000000
SP C
1.000000    0.220000000   1.00000000
D C
1.000000    4.400000000
D C
1.000000    0.870000000
D C
1.000000    0.220000000
S O
1.000000 2000.000000000
S O
1.000000  400.000000000
S O
1.000000  100.000000000
S O
1.000000   25.000000000
SP O
```

```
1.000000    7.800000000   1.00000000
SP O
1.000000    1.560000000   1.00000000
SP O
1.000000    0.390000000   1.00000000
D O
1.000000    7.800000000
D O
1.000000    1.560000000
D O
1.000000    0.390000000
END
ENTER
```

12 Controlling the input Orbitals: The VECTORS Directive

Each of the SCF modules requires one or more sets of trial molecular orbitals or eigenvectors to initiate the iterative process. The analysis routines also require definition of the input set of orbitals to be analysed. In both cases the origin of such a set is defined under control of the VECTORS directive. The syntax and usage of the directive is very much a function of the status of the computation in hand. We may identify two differing situations, where the User must either,

- define a mechanism for generating the orbitals, or rely on the default mechanism (ATOMS, see below);
- nominate a *Section* on either the parent Dumpfile, or some 'foreign' Dumpfile, wherein a suitable set of orbitals may be found. In contrast to previous versions of GAMESS-UK, which required explicit specification of these section numbers, the current release provides a set of default values so that the user may avoid the task of nominating sections. These defaults, which are a function of SCFTYPE, are summarised in Table 2.

12.1 Mechanism Specification

At the outset of a calculation, with no associated SCF computation available, the User must either define a mechanism for generating the trial MOs though keyword specification on the VECTORS line, or rely on the default mechanism (note that the mechanism chosen is often a function of basis set). If presented for such usage, the VECTORS directive comprises a single data line read to the variables TEXT, ATEXT, BTEXT using format (3A).

- TEXT should be set to the character string VECTORS
- ATEXT should be set to the appropriate string (see below) defining the generate mechanism to be employed.

- BTEXT is an optional string that may be set to the character string PRINT when the trial vectors will be printed. If BTEXT is omitted, no vectors will be sent to the printer

Valid ATEXT strings include the following:

- **VECTORS ATOMS** : Construct an initial starting guess based on concatenating the 1-particle density matrices for each of the component atoms of the molecular system. The present implementation of ATOMS represents a significant improvement over that available in previous releases of the code, and should normally be the option of choice. It is now the default option in the absence of the VECTORS directive.

  The following extra optims are recognised in addition to a **print** directive :

  – ALWAYS : The atomic startup is used again in every point in a geometry optimisation to determine the start-orbitals instead of the orbitals of the previous point.

  – GROUND : The atomscf routines try to use the groundstate of the atoms; normally an average of their lowest states is used, which is usually quite adequate.

  – UHF : The atomic startup produces alpha and beta density matrices, which are used straightaway in the UHF scf. Therefore :

    ∗ This option only makes sense if the two density matrices differ, which can be effected, using the CONFIGURATION or SPECIFY subdirective.

    ∗ As the density matrices are not idempotent, the integrated density will not be correct; Therefore an ACCURACY IGNORE should be given in DFT calculations.

  – CONFIGURATION or SPECIFY: Allows the user to specify explicitly the configuration used for certain atoms; this is useful for pseudopotentials, when the user supplies the pseudopotentials under control of the CARDS option or if the user does not like the configuration chosen by the atomscf program. The configuration(s) are specified for each atom on subsequent lines following the VECTORS directive, with the data terminated by END : **ATOM subdirectives** :

    ∗ CONF : specify the required configuration to be used in the atomic scf for the ATOM as (e.g.) d5s1. It is worthwhile checking that the required effect is obtained.

    ∗ DCONF or DENS : specify the configuration to be used in calculating the final atomic density matrix.

    ∗ CHARGE (A,F) : specify the charge for the ATOM. The extra charge is preferably added to the highest open shell. Otherwise it is spread over the closed shells.
      Example:

      ```
      Na charge +1.0
      CL charge -1.0
      ```

    ∗ SPIN (A,A,I,I) Specify the division of the electrons over the alpha and beta density matrices for the specified (s,p,d,f) shell. This directive is only allowed (and then really required) if UHF is specified on the VECTORS card; More spin directives may given to specify the division for different shells. This directive overrides specification by the DENS or CHARGE sub-directives. **Example:**

```
Fe1  CONF d5s1 DENS d5s1 SPIN d 5 0
Fe2  CONF d5s1 DENS d5s1 SPIN d 0 5
```

- ∗ NORE or NONREL Specifies that this atom is to be treated non-relativistic in an ATOMIC ZORA calculation.
- ∗ FORCE Normally the changes specified in these subdirectives are only used to generate a start and are not included when calculating the atomic ZORA corrections. Specifying FORCE will cause the program to use them always. So one might have ZORA corrections for a charged atom.

Example: The following specifies explicitly the configuration for the iron atoms in Fe_2O_3, where the two iron d-shells are either completely alpha or beta and the s open shell is divided. For fun the oxygens are made slightly negative.

```
VECTORS ATOMS CONF UHF
Fe1  CONF d5s1 DENS d5s1 SPIN d 5 0
Fe2  CONF d5s1 DENS d5s1 SPIN d 0 5
O      CHARGE -0.6
END
```

- **VECTORS ATORBS** : Construct starting orbitals as a concatenation of the atomic orbitals of the atoms (not generally recommended). This is useful however for subsequent VB calculations and for calculations on atoms. The SECTION keyword is recognised as input on the same line followed by the section number. If given, the atomic orbitals are written to the section specified and the density matrices are used as in the ATOMS option.

- **VECTORS HCORE** : Diagonalise the 1-electron (core) Hamiltonian. This is the most general mechanism available, but is also the least reliable in that the resulting MOs may often not exhibit the required ordering. We return to this point below.

- **VECTORS MINGUESS** : Construct and diagonalise a Huckel type matrix. This option is limited to minimal basis sets (e.g., BASIS STO3G), but in such cases often leads to a reliable set of MOs. Note that the original MINGUESS implementation has been extended to handle nuclei up to and including Xenon.

- **VECTORS EXTGUESS** : Limited to split-valence basis sets (e.g., 3-21G, 4-31G etc.), leading in general to a reliable set of orbitals. Note that the original EXTGUESS implementation has been extended to handle polarisation basis sets such as 4-31G*, 4-31G** etc. and can now handle nuclei up to and including Xenon.

- **VECTORS ALPHAS** : Trial vectors are generated from a Fock matrix based on a Mulliken-type approximation together with a set of input diagonal Fock elements. Following the VECTORS line, a sequence of NBASIS real numbers (where NBASIS is the number of basis functions) must be input, with the I'th such number set to the negative of the expected value of the I'th diagonal Fock matrix (in the basis representation)

  Example the following ALPHAS data refers to a triple-zeta (TZV) calculation on formaldehyde :

```
ALPHAS
8.9 11.6 5.5 2.7 1.2
```

```
-0.2 0.0 0.2  0.1 0.4 0.8  0.0 0.1 0.4
15.5 15.1 6.6 3.6 1.7
-0.3 -0.1 -0.1  0.0 0.1 0.5  0.0 0.0 0.5
-0.5 0.5 0.5   -0.5 0.5 0.5
```

Having generated the trial MOs (and possibly manipulated them under control of the SWAP directive), the resulting set of vectors is written to the Dumpfile Section(s) nominated on the ENTER directive. Thus a typical data sequence in a closed-shell SCF calculation would be

```
VECTORS ATOMS
ENTER 10
```

or just

```
ENTER 10
```

where the vectors generated by concatenating the atomic SCF densities are written to Section 10 of the Dumpfile. Any subsequent SCF undertaken during the job will refresh this set of vectors.

- **VECTORS NOGEN section** : The NOGEN option is specific to generating a trial set of GVB orbitals, with **section** an integer referencing a section on the Dumpfile where a suitable starting set of either SCF- or localised-MOs may be found. For each GVB pair, two trial orbitals are required, the strongly occupied MO (corresponding to an SCF-occupied MO) and a weakly occupied orbital. The NOGEN facility will generate such weakly occupied orbitals from the strongly occupied counterparts. The user must ensure that, given an n-pair GVB treatment, the top n orbitals from **section** correspond to the strongly occupied MOs of each pair. This might typically be achieved under control of the SWAP directive.

12.2 Section specification from the ‘Parent’ Dumpfile

The above discussion leads in obvious fashion to the second method of specifying the trial orbitals, namely by nominating a Section on the Dumpfile wherein such orbitals may be found, the orbitals having been written to that Section by some preceding run of the program under control of the ENTER directive. In contrast to previous versions of GAMESS-UK, which required explicit specification of these section numbers, the current release provides a set of default values so that the user may avoid the task of nominating sections. These defaults, which are a function of SCFTYPE, are summarised in Table 2.

Clearly a discussion of the VECTORS directive, where we define Sections containing input MOs, should now be linked to that of the ENTER directive, where Sections for orbital output are nominated. We also consider Section specification as a function of SCFTYPE in the notes below, focusing attention initially on the use of orbitals resident on the ‘parent’ Dumpfile.

Table 2: Default Vector Sections as a function of SCFTYPE

SCFTYPE	Number of	Default	
	Sections	Section	Numbers
Closed-shell SCF	1	1	
UHF	2	2	3
Open-shell RHF	2	4	5
GVB	2	4	5
CASSCF	2	6	7
MCSCF	2	8	9

1. *Closed Shell SCF* : Here we are only involved in nominating a single Section on the Dumpfile to contain the closed-shell SCF eigenvectors. Thus given the data sequence

   ```
   VECTORS ATOMS
   ENTER 1
   ```

   or just ENTER 1 in a startup-job, the sequence

   ```
   VECTORS 1
   ENTER 1
   ```

   would be specified in a subsequent SCF restart, in which case the MOs in Section 1 would be updated, or perhaps the sequence

   ```
   VECTORS 1
   ENTER 10
   ```

   if the User wished to keep copies of both initial and final MOs.

2. *UHF SCF* : Now two Sections are required, the first referring to the α-spin orbitals, the second to the β-spin orbitals. When initiating a UHF calculation, the data sequence

   ```
   VECTORS 1
   ENTER 2 3
   ```

   is permitted, indicating that both α- and β-spin trial MOs are to be taken from the same Section. Restarting the above computation would typically involve the sequence

   ```
   VECTORS 2 3
   ENTER 2 3
   ```

3. *Open-shell RHF and GVB Calculations* : Again two Sections are involved. The first is used to hold the 'internal' non-canonicalised MOs, the orbital set used during the RHF or GVB iterations. The second Section is used for output of the 'external' canonicalised orbitals, with energy weighting in the virtual manifold. Again, given a set of trial MOs in Section 1, we typically instigate the RHF/GVB calculation with the data sequence

