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 2. PROGRAM USAGE and SAMPLE DATA

M.F. Guest, J. Kendrick, J.H. van Lenthe and P. Sherwood

Contents

1 Introduction

The main purpose of this chapter is to provide an overall guide to using the program without the tedium often associated with an extensive catalogue of data input requirements. We aim to achieve this by describing a sequence of data files demonstrating use of the program in a variety of calculations on the formaldehyde molecule, H_2CO. The role and specification of the directives found in these examples will be presented in later Parts of the manual. Specifically we consider here data files for performing,

1. Closed-shell SCF calculations.
2. Closed-shell Direct-SCF calculations.
3. An open-shell restricted Hartree-Fock calculation on the formaldehyde cation (H_2CO^+, $^2B_2$), and corresponding direct-ROHF calculation.
4. An open-shell unrestricted Hartree-Fock calculation on the formaldehyde cation (H_2CO^+, $^2B_2$), and corresponding direct-UHF calculation.
5. GVB-1/PP and direct-GVB calculations on the ground state molecule.
6. A pseudopotential calculation on the ground state molecule.
7. A CASSCF calculation on the ground state molecule.
8. MCSCF calculations on the ground state molecule.
9. DFT calculations on the ground state molecule and cation.
10. Møller Plesset MP2 and MP3 calculations, single- and multi-reference.
11. Analysis of the ground state wavefunction.
12. Morokuma Energy Decomposition Analysis.
13. Restarting Integral and SCF Computations.
14. Geometry Optimisation in internal coordinates.
15. Geometry Optimisation in cartesian coordinates.
16. Transition State Optimisation.
17. Numerical Force Constant Evaluation.
18. Analytic Force Constant calculations.
19. Polarisability calculations.
20. Hyperpolarisability calculations.
21. Magnetisability calculations.

22. Calculation of infra–red intensities.
23. Calculation of Raman intensities.
24. Direct-CI calculations on the ground state molecule and ion.
25. Conventional and Semi-direct Table-CI calculations on the ground and first excited state of the molecule.
26. Full-CI calculations on the ground state molecule and ion.
27. Closed-shell coupled-cluster calculations (CCSD and CCSD(T)) on the ground state molecule.
28. A Green's function OVGF calculation of the valence IEs.
29. A Green's function 2ph-TDA calculation of the valence IEs.
30. RPA calculations of excitation energies.
31. Multi-configurational Linear Response (MCLR) calculations of excitation energies.
32. ZORA Relativistic effects.
33. Combining multiple calculations in a single job step.

1.1 Treatment of Molecular Symmetry

Before considering aspects of data specification, it is important that the user has at least a rough idea of the varying methods that GAMESS–UK employs in the treatment of molecular symmetry. The aim is, of course, to try and optimise performance while maintaining simplicity of related data specification. There are two fundamental, and related levels at which symmetry will be employed;

1. at the *molecular* level, when the program will deduce the point group symmetry based on the geometry provided, and subsequently, in default, use that information in minimising the number of integrals that need be constructed, for example, in SCF calculations. At this level, the program in most instances is capable of handling both Abelian and non-Abelian point groups on an equal footing;
2. at the *orbital* level, both at the basis function and molecular orbital (MO) level, when the symmetry characteristics of MOs will be used in optimising subsequent post Hartree-Fock calculations. This requirement is met through the internal use of symmetry-adapted basis functions. While this technique is limited to Abelian point groups, the program will automatically treat non-Abelian groups by resorting to the optimum Abelian group when handing orbital symmetry.

The following points should be noted on the implementation and possible constraints inherent in the use of symmetry:

- The TAGs used to characterise the component nuclei of the system in either the GEOMETRY or ZMATRIX directive play a vital role, in symmetry determination. They act, for example, to define the atomic number of the component nuclei and are used in establishing the effective point group symmetry of the system. Failure to appreciate the rules for TAG specification outlined in the description of these directives can lead to a considerable loss in efficiency.
- In RHF, UHF and Møller Plesset calculations GAMESS–UK will, based on the molecular point group, generate and retain only the unique integrals required, for example, in the process of constructing a 'skeletonised' Fock matrix [1]. Such a symmetry-truncated integral list is, however, *NOT* usable at present in pair-GVB, CASSCF, MCSCF, RPA or CI calculations, and again *considerable caution* should be taken when using an integral file generated in an earlier SCF run directly in a subsequent post-Hartree Fock calculation under control of the BYPASS directive.
- In geometry optimisations the point group deduced is based on the starting geometry, and is not allowed to change during the subsequent optimisation. This can lead to problems if the Z-matrix is constructed in such a way as to allow such changes to occur.
- The present implementation of both MCSCF and CI capabilities assumes that the process of symmetry adaptation is in operation. If for any reason the SCF MOs of differing irreducible representations become mixed, the post Hartree Fock calculations may prove unreliable.

1.2 The Role of the Dumpfile

GAMESS-UK makes extensive use of a number of files, the most important of which is the so-called Dumpfile that in default is routed to ED3. This file is crucial to the program and controls all restart activities. The Dumpfile is organised into variable length sections, with the user typically nominating a number of these for data storage e.g., for eigenvectors. The sections are characterised by integers (in the range 1 - 350), which are specified by the user through data input. Routine usage of the code has normally involved such specification at two points (or more) in the data input, through the VECTORS and ENTER directive. In both cases the directive is used to control the reading and writing of eigenvectors, with VECTORS used in restart jobs to specify the location of suitable vectors for input to some SCF process, and ENTER used in both startup and restart jobs to specify where generated eigenvectors are to be stored.

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 1.

The following points should be noted

- 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.

Table 1: 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

- 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 the table are described in detail at the appropriate point below.

2 Closed-Shell SCF Calculation

We wish to perform an SCF calculation at the geometry r(C-H)=1.099 A, r(C-O)=1.203 A and angle HCO=121.8^o. The geometry is specified through use of the z-matrix [2, 3], where each line of the ZMATRIX directive is responsible for specifying the nature and location of a given nucleus in terms of the position of those nuclei defined by previous lines. Note at the outset that the z-matrix TAGs used to characterise the component nuclei of the system play a vital role in characterising the system. They act, for example, to define the charge of the component nuclei and are used in establishing the effective point group symmetry of the system. The program incorporates a number of 'built-in' basis sets, with the split-valence 3-21G basis due to Pople et al [4] as the default. The following data sequence would be required in performing the SCF calculation using this default basis:

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
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
```

Note that this data sequence assumes a number of default specifications; the corresponding sequence specifying these defaults in-line would be as follows;

```
(*) DUMPFILE ED3 1
(*) MAINFILE ED2
(*) MINBLOCK ED2 1
(*) MAXBLOCK ED2 99999
(*) ADAPT ON
    TITLE
    H2CO - FULL DATA SPECIFICATION
(*) CHARGE 0
(*) MULT 1
(*) SUPER ON
    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 SV 3-21G
(*) RUNTYPE SCF
(*) SCFTYPE RHF
(*) LEVEL 1.0 5 0.3
(*) DIIS ON
(*) VECTORS ATOMS
(*) ENTER 1
```

where the default specifications which apply in the present closed-shell single point geometry calculation are indicated by a (*). In particular

1. DUMPFILE, MAINFILE, MINBLOCK and MAXBLOCK specify the file attributes. Dumpfile output is routed to ED3 commencing at block 1, while Mainfile output is to ED2 commencing at block 1.

2. CHARGE and MULT specify the charge and spin multiplicity of the system, with the default referencing a closed-shell neutral system.

3. The ADAPT directive specifies that the SCF computation is to be performed in a symmetry adapted basis.

4. The SUPER directive specifies the format to be used in generation of the two-electron integral file. The program incorporates three options, namely

   - P-Supermatrix (2J-K)
   - separate J- and K-Supermatrices (in practice 2J-K and K)
   - conventional 2-electron integral format

   In default efficiency considerations are used in deciding the appropriate format based on the particular computation to be undertaken (as defined by the SCFTYPE directive). *Considerable Caution* must be exercised when considering usage of the Mainfile produced in one phase of the computation in some subsequent phase, and specification of the SUPER directive provides some control over this usage. The default and available integral options are summarised in Table 2, where the specified defaults are those appropriate to RUNTYPE SCF.

Table 2: GAMESS–UK Integral Options as a function of SCFTYPE

SCFTYPE	Default	Available
Closed-shell SCF	P	P,J+K,2-electron integral
UHF	J+K	J+K,2-electron integral
Open-shell RHF	J+K	J+K,2-electron integral
GVB	J+K	J+K,2-electron integral
MP2	2-electron integral	2-electron integral
MP3	2-electron integral	2-electron integral
CASSCF	2-electron integral	2-electron integral
MCSCF	2-electron integral	2-electron integral

Thus, for example, attempting to use the integral file produced in default during a closed-shell SCF calculation (P-supermatrix) in a subsequent open-shell computation must be considered an invalid operation, and will lead to an error condition.

5. RUNTYPE and SCFTYPE define the computation to be carried out. RUNTYPE defines the particular task to be undertaken, while SCFTYPE specifies the form of wavefunction calculation to be employed throughout the task. RUNTYPE options are given in Table 3, while the categories of wavefunction that may be requested under control of the SCFTYPE directive are shown in Table 4.

   Note that additional directives may be required in further characterising the SCFTYPE specification. The default program options are

   ```
   RUNTYPE SCF
   SCFTYPE RHF
   ```

   i.e. single point restricted Hartree-Fock SCF computation.

6. LEVEL and DIIS define the convergence aids to apply throughout the computation. Note that the format of the LEVEL directive i.e. the number of level shifters to be specified, is dependent on the SCFTYPE setting.

7. The VECTORS directive determines the method to be used in generating a trial set of eigenvectors for the SCF calculation. The program incorporates many options; in default trial vectors are generated via the ATOMS option, involving the superposition of atomic SCF densities.

8. In default, the final set of converged vectors will be written to Section 1 of the Dumpfile, the default section for housing closed-shell SCF vectors (see Table 1). This could equally be achieved by explicit section specification (i.e. ENTER 1).

9. The following data sequence would be required in performing a minimal basis set (STO-3G) calculation at the above nuclear geometry. Note that the BASIS directive is now used to specify STO3G, while in the absence of the VECTORS directive the default ATOMS option is again used in generating a trial set of vectors.

Table 3: RUNTYPE Options Within GAMESS–UK

RUNTYPE INTEGRAL	Single point integral calculation
RUNTYPE SCF	Single point integral plus SCF calculation
RUNTYPE OPTIMIZE	Geometry optimisation (internal coordinates)
RUNTYPE OPTXYZ	Geometry optimisation (cartesian coordinates)
RUNTYPE SADDLE	Saddle point location
RUNTYPE FORCE	Force constant evaluation
RUNTYPE HESSIAN	Analytic Force constant evaluation
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 ANALYSE	Wavefunction analysis
RUNTYPE TRANSFORM	Integral transformation
RUNTYPE CI	CI calculation
RUNTYPE GF	Green's Function OVGF calculation
RUNTYPE TDA	Green's Function 2ph-TDA calculation
RUNTYPE RESPONSE	Response calculations of Excitation Energies

Table 4: SCFTYPE Specification within GAMESS–UK

SCFTYPE RHF	Restricted Hartree-Fock
SCFTYPE DIRECT	Direct-SCF
SCFTYPE UHF	Unrestricted Hartree-Fock
SCFTYPE DIRECT UHF	Direct-UHF
SCFTYPE GVB	Generalised Valence Bond
SCFTYPE DIRECT GVB	Direct-GVB
SCFTYPE MP2	2nd order Møller Plesset
SCFTYPE MP3	3nd order Møller Plesset
SCFTYPE CASSCF	Complete Active Space SCF
SCFTYPE MCSCF	2nd order MCSCF

```
TITLE
H2CO - MINIMAL STO3G BASIS - CLOSED SHELL SCF
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 STO3G
ENTER
```

The corresponding data for performing an extended, triple-zeta plus polarisation (TZVP) basis is shown below.

```
TITLE
H2CO - EXTENDED TZVP BASIS - CLOSED SHELL SCF
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 TZVP
ENTER
```

2.1 Spherical Harmonic Basis Sets

The default Cartesian angular functions (6 d, 10 f, 15 g) used throughout GAMESS-UK may now be overridden under control of the HARMONIC directive. This provides the option of using spherical-harmonic (5 d, 7 f, 9g) angular functions. Note that such usage is implemented internally through appropriate transformations, and not by computing integrals or derivative integrals over the spherical functions.

Typical usage will involve just presenting the string HARMONIC. Thus the data for performing an extended, triple-zeta plus polarisation (TZVP) spherical harmonic basis is shown below.

```
TITLE
H2CO - EXTENDED TZVP SPHERICAL HARMONIC BASIS - CLOSED SHELL SCF
HARMONIC
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 TZVP
ENTER
```

The following points should be noted:

- The HARMONIC directive, if present, should appear before the BASIS directive.

- A primary use of spherical functions is to help to eliminate problems with linear dependence.
- It is not possible in the present release of the code to employ the HARMONIC option in Table-CI calculations.

3 Closed-Shell Direct-SCF Calculation

We wish to perform a direct-SCF calculation equivalent to that above. A valid data sequence for performing such a calculation is shown below:

```
TITLE
H2CO - EXTENDED TZVP BASIS - DIRECT-SCF
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 TZVP
SCFTYPE DIRECT RHF
ENTER
```

Note that the SCFTYPE directive is now required in requesting the DIRECT option. The third parameter on the data line, RHF, points to the particular category of wavefunction required i.e. closed-shell SCF. The options supported in direct-mode include RHF, UHF, GVB and MP2 (see below). Omitting this parameter and presenting just the line SCFTYPE DIRECT will lead to the default option of a direct-RHF calculation for closed-shell systems. Note that the default file output in such calculations is confined to ED3 and ED7, the Dump- and Scratch-file respectively.

4 RHF Open Shell Calculation

A restricted Hartree-Fock calculation is to be performed on the formaldehyde cation at the geometry specified above. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - OPEN SHELL RHF
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
ENTER
```

In the absence of the RUNTYPE and SCFTYPE directives, a single-point restricted Hartree-Fock calculation will be performed. The following points should be noted:

1. The CHARGE and MULT directives are now required to define the system attributes.

2. In an open-shell system. the default SCF calculation uses the restricted Hartree-Fock RHF method, with the open shell occupancy assumed to correspond to the high-spin configuration. If the user wishes to modify this occupancy and define the shell structure characterising the wavefunction, then the OPEN directive must be used to specify the electronic distribution in the open-shell orbitals. Using the OPEN directive in the present case would lead to the following data sequence:

   ```
   TITLE
   H2CO+ - 2B2 - 3-21G DEFAULT BASIS - OPEN SHELL RHF
   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
   OPEN 1 1
   ENTER
   ```

   The current setting is self-evident - one orbital containing one electron. Again, in more general cases the OPEN directive must be used to define the shell structure characterising the Hartree-Fock wavefunction.

3. Two sets of eigenvectors are generated in an open-shell RHF calculation, the non-canonicalised (locked) eigenvectors that are used during the SCF process, and a second set, the canonicalised vectors, which are generated on termination of the SCF process. The latter exhibit energy weighting in the virtual manifold, and act as the most obvious starting point for a post Hartree-Fock computation.

4. Two sections will be used to house these eigenvectors. In default the non-canonicalised vectors will be written to section 4 of the Dumpfile, while the canonicalised vectors will be written to section 5 (see Table 1).

5. Explicit specification of these sections thus requires two integers on the ENTER directive. Presenting the data line

   ```
   ENTER 4 5
   ```

   will result in the same eigenvector section storage as the default.

6. In the absence of the SUPER directive, the default Mainfile format for an open-shell RHF calculation (J+K) will apply.

While the above data structure appears the most straightforward way of accomplishing the computation, it relies on the initial trial eigenvector guess (generated through the default ATOMS option) providing the required open-shell electronic configuration. Such a situation is unlikely to hold in all cases (note that the present ATOMS implementation is much more likely to yield this configuration than the alternative VECTORS option EXTGUESS), and a more reliable way to proceed is to use the closed-shell vectors generated in section 2 as the starting guess. Such a route requires,

- performing the calculations of §2 and §4 under control of the same Dumpfile.
- in the interests of efficiency, we require the specification of a consistent format for the Mainfile, thus allowing integral generation to be bypassed in section 4, with both closed- and open-shell calculations using the same integral file.

Taking these considerations into account, the following data sequence for the examples of §2 and §4 should be presented, where the Dumpfile created in the closed-shell case is subsequently allocated as ED3 in the open-shell calculation. It is assumed of course that both Mainfile and Dumpfile produced in §2 have been saved.

Run I

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
SUPER FORCE
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
```

Run II

```
RESTART
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - OPEN SHELL RHF
CHARGE 1
MULT 2
BYPASS
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
```

The closed-shell data file now contains the SUPER directive, and by virtue of the FORCE keyword, instructs integral generation to proceed through (J+K) Supermatrices rather than through the default P-supermatrix. The following modifications to the data for the open shell calculation should be noted;

1. The RESTART directive indicates that the calculation is to be driven from a known Dumpfile.
2. The BYPASS directive indicates that the integral generation phase of the SCF processing is to be bypassed - the file generated in **Run I** is to be used. Again , this bypassing is *only* made possible by virtue of requesting the appropriate integral format (J+K) at generation time at the outset of the closed-shell run.
3. Note that the default VECTOR and ENTER sections are still applicable in this example. In deciding on an appropriate set of eigenvectors to initiate an open-shell calculation, the program will in a RESTART job first examine the Dumpfile to see if the default open-shell VECTORS section have been written to by a previous job. If not, as in the present example, the closed-shell default vectors section will be used to provide the starting guess (as written to in **Run I**). If this section is not present, the calculation will revert to an atomic GUESS. The open-shell calculation will proceed to use the default open-shell vectors specification for output of the eigenvectors (see Table 1). Thus the data sequence above is equivalent to presenting the data lines;

```
VECTORS 1
ENTER 4 5
```

4. Assume that we now wish to perform a subsequent calculation on the $^2B_1$ state of the cation. An examination of the input SCF orbitals (the energy-weighted orbitals from the $^2B_2$ calculation) reveals the following ordering of the doubly occupied orbitals (DOMOs)

M.O.	1	2	3	4	5	6	7	8
symmetry	$1a_1$	$2a_1$	$3a_1$	$4a_1$	$1b_2$	$5a_1$	$1b_1$	$2b_2$

The input orbitals must be arranged such that the doubly occupied manifold precedes the open shell orbitals (grouped according to shell). In the present case, with 7 DOMOS and 1 singly occupied orbital, we must reorder the input MOs such that the $1b_1$ orbital occupies the 8th position in the input list. Such a reordering is accomplished by the SWAP directive, as in the following data sequence:

```
RESTART
TITLE
H2CO+ - 2B1 - 3-21G DEFAULT BASIS - OPEN SHELL RHF
CHARGE 1
MULT 2
BYPASS
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
```

```
SWAP
7 8
END
ENTER
```

Note that the ^{2}B$_1$ eigenvectors will again be stored in the default open-shell vectors sections (4 and 5), thus overwriting the ^{2}B$_2$ orbitals generated in **Run II** above. To keep the orbitals from both open-shell calculations will now require explicit specification of the sections to be used in storing the vectors. this could be achieved as follows;

```
VECTORS 5
SWAP
7 8
END
ENTER 6 7
```

where the initial vectors will be the energy ordered ^{2}B$_2$ MOS, and the ^{2}B$_1$ vectors will be written to sections 6 (non-canonicalised) and 7 (energy ordered).

4.1 Direct-RHF Open Shell Calculation

A valid data sequence for performing a direct restricted Hartree-Fock calculation on the formaldehyde cation is shown below:

```
TITLE
H2CO+ - 2B2 - 3-21G BASIS - OPEN SHELL DIRECT-RHF
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 DIRECT
ENTER
```

Note that in contrast to the conventional open shell calculation, the SCFTYPE directive must be presented to request the DIRECT requirement. The third parameter on the SCFTYPE line is not required however, given the default for an open shell system is to perform a restricted Hartree-Fock calculation. Equally the default high-spin open shell occupancy means that the OPEN defaults apply, so that the directive is not required. The above data is thus equivalent to the following data sequence:

```
TITLE
H2CO+ - 2B2 - 3-21G BASIS - OPEN SHELL DIRECT-RHF
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 DIRECT RHF
OPEN 1 1
ENTER
```

In the absence of the RUNTYPE directive, a single-point Hartree-Fock calculation will be performed. Note that the SUPER directive is not relevant in a direct calculation, given that the integrals are re-computed on each iterative cycle of the SCF. Equally, use of the BYPASS directive in such calculations to avoid computation of some pre-computed integral list has no real meaning, and should not be used.

The following data sequence would be required to use the closed shell vectors as a starting guess for the open shell calculation, where each phase of the calculation is conducted in DIRECT fashion (cf. the data sequences in section §2 and §4).

Run I

```
TITLE
H2CO - 3-21G BASIS - CLOSED SHELL DIRECT-SCF
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 DIRECT
ENTER
```

Run II

```
RESTART
TITLE
H2CO+ - 2B2 - 3-21G BASIS - OPEN SHELL DIRECT RHF
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 DIRECT
ENTER
```

Assume that we now wish to perform a subsequent calculation on the $^2B_1$ state of the cation. Again, the input orbitals must be arranged such that the doubly occupied manifold precedes the open shell orbitals. In the present case, with 7 DOMOS and 1 singly occupied orbital, we must reorder the input MOs such that the $1b_1$ orbital occupies the 8th position in the input list through the SWAP directive, thus

```
RESTART
TITLE
H2CO+ - 2B1 - 3-21G BASIS - OPEN SHELL DIRECT-RHF
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 DIRECT
SWAP
7 8
END
ENTER
```

5 UHF Calculation on the formaldehyde cation

The simplest way of conducting an unrestricted Hartree Fock (UHF) calculation is exemplified by the following data sequence for the formaldehyde cation;

```
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - UHF 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
ENTER
```

The following points should be noted.

1. The SCFTYPE directive is now required in requesting the UHF option. Note that the OPEN directive used in RHF calculations should not be introduced when requesting a UHF calculation; if present, it will be ignored.

2. 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 1).

3. Explicit specification of these sections thus requires two integers on the ENTER directive. Presenting the data line

```
ENTER 2 3
```

will result in the same eigenvector section storage as the default.

4. In the absence of the SUPER directive, the default Mainfile format for a UHF calculation (J+K) will apply.

While the above data structure appears the most straightforward way of accomplishing the computation, it relies on the initial trial eigenvector guess (generated through the default ATOMS option) providing the required open-shell electronic configuration. Such a situation is unlikely to hold in all cases; as with open-shell restricted Hartree Fock calculations, a more reliable way to proceed is to use an appropriate set of closed-shell vectors as the starting guess. Assuming again that the closed-shell calculation of §2 has been successfully completed, the following data sequence would be required in performing a UHF calculation on the cation, using the (J+K) supermatrix and eigenvectors from the closed-shell run.

```
RESTART
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - UHF CALCULATION
CHARGE 1
MULT 2
BYPASS
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
ENTER
```

Note that the default VECTOR and ENTER sections are still applicable in this example. In deciding on an appropriate set of eigenvectors to initiate an open-shell UHF calculation, the program will first examine the Dumpfile to see if the default UHF VECTORS sections have been written to by a previous job. If these sections exist, the program will utilise the resident vectors as a starting point for the current UHF calculation. If not, as in the present example, the closed-shell default vectors section will be used to provide the starting guess for both α-spin and β-spin MOs (as written to in **Run I**). If this section is not present, the calculation will revert to an atomic GUESS. The UHF calculation will proceed to use the default open-shell UHF vectors specification for output of the eigenvectors (see Table 1). Thus the data sequence above is equivalent to presenting the data lines;

```
VECTORS 1
ENTER 2 3
```

Had the results of the RHF computation in §4 been available, then the data line

```
VECTORS 5
```

would have provided the canonicalised RHF orbitals as a starting point for the UHF iterative process. Assuming that the SCF calculations of §2, §4 and §5 been performed in sequence, the status of the user–sections on the Dumpfile would appear as follows, with each of the vectors available for subsequent analysis (see Part 8).

Section	Contents
1	X^1A_1 Closed-shell RHF vectors
2	$^2B_2$ α-spin UHF vectors
3	$^2B_2$ β-spin UHF vectors
4	$^2B_2$ Open-shell RHF vectors
5	$^2B_2$ Canonicalised Open-shell RHF vectors
6	$^2B_1$ Open-shell RHF vectors
7	$^2B_1$ Canonicalised Open-shell RHF vectors

5.1 Direct-UHF Calculation on the formaldehyde cation

The following data sequence would be required in performing a direct-UHF calculation on the cation, using the eigenvectors from the closed-shell run. Note that the BYPASS directive shown in the example above is no longer appropriate.

```
RESTART
TITLE
H2CO+ - 2B2 - 3-21G BASIS - DIRECT-UHF 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 DIRECT UHF
VECTORS 1
ENTER
```

Note the revised form of the SCFTYPE directive requesting the direct-UHF option. The default Dumpfile sections are again used for storage of the α-spin (section 2) and β-spin (section 3) orbitals. Had the results of the RHF computation in §4 been available, then the data line

```
VECTORS 5
```

would have provided the canonicalised RHF orbitals as a starting point for the UHF iterative process.

6 GVB Calculation on the formaldehyde molecule

Before considering the detailed data input, we should draw attention to certain aspects of,

Table 5: Orbital Numbering in H_2CO

MO Sequence Number	Symmetry	MO Sequence Number	Symmetry
1	$1a_1$	9	$2b_1$
2	$2a_1$	10	$6a_1$
3	$3a_1$	11	$3b_2$
4	$4a_1$	12	$7a_1$
5	$1b_2$		
6	$5a_1$		
7	$1b_1$		
8	$2b_2$		

- the ordering expected of the trial input molecular orbitals, and
- the treatment of molecular symmetry within the GAMESS–UK program.

The following points should be noted:

1. In the general case of a GVB [5] calculation on an open-shell system, comprising **m** doubly-occupied orbitals, **n** open shell orbitals and **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. In the present case, an examination of the closed-shell SCF-MOs reveals the ordering shown in Table 5.

   Thus to perform a GVB-1/PP [5] calculation for the ($1b_1$, $2b_1$) pair, we must reorder the input MOs, exchanging the $2b_2$ and $1b_1$ MO i.e., the 7th and 8th input orbitals. No reordering of the virtual orbitals is necessary, since the $2b_1$ MO already occupies the 9th position in the list.

2. In the RHF and UHF examples above, 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 [1]. Such a symmetry-truncated integral list is, however, *NOT* usable in pair-GVB, CASSCF, MCSCF or CI calculations, and again *considerable caution* should be exercised when considering use of an integral file generated in an earlier SCF run directly in a subsequent post-Hartree Fock calculation under control of the BYPASS directive. There are several ways to proceed in such cases:

- simply omit the BYPASS directive in the subsequent run. The program will choose the correct format based on the SCFTYPE specification, and regenerate the integral file in the appropriate way.
- In many instances this regeneration process is too expensive, and the user must suppress the 'skeletonisation' process when the integral file is first generated. This is again achieved under control of the SUPER directive. Specifically the data line

  ```
  SUPER FORCE NOSYM
  ```

  presented in the initial closed-shell calculation would result in a full (J+K) supermatrix of direct use in a subsequent GVB calculation, although of course increasing the cost of the original SCF computation.
- An examination of the available integral options as a function of SCFTYPE (Table 2) reveals that the only option covering all possible wavefunction activities is the conventional two-electron integral format. If the use of symmetry is suppressed, then such a file would also be suitable for direct input to the integral transformation program preceding MRD-CI or DIRECT-CI calculations. This format may be requested in all SCF computations by presenting the data line

  ```
  SUPER OFF NOSYM
  ```

  Note that this directive must not only be specified in the run that generated the integrals, but in *all* subsequent runs in which the file is used.
- Note that this suppressed symmetry integral file would not be usable in Møller Plesset or analytic derivative calculations. The required format in such cases requires a skeletonised integral list, and follows from presenting the data line

  ```
  SUPER OFF
  ```

The following data sequences would perform the initial closed shell and subsequent GVB-1/PP calculation. Note the form of the SCFTYPE directive in the GVB run - the integer specified after the GVB keyword indicates the number of GVB pairs - in the present case, just 1.

Closed-shell SCF

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
SUPER FORCE 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
```

GVB/1-PP Data

```
RESTART
TITLE
H2CO  - GVB/1-PP - 3-21G BASIS - (1B1->2B1*)
BYPASS
SUPER FORCE 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
SCFTYPE GVB 1
SWAP
7 8
END
ENTER
```

Note that the GVB module uses the same Dumpfile sections for storage of the locked eigenvectors (section 4) and energy ordered vectors (section 5) as the open-shell RHF module.