```
VECTORS 1
ENTER 4 5
```

and continue the processing with the sequence

```
VECTORS 4 5
ENTER 4 5
```

4. *CASSCF Calculations* : Again two Sections are required, with both Sections containing information crucial to the internal running of the CASSCF module. The first Section is assumed to contain just the orbital set, while the second contains the canonicalised MOs (relevant to subsequent CI studies), *plus* a set of CI coefficients which may be used in assisting CASSCF restarts. Given a trial set of MOs in Section 1, we would typically initiate the CASSCF calculation with the sequence

   ```
   VECTORS 1
   ENTER 6 7
   ```

   On completion of the job, Section 6 will contain the CASSCF MOs, while Section 7 will hold canonicalised orbitals *plus* the current set of CI coefficients. In a subsequent restart we would specify

   ```
   VECTORS 6 7
   ENTER 6 7
   ```

   when the orbitals will be read from Section 6, and the CI coefficients from Section 7. Note that the program assumes, given two Sections on the VECTORS line, that the second may be used as a source of CI coefficients - if that Sections contains no such data, or coefficients from some different CASSCF calculation with a different CI space), an error condition will result.

There are times when it is useful to see the starting vectors, in which case the character string PRINT may be specified on the VECTORS directive line. eg. VECTORS ATOMS PRINT If the vectors are read from a section, one may forego the orthogonalisation of these vectors by specifying the keyword NOORTH e.g., VECTORS 19 NOORTH. This may be used for example in calculating coulomb energies, where the (non)-orthogonal vectors are combined by the SERVEC subprogram.

12.3 Using Default Sections under VECTORS and ENTER

In all the examples above, we have assumed that the User is explicitly defining the sections of the Dumpfile to be used for vector retrieval (under control of the VECTORS directive) and vector storage (under control of the ENTER directive). While this mechanism provides an additional degree of user control, it is now possible to use a set of default sections that avoids the need for explicit specification on both VECTORS and ENTER directives, and in most cases removes the need for presenting the VECTORS directive altogether. The following points should be noted regarding use of these defaults;

- This default usage is not designed to completely remove the need for section specification, and is intended primarily to cover simple operations e.g. a simple SCF or geometry optimisation.

- While an expanded summary of section usage is now routinely printed on job termination, the user should be aware of the attributes of the various vector sections before mixing default and input-driven section specification.

- the contents of each of the sections specified in Table 2 have been described in some detail in Part 2 of the manual. With the exception of "RUNTYPE ANALYSE", the choice of default section(s) is determined solely by the SCFTYPE that has been requested. Given this, the decision on the choice of input eigenvectors is made in three stages. Thus when deciding on the section(s) to be used for these orbitals (i.e the VECTORS section(s)), the default sections appropriate to the nominated SCFTYPE will be examined first. If found to exist, and in the absence of explicit section specification on the VECTORS directive, the input eigenvectors will be taken from these default section(s). If these default sections have not been written to in some previous job or job step, then the default closed-shell eigenvector section (section 1) will be examined, and the input eigenvectors taken from this section assuming again that this section has been written to previously. If the closed-shell vectors section does not exist, then the eigenvectors will be generated from an atomic-guess. The choice of section number(s) for the output of eigenvectors (i.e. those sections typically nominated on the ENTER directive) is taken directly from Table 2; clearly this can lead to the final vectors over-writing the input eigenvectors. We illustrate these choices by considering a number of cases below as a function of SCFTYPE.

    1. *Closed Shell SCF* : Here we are only involved in considering a single Section on the Dumpfile that contains the closed-shell SCF eigenvectors. Thus presenting the single data line

        ```
        ENTER
        ```

        in a startup-job would act to request an atomic GUESS for generating the initial SCF eigen vectors, that would be stored, and subsequently updated and written to section 1 of the Dumpfile (see Table 2) during the SCF process. Presenting the same data line in a RESTART job would result in the eigenvectors of section 1 being used as the trial vector set, and subsequently updated and overwritten during the SCF process. In practice this default section would be examined for content at job outset, and the contents used for the SCF process assuming the section had been written to by a previous job. If the section is not present on the Dumpfile, then a trial set will be generated using the "VECTORS ATOMS" mechanism.

        If the User wished to keep copies of both initial and final MOs, then the single data line of the form

        ```
        ENTER 10
        ```

        would be required, with the final set of MOS being written to section 10. Note that usage of this set in some subsequent job would require explicit introduction of the data line "VECTORS 10" to avoid use of the default section.

2. *UHF SCF* : Two sets of eigenvectors are generated in an open-shell unrestricted Hartree Fock (UHF) calculation, the α-spin SCF MOs and β-spin orbitals. In default the α-spin MOs will be written to section 2 of the Dumpfile, and the β-spin MOs to section 3. (see Table 2). Thus presenting the single data line

   ```
   ENTER
   ```

   in a startup-job would act to request an atomic GUESS for generating a set of SCF eigen vectors, that would be be used to initiate the UHF process, with the α- and β-spin MOs subsequently updated and written to sections 2 and 3 of the Dumpfile (see Table 2) during the SCF process.

   Presenting the same data line in a subsequent RESTART job would result in the eigenvectors of section 2 and 3 being used as the trial vector set, and subsequently updated during the UHF process. In practice these default sections would be examined for content at job outset, and the contents used for the SCF process assuming the sections had been written to by a previous job. If the sections are not present on the Dumpfile, then either (i) the closed shell MOS (if present) in section 1 will be used as the trial orbitals for both α- and β-spin, or (ii) with no closed shell section, a trial set will be generated using the "VECTORS ATOMS" mechanism.

   If the User wished to keep copies of both initial and final UHF MOs, then the single data line of the form

   ```
   ENTER 10 11
   ```

   would be required, with the final set of α-spin MOs MOS being written to section 10 and β-spin orbitals to section 11. Note that usage of this set in some subsequent job would require explicit introduction of the data line "VECTORS 10 11" to avoid use of the default section.

3. *Open-shell RHF and GVB Calculations* : Again two Sections are involved, the first holding the 'internal' non-canonicalised MOs, the orbital set used during the RHF or GVB iterations, while the second is used for output of the 'external' canonicalised orbitals on termination of the SCF process. In default the 'internal' MO set will be written to section 4 and the canonicalised orbital set to section 5 of the Dumpfile. (see Table 2). Thus presenting the single data line,

   ```
   ENTER
   ```

   in a startup-job would act to request an atomic GUESS for generating a set of SCF eigen vectors, that would be be used to initiate the open-shell SCF, with the 'internal' non-canonicalised MOs subsequently updated and written to section 4 of the Dumpfile and the canonicalised orbital set written to section 5.

   Presenting the same data line in a subsequent RESTART job would result in the eigenvectors of section 4 and 5 being used as the trial vector set, and subsequently updated and overwritten during the open-shell SCF. In practice these default sections would be examined for content at job outset, and the contents used for the SCF process assuming the sections had been written to by a previous job. If the sections are not present on the Dumpfile, then either (i) the closed shell MOS (if present) in section 1 will be used as the trial orbitals, (ii) with no closed shell section, a trial set will be generated using the "VECTORS ATOMS" mechanism. If the User wished to keep copies of both sets of initial and final open-shell MOS, then a single data line of the form