6.1 Direct-GVB Calculation on the formaldehyde molecule

The following data sequences would perform in direct fashion the initial closed shell and subsequent GVB-1/PP calculation shown above. Note the form of the SCFTYPE directive in the GVB run, and the appearance of the DIRECT keyword. The integer specified after the GVB keyword again indicates the number of GVB pairs - in the present case, just 1.

Closed-shell SCF

```
TITLE
H2CO - 3-21G BASIS - DIRECT CLOSED SHELL SCF
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 DIRECT
ENTER
```

GVB/1-PP Data

```
RESTART
TITLE
```

```
H2CO  - DIRECT GVB/1-PP - 3-21G BASIS - (1B1->2B1*)
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 DIRECT GVB 1
SWAP
7 8
END
ENTER
```

7 ECP Calculation on the formaldehyde molecule

We outline below the data for performing valence-only molecular orbital treatments using both the semi-local [6] and non-local [7] formalisms. Note that in the non-local procedure the required library of pseudopotentials is held on a Library File, which in default is assumed to be available on ED0 commencing at block 1. Overriding this default is described under the PSEUDO directive. No such file is required when performing semi-local ECP calculations.

7.1 Non-Local Pseudopotential Calculation

The following data sequence would be required in performing a non-local ECP calculation:

```
TITLE
H2CO - 1A1 - NON-LOCAL ECP CALCULATION
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 ECPDZ
PSEUDO NONLOCAL
O O
C C
ENTER
```

The following points should be noted:

- The non-local pseudopotential is requested on the PSEUDO directive: the subsequent data lines of this directive allocate an ECP stored in the Library to the atoms specified in the z-matrix through TAG specification. This explains, hopefully, the somewhat confusing syntax above: the library ECPs for carbon and oxygen are tagged as C and O respectively i.e., the same tags applied in the z-matrix. The first TAG on such a data line refers to the library TAG, all subsequent fields to the unique nuclei tags specified in the z-matrix.
- the ECPDZ tag on the BASIS line requests use of a double zeta contraction of the appropriate library of ECP basis sets [6].

- in the absence of the VECTORS directive, the ATOMS option is requested by default. Note that at present only the ATOMS and HCORE options are available in ECP calculations as the basis-specific options such as EXTGUESS and MINGUESS have not been modified to account for the valence only nature of the computation.

7.2 Local Pseudopotential Calculation

In contrast to Version 6.2 of the code, where only two libraries of ECPs (CEP and the Hay-Wadt ECP, now code-named LANL) were included, seven semi-local ECP sets are now available, code-named as follows;

1. CEP or SBKJC – available in Version 6.2 of the code, the the Compact Effective Potentials (CEPs) are due to (i) Stevens et al [8] for the elements Li-Ar, (ii) Stevens at al [9] for the elements K-Rn, and (iii) Cundari et al [10] for the Lanthanides.
2. LANL – available in version 6.2 of the code, the LANL ECPs are due to Hay and Wadt [6] and cover the elements Na-Bi.
3. LANL2 – The Hay and Wadt ECPs, with the inner-valence forms used for transition metals etc. These are as provided in the Gaussian and NWChem suite of programs.
4. CRENBL – The small core potential due to Christiansen et al [11]. These ECPs are sometimes referred to as shape consistent because they maintain the shape of the atomic orbitals in the valence region.
5. CRENBS – The averaged relativistic, large core ECPs due to Ermler and co-workers [12].
6. STRLC – The Stuttgart relativistic, large core ECP due to Preuss at al. [13].
7. STRSC – The Stuttgart relativistic, small core ECP due to Preuss at al. [14].

A full list of the elements for which ECPs and associated basis sets are available for each of the seven library sets above is given in Tables 5 and 6 of Part 3 of the manual. Considering the non-local ECP example given above, the following data sequence would be required to perform the corresponding local ECP calculation using the LANL2 ECP:

```
TITLE
H2CO - 1A1 - LOCAL LANL2 ECP CALCULATION
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 ECP LANL2
PSEUDO ECP
O LANL2 O
C LANL2 C
ENTER
```

The local ECP is requested through the ECP field on the PSEUDO directive. The subsequent data lines of this directive again allocate an ECP stored in one of the ECP Libraries (program-resident) to the atoms specified in the z-matrix through TAG specification. In this case the LANL2 parameter requests the ECP due to Hay and Wadt [6]; the LANL2 library ECPs for carbon and oxygen are again tagged as C and O respectively.

Note that compatibility with the previous versions of the code has been maintained, so that presenting the following data sequence:

```
TITLE
H2CO - 1A1 - LOCAL ECP CALCULATION
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 ECPDZ
PSEUDO ECP
O O
C C
ENTER
```

will, as before, result in a LANL ECP calculation with the DZ ECP basis set due to Hay and Wadt.

8 In-core SCF Calculations

In all the conventional SCF calculations described above, we have assumed that the 2–electron integral file is written to disk (the Mainfile), prior to repeated processing associated with the SCF iterations. One alternative, confined at present to closed–shell SCF calculations, is to use the direct-SCF algorithm, which removes the I/O processing associated with conventional SCF at the cost of an increased CPU requirement involved in recalculating the integrals on each SCF iteration. In-core SCF calculations provide a further alternative, where the I/O overhead associated with the conventional route is removed by mapping the two–electron integral list directly to memory, rather than disk storage. While obviously limited to small–medium sized molecules, even on machines with sizable amounts of central memory i.e., 128 MBytes or more, this technique can provide a significant increase in processing efficiency, particularly on those machines with a poor I/O subsystem.

In–core SCF calculations are requested by modifications to the MFILE directive. Two possible options are provided;

1. The simplest option is to present the data line

   ```
   MFILE MEMORY
   ```

   requesting that the complete 2–electron integral file be routed to memory, rather than disk. Note that the program will, in the presence of the above data line, use a rather conservative algorithm to determine whether sufficient central memory is available to house the complete integral file; if this memory is judged not to be available, the conventional I/O route will be followed, with integrals routed to ED2.

2. A more flexible option is to nominate a data set to receive any integrals that will not fit into local memory, thus

   ```
   MFILE MEMORY ED4
   ```

   whereby the data set assigned as ED4 will be used to store those integrals that exceed the local available memory capacity.

Note also the following;

- this capability is at present limited to UNIX implementations of the code;
- in–core processing of the integral file is available within all SCFTYPEs, RUNTYPEs and integral file formats;
- such jobs may not be restarted, since the memory–resident integral file is not, at present, routed to disk.
- the data set ED2 *cannot* be used as the memory backup file, since the program uses the internal tables associated with this file to map memory resident integrals i.e. the data line

  ```
  MFILE MEMORY ED2
  ```

  is invalid.

The following data sequence would be required in performing an in–core SCF calculation on H_2CO;

```
TITLE
H2CO - 3-21G DEFAULT BASIS - IN-CORE SCF
MFILE MEMORY
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
```

9 CASSCF Calculations

The present release of GAMESS–UK contains two distinct modules for performing MCSCF calculations, namely:

- a CASSCF-only module, hereafter referred to as CASSCF

- a more general MCSCF module, capable also of performing CASSCF calculations [15].

The first module is invoked through the SCFTYPE CASSCF option, the second through the SCFTYPE MCSCF option. In choosing between the two alternative codes, the user should note that the MCSCF module is significantly more flexible and efficient than the original CASSCF module, capable of handling far larger MCSCF expansions. Geometry optimisations and numerical force constant calculations are possible with both options. This section will deal with driving the CASSCF option, with the following section dealing with MCSCF execution.

We wish to perform a CASSCF [16] calculation on the X^{1}A$_1$ state of formaldehyde, using a full-valence criterion in specifying the active space. Thus, in addition to the doubly occupied SCF MOs, 1a$_1$-5a$_1$, 1b$_1$ and 1b$_2$-2b$_2$, the formally vacant SCF virtual orbitals, the 6a$_1$ and 7a$_1$, 2b$_1$ and 3b$_2$ MOs are to be permitted variable occupancy in the MCSCF treatment. Restricting the 1a$_1$ (O1s), 2a$_1$ (C1s) and 3a$_1$ (O2s) MOs to be doubly occupied throughout yields a CASCI secular space of 1408 configurations for this so-called '10 electron in 9 orbital' CASSCF calculation. We again assume that the calculation is to proceed in two stages, with initial generation of the closed-shell SCF orbitals, followed by the CASSCF computation. As mentioned above, generation of a valid Mainfile for direct use in the CASSCF calculation requires the data line

```
SUPER OFF NOSYM
```

in the closed-shell run, hence allowing BYPASS'ing in the MCSCF computation.

Closed shell SCF Data

```
TITLE
H2CO - 3-21G  CLOSED SHELL SCF  -  SUPPRESS SKELETONISATION
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
ENTER
```

CASSCF Data

```
RESTART
TITLE
H2CO  - CASSCF   - 3-21G BASIS - (10E IN 9 M.O.)
BYPASS
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 CASSCF
CONFIG PRINT
```

```
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

Note the additional CONFIG directive specified in the CASSCF data, responsible for defining;

1. the active orbital space for the calculation. This involves classifying the input MOs as either primary or secondary in character, with the primary orbitals classified by type;
2. an initial reference configuration (typically the Hartree-Fock configuration) to be used in generating the complete CI expansion, and hence the loop-formulae tape. This involves assigning occupation numbers, through orbital TAGs, to the primary orbitals;

The active orbital space and initial reference configuration are defined by CONFIG - each orbital in the primary space (MOs 1-12) is classified by type, where the following types are introduced:

- FZC – frozen core orbital, an orbital which will remain doubly occupied in all configurations;
- DOC – orbitals in the active space which are formally doubly occupied, and which will be permitted variable occupancy in the MCSCF treatment;
- UOC – orbitals in the active space which are formally unoccupied, corresponding to SCF virtual MOs, but which will be permitted variable occupancy in the MCSCF.

Other valid orbital TAGs used in characterising open-shell configurations include ALP, AOS and BOS (see 4.3.2).

The following points should be noted:

- An examination of the input SCF MOs reveals that the 12 orbitals to be included in the primary space correspond to the first 12 SCF MOs. This may not always be the case, and the user may have to resort to the SWAP directive to ensure the primary MOs occur first in the list. Note also that the FZC orbitals must precede the orbitals permitted variable occupancy in the active list.
- Note the PRINT keyword on the CONFIG directive. This requests output of the complete list of configurations, characterised by occupation number pattern.
- Two sets of eigenvectors are generated in a CASSCF calculation, the non-canonicalised CASSCF MOs that are used during the CASSCF process, and a second set, the canonicalised vectors, which are generated on termination of the CASSCF process. The latter exhibit energy weighting in the virtual manifold, and act as the most obvious starting point for a post Hartree-Fock computation.
- Two sections will be used to house these eigenvectors. 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 1). Note that the latter section also contains the current CI coefficients.

- Explicit specification of these sections thus requires two integers on the ENTER directive. Presenting the data line

  ```
  ENTER 6 7
  ```

  will result in the same eigenvector section storage as the default.

- The present CASSCF module [16] incorporates several optimisation techniques, each of which tends to be most effective at differing stages of the MCSCF procedure. We note here that Super-CI, 2-step Newton-Raphson (NR) and 1-step Newton-Raphson optimisation may be requested. In previous releases of GAMESS-UK, the user was responsible for determining just which of the possible optimisation methods was to apply at each CASSCF iteration, and when to carry out explicit Hessian construction. The optimisation techniques were controlled by the SUPERCI, NEWTON and HESSIAN directives. In the current release the appropriate method is chosen dynamically, and the user need no longer drive this process; however, such control is still possible if the dynamic method runs into trouble. Thus the following data sequence would be required to force Super-CI optimisation on cycles 1-7 of the iterative process, and 2-step Newton Raphson on cycles 8-20, with Hessian construction conducted on each NR cycle.

  ```
  RESTART
  TITLE
  H2CO  - CASSCF   - 3-21G BASIS - (10E IN 9 M.O.)
  BYPASS
  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 CASSCF
  CONFIG PRINT NOSORT
  FZC 1 TO 3
  DOC 4 TO 8
  UOC 9 TO 12
  END
  SUPERCI 1 TO 7
  NEWTON 8 TO 20
  HESSIAN 8 TO 20
  ENTER
  ```

  Note the NOSORT keyword now appearing on the CONFIG directive. This parameter deactivates generation of the reordered Loop Formulae tape (ED10) - this file is only required during 1-step Newton Raphson optimisation, which has not been requested here. The SIMUL Directive may be used to specify the 1-step NR method.

10 MCSCF Calculation

We now consider the data input requirements for the 2nd-order MCSCF module. We again wish to perform a CASSCF calculation on the X^1A_1 state of formaldehyde, using a full-valence criterion in specifying the active space. We assume that the calculation is to proceed in two stages, with initial generation of the closed-shell SCF orbitals, followed by the MCSCF computation. As mentioned above, generation of a valid Mainfile for direct use in the MCSCF calculation requires the data line

```
SUPER OFF NOSYM
```

in the closed-shell run, hence allowing BYPASS'ing in the MCSCF computation.

Closed shell SCF Data

```
TITLE
H2CO - 3-21G  CLOSED SHELL SCF  -  SUPPRESS SKELETONISATION
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
```

MCSCF Data

```
RESTART
TITLE
H2CO  - CASSCF   - 3-21G BASIS - (10E IN 9 M.O.)
SUPER OFF NOSYM
NOPRINT
BYPASS
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 MCSCF
THRESH 4
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
PRINT ORBITALS VIRTUALS NATORB
CANONICAL 10 FOCK DENSITY FOCK
ENTER
```

Data input characterising the MCSCF calculation is introduced by the MCSCF directive, and comprises the PRINT, ORBITAL and CANONICAL directives. This data defines:

1. the active orbital space for the calculation. This involves classifying the input MOs as either primary or secondary in character, with the primary orbitals classified by type;
2. an initial reference configuration (typically the Hartree-Fock configuration) to be used in generating the complete CI expansion. This involves the explicit assignment of occupation numbers and symmetries, through orbital TAGs, to the primary orbitals.

Both definitions are controlled by the ORBITAL directive - each orbital in the primary space (MOs 1-12) are classified by type, where the following types are introduced:

- COR - core orbital, an orbital which will remain doubly occupied in all configurations. Note the contrast to the CASSCF module: the FZC tag may now be used to specify an orbital that remains strictly frozen at its input expansion.
- DOC - orbitals in the active space which are formally doubly occupied, and which will be permitted variable occupancy in the MCSCF treatment.
- UOC - orbitals in the active space which are formally unoccupied, corresponding to SCF virtual MOs, but which will be permitted variable occupancy in the MCSCF.

Other valid orbital TAGs used in characterising open-shell configurations include ALP and BET (see 4.3.2). The integer tag appended to each three–character identifier specifies the symmetry of the active orbital, and corresponds to the irreducible representation (IRrep) of the MO, information generated through the symmetry adaption process. Considering the output from the closed shell SCF calculation on H_2CO, in particular the symmetry adapted basis set information provided prior to generating the trial set of vectors:

```
==============================
IRREP  NO. OF SYMMETRY ADAPTED
       BASIS FUNCTIONS
==============================
  1          12
  2           4
  3           6
==============================
```

and the list of MOs printed at convergence:

```
===============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
===============================================
   1     1    -20.48361193          2.0000000
   2     1    -11.28387037          2.0000000
   3     1     -1.42107323          2.0000000
   4     1     -0.86257503          2.0000000
   5     3     -0.69835324          2.0000000
   6     1     -0.63657557          2.0000000
   7     2     -0.52867413          2.0000000
   8     3     -0.43116746          2.0000000
   9     2      0.14867301          0.0000000
  10     1      0.27270470          0.0000000
  11     3      0.36811563          0.0000000
  12     1      0.45408083          0.0000000
  13     2      0.93005840          0.0000000
  14     3      1.01802548          0.0000000
  15     1      1.04135899          0.0000000
  16     1      1.15883011          0.0000000
  17     3      1.26989260          0.0000000
  18     1      1.56946779          0.0000000
  19     2      1.86711355          0.0000000
  20     1      1.90074847          0.0000000
  21     3      1.98283110          0.0000000
  22     1      3.32677642          0.0000000
===============================================
```

The following ORBITAL specification:

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

follows straightforwardly from the above output. Note that a list of irreducible representations (IRreps) and their associated indices for each of the abelian point groups are given in Table 7.

The following points should be noted:

- An examination of the input SCF MOs reveals that the 12 orbitals to be included in the primary space correspond to the first 12 SCF MOs. If this not the case, the ordering specified by the ORBITAL directive will be imposed by the program so, in contrast to the CASSCF module, the user need not resort to the SWAP directive. Note however that the FZC orbitals must precede the orbitals permitted variable occupancy in the active list.
- The integer specified on the CANONICAL directive defines the section number on the Dumpfile for output of the MCSCF natural orbitals. Note that the two data lines

  ```
  PRINT ORBITALS VIRTUALS NATORB
  CANONICAL 10 FOCK DENSITY FOCK
  ```

  now define the defaults (Version 6.3 onwards) and may be omitted, when the MCSCF natural orbitals will be written to section 10 of the Dumpfile. Such orbitals may be subsequently retrieved in, for example, Direct-CI calculations.
- Two sections on the Dumpfile that may be specified via the ENTER directive are again written during an MCSCF calculation, the first containing the non-canonicalised orbitals that are used during the MCSCF process, and the second a variety of restart information generated on termination of the MCSCF process. Both sections contain information that is used internally within the MCSCF module, and in contrast to the CASSCF module, are not designed for reference by other modules within GAMESS-UK. Now the most obvious starting point for a post Hartree-Fock computation are the MCSCF natural orbitals that are written to the section specified by the CANONICAL directive (see above).

- In default the non-canonicalised MCSCF vectors will be written to section 8 of the Dumpfile, while the restart information will be written to section 9 (see Table 1).
- Explicit specification of these sections thus requires two integers on the ENTER directive. Presenting the data line

  ```
  ENTER 8 9
  ```

  will result in the same eigenvector section storage as the default.

11 DFT Calculations

Background material on Density Functional Theory (DFT) and a description of the GAMESS-UK implementation and issues relating to the choice of functionals, integration grids and associated performance, together with a full description of the available directives are given in Part 4 of the manual. The user is advised to consult that material prior to using the code.

11.1 The DFT Directive and Default Settings

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)
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; omitting the directive 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 [17]
- the Lee, Yang and Parr (LYP) correlation functional [18]
- 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 [19] and Gauss-Legendre angular grid, using the SSF weighting scheme with screening [21] and MHL angular grid pruning [20]. Note that this choice corresponds to the "QUADRATURE MEDIUM" setting described below.

The following points should be noted;

1. DFT calculations on closed shell systems may be performed using either the conventional (SCFTYPE RHF) or direct-SCF (SCFTYPE DIRECT RHF or SCFTYPE DIRECT) modules.
2. DFT calculations on open shell systems are only available using conventional UHF (SCFTYPE UHF) or direct-UHF (SCFTYPE DIRECT UHF) modules. Restricted ROHF calculations are not possible in the current release of the code.
3. Only conventional 2-electron integral format is available when performing either closed- or open-shell DFT calculations. Neither P-supermatrix nor separate J- and K-Supermatrices are currently supported.

11.2 DFT Basis Sets

In addition to the standard basis sets available (see Part 3), the polarized DFT orbital basis sets due to Godbout et al. [29] may be requested through simple keyword specification on the BASIS directive. Three such sets are available through this mechanism, the DZVP, DZVP2 and TZVP basis; each is requested in straightforward fashion by a BASIS directive of the form:

```
BASIS DFT DZVP

BASIS DFT DZVP2

BASIS DFT TZVP
```

Omitting the third string and presenting the data line "BASIS DFT" will realise the DZVP basis.

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 [17] and Lee, Yang and Parr correlation (LYP) correlation energy functional [18]. Over-riding this default may be achieved through the following DFT keywords:

- HFEX; The keyword HFEX selects the Hartree-Fock exchange term as the exchange functional.
- BECKE88; The keyword BECKE88 selects the default Becke88 exchange functional This is a gradient-corrected exchange energy functional with correct *1/r* asymptotic behaviour of the exchange-energy density.
- LYP; The keyword LYP selects the default Lee, Yang and Parr correlation energy functional
- NOCORR; The keyword directive NOCORR selects the null functional for the correlation energy, i.e. it switches off all correlation energy functionals
- B3LYP; The keyword B3LYP selects the hybrid exchange-correlation energy functional due to Becke [22]
- S-VWN or SVWN; The keyword S-VWN or SVWN selects the LDA exchange functional and the Vosko, Wilk, and Nusair (VWN) correlation functional [23]
- B-P86 or BP86; The keyword B-P86 or BP86 selects the Becke88 exchange energy functional [17] and the Perdew 1986 gradient corrected correlation functional [24].
- B97; The keyword B97 selects the Becke97 hybrid exchange-correlation energy functional [25]
- B97-1; The keyword B97-1 selects the Becke97 hybrid exchange-correlation energy functional as it was reparametrised by Hamprecht et al. [26, 25]
- B97-2; The keyword B97-2 selects the Becke97 hybrid exchange-correlation energy functional as it was reparametrised by Wilson et al. [27, 25]
- HCTH; The keyword HCTH selects the Hamprecht, Cohen, Tozer and Handy exchange-correlation energy functional [26]

11.5 Specification of Integration Grids

While a large number of options are available in specifying possible integration grids (see Part 4), the inexperienced user is strongly advised to use just the QUADRATURE directive for this purpose.

11.5.1 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 [19] and Gauss-Legendre angular grid, using the SSF weighting scheme with screening [21] and MHL angular grid pruning [20]. 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
```

Directives permitting a more detailed specification of the quadrature grids are described in part 4 of the manual.

11.6 Coulomb fitting

The efficiency of DFT calculations on medium sized molecules can be enhanced by avoiding the evaluation of 4-center 2-electron integrals. This requires the user (i) to request a functional without Hartree-Fock exchange, and (ii) to fit the total electron density to an auxiliary basis set. The Coulomb energy contributions can then be evaluated using the fitted density, requiring at worst 3-center 2-electron integrals. The basic theory behind this has been published by Dunlap *et al.* [28].

The technology implemented allows Coulomb fitting to be used in both energy and gradient evaluations using the JFIT and JFITG directives respectively. Furthermore the number of 3-center 2-electron integrals that will be evaluated can be reduced using the Schwarz inequality to discard small integrals. Below is an example of a formaldehyde calculation using Coulomb fitting, with explicit specification of the auxiliary basis, the DGauss A1 set [29]:

```
TITLE
H2CO - 6-31G/BLYP CLOSED SHELL DFT WITH 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 JFITG
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
```

The following points should be noted:

- Specification of the Auxiliary Fitting basis under control of the JBAS directive is based upon the same format as used in explicit specification under the BASIS directive. As with the latter (see Part 3), alternative formats (in terms of atom / orbital and coefficient / exponent orderings) are supported to provide compatibility with other packages and basis set libraries. Thus the above JBAS data may also be presented (through addition of the NWCHEM keyword) as follows;

  ```
  TITLE
  H2CO - 6-31G/BLYP  CLOSED SHELL DFT WITH 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 JFITG
DFT SCHWARZ 6
DFT JBAS NWCHEM
# BASIS "DGauss A1 DFT Coulomb Fitting"
H    S
   45.000000000      1.00000000
H    S
    7.500000000      1.00000000
H    S
    1.500000000      1.00000000
H    S
    0.300000000      1.00000000
C    S
 1114.000000000      1.00000000
C    S
  223.000000000      1.00000000
C    S
   55.720000000      1.00000000
C    S
   13.900000000      1.00000000
C   SP
    4.400000000      1.00000000      1.00000000
C   SP
    0.870000000      1.00000000      1.00000000
C   SP
    0.220000000      1.00000000      1.00000000
C    D
    4.400000000      1.00000000
C    D
    0.870000000      1.00000000
C    D
    0.220000000      1.00000000
O    S
 2000.000000000      1.00000000
O    S
  400.000000000      1.00000000
O    S
  100.000000000      1.00000000
O    S
   25.000000000      1.00000000
O   SP
    7.800000000      1.00000000      1.00000000
O   SP
    1.560000000      1.00000000      1.00000000
O   SP
    0.390000000      1.00000000      1.00000000
O    D
    7.800000000      1.00000000
O    D
    1.560000000      1.00000000
O    D
    0.390000000      1.00000000
END
ENTER
```

- Although the Coulomb fitting may improve efficiency, a significant additional improvement can be obtained if the 3-center 2-electron integrals can be stored in memory. The MEMORY subdirective sets aside the maximum amount of memory that is not needed for other purposes to hold the 3-center 2-electron integrals. If not all the integrals fit in memory then those that cannot be stored will be recomputed. To switch this option on, replace the data line

  ```
  DFT JFIT JFITG
  ```

  in the above example with the line;

  ```
  DFT JFIT MEMORY JFITG
  ```

- The example above involves explicit specification of the fitting basis set. A simpler mode of specification is also supported, enabling the user to request either the DGauss A1 or A2 basis sets, the Demon fitting basis [29], or the fitted basis sets tabulated by Ahlrichs and co-workers [30]. These are requested through keyword specification on the JBAS data line, thus;

  ```
  DFT JBAS A1-DGAUSS

  DFT JBAS A2-DGAUSS

  DFT JBAS DEMON

  DFT JBAS AHLRICHS
  ```

  The following data set is thus equivalent to those presented above;

  ```
  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 JFITG
  DFT SCHWARZ 6
  DFT JBAS A1-DGAUSS
  ENTER
  ```

12 Møller Plesset MP2 and MP3 Calculations

In this section the traditional Møller Plesset calculations are discussed. These calculations are based on a single closed shell Hartree-Fock reference determinant. Also available are multi-reference MP2 and MP3 calculations. Because the latter option is part of the Direct-CI module these calculations are described in section 23.