```
ENTER 10 11
```

would be required, with the final set of 'internal' MOS being written to section 10 and the canonicalised orbitals to section 11. Note that usage of this set in some subsequent job would require explicit introduction of the data line "VECTORS 10 11" to avoid use of the default section.

4. *CASSCF Calculations* : Again, two Sections are required; the first housing the non-canonicalised CASSCF MOs that are used during the CASSCF process, and the second set, the canonicalised vectors that are generated on termination of the CASSCF process. In default the non-canonicalised vectors will be written to section 6 of the Dumpfile, while the canonicalised vectors will be written to section 7 (see Table 2). Note that the latter section also contains the current CI coefficients. Presenting the single data line,

   ```
   ENTER
   ```

   in a startup-job would act to request an atomic GUESS for generating a set of eigen vectors, that would be be used to initiate the CASSCF process, with the non-canonicalised CASSCF MOs subsequently written to section 6 of the Dumpfile and the final canonicalised orbital set written to section 7.

   Presenting the same data line in a subsequent RESTART job would result in the eigenvectors of section 6 and CI coefficients of section 7 being used to initiate the CASSCF process, each being subsequently updated and overwritten. In practice these default sections would be examined for content at job outset, and the contents used for the CASSCF process assuming the sections had been written to by a previous job. If the sections are not present on the Dumpfile, then either (i) the closed shell MOS (if present) in section 1 will be used as the trial orbitals, (ii) with no closed shell section, a trial set will be generated using the "VECTORS ATOMS" mechanism.

5. *Wavefunction Analysis* : The variety of wavefunction analysis and property modules of GAMESS–UK (see Part9) also rely on the VECTORS directive to define the section number of the eigenvectors to be analysed. Note that in contrast to SCF processing, *no default specification* is available under "RUNTYPE ANALYSE". Explicit specification is required even if this section had been written to using one of the default options during, for example, previous SCF processing.

12.4 Specification from a 'Foreign' Dumpfile : GETQ

The GETQ option causes restoration of eigenvectors from a 'foreign' Dumpfile. Characterisation of such a Dumpfile is achieved by specification, through data input, of

- the LFN of the data set on which this Dumpfile resides. The foreign Dumpfile may reside on a direct data set assigned using a DDNAME in the range ED0 to ED19, or on a sequential data set assigned using a DDNAME in the range MT0 to MT19;
- the starting block of the Dumpfile;
- the Section number wherein the required vectors are to be found, this integer having been that associated with the ENTER directive (whether by default or by explicit specification) of the run which created the trial vectors.

A given set of eigenvectors is deemed to represent valid input to GETQ if it has the following attributes;

1. It is derived from a previous calculation performed at either the same or at a different geometry.
2. It is derived from a previous calculation where the symmetry adapted option is different
3. It is derived from a previous calculation conducted in a *smaller* basis set. The program maps the 'old' basis onto the 'new' using standard projection techniques, and it is now routine practice to use, for example, the vectors generated in a SV 3-21G basis to initiate an SCF calculation in a triple zeta + polarisation basis set. Again the symmetry adapted option may be different in the 'old' and 'new' calculation.
4. The role of the EXTRA option (see ATMOL3 9) is largely redundant given the availability of the revised GETQ, and it is suggested that GETQ be used in such cases.
5. The only major restriction in the use of GETQ is that the ordering of the nuclei presented in the z-matrix definition lines be the same in both 'old' and 'new' calculation.

The syntax of the GETQ option is again dependent on the number of sets of vectors to be retrieved from 'foreign' Dumpfiles.

1. When restoring a single set of vectors, as in closed and open-shell restricted Hartree Fock calculations, the VECTORS directive consists of a single data line, as follows

   VECTORS GETQ *lfnvec iblkv isectv*

   where,

   - *lfnvec* : should be set to a valid character string (ED0-ED19, MT0-MT19) specifying the LFN of the data set on which the 'foreign' Dumpfile resides;
   - *iblkv* : should be set to the starting block of the 'foreign' Dumpfile;
   - *isectv* is an integer used to specify the section where the required vectors are to be found on the 'foreign' Dumpfile. Note that *isectv* must be specified even if referring to the appropriate default section.

2. When restoring two sets of eigenvectors from a 'foreign' Dumpfile, for example the α-spin and β-spin vectors in a UHF calculation, then the VECTORS directive is a straightforward extension of the single-vector case, with the location of each set specified. Thus the data line is as follows

   VECTORS GETQ *lfnveca iblka isecta lfnvecb iblkb isectb*

   where,

   - *lfnveca*, *iblka* and *isecta* : The trial α-spin vectors are to be restored from the Section specified by *isecta* on a 'foreign' Dumpfile commencing at the block specified by *iblka*, with the Dumpfile residing on the data set assigned using the LFN *lfnveca* .
   - *lfnvecb*, *iblkb* and *isectb* : The trial β-spin vectors are to be restored from *isectb* on a foreign Dumpfile residing on a data set assigned using the LFN *lfnvecb*, commencing at block *iblkb*.

   Note that

   - the trial β-spin orbitals will be set equal to the trial α-spin orbitals if *lfnvecb*, *iblkb* and *isectb* are omitted from the data line.
   - there is currently no provision for restoring the CI Vector Section in a CASSCF calculation under control of the GETQ option. In such cases it is only possible to restore a single set of vectors, the CASSCF orbitals.

Example 1

A geometry optimisation has been performed with a closed shell SCF calculation in a 3-21G basis set. A single point calculation in a TZVP basis set is to be performed at the optimised geometry. Assuming that the Dumpfile from the optimisation run is located on a data set assigned to the present calculation as ED4, starting at block 1, and that the converged 3-21G vectors are stored in Section 2 of this Dumpfile, then these vectors will provide a satisfactory starting point for the TZVP study, and may be restored using a directive of the form

```
VECTORS GETQ ED4 1 2
```

where the data set used to hold the 'foreign' Dumpfile is assigned to the program using the LFN ED4.

Example 2

The Dumpfile for the TZVP calculation above is to be located on the *same* data set used for performing the the 3-21G calculation. Assume that on completion of the original optimisation the total length of the Dumpfile was found to be smaller than 250 blocks. Siting the TZVP calculation at block 250 on this data set, by means of the directive

```
DUMPFILE ED3 250
```

permits use of the following GETQ directive

```
VECTORS GETQ ED3 1 2
```

in making the 3-21G vectors available to the present calculation.

Example 3

A UHF calculation has been performed in a 3-21G basis, with the converged α- and β-spin orbitals stored in Sections 8 and 9 of the Dumpfile which had started at block 50. We now wish to perform the same calculation using a TZV basis set. Assigning the data set containing this Dumpfile from the previous calculation as ED4, we may commence the TZV UHF calculation by means of a directive of the form

```
VECTORS GETQ ED4 50 8 ED4 50 9
```

12.5 LOCK

This directive consists of one data line, with the character string LOCK in the first data field. In the presence of the LOCK directive, the program will seek a stationary value for the functional, and proceed in the direction of 'minimum change'. The use of the LOCK directive is automatic in open shell, GVB and CASSCF/MCSCF calculations.

12.6 SWAP

The first data line should contain the character string SWAP in the first data field. Subsequent data lines are read to variables I,J using format (2I). The effect is that the I'th and J'th molecular orbitals, as generated by the VECTORS directive, are interchanged. When all the interchanging lines have been presented, the directive should be terminated by a data line containing the character string END in the first data field. The following notes may prove helpful:

- The SWAP directive is normally used to switch from a configuration known not to be the ground state into the ground state.
- upon completion of the SWAP directive, revised molecular orbital lists are held in memory, but not in the Dumpfile.

Example

```
SWAP
10 12
11 13
END
```

Molecular orbitals 10 and 11 are interchanged with molecular orbitals 12 and 13 respectively.

In the case of UHF calculations, the syntax of this directive is modified to reflect the presence of both α and β orbitals. Now the directive initiator is read to variables TEXT,SPIN using format (2A).

- TEXT should be set to character string SWAP.
- SPIN should be set to one of the character strings ALPHA or BETA, and will cause swapping of either α or β spin vectors respectively. The SPIN parameter may be omitted (thus the directive is the same as that above), and will cause α-spin vectors to be interchanged.

13 Controlling Geometry and Transition State Optimization

There are a variety of methods available for controlling the search for a stationary point on a potential energy surface. Each may be requested through appropriate keyword specification on the RUNTYPE directive, and in the following section we detail subsequent data requirements of each method. In most cases adequate control of the optimisation pathways is provided through a set of 'built-in' parameters, and the user need only consider overriding these defaults in troublesome cases, through use of the directives described below. An introduction to the various methods and their usage has already been presented in Part 2, and should be used in conjunction with the notes below.

13.1 Geometry Optimisation and RUNTYPE Specification

Three methods are available to search for a minimum on a potential Surface,

1. the recommended method, a quasi-Newton rank–2 update procedure, is driven through the specification

   ```
   RUNTYPE OPTIMIZE
   ```

   This method performs optimisation in internal coordinates, and thus requires initial ZMATRIX and VARIABLES specification of the molecular geometry, or ZMATRIX construction from a set of cartesian coordinates supplied under control of the GEOMETRY directive.

2. the second internal coordinate–driven method is that based on the hill-walking algorithm due to Simons and Jorgensen [6]. While primarily intended for transition state usage, it may also be employed in geometry optimisation. A more detailed account of the method and associated data is given below in section 10. We note here that the procedure is driven through additional keyword specification on the RUNTYPE directive, thus;

   ```
   RUNTYPE OPTIMIZE JORGENSEN
   ```

3. the third method, perhaps less robust and flexible than the others, is a cartesian-driven update method. This is requested through the following RUNTYPE specification,

```
                RUNTYPE OPTXYZ
```

We shall use the RUNTYPE keywords – OPTIMIZE, JORGENSEN and OPTXYZ – to subsequently refer to the the three methods.

13.2 Stopping an optimisation when molecule dissociates - CHECK, DISS or DIST

Appending the keywords "CHECK", "DISS" or "DIST" in A format to any of the following runtypes:

- OPTIMIZE
- SADDLE
- OPTX

Will cause a check to be carried out as to whether the number of bonds within the molecule changes (defined as the distance between two bonded atoms changing by more than 15% over the course of the optimisation). If the number of bonds changes, then the optimisation will abort. This can often be useful to halt a geometry optimisation when a molecule appears to be dissociating.

13.3 OPTIMIZE Data Specification

Four directives are provided to control the OPTIMIZE search procedure, MINMAX, XTOL, STEPMAX and VALUE. The user should note that the present implementation is based on maintaining a history of the optimisation pathway, that will be worked through on each restart of the optimisation. This appears on the output as a sequence of both 'old' and 'new' calculations, with the history printed on each restart. This printing may be suppressed through use of the NOPRINT directive, with specification of the HISTORY keyword.

13.3.1 OPTIMIZE Data - MINMAX

This directive may be used to control the number of energy evaluations and line-searches permitted in optimising a given structure, and consists of a single data line read to the variables TEXT, IVAL, LINE using format (A,2I).

- TEXT is set to the character string MINMAX.
- IVAL is an integer specifying the maximum number of energy evaluations allowed in the optimisation.

- LINE is an integer defining the maximum number of line searches permitted in the optimisation.

The MINMAX directive may be omitted, when both IVAL and LINE will be set to the maximum allowed value of 60. The following specification is thus equivalent to the default;

```
MINMAX 60 60
```

Example

In some cases the user may wish to perform just the initial point on the optimisation pathway to gauge the quality of the starting geometry though the magnitude of the gradient at that point. This may be achieved though use of MINMAX, as shown below:

Start-up Job

```
TITLE\H2O DZ OPTIMIZATION STARTUP
ZMAT ANGSTROM\O\H 1 ROH\H 1 ROH 2 THETA
VARIABLES\ROH 0.956 HESSIAN 0.7\THETA 104.5 HESSIAN 0.2 \END
BASIS DZ
RUNTYPE OPTIMIZE\MINMAX 1 1
ENTER
```

Here we are using the MINMAX directive to terminate the optimisation after the first point. This may then be restarted as shown below, where the default MINMAX settings will apply.

Restart Job

```
RESTART OPTIMIZE
TITLE\H2O DZ OPTIMIZE
ZMAT ANGSTROM\O\H 1 ROH\H 1 ROH 2 THETA
VARIABLES\ROH 0.956 HESSIAN 0.7\THETA 104.5 HESSIAN 0.2 \END
BASIS DZ
RUNTYPE OPTIMIZ\ENTER
```

13.3.2 OPTIMIZE Data - XTOL

This directive may be used to define the convergence thresholds for the optimisation, and consists of a single data line read to the variables TEXT, TOL using format (A,F):

- TEXT should be set to the character string XTOL;
- TOL should be set to the value to be used in defining the four acceptance criteria for the convergence of the optimisation algorithm. These criteria are:

```
maximum change in variables  <  TOL
average change in variables  <  TOL * 2/3
maximum gradient             <  TOL * 1/4
average gradient             <  TOL * 1/6
```

The XTOL directive may be omitted, when TOL will be set to 0.003. The default thus corresponds to presenting the data line