Møller Plesset calculations are performed under control of SCFTYPE specification, with the level of treatment, either MP2 or MP3, nominated by keyword. A second keyword may also be required, requesting the level of underlying SCF, either RHF (closed-shells) or UHF (open-shell systems). Before detailing example data files for performing such calculations, we mention some general points:

1. Møller Plesset processing involves a combination of tasks, including integral generation, SCF, integral transformation and, finally, the Møller Plesset calculation itself.

2. Several files will be generated under such processing. These include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Scratch file (ED7).
   - temporary files for sorting transformed integrals (the Sortfile) and intermediate matrices in the Møller Plesset calculation. The number of such files is a function of the underlying SCF (RHF (closed-shells) or UHF (open-shell systems)), and the level of Møller Plesset theory requested, as follows;
     - MP2/RHF:- no additional files required;
     - MP2/UHF:- ED16, ED17, ED18 and ED19;
     - MP3/RHF:- ED16 and ED17;
     - MP3/UHF:- ED16, ED17, ED18, ED19, MT0 and MT1.

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

3. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF
   ```

   in the SCF run. Note that in contrast to CASSCF and MCSCF calculations, MP processing is driven off the skeletonised list of 2-electron integrals, so that the NOSYM parameter specification is not required on the SUPER directive.

4. In default all molecular orbitals will be deemed active in the MP calculation.

12.1 MP2 Calculations

A closed-shell MP2 calculation is to performed on the formaldehyde molecule, A valid data sequence for performing such a calculation is shown below, where we are performing all the computation in a single job.

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF
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 MP2 RHF
ENTER
```

Note that the SCFTYPE specification above may be simplified to just

```
SCFTYPE MP2
```

with RHF the default level of underlying SCF for closed–shell systems. Now let us consider performing the above calculation in two steps, where the first carries out the SCF, the second the transformation and MP2 calculation. Assuming we wish to avoid recalculating the 2e-integrals in the MP2 calculation, then the SUPER OFF data line should be presented in the SCF job, allowing bypassing of integral evaluation in the subsequent calculation. First the closed shell case: valid data sequences for performing the calculation are given below.

Closed shell SCF Data

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
SUPER 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
ENTER
```

MP2 Data

```
RESTART
TITLE
H2CO - X1A1 - 3-21G DEFAULT BASIS - MP2/RHF CALCULATION
SUPER OFF
BYPASS
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 MP2
ENTER
```

Let us now consider a MP2 calculation on the ${}^2B_2$ state of H_2CO^+, now using the UHF formalism. A valid data sequence for performing such a calculation is shown below, where we are performing all the computation in a single job.

```
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - MP2/UHF 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 MP2 UHF
ENTER
```

Again the UHF flag may be omitted from the SCFTYPE data line, given that UHF is the underlying SCF for open–shell systems. Consider performing the above calculation in two steps, where the first carries out the UHF, the second the transformation and MP2 calculation. Valid data sequences for performing the calculation are shown below, where we again BYPASS integral evaluation in the MP2 job. Note again that this BYPASS'ing of integral evaluation necessitates the introduction of the SUPER OFF data line in the initial UHF job.

UHF Data

```
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - UHF CALCULATION
SUPER OFF
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
ENTER
```

MP2 Data

```
RESTART
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - MP2/UHF CALCULATION
```

```
SUPER OFF
BYPASS
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 MP2
ENTER
```

12.2 MP3 Calculations

The MP3 level of treatment is requested in equivalent fashion to the MP2 calculations detailed above, with use of the MP3 keyword on the SCFTYPE data line. Data sequences for performing a closed-shell MP3 calculation on the formaldehyde molecule, and an open-shell calculation on the $^2B_2$ state of H_2CO^+ are given below.

MP3/RHF Data

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP3/RHF
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 MP3
ENTER
```

MP3/UHF Data

```
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - MP3/UHF 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 MP3
ENTER
```

12.3 Freezing and Discarding Orbitals

In the examples above we have assumed that all SCF–MOs are active in the subsequent Møller Plesset calculation. In many instances however this will not be the case, for the user may wish to

- 'freeze' inner-shell orbitals, performing a 'valence-only' Møller Plesset calculation.
- discard certain virtual orbitals from the Møller Plesset calculation, typically the high-energy inner-shell complement orbitals.

The ACTIVE directive is provided for controlling the final subset of orbitals for inclusion in the Møller Plesset calculation. The freezing of core , or inner-shell, orbitals and the discarding of virtual orbitals is achieved by nominating under control of the ACTIVE directive, the sequence nos. of the active set of SCF orbitals to appear in the calculation. Consider the MP2/RHF H_2CO calculation above. The following data sequence would be required to freeze both the O1s and C1s orbitals (with SCF sequence numbers 1 and 2 respectively) and to discard the highest virtual orbital (with SCF sequence number 22):

```
TITLE
H2CO - 3-21G BASIS - MP2/RHF (VALENCE ONLY)
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 MP2
ACTIVE
3 TO 21 END
ENTER
```

12.4 Direct-MP2 Calculations

We wish to perform a direct-MP2 calculation equivalent to that above. A valid data sequence for performing such a calculation is shown below:

```
TITLE
H2CO - 3-21G DEFAULT BASIS - DIRECT-MP2
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 DIRECT MP2
ENTER
```

Note the change in syntax of the SCFTYPE directive when specifying the DIRECT option. The third parameter on the data line, MP2, points to the particular category of Møller Plesset wavefunction required i.e., MP2. At present this is the only option supported in direct-mode, and is only applicable to closed–shells. Note that the default file output in such calculations is confined to ED3 and ED7, the Dump- and Scratch-file respectively. Note also that such calculations are memory intensive, with the MEMORY pre–directive in the above example (see the machine–specific Parts) used to increase the default memory allocation in the present run.

13 Analysing the Wavefunction

GAMESS–UK includes a variety of tools for analysing wavefunctions, driven by the RUNTYPE ANALYSE directive. It is now possible to:

- Calculate a variety of 1-electron properties.
- Generate a localised orbital representation of an SCF wavefunction using either the dipole-centroid technique due to Foster and Boys, or the overlap-based criterion due to Pipek and Mezey [31].
- Provide graphical analysis of molecular wavefunctions. The program is capable of generating contour and perspective plots which depict:
  - the electron density associated with one or more molecular orbitals;
  - the amplitude of a molecular orbital;
  - a comparison of the density distribution in two or more molecular systems;
  - the interaction energy between a molecular distribution and a hypothetical point charge, generating the so–called electrostatic potential plot.

  Two types of plot may be generated by the program to provide a pictorial representation of a given density or potential function in a specified molecular plane:

  - a contour plot, with contours representing lines of constant value, depicting the spatial characteristics of the given function;
  - a perspective plot, with the values of the function in a given plane, displayed as a 3–D perspective picture.

  Both types of plot are generated from a grid of function values produced by the program.
- Perform a distributed multipole analysis (DMA) of an SCF wavefunction [32].
- Perform a more detailed Mulliken analysis, including both bond and orbital properties.

The user should note the following;

- While the various SCF modules provide default sections for eigenvector information, it will be necessary in the ANALYSE modules to specify, via the VECTORS directive, the specific eigenvectors to be analysed.

- Each mode of analysis typically requires one or more directives to specify the particular tasks required.

At present we restrict ourselves to sample data files for property evaluation, localised orbital analysis, graphical analysis, DMA and extended-Mulliken analysis. In each case we assume that the closed-shell SCF calculation on formaldehyde (I) has been successfully completed, and perform the requested analysis based on the SCF-MOs, as written to the Section 1 of the Dumpfile.

13.1 One-electron Property Evaluation

The following data sequence would be required in evaluating the electric field gradient at the carbon and oxygen nuclei.

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - 1-E PROPERTIES
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
PROPERTY
4 C
4 O
END
VECTORS 1
ENTER
```

Each one-electron operator is known to the user by an operator number; a full list of the available operators and associated numbers in given in Table 6 . The user specifies, under control of the PROPERTY directive, those properties to be be computed at any of the nuclei known to the system, by virtue of the TAGs defined in the z-matrix.

The example above typifies the case where a single set of MOs are associated with the particular SCFTYPE, and as such may be input under control of the VECTORS directive to the properties package. A somewhat different approach is required when computing the one-electron properties derived from a wavefunction with more than one set of MOs (e.g., a UHF wavefunction), or in cases where only the total density matrix, and not an associated set of MOs, is available (e.g., in an MP2 calculation). In both cases, the user will need to generate the associated set of spinfree natural orbitals, and present these as input to the analysis module. Such orbitals are generated under control of the NATORB directive, which may used to route the natural orbitals to a nominated section on the Dumpfile.

The following data sequences would be required when evaluating the properties based on a UHF wavefunction. First, the data for the UHF calculation itself:

Table 6: The One-electron Operators and Operator Numbers

Operator Number	Operator	Operator Number	Operator
1	Potential	11	Third Moment (combined)
2	Diamagnetic Shielding	12	Hexadecapole Moment
3	Electric Field	13	Fourth Moment (even)
4	Electric Field Gradient	14	Fourth Moment (odd)
5	Dipole Moment	15	Overlap
6	Quadrupole Moment	16	Planar Charge Density
7	Diamagnetic Susceptibility	17	Line Charge Density
8	Second Moment	18	Charge Density
9	Octupole Moment	19	Isotropic ESR Coupling Constants
10	Third Moment	20	Anisotropic ESR Coupling Constants

```
TITLE
H2CO - 3A2 UHF - 3-21G DEFAULT BASIS
MULT 3
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
NATORB 10 PRINT
ENTER
```

Having routed the spinfree natural orbitals to section 10 on the Dumpfile, the properties calculation proceeds by nominating this section on the VECTORS line, thus:

```
RESTART NEW
TITLE
H2CO - 3A2 UHF - 3-21G DEFAULT BASIS - 1-E PROPERTIES
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
SCFTYPE UHF
PROPERTY
4 C
4 O
END
VECTORS 10
ENTER
```

As presented above, the NATORB directive will request generation of the spinfree natural orbitals. Two variants of the directive allow for (i) generation of the spin natural orbitals, and (ii) annihilation of the UHF wavefunction and subsequent generation of both spin- and spinfree NOs. The associated data requirements are straightforward:

- Generation of the spin NOs is driven by the keyword SPIN presented immediately after the NATORB initiator. Thus the data line

  ```
  NATORB SPIN 11 PRINT
  ```

  would result in the routing of the spin NOs to section 11 of the Dumpfile.

- Annihilation of the UHF wavefunction, and subsequent generation of the NOs from the annihilated density matrices is driven by specification of the keyword ANNIHILATE as the final character string on the NATORB data line. Thus the data sequence:

  ```
  NATORB 12 PRINT ANNIHILATE
  ```

  would route the spinfree NOs of the annihilated UHF wavefunction, AUHF, to section 12 of the Dumpfile.

The theory behind the AUHF analysis can be found in [33]. Note that the NOs of the UHF and AUHF wave function are in fact identical, the only difference lying in the occupation numbers.

Now let us consider the date requirements when computing properties at the optimum geometry derived from an MP2 calculation. First, the data for the MP2 geometry optimisation, where the spinfree natural orbitals at the optimised geometry are to be routed to section 20.

```
TITLE
H2CO - X1A1 - MP2 (DZ BASIS)
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 OPTIMISE
SCFTYPE MP2
NATORB 20 PRINT
ENTER
```

Having routed the spinfree natural orbitals to section 20 on the Dumpfile, the properties calculation proceeds by nominating this section on the VECTORS line, thus:

```
RESTART
TITLE
H2CO - X1A1 - MP2/DZ BASIS - 1-E PROPERTIES
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 ANALYSE
SCFTYPE MP2
PROPERTY
4 C
4 O
END
VECTORS 20
ENTER
```

Note the use of RESTART in restoring the optimized geometry from the Dumpfile.

13.2 Simplified Property Specification

In the examples above we have assumed that property evaluation is to be conducted under control of RUNTYPE ANALYSE, with explicit specification of the required one-electron properties. A simplified mechanism for property evaluation can be requested through presenting the data line

```
PROPERTY ATOMS
```

after RUNTYPE and SCFTYPE specification. This will result in the default wavefunction analysis conducted after RUNTYPE processing being augmented with the computation of certain one-electron properties. The following points should be noted:

- the properties evaluated include the electrostatic potential, electric field, electric field gradient, and electron density at each of the atomic centres, plus the dipole, second moment, quadrupole moment, third and octupole moments, at the computed centre of mass of the system under study. In addition the spin densities will also be computed in the case of open shell systems.
- this analysis, if requested, is available on completion of SCF, OPTIMIZE, OPTXYZ, SADDLE, and CI processing.

The following data sequence would be required to generate the above list of properties on completion of an SCF calculation of the formaldehyde molecule.