```
XTOL 0.003
```

13.3.3 OPTIMIZE Data - STEPMAX

This directive may be used to define the the maximum allowed movement in any of the variables in a single step of the geometry optimisation. Note that the internal units of the variables are bohr for bond lengths and radians for angles. The directive consists of a single data line read to the variables TEXT, STEP using format (A,F):

- TEXT should be set to the character string STEPMAX;
- STEP should be set to the maximum permitted movement in any variable.

The STEPMAX directive may be omitted, when STEP will be set to 0.2. The default thus corresponds to presenting the data line

```
STEPMAX 0.2
```

There is certainly at least one circumstance where changes to the default setting will prove crucial in achieving controlled convergence. If the starting geometry is known to be poor, or if ZMATRIX specification is such that a specific bond is not explicitly defined (as can happen for example with aromatic compounds), then the first step taken on the optimisation can cause the energy to go *up* and, at best, several extra points will be required to recover from this effect. This effect is fairly common if in addition the starting hessian is also poorly defined. When this happens, the user should consider starting the optimisation again, presenting a STEPMAX directive of the form:

```
STEPMAX 0.1
```

An additional side effect of excessive steps in the optimisation is a possible change of state in the SCF calculation, particularly in closed–shell wavefunctions. If this happens, the subsequent optimisation will almost certainly prove meaningless. Presenting the LOCK directive in the closed–shell case may act to minimise this occurrence.

13.3.4 OPTIMIZE Data - VALUE

This directive may be used to control the accuracy of the search for a turning point during a line search, and consists of a single data line read to the variables TEXT, TURN using format (A,F);

- TEXT should be set to the character string VALUE;

- TURN should be set to a value between 0.0 and 1.0 that will control the accuracy of the line search procedure. Note that the smaller TURN, the more accurate the line search.

The VALUE directive may be omitted, when TURN will be set to 0.6. The default thus corresponds to presenting the data line

```
VALUE 0.6
```

13.4 Modifying the Optimisation Pathway

In some cases the user may wish to modify the parameters controlling geometry optimisation, though XTOL, VALUE and STEPMAX specification, during the course of the optimisation. Note that these parameters may only be modified between line searches, and not between individual energy or gradient evaluations. The most straightforward example would be initialising the optimisation with stringent controls, then relaxing these controls as the optimisation proceeds in some subsequent restart job. The startup job may either be interrupted under control of the MINMAX directive, or through time specification on the TIME pre-directive (see Parts 12-16 of the manual).

Consider the example of Part 3 section 8.4 on the optimisation of HCN. The following data file requests termination after two line searches through the MINMAX specification, during which conservative settings of the optimisation parameters will apply:

```
TITLE
HCN  DUNNING DZ + BOND(S,P)
ZMAT ANGSTROM
C
BQ 1 RCN2
X 2 1.0 1 90.0
N 2 RCN2 3 90.0 1 180.0
X 1 1.0 2 90.0 3 0.0
H 1 RCH 5 90.0 4 180.0
VARIABLES
RCN2 0.580
RCH 1.056
END
BASIS
DZ H
S BQ
1.0 1.0
P BQ
1.0 0.7
DZ C
DZ N
END
RUNTYPE OPTIMIZE
MINMAX 60 2
XTOL 0.005
STEPMAX 0.1
VALUE 0.3
ENTER
```

In the restart job shown below, the modified parameter settings will apply from the third line search onwards.

```
RESTART OPTIMIZE
TITLE
HCN  DUNNING DZ + BOND(S,P)
ZMAT ANGSTROM
C
BQ 1 RCN2
X 2 1.0 1 90.0
N 2 RCN2 3 90.0 1 180.0
X 1 1.0 2 90.0 3 0.0
H 1 RCH 5 90.0 4 180.0
VARIABLES
RCN2 0.580
RCH 1.056
END
BASIS
DZ H
S BQ
1.0 1.0
P BQ
1.0 0.7
DZ C
DZ N
END
RUNTYPE OPTIMIZE
XTOL 0.0005
STEPMAX 0.2
VALUE 0.6
ENTER
```

A somewhat more complex situation may arise when the user wishes to modify optimisation processing already performed in the startup job, either because

- problems have been encountered that can only be remedied by modifying the optimisation control parameters, or
- the user wishes to terminate the optimisation in a controlled fashion, since the optimisation is deemed to be complete.

The first case would not be cured merely by presenting revised parameters in a restart job, since the program will initially run through the pathway prior to applying the new parameters, by which time the position may not be recoverable. In such cases the user must identify from the output of the startup job a line search in the optimisation pathway prior to the problem, and present this information to the program via a revised form of the MINMAX directive in a restart job. The revised format consists of a single data line read to the variables TEXT, TEXT1, LINES using format (2A,I):

- TEXT is set to the character string MINMAX;

- TEXT1 is set to the character string REVISE;
- LINES is an integer defining the line search after which any optimisation parameters presented in the input stream are to apply.

The following specification would cause the original optimisation pathway to be followed for the first three line searches only; beyond that point the optimisation would be restarted based on any revised control parameters.

```
MINMAX REVISE 4
```

In the example below we consider the SCF geometry optimisation of C_4F_4. In fact this optimisation proceeds smoothly, converging to the default accuracy (XTOL = 0.003) on the seventh line search using the data shown below.

```
TITLE
**** C4F4  3/21G ****
ZMAT ANGS
X
C 1  R1
C 1  R2  2  90.
C 1  R1  3  90.   2 180.
C 1  R2  4  90.   3 180.
X 2 1.   1 90.    3 0.
F 2 R3   6 90.    3 180.
X 4 1.   1 90.    3 0.
F 4 R3   8 90.    3 180.
X 3 1.   1 90.    4 0.
F 3 R3   10 90.   4 180.
X 5 1.   1 90.    4 0.
F 5 R3   12 90.   4 180.
VARIABLES
R1 1.2
R2  1.3
R3 1.313
END
RUNTYPE OPTIMIZE
LEVEL 2.0 40 1.0
ENTER
```

Now let us assume we wish to tighten the convergence threshold, using an XTOL setting of 0.001. Merely presenting a revised XTOL specification in a restart job will not have the desired effect, for the job will merely work through the optimisation pathway, reaching convergence before the revised XTOL setting will come into effect. The user must inform the optimisation that the previous actions on the seventh line search are to be ignored, and repeated with the revised XTOL setting, by presenting the following data file:

```
RESTART OPTIMIZE
TITLE
**** C4F4  3/21G ****
```

```
ZMAT ANGS
X
C 1  R1
C 1  R2  2  90.
C 1  R1  3  90.   2 180.
C 1  R2  4  90.   3 180.
X 2 1.   1 90.    3 0.
F 2 R3   6 90.    3 180.
X 4 1.   1 90.    3 0.
F 4 R3   8 90.    3 180.
X 3 1.   1 90.    4 0.
F 3 R3   10 90.   4 180.
X 5 1.   1 90.    4 0.
F 5 R3   12 90.   4 180.
VARIABLES
R1 1.2
R2  1.3
R3 1.313
END
RUNTYPE OPTIMIZE
LEVEL 1.0
MINMAX REVISE 7
XTOL 0.001
ENTER
```

13.5 OPTXYZ Data; MINMAX, XTOL and STEPMAX

Three directives are provided to control the OPTXYZ search procedure, MINMAX, XTOL and STEPMAX. Note that directive specifications are similar, but not identical to, the descriptions above. Note also that algorithm employed *only* guarantees convergence to a *stationary point*, not necessarily to a minimum.

13.5.1 OPTXYZ Data - MINMAX

This directive may be used to control the number of energy evaluations and searches permitted in optimising a given structure, and consists of a single data line read to the variables TEXT, NPTS, NSERCH using format (A,2I):

- TEXT is set to the character string MINMAX;
- NPTS is an integer specifying the maximum number of energy evaluations allowed in the optimisation;
- NSERCH is an integer defining the maximum number of searches permitted in the BFGS update procedure.

The MINMAX directive may be omitted, when both NPTS and NSERCH will be set to the maximum allowed value of 60. The following specification is thus equivalent to the default;

```
MINMAX 60 60
```

Example

In some cases the user may wish to perform just the initial point on the optimisation pathway to gauge the quality of the starting geometry though the magnitude of the gradient at that point. This may be achieved though use of MINMAX, as shown below:

Start-up Job

```
TITLE\H2O DZ OPTIMIZATION STARTUP
ZMAT ANGSTROM\O\H 1 ROH\H 1 ROH 2 THETA
VARIABLES\ROH 0.956 HESSIAN 0.7\THETA 104.5 HESSIAN 0.2 \END
BASIS DZ
RUNTYPE OPTXYZ\MINMAX 1 1
ENTER
```

Here we are using the MINMAX directive to terminate the optimisation after the first point. This may then be restarted as shown below, where the default MINMAX settings will apply.

Restart Job

```
RESTART OPTXYZ
TITLE\H2O DZ OPTIMIZE
ZMAT ANGSTROM\O\H 1 ROH\H 1 ROH 2 THETA
VARIABLES\ROH 0.956 HESSIAN 0.7\THETA 104.5 HESSIAN 0.2 \END
BASIS DZ
RUNTYPE OPTXYZ
ENTER
```

13.5.2 OPTXYZ Data - XTOL

This directive may be used to define the convergence threshold for the optimisation, and consists of a single data line read to the variables TEXT, TOL using format (A,F):

- TEXT should be set to the character string XTOL;
- TOL should be set to the value to be used in defining the criteria for convergence of the optimisation algorithm, the maximum component of the gradient.

The XTOL directive may be omitted, when TOL will be set to 0.001. The default thus corresponds to presenting the data line

```
XTOL 0.001
```

13.5.3 OPTXYZ Data - STEPMAX

This directive may be used to define the the maximum allowed movement in any of the cartesian coordinates in a single step of the geometry optimisation (in units of bohr). The directive consists of a single data line read to the variables TEXT, STEP using format (A,F):

- TEXT should be set to the character string STEPMAX;
- STEP should be set to the maximum permitted movement in any coordinate.

The STEPMAX directive may be omitted, when STEP will be set to 0.2. The default thus corresponds to presenting the data line

```
STEPMAX 0.2
```

13.6 JORGENSEN Data Specification

An extended discussion of the data requirements for use when invoking the Simons and Jorgensen algorithm is given in section 10. For completeness we include here the data file required in optimising the geometry of H_2CO, noting that in most cases the default settings will prove satisfactory.