```
TITLE
H2CO - 3-21G BASIS - SCF + DEFAULT 1-E PROPERTIES
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 SCF
PROPERTY ATOMS
ENTER
```

In this example the set of MOs to be used in the property evaluation will be retrieved from that section written in the SCF process, namely section 1 of the Dumpfile i.e. the default section number for the underlying closed-shell SCFTYPE (see Table 1).

A somewhat different approach may be required when computing the one-electron properties derived from a wavefunction with more than one set of MOs (e.g., a UHF wavefunction), or in cases where only the total density matrix, and not an associated set of MOs, is available (e.g., in an MP2 calculation). In both cases, the user may need to ensure that the associated set of spinfree natural orbitals and, where relevant SPIN natural orbitals, are generated by specification of the NATORB directive(s), used to route the NOs to a nominated section on the Dumpfile.

We illustrate this effect by first considering the data requirements when performing a UHF wavefunction. The following data sequence would be required when evaluating the properties based on a direct-UHF calculation, with the computation based on the alpha- and beta-UHF MOs routed to the default sections 1 and 2 respectively under implicit control of the ENTER directive.

```
TITLE
H2CO - 3A2 UHF PROPERTIES - 3-21G BASIS
MULT 3
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 DIRECT UHF
PROPERTY ATOMS
ENTER
```

The same calculation may be performed based on the spinfree and spin natural orbitals of the UHF wavefunction; in this case the NATORB data lines will be used to route the spinfree and spin natural orbitals to sections 10 and 11 of the Dumpfile respectively, and these orbitals will be used in computing the 1-electron properties, thus:

```
TITLE
H2CO - 3A2 UHF NO-BASED PROPERTIES - 3-21G BASIS
MULT 3
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 DIRECT UHF
PROPERTY ATOMS
NATORB 10
NATORB SPIN 11
ENTER
```

The following data sequence would be required if the user wished to compute the properties of the annihilated UHF wavefunction:

```
TITLE
H2CO - 3A2 annihilated UHF properties 3-21G BASIS
MULT 3
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 DIRECT UHF
PROPERTY ATOMS
NATORB 10 ANNIHILATE
NATORB SPIN 11 ANNIHILATE
ENTER
```

Note again that the NOs of the UHF and AUHF wave function are in fact identical, the only difference lying in the occupation numbers.

Now let us consider the date requirements when computing properties at the optimum geometry derived from an MP2 calculation.

```
TITLE
H2CO - X1A1 - MP2 DZ BASIS - PROPERTIES
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 OPTIMISE
PROPERTY ATOMS
SCFTYPE MP2
NATORB 20
ENTER
```

Having generated the MP2 optimised geometry, the spinfree natural orbitals will be routed to section 20 on the Dumpfile, and used in the subsequent properties calculation.

13.3 Localised Orbitals

The following data sequence would be required in localising the valence SCF-MOs using the Foster-Boys algorithm, where the LOCAL directive specifies those orbitals deemed to be active in the localisation process.

```
RESTART
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
VECTORS 1
ENTER 20
```

Note that the localised orbital module is the only analysis module that creates a new set of eigenvectors, and the user must specify the destination section on the Dumpfile for these orbitals i.e. no default section will be employed. In this case the final set of LMOs will be output to Section 20 of the Dumpfile.

13.4 Distributed Multipole Analysis

The following data sequence would be required in requesting a distributed multipole analysis of the SCF-MOs [32], where the DMA directive instigates the process.

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - DMA ANALYSIS
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
DMA
VECTORS 1
ENTER
```

13.5 Graphical Analysis

The following data sequence would be required in generating grids of total density, atom-difference density, electrostatic potential and orbital amplitude for subsequent graphical analysis.

The GRAPHICS directive introduces data defining the required graphics processing, with GDEF data defining the grid of points involved, and subsequent CALC and PLOT directives introducing data specifying the required computation associated with the grid (CALC) and corresponding graphical output to be generated (PLOT).

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - GRAPHICAL ANALYSIS
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
GRAPHICS
GDEF
TYPE 2D
POINTS 99
TITLE
SQUARE 2D GRID (99*99)
CALC
TYPE ATOM
TITLE
H2CO -ATOM DIFFERENCE
SECTION 150
PLOT
TYPE LINE
TITLE
ATOM DIFFERENCE DENSITY LINEPRINTER PLOT
CALC
TYPE DENS
SECTION 151
TITLE
H2CO - TOTAL DENSITY
PLOT
TYPE LINE
TITLE
DENSITY LINEPRINTER PLOT
CALC
TYPE MO 2
TITLE
H2CO MO 2 AMPLITUDE
SECTION 152
PLOT
TYPE LINE
TITLE
MO 2 LINEPRINTER PLOT
GDEF
TYPE 2D
POINTS 25
TITLE
SQUARE 2D GRID (25*25)
CALC
TYPE POTE
```

```
TITLE
H2CO - POTENTIAL
SECTION 153
PLOT
TYPE LINE
TITLE
POTENTIAL LINEPRINTER PLOT
VECTORS 1
ENTER
```

The resolution of each plot is controlled by the size of the grid, via the POINTS sub–directive of GDEF. Note that the TYPE sub–directive of CALC defines the type of grid (ATOM, DENS, MO and POTE for atom–difference, total density, orbital amplitude and electrostatic potential respectively). In the present example output is restricted to the line–printer, through the LINE parameter in the PLOT data.

13.6 Population Analysis

The following data sequence would be required in performing an extended population analysis of the valence SCF-MOs, where the MULLIKEN directive specifies those orbitals for which printed output is required. The ATOM and ORBITAL keyword request the emphasis in the analysis generated through the grouping of basis functions [34]. The sequence of integers specified on the MULLIKEN line specifies those MOs for which printed output is required.

```
RESTART
TITLE
H2CO - 3-21G BASIS - ANALYSIS OF VALENCE MOs
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
MULLIKEN ATOM ORBITAL 3 TO 8 END
VECTORS 1
ENTER
```

Note that it is also possible to define the groups of basis functions through user input. Thus the following data would perform the same analysis as the ATOM specification above, where the GROUP keyword on the MULLIK data line indicates that subsequent data lines will follow, terminated by the END keyword, that will assign the basis functions to user-defined groups.

```
RESTART
TITLE
H2CO - 3-21G BASIS - INPUT GROUPS FOR ANALYSIS
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
MULLIKEN GROUP  3 TO 8 END
CATOM 1 TO 9
OATOM 10 TO 18
H1ATOM 19 20
H2ATOM 21 22
END
VECTORS 1
ENTER
```

13.7 Morokuma Energy Decomposition Analysis

The following example illustrates how the Morokuma EDA is performed using a sequence of three separate GAMESS-UK input decks, one for each of the two fragments and an analysis job for the supermolecule.

The Class 2 MOROKUMA directive controls the job, and may take one of two forms, depending on whether a fragment SCF or an interaction calculations required. The directive sequence:

```
MOROKUMA FRAG NUMBER TAG
```

specifies the RHF calculation on one of the fragments. NUMBER should be 1 or 2, indicating the position of the fragment in the supermolecule. TAG is replaced with a string to identify the fragment; the job will result in a file of this name containing the fragment basis and wavefunction information being written in the working directory of the job. There is currently an 8 character limit on TAG. The sequence:

```
MOROKUMA INTERACT TAG1 TAG2
```

requests that an interaction energy analysis be performed. The geometry is assumed to be that of the supermolecule, and the two tags denote the fragment files from two previous runs under control of MOROKUMA FRAG as above.

A number of restrictions should be noted when using the morokuma analysis module:

- The implementation is restricted to RHF calculations using the conventional (non-direct) SCF module.
- The use of symmetry (including symmetry adaption) must be disabled for all component jobs. The atoms in the supermolecule must be presented in the same order as that obtained by concatenating the two fragments and the basis sets specified for the separate tasks must correspond.
- The code is developmental, although it is believed to work correctly within the above limits, prospective users are advised to contact the authors.

- Morokuma EDA jobs cannot be restarted.

```
TITLE
MOROKUMA TEST FRAG2
ADAPT OFF
NOSYM
GEOMETRY
 0.00000000  -1.10092542 -1.43475395  1.0 H
 0.00000000  -1.10092542  1.43475395  1.0 H
 0.00000000   0.00000000  0.00000000  8.0 O
END
MOROKUMA FRAG 1 FRAG1
BASIS SV 4-31G
ENTER


TITLE
MOROKUMA TEST
ADAPT OFF
NOSYM
GEOMETRY
 3.24201636   2.02583666  0.00000000  1.0 H
 4.24693920   4.71362490  0.00000000  1.0 H
 4.77568401   2.98417857  0.00000000  8.0 O
END
MOROKUMA FRAG 2 FRAG2
BASIS SV 4-31G
ENTER


TITLE
MOROKUMA TEST
ADAPT OFF
NOSYM
GEOMETRY
 0.00000000  -1.10092542 -1.43475395  1.0 H
 0.00000000  -1.10092542  1.43475395  1.0 H
 0.00000000   0.00000000  0.00000000  8.0 O
 3.24201636   2.02583666  0.00000000  1.0 H
 4.24693920   4.71362490  0.00000000  1.0 H
 4.77568401   2.98417857  0.00000000  8.0 O
END
BASIS SV 4-31G
MOROK INTERACT FRAG1 FRAG2
VECTORS ATOMS
ENTER
```

14 Restarting Integral and SCF Computations

In all the examples considered so far, we have assumed that the particular activity requested, in general some SCF computation, completes in the time allocated to the job. This is often not the case, and we need to consider restarting the computation in a controlled fashion. Such a requirement is most often met in SCF computations when either

- integral evaluation has not been completed, or
- SCF convergence has not been achieved, either due to lack of time, or to convergence problems, when the maximum number of iterative cycles has been exceeded.

Restarting the computation is achieved under control of the RESTART directive, which now nominates the task to be restarted i.e., that in progress when the previous job dumped. The following data files would be required in restarting the computation described in §2, §6 and §9 above:

Closed-Shell SCF Restart Data

```
RESTART SCF
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
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
VECTORS 1
ENTER 1
```

GVB/1-PP Restart Data

```
RESTART SCF
TITLE
H2CO  - GVB/1-PP - 3-21G BASIS - (1B1->2B1*)
BYPASS
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 1 REST
VECTORS 2 3
ENTER 2 3
```

CASSCF Restart Data

```
RESTART SCF
TITLE
H2CO  - CASSCF   - 3-21G BASIS - (10E IN 9 M.O.)
BYPASS
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 CASSCF
CONFIG BYPASS
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
VECTORS 2 3
ENTER 2 3
```

The following points should be noted:

- Given the default Dumpfile settings, all examples commence with the data line

  ```
  RESTART SCF
  ```

- Note that the default VECTORS and ENTER attributes will, in restart mode, still apply. Thus in the closed-shell case the default VECTORS section, section 1, will be examined at the outset of processing, and if found to exist from a previous job (i.e. the startup job), will be used as a source of eigenvectors to restart the SCF computation. The same process will be undertaken in the GVB and CASSCF restart jobs, with sections 4 and 5 (GVB), and sections 7 and 8 (CASSCF), being used to restart SCF processing (see Table 1). Note also that at the outset of a specific computation, the program generates the trial MOs and stores these in the default vectors section, or that Section nominated on the ENTER line. This activity precedes integral evaluation, so even if the starting job had dumped during computation of the two-electron integrals, an appropriate set of eigenvectors will have already been generated.

- The REST parameter on the SCFTYPE directive of the GVB run instructs the program to restore the set of CI pair coefficients from the Dumpfile, and not to use the default settings. This is crucial when restarting GVB geometry optimisations.

- Several changes in the CASSCF data file should be noted. The BYPASS keyword on the CONFIG data line instructs the program to bypass generation of the Loop Formula tape (ED9) - assuming of course that this file had been saved from the startup run.

15 Geometry Optimisation

In the examples above we have considered performing a single point calculation i.e., at a particular geometry, using the various SCFTYPE options available within GAMESS–UK. Each category of wavefunction may, in addition, be used in optimising the molecular geometry through calculation of not only the energy, but also the gradient of the energy. In the present section we consider the format of the data required when carrying out such calculations using the quasi-Newton optimisation procedures available in GAMESS–UK. The following points should be noted;

1. Geometry optimisation may be conducted in either a framework of internal coordinates (as defined by the ZMATRIX and VARIABLES specification lines) or directly in a framework of cartesian coordinates (as generated from the ZMATRIX or defined through the GEOMETRY directive).

2. Cartesian coordinate optimisation is requested by specifying the OPTXYZ option of the RUNTYPE directive. Restarting such calculations after a controlled dump again involves the OPTXYZ specification on the RESTART directive.

3. Internal coordinate optimisation is requested by specifying the OPTIMIZE option of the RUNTYPE directive. Restarting such calculations after a controlled dump again involves the OPTIMIZE specification on the RESTART directive.

15.1 Internal Coordinate Optimisation

Under control of the RUNTYPE OPTIMIZE specification, geometry optimisation is conducted in a system of internal coordinates - bond lengths, bond angles and dihedral angles - defined by the z-matrix. This is controlled through the introduction of so-called VARIABLES in the z-matrix. Any internal coordinate whose value is to be varied during optimisation *must* be specified as a VARIABLE, and an initial value assigned to it through the VARIABLE definition lines of the ZMATRIX directive. Consider the data from the SCF computations on formaldehyde:

```
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
```

The following data depicts the corresponding ZMATRIX required when optimising the geometry;