```
TITLE
H2CO - DZ BASIS - JORGENSEN OPTIMISATION
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
END
BASIS DZ
RUNTYPE OPTIMIZE JORGENSEN
ENTER
```

13.7 Transition State Location and RUNTYPE Specification

Three methods are available to search for a transition state on a potential Surface, each driven through SADDLE specification on the RUNTYPE directive and each relying on internal coordinate specification through the ZMATRIX directive.

1. the recommended method, a modification to the Cerjan and Miller ‘trust-region’ algorithm, is driven through the specification

   ```
   RUNTYPE SADDLE
   ```

   This method performs optimisation in internal coordinates, and thus requires initial ZMATRIX and VARIABLES specification of the molecular geometry, or ZMATRIX construction from an initial set of cartesian coordinates supplied under control of the GEOMETRY directive. Note that the success of the method is dependent on the quality of the initial

Hessian, and the user is reminded of the need to address this issue through appropriate TYPE specifications on the VARIABLE definition lines of the ZMATRIX (see Part 3 section 8).

2. the second internal coordinate–driven method is that based on the hill-walking algorithm due to Jorgensen and coworkers [6]. A more detailed account of the method and associated data is given below in section 10. We note here that the procedure is driven through additional keyword specification on the RUNTYPE directive, thus;

   ```
           RUNTYPE SADDLE JORGENSEN
   ```

   The method is again reliant on a quality initial hessian for success.

3. the third method, perhaps less reliable and requiring additional input data than the others, is the synchronous–transit internal coordinate based method due to Bell and Crighton. This is again requested through the RUNTYPE SADDLE specification, together with appropriate usage of the LSEARCH directive (see below).

We consider the data input requirements for the trust-region and synchronous–transit methods below, and those for the Jorgenson algorithm in section 10.

13.8 SADDLE Data Specification: Trust-Region

Four directives are provided to control the trust–region search procedure, MINMAX, XTOL, STEPMAX and VALUE, with specifications very similar to the corresponding directives described above for OPTIMIZE usage. The user should note that the implementation is also based on maintaining a history of the optimisation pathway, that will be worked through on each restart of the optimisation. This appears on the output as a sequence of both 'old' and 'new' calculations, with the history printed on each restart. This printing may be suppressed through use of the NOPRINT directive, with specification of the HISTORY keyword.

13.8.1 SADDLE Data - MINMAX

This directive may be used to control the number of energy evaluations and line-searches permitted in the location of the transition state, and consists of a single data line read to the variables TEXT, IVAL, LINE using format (A,2I).

- TEXT is set to the character string MINMAX.
- IVAL is an integer specifying the maximum number of energy evaluations allowed in the optimisation.
- LINE is an integer defining the maximum number of line searches permitted in the optimisation.

The MINMAX directive may be omitted, when both IVAL and LINE will be set to the maximum allowed value of 60. The following specification is thus equivalent to the default;

```
MINMAX 60 60
```

Example

In some cases the user may wish to perform just the initial point on the optimisation pathway to gauge the quality of the starting geometry though the magnitude of the gradient at that point. This may be achieved though use of MINMAX, as shown below:

Start-up Job

```
TITLE\HCCH/CCH2 . RHF3-21G . START-UP JOB.
ZMAT ANGS\C\ C 1 L1\ H 2 L2  1 A1
X 2 1.0 1 90.0 3 180.0\ H 2 L3  4 A2   1 180.0
VARIABLES
L1 1.24054 TYPE 3
L2 1.65694 TYPE 3
L3 1.06318 TYPE 3
A1 60.3568 TYPE 3
A2 60.3568 TYPE 3
END
RUNTYPE SADDLE
XTOL 0.002\MINMAX 60 1
ENTER
```

Here we are using the MINMAX directive to terminate the optimisation after the first line search. Note that using the number of energy evaluations as the criterion may not be productive, for this will cause termination at the first point in the evaluation of the 2nd-derivative matrix requested through the TYPE 3 specifications. This may then be restarted as shown below, where the default MINMAX settings will apply.

Restart Job

```
RESTART SADDLE
TITLE\HCCH/CCH2 . RHF3-21G . START-UP JOB.
ZMAT ANGS\C\ C 1 L1\ H 2 L2  1 A1
X 2 1.0 1 90.0 3 180.0\ H 2 L3  4 A2   1 180.0
VARIABLES
L1 1.24054 TYPE 3
L2 1.65694 TYPE 3
L3 1.06318 TYPE 3
A1 60.3568 TYPE 3
A2 60.3568 TYPE 3
END
RUNTYPE SADDLE
XTOL 0.002
ENTER
```

13.8.2 SADDLE Data - XTOL

This directive may be used to define the convergence thresholds for the optimisation, and consists of a single data line read to the variables TEXT, TOL using format (A,F):

- TEXT should be set to the character string XTOL;
- TOL should be set to the value to be used in defining the four acceptance criteria for the convergence of the optimisation algorithm. These criteria are:

```
maximum change in variables  <  TOL
average change in variables  <  TOL * 2/3
maximum gradient             <  TOL * 1/4
average gradient             <  TOL * 1/6
```

The XTOL directive may be omitted, when TOL will be set to 0.001. The default thus corresponds to presenting the data line

```
XTOL 0.001
```

13.8.3 SADDLE Data - STEPMAX

This directive may be used to define the the maximum allowed movement in any of the variables in a single step of the transition state location. Note that the internal units of the variables are bohr for bond lengths and radians for angles. The directive consists of a single data line read to the variables TEXT, STEP using format (A,F);

- TEXT should be set to the character string STEPMAX;
- STEP should be set to the maximum permitted movement in any variable.

The STEPMAX directive may be omitted, when STEP will be set to 0.2. The default thus corresponds to presenting the data line

```
STEPMAX 0.2
```

There is certainly at least one circumstance where changes to the default setting may prove crucial in achieving controlled convergence. If the starting geometry is known to be poor, or if ZMATRIX specification is such that a specific bond is not explicitly defined (as can happen for example with aromatic compounds), then the first step taken on the optimisation may prove both excessive and counter-productive; at best several extra points will be required to recover from this effect, at worst a change of state may be induced in the SCF wavefunction. This effect is fairly common if in addition the starting hessian is also poorly defined. When this happens, the user should consider starting the optimisation again, presenting a STEPMAX directive of the form:

```
STEPMAX 0.1
```

13.8.4 SADDLE Data - VALUE

This directive may be used to control the accuracy of the search for a turning point during a line search, and consists of a single data line read to the variables TEXT, TURN using format (A,F);

- TEXT should be set to the character string VALUE;
- TURN should be set to a value between 0.0 and 1.0 that will control the accuracy of the line search procedure. Note that the smaller TURN, the more accurate the line search.

The VALUE directive may be omitted, when TURN will be set to 0.3. The default thus corresponds to presenting the data line

```
VALUE 0.3
```

13.9 Synchronous–Transit Data

The present implementation of the synchronous–transit method is driven under specification of the LSEARCH directive. Successful results from the method rely on the specification of not only a reasonable guess for the initial geometry, but on presenting the equilibrium geometries as data for the two minima involved on the potential surface. These minima are specified on the variable definition lines of the ZMATRIX, and require that the form of the ZMATRIX has been constructed in such a way as to yield VARIABLES that transform smoothly from one minima, though the transition state and onto the second minima. This is shown below for the transition state involved in the HCN to HNC isomerisation process.

```
TITLE
HCN SADDLE POINT - SYNCHRONOUS TRANSIT
ZMAT ANGS
C
X 1 1.0
N 1 CN 2 90.0
H 1 CH 2 90.0 3 HCN
VARIABLES
CN 1.1484 MINIMA 1.1371 1.1597
CH 1.5960 MINIMA 1.0502 2.1429
HCN 90.0   MINIMA 180.0  0.0
END
BASIS SV 4-31G
RUNTYPE SADDLE
LSEARCH 0 4
ENTER
```

13.9.1 Synchronous–Transit Data - LSEARCH

The LSEARCH directive may be used to request and characterise the synchronous–transit method, overriding the default trust-region, and consists of a single data line read to the variables TEXT, LINE, IPOL using format (A,2I)

- TEXT should be set to the character string LSEARCH
- LINE is an integer used to identify the type of line search to be performed. The two valid settings are LINE=0, requesting subsequent line searches be based on energy evaluation alone, and LINE=1, when both the energy and gradient of the energy will be used at each point in each line search.
- IPOL is an integer used to specify the form of polynomial be employed for the principal direction of negative curvature [4], The two valid settings are LINE=2 (quadratic polynomial) or LINE=4 (quartic polynomial).

The synchronous–transit method is invoked by presenting the data line

```
LSEARCH 0 4
```

which must be presented *after* the RUNTYPE directive. Note that the default trust region method corresponds to the specification

```
LSEARCH 0 5
```

The LSEARCH directive may also be used to influence OPTIMISE and OPTXYZ runs. For an OPTIMISE run, a LSEARCH 1 specification requests (as above) that function and gradient evaluations are used during a line search. For an OPTXYZ run, which usually tries to do a line search until the energy goes down, the LSEARCH 1 specification indicates that a gradient evaluation is requested and each point is used, even if the energy is higher. For OPTXYZ specifying the first parameter as 0 (function evaluations) and specifying the second variable allows one to override the standard maximum number of steps along a line (normally 3), before the hessian is reset and the program tries again. To try for 7 points in OPTXYZ one specifies

```
LSEARCH 0 7
```

13.9.2 Synchronous–Transit Data - TOLMAX

The TOLMAX directive may be used to control how far a search for a minimum in the n-1 subspace [4] may proceed before another search for a maximum is performed. Smaller values will cause the program to search for a maximum more often. The default is

```
TOLMAX 0.1
```

13.9.3 Synchronous–Transit Data - TOLSTEP

The TOLSTEP directive is used to maintain 'good' conjugate directions to the principal direction of negative curvature. If the step along this direction was too large in the previous iteration then the program takes a small step in order to estimate a better set of conjugate directions. The default is equivalent to

```
TOLSTEP 0.1
```

13.9.4 Synchronous–Transit Data - TANSTEP

The TANSTEP directive may be used when the TOLSTEP test requires the conjugate directions to be recalculated. A step (TANSTEP*previous step length) is taken along the current tangent to the polynomial and the function and gradients are calculated at this point. The default is

```
TANSTEP 0.1
```

13.9.5 Synchronous–Transit Data - MINMAX

This directive may be used to control the number of energy evaluations and line-searches permitted in optimising a given structure. The default is