```
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
```

where r(C-H), r(C-O) and angle HCO have been declared as variables with symbolic names CH, CO and HCO respectively. Both of the above data sequences are equivalent in the context of a single point SCF calculation. The following points should be noted;

1. In the sequence above the dihedral angle of 180.0^o has been used to define the required planar geometry, while use of a single variable for both CH bonds and HCO angles leads to a system of C_{2v} geometry. The optimisation will be conducted subject to these constraints, with the symmetry of the starting geometry maintained throughout optimisation.

Any attempt to change molecular point group during optimisation will lead to an error condition.

2. As with the dihedral angle above, any parameter in the z-matrix which is not declared a variable will remain fixed during optimisation. This may be controlled either by specifying the parameter by value in the definition lines of the z-matrix, or through CONSTANTS data lines. Thus the data sequences

```
ZMATRIX ANGSTROM
C
O 1 CO
H 1 1.099 2 HCO
H 1 1.099 2 HCO 3 180.0
VARIABLES
CO 1.203\HCO 121.8
END
```

and

```
ZMATRIX
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 PHI
CONSTANTS
PHI 180.0\CH 1.099
VARIABLES
CO 1.203\HCO 121.8
END
```

are equivalent in constraining the optimisation such that only r(C-O) and HCO are varied.

3. Finally we consider three complete data files for carrying out the H_2CO optimisation. First, we show 2 possible jobs for an SCF optimisation, the first representing the startup job, the second a possible restart job to complete the computation assuming the first had dumped on time.

Optimisation - Startup Data

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - 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
RUNTYPE OPTIMIZE
ENTER
```

Optimisation - Restart data

```
RESTART OPTIMIZE
TITLE
H2CO  - 3-21G - RESTART 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
RUNTYPE OPTIMIZE
ENTER
```

DFT Geometry Optimisation

The following data file is for performing the corresponding DFT optimisation, using the B3LYP functional in direct-SCF mode;

```
TITLE
H2CO - 3-21G BASIS - DFT/B3LYP 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
RUNTYPE OPTIMIZE
SCFTYPE DIRECT
DFT B3LYP
ENTER
```

15.2 Determining the Initial Hessian

In addition to specifying the starting geometry and internal coordinates, the user need also consider defining an initial force constant matrix (Hessian or second-derivative matrix), bearing in mind that a drastic reduction in the number of energy and gradient calculations required in the optimisation pathway can be realised through an accurate estimate of this Hessian. This is particularly true in the location of transition states. The following options are available

1. In default mode (as in the example above) the program provides an estimate of the diagonal force constant matrix, based on a look-up table of bond-stretches, bending angle etc. involving the component nuclei of the molecule.
2. These defaults are, in most cases, perfectly adequate in determining equilibrium geometries. They may, however, be overridden by providing additional information on the corresponding VARIABLE definition lines. In the formaldehyde examples above, the default value for the C-H bond variable may be replaced by the data line

```
CH 1.099 HESSIAN 0.5
```

whereby the diagonal force constant for the CH variable is set to 0.5.

3. A more accurate, but clearly more expensive, estimate of the initial force constant matrix may be generated through explicit computation, either numerically or analytically of all or part of the trial hessian;

   (a) NUMERICAL DETERMINATION of some subset of the hessian may be performed *in situ* prior to commencing the geometry optimisation. This is requested through specification of the TYPE keyword on the VARIABLE definition lines. In such cases the corresponding part of the Hessian will be evaluated numerically, prior to commencing optimisation. Two settings are possible

   - TYPE 2 ; requests calculation of the diagonal force constant and involves an additional energy calculation.
   - TYPE 3 ; requests calculation of the diagonal force constant and all off-diagonal elements involving the variable. This requires an additional energy-plus-gradient calculation for each variable nominated.

   Thus the following data sequence would lead to explicit calculation of the complete Hessian for formaldehyde.

```
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO 1.203 TYPE 3
CH 1.099 TYPE 3
HCO 121.8 TYPE 3
END
```

   (b) ANALYTIC DETERMINATION of the complete trial force constant matrix may be performed under control of a separate RUNTYPE, HESSIAN, with the resulting Hessian matrix subsequently restored from the Dumpfile in the optimisation job. This method of determination is considered in more detail below (see for example, 2.16.2, Example 3); it is rarely justified in equilibrium geometry determination, but is probably the most efficient method in the more complex process of locating transition structures. At this point we merely present an example, where the Hessian is restored though the FCM specification on the RUNTYPE directive in the optimisation job. The FCM input is identical to that in the FCMIN directive (see part4)

Determining the Trial Hessian

```
TITLE
H2CO - 3-21G - DETERMINING THE INITIAL HESSIAN
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
RUNTYPE HESSIAN
ENTER
```

Optimisation data - Restoring the Initial Hessian

```
RESTART NEW
TITLE
H2CO - 3-21G - RESTORING THE INITIAL HESSIAN
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
RUNTYPE OPTIMIZE FCM
ENTER
```

Note that if this restoring of the Hessian is carried out in initialising the optimisation, an identical RUNTYPE specification must be presented in any subsequent restart of the optimisation; failure to adhere to this will lead to a 'PARAMETER ERROR' diagnostic, and job failure.

Experience suggests that the most efficient way to proceed when performing geometry optimisation on small-medium sized molecules with extended basis sets (larger than minimal or 3-21G) is to

- perform the geometry optimisation in a 'smaller' basis set
- input the resulting Hessian to the extended basis set calculation, by reading the matrix directly from the Dumpfile associated with the 'small' calculation. Such input is controlled through the RUNTYPE directive, which takes the form

  ```
  RUNTYPE OPTIMIZE ED4 1
  ```

  Here is it assumed that the data set containing the Dumpfile from the 'small' calculation has been assigned to the 'extended' calculation with the LFN ED4, commencing at block 1. This Hessian may be generated in a previous optimise run or in a separate analytical hessian calculation. The data on the runtype card is read like in the FCMIN directive (part4).This is a particularly attractive way of minimising overall CPU requirements, given that the 'small' basis eigenvectors may also be input to initiate the 'extended' SCF calculation under control of the GETQ directive (see section 4.8). Note that if this restoring of the Hessian is performed in an optimisation startup, the identical RUNTYPE directive must be presented in any subsequent restarts – failure to do so will lead to a 'PARAMETER ERROR' diagnostic, and job failure.

15.3 Cartesian Coordinate Optimisation

A second optimisation procedure is provided specifically for those cases where problems arise with the internal coordinate driven scheme or when direct input of cartesian coordinates (e.g., from a data base) is more convenient. While the cartesian procedure is less flexible than the internal coordinate method, in that

- it is not possible to define the starting Hessian or to restore a Hessian from, say, a smaller basis calculation,
- the algorithm employed is only guaranteed to converge to a stationary point, not necessarily a minimum,

nevertheless the quasi–Newton method, driven under RUNTYPE OPTXYZ control, has proved moderately robust and reliable. The following data demonstrates OPTXYZ usage, where the x, y and z coordinates of the component atoms are input under control of the GEOMETRY directive (here in atomic units).

```
TITLE
H2CO GEOMETRY TEST
GEOMETRY
 0.0000000      0.0000000      0.9998722  6 C
 0.0000000      0.0000000     -1.2734689  8 O
 0.0000000      1.7650653      2.0942591  1 H
 0.0000000     -1.7650653      2.0942591  1 H
END
RUNTYPE OPTXYZ
ENTER
```

The following points should be noted;

- It is now possible to "freeze" coordinates under OTPXYZ control. This is achieved by appending the keyword NOOPT to the geometry definition lines of the appropriate centres as specified by the GEOMETRY directive. Thus in the example above, the H atoms may be held at their input geometry during optimisation using the following data;

  ```
  TITLE
  H2CO GEOMETRY TEST
  GEOMETRY
   0.0000000      0.0000000      0.9998722  6 C
   0.0000000      0.0000000     -1.2734689  8 O
   0.0000000      1.7650653      2.0942591  1 H NOOPT
   0.0000000     -1.7650653      2.0942591  1 H NOOPT
  END
  RUNTYPE OPTXYZ
  ENTER
  ```

- Note that is it is still possible to use the ZMATRIX to input the molecular geometry in such calculations, although the starting variables provided will not be explicitly updated during the course of the subsequent optimisation.

15.4 Mixed Z-matrix and Cartesian Optimisation

It is also possible to perform mixed z-matrix/cartesian optimisations. In the example below the structure of ethylene has been defined using a z-matrix for two of the hydrogen atoms; the remaining four atoms are input as cartesian coordinates. Cartesian or internal coordinates can be defined as variables in the geometry optimisation. Constructing molecules using mixed z-matrix and cartesian input is discussed in greater detail in Part 3.

```
TITLE
ETHYLENE
ZMATRIX ANGSTROM
CARTESIANS
C   0.000    0.000    0.000
C   0.000    0.000    CC
H   WIDTH    0.000   -DEPTH
H  -WIDTH    0.000   -DEPTH
INTERNALS
H 2  CH  1  CCH  3  TWIST
H 2  CH  1  CCH  5  180.0
VARIABLES
CC 1.4
CH 1.0
WIDTH 0.8
DEPTH 0.5
CCH 120.0
TWIST 10.0
END
RUNTYPE OPTIMISE
ENTER
```

15.5 Energy-only Geometry Optimisation

Finally, an optimisation procedure is available which uses a modified Fletcher Powell method with numerical differentiation of energies to produce gradients. This procedure is intended for use with methods for which analytic gradients are not available, and is requested under RUNTYPE OPTIMIZE control, through specification of and additional keyword, FP, on the RUNTYPE data line. The following example demonstrates FP usage.

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - 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
RUNTYPE OPTIMIZE FP
ENTER
```

15.6 Post Hartree-Fock Geometry Optimisation

While the examples above of internal coordinate, cartesian coordinate and energy-only optimisation have used only the closed-shell SCF case, the user should note that such optimisations are also available for CASSCF, MCSCF and MP2/MP3 wavefunctions, in addition to UHF and GVB. Energy-only optimization capabilities for Direct-CI, CCSD and Full-CI calculations are considered under the appropriate CI section further in this chapter. Data input requirements for the cases under consideration at this point follow straightforwardly from the preceding sections, with appropriate SCFTYPE specification. We illustrate such usage below for the cases of CASSCF, MCSCF, and MP2/RHF and MP2/UHF wavefunctions.

15.6.1 CASSCF Geometry Optimisation

When performing either CASSCF or MCSCF geometry optimisations, the user should initially generate an appropriate set of trial MOS for input to the subsequent geometry optimisation, using these orbitals as the basis for the CONFIG or ORBITAL data.

Run I: The initial SCF Calculation

```
TITLE
H2CO - 3-21G  - SCF STARTUP FOR CASSCF GEOM. 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
ENTER
```

Run II: The CASSCF Optimisation

```
RESTART
TITLE
H2CO - CASSCF GEOM OPT. (10E IN 9 M.O.)/TOTAL ENERGY  -113.359134855
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
RUNTYPE OPTIMIZE
SCFTYPE CASSCF
```

```
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

Assume that the above optimisation had converged, and at some point after the user wishes to compute a number of properties of the CASSCF wavefunction, but no longer has access to the Dumpfile used above. This may be accomplished using the following data sets, the first to compute an initial SCF wavefunction at the optimised CASSCF geometry, the second to re-do a single-point CASSCF calculation at this geometry, using the "PROPERTY ATOMS" directive (see §12) to obtain a variety of one-electron properties.

Run I: The Initial SCF Calculation

```
TITLE
H2CO - 3-21G  - SCF AT CASSCF GEOMETRY
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO    1.2406313\CH    1.1136939\HCO 123.1820211
END
ENTER
```

Run II: The CASSCF Property Analysis

```
RESTART
TITLE
H2CO - CASSCF PROPERTIES AT OPTIMISED GEOM.
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO    1.2406313\CH    1.1136939\HCO 123.1820211
END
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
PROPERTY ATOMS
ENTER
```

15.6.2 MCSCF Geometry Optimisation

Run I: The Initial SCF Calculation

```
TITLE
H2CO - 3-21G  - SCF STARTUP FOR MCSCF GEOM. 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
ENTER
```

Run II: The MCSCF Optimisation

```
RESTART
TITLE
H2CO - MCSCF GEOM OPT. (10E IN 9 M.O.)/TOTAL ENERGY -113.359134855
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
RUNTYPE OPTIMIZE
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
XTOL 0.0005
ENTER
```

The following points should be noted:

- Note the XTOL directive; this is used to converge the geometry optimisation more stringently, a typical tactic when subjecting the optimised geometry to a subsequent frequency analysis.
- It is **NOT** possible to use FZC orbitals (frozen core orbitals, those that will remain doubly occupied in all configurations) when performing MCSCF geometry optimisations. Presenting such a designator in the ORBITAL data e.g.

  ```
  FZC1 FZC1 FZC1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
  ```

  will lead to the following error diagnostic;

  ```
  ***************************************************
  **** cannot use FZC mos with MCSCF gradients   ****
  **** use the COR descriptor for these orbitals ****
  ***************************************************
  ```

- Note that the MCSCF module is approximately twice as fast as the CASSCF module for this calculation.
- Assume that the above optimisation had converged, and at some point after the user wishes to compute a number of properties of the CASSCF wavefunction, but no longer has access to the Dumpfile used above. This may be accomplished using the following data sets, the first to compute an initial SCF wavefunction at the optimised MCSCF geometry, the second to re-do a single-point MCSCF calculation at this geometry, using the "PROPERTY ATOMS" directive (see §12) to obtain a variety of one-electron properties.