```
MINMAX 60 60
```

The first integer refers to the maximum number of energy evaluations allowed, and the second to the maximum number of line searches.

13.9.6 Synchronous–Transit Data - XTOL

Defines the four acceptance criteria for the convergence of the synchronous–transit algorithm. The criteria are:

```
maximum change in variables  <  xtol
average change in variables  <  xtol*2/3
maximum gradient             <  xtol*1/4
average gradient             <  xtol*1/6
```

The default corresponds to presenting the data line

```
XTOL 0.001
```

13.9.7 Synchronous–Transit Data - STEPMAX

Defines the maximum allowed movement in any of the variables in a single step. The internal units of the variables are bohr for bond lengths and radians for angles. The default is equivalent to

```
 STEPMAX 0.2
```

13.10 Restoring the Force Constant Matrix - FCM

An FCM directive may be given to specify the Force Constant Matrix (Hessian) to be restored in an OPTIMISE or SADDLE calculation. Specifying FCM or a dumpfile name on a RUNTYPE OPTIMISE or RUNTYPE SADDLE directive is equivalent to providing a separate FCM directive. The directive consists of a single line containing

```
        FCM  DD  IBLOCK  TYPE
```

where DD is the name of the dumfile containing the Hessian and IBLOCK is its starting block. TYPE is the type of Hessian requested and can be any of : MP2, SCF or OPTIMISE, where the former denote analytical hessians generated using MP2 or SCF respectively and OPTIMISE requests the Hessian produced during an OPTIMISE or SADDLE run. If the TYPE is omitted, the dumpfile is searched for hessian sections in the order given above. The Dumpfile specification may be omitted, in which case the current dumpfile is searched. For example the next sets of input lines are equivalent, assuming the default dumpfile :

```
      RUNTYPE OPTIMISE
      FCM ED3 1
      ........
      RUNTYPE OPTIMISE FCM
      ........
      RUNTYPE OPTIMISE ED3 1
```

The FCM directive may also read

```
      FCM UNIT7
```

in which case the Force Constant Matrix is read from the punchfile.

14 Jorgensen and Simons Optimisation Algorithm

An alternative stationary point optimisation procedure, code-named JORGENSEN, is available within GAMESS–UK. An efficient quasi Newton–Raphson algorithm for locating transition states, this procedure is based on a modification to the Newton–Raphson step first proposed by Cerjan and Miller [5], although the major part of the algorithm is founded on the later developments of Simons, Jorgensen and coworkers [6]. The algorithm is capable of locating transition states even if started in the 'wrong' region of the energy surface, and, by invoking Hessian mode following, can locate transition states for alternative rearrangement and/or dissociation reactions from the same starting point. The algorithm may also be used to locate minima on a surface.

A description of the formalism and the ideas behind it, together with a description of the algorithm and some practical examples are given in reference [7].

14.1 Invocation of The Algorithm and Associated Parameters

The Jorgensen and Simons optimisation algorithm is invoked by appending the keyword JORGENSEN on the appropriate RUNTYPE directive i.e.

```
   RUNTYPE SADDLE  JORGENSEN  ......  Search for saddle point.

   RUNTYPE OPTIMIZE JORGENSEN ......  Search for minimum point.
```

Omission of the keyword will cause the program to default to the optimisation algorithms described above. The following discussion makes the assumption that a search is being made for such a stationary point. If no other relevant directives are specified the default is to evaluate the Hessian according to the TYPE specifications in the VARIABLES definition lines of the ZMATRIX directive. A Hessian may still, however, be restored from a foreign dumpfile by specifying the LFN, starting block and section number of the foreign dumpfile in the 4th and 5th fields on the RUNTYPE directive line, e.g.

```
RUNTYPE SADDLE JORGENSEN ED4 1
```

In addition to the keyword JORGENSEN, there are a number of sub-directives that may be used to control the search procedure. These should be specified immediately following the RUNTYPE directive, and are summarised below.

14.2 POWELL

The POWELL directive consists of a single data line comprising the keyword POWELL in the first data field. The directive may be used to request updating of the Hessian using the Powell update procedure, and should be specified in Transition state location.

14.3 BFGS

The BFGS directive consists of a single data line comprising the keyword BFGS in the first data field. The directive may be used to request updating of the Hessian using the BFGS update procedure, and should be specified in geometry optimisation.

14.4 BFGSX

The BFGSX directive may be used to request use of a modified BFGS update procedure in geometry optimisation calculations, with safeguards to ensure retention of a positive definite hessian.

14.5 RECALCULATE

This directive consists of a single data line read to the variables TEXT, NSTEP using format (A,I).

- TEXT should be set to the character string RECALCULATE
- NSTEP is an integer used to specify the frequency of recalculation of the Hessian using the method dictated by the "TYPE" keywords of the Variables Definition line of the ZMATRIX directive.

The directive may be omitted, when the default of updating the hessian only will hold throughout. An alternative specification permits this recalculation to be suppressed through a data line of the form

```
        RECALCULATE OFF
```

14.6 CUTOFFS

This directive may be used to request relaxation of certain constraints in optimisation, and should be used with caution.

14.7 EIGMIN

The EIGMIN directive consists of a single data line read to variables TEXT, VMIN using format (A,F):

- TEXT should be set to the character string EIGMIN;
- VMIN should be set to the minimum allowed value for eigenvalues of the Hessian matrix

14.8 EIGMAX

The EIGMAX directive consists of a single data line read to variables TEXT, VMAX using format (A,F):

- TEXT should be set to the character string EIGMAX;
- VMAX should be set to the maximum allowed value for eigenvalues of the Hessian matrix

14.9 MAXJORGEN

The MAXJORGEN directive consists of a single data line read to variables TEXT, MAXJOR using format (A,I):

- TEXT should be set to the character string MAXJORGEN;
- MAXJOR should be set to the maximum number of allowed cycles in the optimisation.

In the absence of this directive, MAXJOR will be set to 40.

14.10 RFO

The RFO directive consists of a single data line read to variables TEXT, TXTRFO using format (2A);

- TEXT should be set to the character string RFO;
- TXTRFO should be used to control the nature of the steps taken in the search procedure, and may be set to the character string ON or OFF. The default specification, corresponding to TXTRFO=ON, results in the RFO or P-RFO steps only (see [7]). Specifying OFF will result in the taking of Newton-Raphson steps where appropriate instead of RFO or P-RFO steps.

14.11 OPTPRINT

This directive may be used to control diagnostic output during optimisation. A data line of the form

```
OPTPRINT ON
```

may be used to increase diagnostic output.

14.12 Example 1

The following data file may be used to perform the search for the FCN/FNC transition state, using an STO-3G basis set. The initial Hessian is computed through TYPE 3 specification, with the Powell update requested throughout Jorgensen optimisation.

```
TITLE
FCN/FNC TS SEARCH . STO3G BASIS.
ZMAT ANGS
C
N 1 L1
F 1 L2 2 A1
VARIABLES
L1 1.2   TYPE 3
L2 1.3   TYPE 3
A1 135.0 TYPE 3
END
BASIS  STO3G
RUNTYPE SADDLE JORGENSEN
POWELL
MAXJOR 55
RECALC OFF
RFO OFF
CUTOFFS
OPTPRINT ON
XTOL 0.0018
ENTER
```

15 Optimisation using DL-FIND

DL-FIND is a self-contained module that is included into GAMESS-UK. Additional information on DL-FIND, as well as the developmental version of the code can be found on the DL-FIND website at:

http://ccpforge.cse.rl.ac.uk/projects/dl-find/

DL-FIND offers several methods for geometry optimisation and transition state search. The geometry can be optimised in a range of coordinate systems: Cartesian coordinates, Mass-weighted Cartesian coordinates, delocalised (redundant) internal coordinates, and hybrid delocalised internal coordinates.

L-BFGS is the recommended method for minimum search.

Transition state search may be done by P-RFO, an improved version of the dimer method, or nudged-elastic band using a climbing image.

DL-FIND is invoked by the runtype optimize (or saddle) and the directive

```
DLFIND
   DL-FIND directives
END
```

15.1 DL-FIND directives

15.1.1 COORDINATES

The directive COORDINATES TEXT specifies the coordinate system in which the optimisation should be performed. Possible choices for TEXT are:

CART Cartesian coordinates (default)

MASS Mass-weighted Cartesian coordinates

DLC Delocalised internal coordinates. A redundant set of bonds, angles, torsions, and sometimes inversions is created. A non-redundant combination of them is found by diagonalising the spectroscopic G-matrix. The optimisation is performed in this non-redundant set.

TC Same as DLC, but the redundant set consists only of bonds – all atoms in the system are connected.

HDLC Hybrid delocalised internal coordinates. The system is partitioned into fragments. Delocalised coordinates (as in DLC) are used within each fragment. The fragments are coupled via Cartesian coordinates. This version is recommended for large systems, as the coordinate transformation scales linearly with the number of atoms.

15.1.2 DIMER

The directive DIMER is used to start a transition state search using the dimer method. Two images of the system (their distance is specified by the DELTA directive) are calculated. They are optimised along the force in all directions perpendicular to the dimer axis, and against the force along the dimer axis. Thus the system converges to a first order saddle point.

After a rotation step, the gradient can be interpolated (default) or recalculated. In this case, specify the directive as DIMER NOINTERPOLATION.

15.1.3 DELTA

DELTA specifies the dimer distance and the elongation for finite-difference Hessian evaluation. The units are atomic units if Cartesian coordinates are used, and undefined if delocalised internals are used. Default:

```
DELTA 0.01
```

15.1.4 NEB

The directive NEB specifies a nudged elastic band calculations. The improved-tangent NEB algorithm with a climbing image is implemented. The L-BFGS optimiser is recommended with NEB, although other optimisers may as well be tried. The syntax is:

```
NEB NIMAGE K
```

Where NIMAGE is an integer specifying the number of images (default: 10), and K is a real number specifying the NEB force constant (default: 0.01).

Two structures have to be provided for an NEB calculation. One endpoint will be the geometry provided in the normal GAMESS input. The other structure will be provided by the directive GEOMETRY.

15.1.5 NEBMODE

The directive NEBMODE specifies details of an NEB calculation.

```
NEBMODE MODE CART
```

Where MODE can be FREE, FROZEN, or PERP. It specifies if the endpoints of the NEB path will be freely minimised, frozen, or only free to move perpendicular to the path, respectively. FROZEN is the default.

If CART is specified, only the initialisation of the path will be done in the coordinate system specified by COORDINATES. The following minimisation will be done in Cartesian coordinates.

15.1.6 GEOMETRY

The directive GEOMETRY specifies the endpoint of an initial NEB path. The other endpoint is specified by the normal input geometry. The directive GEOMETRY can also be used to specify an initial dimer direction.

```
GEOMETRY TEXT
```

TEXT is the name of a file in xyz format. It has to contain the same atoms as the GAMESS input geometry.

15.1.7 OPTIMISER

The directive OPTIMISER TEXT specifies the optimisation algorithm. Possible choices for TEXT are:

LBFGS Limited-memory Broyden–Fletcher–Goldfarb–Shanno optimisation. Recommended for minimum search, NEB, and the dimer method. The time and memory requirements spent for determining the search direction and step length scale linearly with the system size. And additional integer parameter can be specified, which determines the number of steps kept in memory. Default is the number of degrees of freedom.

PRFO The partitioned rational function optimisation method. Recommended for transition state search (unless done by the dimer or NEB methods). It requires the calculation of the Hessian. The UPDATE directive (see below) can be used to specify how the Hessian should be updated.

CG Conjugate gradient method following Polak–Ribière [?].

SD Steepest descent.

DYN Damped molecular dynamics. Four additional real parameters can be specified, which determine (1) the time step in atomic units (default 1.0), (2) the start friction (default: 0.3), (3) the factor to reduce the friction each time the energy decreases (default: 0.95), and the friction to apply if the energy increases (default 0.3). The frictions are defined so that 0 corresponds to free (undamped) dynamics, and 1 corresponds to steepest descent. This variable friction facilitates convergence to an energy minimum.

15.1.8 UPDATE

The directive UPDATE specifies the Hessian update details in case an explicit Hessian is calculated (i.e. for the PRFO optimiser). Syntax:

```
UPDATE METHOD RECALC FD SOFT
```

Where RECALC is the (integer) number of updates before the Hessian is recalculated (default: 100), FD is 1 or 2 to use a one-point or two-point formula to calculate the finite-difference Hessian (default: 2). SOFT is a positive real number (default: 0.003). Eigenmodes with an absolute eigenvalue below SOFT are ignored by the P-RFO algorithm. This avoids steps into the translation and rotation directions. METHOD may be one of

BOFILL Bofill update of the Hessian (default)

POWELL Powell update of the Hessian

NONE No update, recalculate the Hessian in each step

15.1.9 XTOL

In accordance to the other optimisers, XTOL specifies the convergence criterion applied to the optimisation. Note that XTOL has to be specified within the DLFIND block. Default:

```
XTOL 0.003
```

15.1.10 MINMAX

In accordance to the other optimisers, MINMAX specifies the maximum number of energy evaluations. Note that MINMAX has to be specified within the DLFIND block. Default:

```
MINMAX 60
```

15.1.11 STEPMAX

In accordance to the other optimisers, STEPMAX specifies the maximum step in one coordinate component. Note that STEPMAX has to be specified within the DLFIND block. Default:

```
STEPMAX 0.5
```

15.2 Example 1

The following data file performs a simple minimisation of H_2CO in redundant internal coordinates using the L-BFGS algorithm.

```
TITLE
H2CO - DZ - DL-FIND energy = -113.8307609
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
```

```
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
END
BASIS DZ
RUNTYPE OPTIMIZE
DLFIND
  COORDINATES DLC
END
ENTER
```

15.3 Example 2

The following data files illustrate a NEB optimisation of part of the reaction path of H_2CO dissociation.

```
TITLE
H2CO - DZ - DL-FIND NEB energy = -113.6497325
ZMATRIX ANGSTROM
C
O 1 CO
H 1 L2 2 A1
H 1 L3 2 A2 3 0.0
VARIABLES
CO 1.25
L2 1.5
L3 1.1
A1 120.0
A2 170.0
END
BASIS DZ
RUNTYPE OPTIMIZE
DLFIND
  NEB 6 0.01
  GEOMETRY neb_endpoint.xyz
END
ENTER
```

An additional file with the name neb_endpoint.xyz is required. It has the contents:

```
           4

C    0.3526549   0.4601460   0.0000000
O    0.1899991  -0.6367263   0.0000000
H   -1.3596765   1.1431771   0.0000000
H   -0.7056650   1.4269885   0.0000000
```

16 The ENTER Directive

The ENTER directive should be the last directive specified, since it initiates the calculation. The directive may also be used to nominate the section(s) on the Dumpfile where eigenvectors generated by the processing requested through SCFTYPE during the present run of the program are to be written. Since more than one set of vectors may be generated by several of the SCF options (two, for example, in open-shell RHF, GVB, UHF, MCSCF and CASSCF calculations) it follows that multiple section specification may feature on the ENTER directive.

The directive consists of a single data line read to variables TEXT, ISECT1 and ISECT2 using format (A,2I).

- TEXT should be set to the character string ENTER;
- ISECT1 may be used to specify the section number on the Dumpfile where the first set of vectors associated with the requested SCFTYPE are to be written;
- ISECT2 may be used to specify the section number on the Dumpfile where the second set of vectors associated with the requested SCFTYPE are to be written.

If both ISECT1 and ISECT are omitted, it will be assumed that the default sections of Table 2 are in effect. Note that if explicit section specification is used, then ISECT2 should be omitted in those cases involving a single set of eigenvectors. We illustrate below typical ENTER data lines as a function of SCFTYPE, assuming the defaults of Table 2 are to be overwritten;

- For a closed shell SCF calculation the data line

  ```
  ENTER 10
  ```

  will route the eigenvectors to section 10 of the Dumpfile.
- For a UHF calculation the data line

  ```
  ENTER 12 13
  ```

  will route the α-spin eigenvectors to section 12 of the Dumpfile β-spin vectors to section 13.
- For a GVB or open shell calculation the data line

  ```
  ENTER 16 17
  ```

  will route the un-canonicalised vectors to section 16, and the canonicalised orbitals to section 17. Omitting the second section in such calculations will result in the canonicalised set *only* appearing on the Dumpfile, the un-canonicalised set being overwritten on convergence of the SCF.
- For a CASSCF calculation the data line

  ```
  ENTER 18 19
  ```

  will route the CASSCF orbitals to section 18, and the canonicalised CASSCF vectors to section 19. Note that the present set of CI coefficients will be appended to this section.

17 STOP

The STOP directive may be presented as an alternative to ENTER and, with the same syntax, may be used simply as a check of the data input. No modifications are made to the Dumpfile, execution terminating once data processing is complete.

References

[1] P. Pulay, Chem. Phys. Lett. **73** (1980) 393, doi:10.1016/0009-2614(80)80396-4; P. Pulay, J. Comp. Chem. **3** (1982) 556–560, doi:10.1002/jcc.540030413.

[2] F.W. Bobrowicz and W.A. Goddard, in 'Modern Theoretical Chemistry', Vol. 3, ed. H.F. Schaefer, Plenum, New York (1977) 79.

[3] B. Jonsson, B.O. Roos, P.R. Taylor and P.E.M. Siegbahn, J. Chem. Phys. **74** (1981) 4566, doi:10.1063/1.441645; B.O. Roos, P. Linse, P.E.M. Siegbahn and M.R.A. Blomberg, Chem. Phys. **66** (1982) 197, doi:10.1016/0301-0104(82)88019-1; P.J. Knowles, G.J. Sexton and N.C. Handy, Chem. Phys. **72** (1982) 337, doi:10.1016/0301-0104(82)85131-8: The original CASSCF module, as developed by Dr. P.J. Knowles was incorporated into GAMESS in April 1983.

[4] S. Bell and J.S. Crighton, J. Chem. Phys. **80** (1984) 2464, doi:10.1063/1.446996.

[5] C.J. Cerjan and W.H. Miller, J. Chem. Phys. **75** (1981) 2800, doi:10.1063/1.442352.

[6] J. Simons, P. Jorgensen, H. Taylor and J. Ozment, J. Phys. Chem. **87** (1983) 2745, doi:10.1021/j100238a013; A. Banerjee, N. Adams, J. Simons and R. Shepard, J. Phys. Chem. **89** (1985) 52, doi:10.1021/j100247a015.

[7] J. Baker, J. Comp. Chem. **7** (1986) 385–395, doi:10.1002/jcc.540070402.

[8] D. Moncrieff and V.R. Saunders, ATMOL-Introduction Notes, UMRCC, May, 1986; Cyber-205 Note Number 32, UMRCC, September, 1985; V.R. Saunders and M.F. Guest, ATMOL3 Part 9, RL-76-106 (1976); M.F. Guest and V.R. Saunders, Mol. Phys. **28** (1974) 819, doi:10.1080/00268977400102171.

[9] M. Dupuis and H.F. King, Int. J. Quantum Chem. **11** (1977) 613, doi:10.1002/qua.560110408; M. Dupuis and H.F. King, J. Chem. Phys **68** (1978) 3998, doi:10.1063/1.436313; H.F. King and M. Dupuis, J. Comp. Phys. **21** (1976) 144, doi:10.1016/0021-9991(76)90008-5; M. Dupuis, J. Rys and H.F. King, J. Chem. Phys. **65** (1976) 111, doi:10.1063/1.432807.

[10] V.R. Saunders and I.H. Hillier, Int. J. Quant. Chem. **7** (1973) 699, doi:10.1002/qua.560070407.

[11] R.W. Warren and B.I. Dunlap, *Fractional occupation numbers and density functional energy gradients with the linear combination of Gaussian-type orbitals approach*, Chem. Phys. Lett. **262** (1996) 384–392, doi:10.1016/0009-2614(96)01107-4.

[12] P.J. Stephens, F.J. Devlin, C.F. Chabalowski, and M.J. Frisch, *Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields*, J. Phys. Chem. **98** (1994) 11623-11627, doi:10.1021/j100096a001.

[13] R.H. Hertwig, and W. Koch, *On the parameterization of the local correlation functional: What is Becke-3-LYP?*, Chem. Phys. Lett. **268** (1997) 345-351, doi:10.1016/S0009-2614(97)00207-8.

[14] Y. Zhao, B.J. Lynch, and D.G. Truhlar, *Development and assessment of a new hybrid density functional model for thermochemical kinetics* J. Phys. Chem. **A108** (2004) 2715-2719, doi:10.1021/jp049908s.

[15] A.D. Becke, *Density-functional thermochemistry. IV. A new dynamical correlation functional and implications for exact-exchange mixing*, J. Chem. Phys. **104** (1996) 1040-1046, doi:10.1063/1.470829.

[16] R.D. Adamson, P.M.W. Gill, and J.A. Pople, *Empirical density functionals*, Chem. Phys. Lett. **284** (1998) 6-11, doi:10.1016/S0009-2614(97)01282-7.

[17] F.A. Hamprecht, A.J. Cohen, D.J. Tozer, and N.C. Handy, *Development and assessment of new exchange-correlation functionals*, J. Chem. Phys. **109** (1998) 6264-6271, doi:10.1063/1.477267.

[18] A.D. Boese, N.L. Doltsinis, N.C. Handy, and M. Sprik, *New generalized gradient approximation functionals*, J. Chem. Phys. **112** (2000) 1670-1678, doi:10.1063/1.480732.

[19] A.D. Boese and N.C. Handy, *A new parametriztion of exchange-correlation generalized gradient approximation functionals*, J. Chem. Phys. **114** (2001) 5497-5503, doi:10.1063/1.1347371.

[20] J.P. Perdew, K. Burke, and M. Ernzerhof, *Generalized gradient approximation made simple*, Phys. Rev. Lett. **77** (1996) 3865-3868, doi:10.1103/PhysRevLett.77.3865.

[21] J.P. Perdew, J.A. Chevary, S.H. Vosko, K.A. Jackson, M.R. Pederson, D.J. Singh, and C. Fiolhais, *Atoms, molecules, solids and surfaces: Applications of the generalized gradient approximation for exchange and correlation*, Phys. Rev. **B46** (1992) 6671–6687, doi:10.1103/PhysRevB.46.6671.

[22] J.P. Perdew, and Y. Wang, *Accurate and simple analytic representation of the electron-gas correlation energy*, Phys. Rev. **B45** (1992) 13244-13249, doi:10.1103/PhysRevB.45.13244.

[23] A.D. Becke, J. Chem. Phys. **107** (1997) 8554, doi:10.1063/1.475007.

[24] P.J. Wilson, T.J. Bradley, and D.J. Tozer, *Hybrid exchange-correlation functional determined from thermochemical data and* ab initio *potentials*, J. Chem. Phys. **115** (2001) 9233-9242, doi:10.1063/1.1412605.

[25] M.E. Mura, and P.J. Knowles, *Improved radial grids for quadrature in molecular density-functional calculations*, J. Chem. Phys. **104** (1996) 9848-9858, doi:10.1063/1.471749.

[26] R.E. Stratmann, G.E. Scuseria, and M.J. Frisch, *Achieving linear scaling in exchange-correlation density functional quadratures*, Chem. Phys. Lett. **257** (1996) 213-223, doi:10.1016/0009-2614(96)00600-8.

[27] B.G. Johnson, *Development, implementation and applications of efficient methodologies for density functional calculations* pages 169 and further, In: J. M. Seminario and P. Politzer (Ed.), *Modern Density Functional Theory: A Tool for Chemistry*, Theoretical and Computational Chemistry, Vol. 2 (Elsevier Science B.V., 1995)

[28] A.D. Becke, *Density-functional thermochemistry. III. The role of exact exchange*, J. Chem. Phys. **98** (1993) 5648, doi:10.1063/1.464913.

[29] P.M.W. Gill, B.G. Johnson, and J.A. Pople, *A standard grid for density functional calculations*, Chem. Phys. Lett. **209** (1993) 506-512, doi:10.1016/0009-2614(93)80125-9.

[30] B.G. Johnson, P.M.W. Gill, J.A. Pople, *The performance of a family of density functional methods*, J. Chem. Phys. **98** (1993) 5612-5626, doi:10.1063/1.464906.

[31] C.W. Murray, N.C. Handy, and G.J. Laming, *Quadrature schemes for integrals of density functional theory*, Mol. Phys. **78** (1993) 997, doi:10.1080/00268979300100651.

[32] A.D. Becke, *Density-functional exchange-energy approximation with correct asymptotic behaviour*, Phys. Rev. **A38** (1988) 3098, doi:10.1103/PhysRevA.38.3098.

[33] A.D. Becke, *A multicenter numerical integration scheme for polyatomic molecules*, J. Chem. Phys. **88** (1988) 2547, doi:10.1063/1.454033.

[34] C. Lee, W. Yang, and R.G. Parr, *Development of the Colle-Salvetti correlation-energy formula into a functional of the density*, Phys. Rev. **B37** (1988) 785-789, doi:10.1103/PhysRevB.37.785.

[35] John P. Perdew, *Density-functional approximation for the correlation energy of the inhomogeneous electron gas*, Phys. Rev. **B33** (1986) 8822-8824, doi:10.1103/PhysRevB.33.8822.

[36] John P. Perdew, and Alex Zunger, *Self-interaction correction to density-functional approximations for many-electron systems*, Phys. Rev. **B23** (1981) 5048-5079, doi:10.1103/PhysRevB.23.5048.

[37] S.J. Vosko, L. Wilk, and M. Nusair, Can. J. Phys. **58** (1980) 1200, doi:10.1139/p80-159.

[38] M. Filatov, and W. Thiel, *A nonlocal correlation energy density functional from a Coulomb hole model*, Int. J. Quant. Chem. **62** (1997) 603-616.

[39] M. Filatov, and W. Thiel, *A new gradient-corrected exchange-correlation density functional*, Mol. Phys. **91** (1997) 847-859, doi:10.1080/002689797170950.

[40] B.I. Dunlap, J.W.D. Connolly, and J.R. Sabin, *On some approximations in applications of $X\alpha$ theory*, J. Chem. Phys. **71** (1979) 3396-3402, doi:10.1063/1.438728.

[41] N. Godbout, D. R. Salahub, J. Andzelm and E. Wimmer, Can. J. Chem. **70**, (1992) 560, doi:10.1139/v92-079.

[42] K. Eichkorn, O. Treutler, H. Ohm, M. Haser and R. Ahlrichs, Chem. Phys. Lett. **240** (1995) 283, doi:10.1016/0009-2614(95)00621-A; K. Eichkorn, F. Weigend, O. Treutler and R. Ahlrichs, Theor. Chim. Acta **97** (1997) 119, doi:10.1007/s002140050244.

[43] V. I. Lebedev and D. N. Laikov, *A quadrature formula for the sphere of the 131st algebraic order of accuracy*, Doklady Mathematics **59** (1999) 477-481.

[44] W. Kohn, and L. J. Sham, *Self-Consistent Equations Including Exchange and Correlation Effects*, Phys. Rev. **140** (1965) A1133-A1138, doi:10.1103/PhysRev.140.A1133.

[45] P. Hohenberg, and W. Kohn, *Inhomogeneous Electron Gas*, Phys. Rev. **136** (1964) B864, doi:10.1103/PhysRev.136.B864.

[46] J. C. Slater, Atomic Radii in Crystals, J. Chem. Phys. **41** (1964) 3199-3204, doi:10.1063/1.1725697. **Note:** For some elements there are no Bragg-Slater radii available. In that case radii are taken from, E. Clementi, D. L. Raimondi, and W. P. Reinhardt, Atomic Screening Constants from SCF Functions. II. Atoms with 37 to 86 Electrons, J. Chem. Phys. **47** (1967) 1300-1307, doi:10.1063/1.1712084.

[47] S. Grimme, *Accurate Description of van der Waals Complexes by Density Functional Theory Including Empirical Corrections*, J. Comp. Chem. **25** (2004) 1463-1473, doi:10.1002/jcc.20078.

[48] S. Grimme, *Semi-empirical GGA-type density functional constructed with a long-range dispersion correction*, J. Comp. Chem. **27** (2006) 1787-1799, doi:10.1002/jcc.20495.

[49] J. Antony, S. Grimme, *Density functional theory including dispersion corrections for intermolecular interactions in a large benchmark set of biologically relevant molecules*, Phys. Chem. Chem. Phys. **8** (2006) 5287-5293, doi:10.1039/b612585a.

[50] "The Construction and Interpretation of MCSCF wavefunctions", M.W. Schmidt and M.S. Gordon, Ann. Rev. Phys. Chem. **49** (1998) 233-266, doi:10.1146/annurev.physchem.49.1.233; "The Multiconfiguration SCF Method", B.O.Roos, in "Methods in Computational Molecular Physics", edited by G.H.F.Diercksen and S.Wilson, D.Reidel Publishing, Dordrecht, Netherlands, 1983, pp 161-187; "The Multiconfiguration SCF Method" B.O.Roos, in "Lecture Notes in Quantum Chemistry", edited by B.O.Roos, Lecture Notes in Chemistry v58, Springer-Verlag, Berlin, 1994, pp 177-254; "Optimization and Characterization of a MCSCF State", J.Olsen, D.L. Yeager, P. Jorgensen, Adv. Chem. Phys. **54** (1983) 1-176, doi:10.1002/9780470142783.ch1; "Matrix Formulated Direct MCSCF and Multiconfiguration Reference CI Methods", H.-J. Werner, Adv. Chem. Phys. **69** (1987) 1-62, doi:10.1002/9780470142943.ch1; "The MCSCF Method", R. Shepard, Adv. Chem. Phys. **69** (1987) 63-200, doi:10.1002/9780470142943.ch2; "The CASSCF Method and its Application in Electronic Structure Calculations", B.O. Roos, in "Advances in Chemical Physics", vol.69, edited by K.P.Lawley, Wiley Interscience, New York, 1987, pp 339-445, doi:10.1002/9780470142943.ch7.

[51] "General second order MCSCF theory: A Density Matrix Directed Algorithm", B.H.Lengsfield, III, J. Chem. Phys. 73 (1980) 382-390, doi:10.1063/1.439885; "The use of the Augmented Matrix in MCSCF Theory", D.R.Yarkony, Chem. Phys. Lett. **77**,634-635(1981), doi:10.1016/0009-2614(81)85223-2; M. Dupuis, P. Mougenot, J.D. Watts, in "Modern Techniques in Theoretical Chemistry", E.Clementi, editor, ESCOM, Leiden, 1989, chapter 7.

[52] "Direct configuration interaction and multiconfigurational self-consistent field method for multiple active spaces with variable occupations. I. Method", J. Ivanic, J. Chem. Phys. **119**, 9364-9376(2003), doi:10.1063/1.1615954.

[53] "Direct configuration interaction and multiconfigurational self-consistent field method for multiple active spaces with variable occupations. II. Application to oxoMn(Salen) and N2O4", J. Ivanic, J. Chem. Phys. **119**, 9377-9385(2003), doi:10.1063/1.1615955.