Run I: The Initial SCF Calculation

```
TITLE
H2CO - 3-21G  - SCF AT MCSCF OPT. GEOM.
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO     1.2406315\CH     1.1136940\HCO  123.1819948
END
ENTER
```

Run II: The MCSCF Property Analysis

```
RESTART NEW
TITLE
H2CO -  MCSCF PROPERTIES AT OPTIMISED GEOM. (10E IN 9 M.O.)
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO     1.2406315\CH     1.1136940\HCO  123.1819948
END
SCFTYPE MCSCF
PROPERTY ATOMS
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
ENTER
```

15.6.3 MP2 Geometry Optimisation

MP2/RHF Optimisation Data

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF - 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
RUNTYPE OPTIMIZE
SCFTYPE MP2
XTOL 0.0001
ENTER
```

The following points should be noted:

- Use of the XTOL directive; this is to converge the geometry optimisation more stringently, a typical tactic when subjecting the optimised geometry to a subsequent frequency analysis.
- It is **NOT** possible to either freeze or discard orbitals when performing Møller Plesset geometry optimisations. Presenting the ACTIVE directive in such calculations will terminate the run.

MP3/RHF Optimisation Data

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP3/RHF - 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
RUNTYPE OPTIMIZE
SCFTYPE MP3
XTOL 0.0001
ENTER
```

MP2/UHF Optimisation Data

We consider performing the calculation in two steps, where the first carries out a UHF calculation, the second the MP2/UHF geometry optimisation. Valid data sequences for performing the calculation are shown below, where we again BYPASS initial integral evaluation in the MP2 job. Note again that this BYPASS'ing of the integral evaluation necessitates the introduction of SUPER OFF data line in the initial UHF job.

Run I: The UHF Calculation

```
TITLE
H2CO+ - DEFAULT BASIS - UHF CALCULATION
CHARGE 1
MULT 2
SUPER OFF
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
SCFTYPE UHF
ENTER
```

Run II: The MP2/UHF Calculation

```
RESTART NEW
TITLE
H2CO+ - 2B2 - 3-21G DEFAULT BASIS - MP2/UHF OPTIMISATION
CHARGE 1
MULT 2
BYPASS
SUPER OFF
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
RUNTYPE OPTIMIZE
XTOL 0.0001
SCFTYPE MP2
THRESH 7
ENTER
```

16 Transition State Optimisation

In the present section we consider the format of the data required when performing transition state optimisations using the modified trust region procedure [35] available in GAMESS–UK. We provide appropriate data for locating both

- the saddle point for the isomerisation process H_2CO to hydroxycarbene (t-HCOH,$^{1}\mathrm{A}'$), and

- the H_2CO (X^1A_1) to H_2 + CO molecular dissociation transition state

The following points should be noted:

1. transition state optimisation is requested by specifying the SADDLE option of the RUNTYPE directive. Restarting such calculations after a controlled dump again involves the SADDLE specification on the RESTART directive.
2. Optimisation is again conducted in a system of internal coordinates, specified through the VARIABLES and VARIABLE definition lines of the Z-matrix. The user must define both the geometry of the initial structure and the method(s) to be used in generating the initial Hessian matrix. While we confine the discussion to numerical evaluation of this initial Hessian at present, the user should consult §16.2 for guidance in performing this task by the more efficient analytic route.

In both examples below we follow the established technique of

- locating the transition state initially in a 'small' (STO-3G) basis
- using the final STO-3G Hessian, geometry and eigenvectors to instigate the 'larger' (3-21G) calculation

Example 1 : The (H_2CO to HCOH) transition structure

STO-3G Calculation

```
TITLE
HCOH <-> H2CO 1A' TS - STO3G
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3  TYPE 3
CO   1.27  TYPE 3
CH1  1.22  TYPE 3
CH5  1.10  TYPE 3
H5CO 115.8 TYPE 3
END
BASIS STO3G
RUNTYPE SADDLE
ENTER
```

3-21G Calculation

```
DUMPFILE ED3 350
TITLE
H2CO <-> HCOH 1A' TS 3-21G BASIS
ZMAT ANGS
```

```
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 57.236     TYPE 3
CO 1.299456     TYPE 3
CH1 1.201293    TYPE 3
CH5 1.115436    TYPE 3
H5CO 116.882    TYPE 3
END
RUNTYPE SADDLE ED3 1
VECTORS GETQ ED3 1 1
ENTER
```

The following points should be noted:

1. The Dumpfile for both calculations are sited on the same data set, that for the extended basis calculation commencing at block 350.

2. The initial Hessian for the STO-3G study is constructed numerically through the TYPE 3 specifications on the VARIABLE Definition lines.

3. The initial VARIABLE specifications in the 3-21G case are taken from the optimised STO-3G structure.

4. The Hessian in the 3-21G case is taken from the optimised STO-3G structure through the data line

   ```
   RUNTYPE SADDLE ED3 1
   ```

   Note that such data overrides the TYPE 3 specification on the VARIABLE definition lines.

5. The eigenvectors from the STO-3G calculation are taken to initiate the extended calculation through the data line

   ```
   VECTORS GETQ ED3 1 1
   ```

   where the first integer specified defines the starting block of the Dumpfile from the minimal basis calculation, the second integer the section wherein lies the minimal basis eigenvectors (the closed-shell SCF default vectors section).

Example 2 : The (H_2CO to H_2 + CO) transition structure

STO-3G Calculation

```
TITLE
H2CO <-> H2 + CO 1A' TS - STO3G
ZMAT ANGS
```

```
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134  TYPE 3
ANG1 43.7  TYPE 3
ANG2 57.8  TYPE 3
CHH  1.292 TYPE 3
XH 0.664   TYPE 3
END
BASIS STO3G
RUNTYPE SADDLE
ENTER
```

3-21G Calculation

```
DUMPFILE ED3 350
TITLE
H2 + CO <-> H2CO 1A' TS 3-21G BASIS
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134  TYPE 3
ANG1 43.7  TYPE 3
ANG2 57.8  TYPE 3
CHH  1.292 TYPE 3
XH 0.664   TYPE 3
END
RUNTYPE SADDLE ED3 1
VECTORS GETQ ED3 1 1
ENTER
```

Note that alternative methods for locating transition states are available within the program, namely:

1. a modified variant of the synchronous transit algorithm due to Bell and co-workers [36]. While we delay a detailed account of using this method until Part 4, we note here that;
    - the algorithm may be invoked through the LSEARCH directive (see Part 4, section 9.8);

  - additional data input is required, in particular specification of the two minima involved. This is achieved through the VARIABLE definition lines of the ZMATRIX directive (see for example 4.9.8);

2. a modified variant of the hill-walking algorithm due to Jorgensen and Simons [37, 38].

16.1 DFT Transition State Optimisation

We again use the (H_2CO to H_2 + CO) transition structure location to illustrate performing DFT optimisations. We perform the calculation in two steps, initially locating the transition state at the HF/SCF level, then using the resulting geometry and (in some cases) the hessian as a starting point for the DFT calculation. Note that it is not possible at present to use ”RUNTYPE HESSIAN” in DFT calculations, and the user should typically (i) restore the initial hessian in such a calculation from an initial HF transition state location, or (ii) use the ”TYPE 3” VARIABLES specification to compute an initial DFT hessian.

HF STO-3G Optimisation

```
TITLE
H2 + CO <-> H2CO 1A’ TS STO3G SCF TOTAL ENERGY = -112.1291164 AU
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134  TYPE 3
ANG1 43.7  TYPE 3
ANG2 57.8  TYPE 3
CHH  1.292 TYPE 3
XH 0.664   TYPE 3
END
BASIS STO3G
RUNTYPE SADDLE
ENTER
```

6-31G DFT S-VWN Optimisation

```
DUMPFILE ED3 350
TITLE
H2 + CO <-> H2CO 1A’ TS 6-31G DFT/S-VWN TOTAL ENERGY = -113.4117918572 au
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
```

```
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO            1.1525832 TYPE 3
CHH           1.2981078 TYPE 3
XH            0.6596229 TYPE 3
ANG1         43.4018534 TYPE 3
ANG2         57.3815232 TYPE 3
END
BASIS 6-31G
RUNTYPE SADDLE ED3 1
DFT S-VWN
VECTORS GETQ ED3 1
ENTER
```

The following points should be noted:

1. the starting variables for the initial geometry in the DFT calculation have been taken from the output of the previous HF optimisation.
2. the initial hessian is that from the previous HF/STO-3G calculation, restored from the STO3G Dumpfile using the "RUNTYPE SADDLE ED3" data line.
3. the initial vectors in the DFT calculation will be the final set of HF/STO3G SCF orbitals restored from section 1 of the STO3G Dumpfile.

16.2 CASSCF Transition State Optimisation

We again use the (H_2CO to H_2 + CO) transition structure location to illustrate performing CASSCF and MCSCF optimisations. In both cases we perform the calculation in two steps, initially locating the transition state at the HF/SCF level, then using the resulting geometry and (in some cases) the hessian as a starting point for the CASSCF calculation. Note that it is not possible at present to use "RUNTYPE HESSIAN" in CASSCF or MCSCF calculations, and the user should typically (i) restore the initial hessian in such a calculation from an initial HF transition state location, or (ii) use the "TYPE 3" VARIABLES specification to compute an initial DFT/CASSCF/MCSCF hessian.

HF Optimisation

```
TITLE
H2 + CO <-> H2CO 1A' TS 3-21G SCF TOTAL ENERGY = -113.0500312
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
```

```
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134  TYPE 3
ANG1 43.7  TYPE 3
ANG2 57.8  TYPE 3
CHH  1.292 TYPE 3
XH 0.664   TYPE 3
END
RUNTYPE SADDLE
ENTER
```

CASSCF Optimisation

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A’ TS 3-21G CASSCF TOTAL ENERGY = -113.22306125
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO             1.1525832 TYPE 3
CHH            1.2981078 TYPE 3
XH             0.6596229 TYPE 3
ANG1          43.4018534 TYPE 3
ANG2          57.3815232 TYPE 3
END
RUNTYPE SADDLE
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

The following points should be noted:

1. the starting variables for the initial geometry in the CASSCF calculation have been taken from the output of the previous HF optimisation.

2. the initial hessian is to be constructed numerically through the TYPE 3 specifications on the VARIABLE Definition lines. Using that from the previous HF/3-21G calculation could have been achieved by specifying "RUNTYPE SADDLE ED3".

3. the initial vectors in the CASSCF calculation will be the final set of HF/3-21G closed shell SCF orbitals restored from the default section 1 of the Dumpfile.

16.3 MCSCF Transition State Optimisation

We again use the (H_2CO to H_2 + CO) transition structure location to illustrate performing MCSCF optimisations. First we follow the CASSCF example, performing the calculation in two steps. Having located the transition state at the HF/SCF level, we then use the resulting geometry (and possibly the hessian) as a starting point for the MCSCF calculation. The data for the HF optimisation is as given above, that for the MCSCF optimisation as follows:

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A’ TS 3-21G BASIS/MCSCF TOTAL ENERGY = -113.223061249
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO        1.1525832 TYPE 3
CHH       1.2981078 TYPE 3
XH        0.6596229 TYPE 3
ANG1     43.4018534 TYPE 3
ANG2     57.3815232 TYPE 3
END
RUNTYPE SADDLE
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
ENTER
```

The following points should be noted:

1. the starting variables for the initial geometry in the CASSCF calculation have been taken from the output of the previous HF optimisation.

2. the initial vectors in the CASSCF calculation will be the final set of HF/3-21G SCF orbitals restored from the default closed-shell SCF vectors section 1 of the Dumpfile.

3. the initial hessian is again constructed numerically. Note that using the HF hessian (through ”RUNTYPE SADDLE ED3” specification) may not provide the optimal choice for post-HF calculations. With systems of fewer than 10 atoms, it is often more efficient, as above, to utilise the ”TYPE 3” feature to compute the initial hessian numerically, assuming the initial geometry is close to the final structure. Using the HF /3-21G hessian would be accomplished with the following data, where we are using the HF/3-21G TS geometry:

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' TS 3-21G / MCSCF TOTAL ENERGY = -113.223061249
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO        1.1525832
CHH       1.2981078
XH        0.6596229
ANG1     43.4018534
ANG2     57.3815232
END
RUNTYPE SADDLE ED3
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
ENTER
```

A total of 9 energy and gradient calculations are required in locating the transition state using the TYPE 3 specification, compared to 12 when using the initial SCF hessian. Note that the MCSCF module is approximately 3 times faster than the CASSCF module for this calculation.

16.4 MP2 Transition State Optimisation

We again use the (H_2CO to $H_2 + $CO) transition structure location to illustrate performing MP2 optimisations. First we follow the CASSCF and MCSCF examples, performing a 6-31G basis calculation in two steps. Having located the transition state at the HF/SCF level, we then use the resulting geometry (and possibly the hessian) as a starting point for the MP2 calculation. The data for the HF and MP2 optimisations are as follows:

HF Optimisation

```
TITLE
H2 + CO <-> H2CO 1A' TS 6-31G SCF TOTAL ENERGY = -113.629882925
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
```

```
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134  TYPE 3
ANG1 43.7  TYPE 3
ANG2 57.8  TYPE 3
CHH  1.292 TYPE 3
XH 0.664   TYPE 3
END
BASIS 6-31G
RUNTYPE SADDLE
ENTER
```

MP2 Optimisation

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' TS 6-31G BASIS / MP2 TOTAL ENERGY = -113.8779369833
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO            1.1565619 TYPE 3
CHH           1.2935171 TYPE 3
XH            0.6562584 TYPE 3
ANG1         42.5942740 TYPE 3
ANG2         57.8778292 TYPE 3
END
BASIS 6-31G
RUNTYPE SADDLE
XTOL 0.0005
SCFTYPE MP2
ENTER
```

The following points should be noted:

1. the starting variables for the initial geometry in the MP2 calculation have been taken from the output of the previous HF optimisation.

2. the initial vectors in the MP2 calculation will be the final set of HF/6-31G SCF orbitals restored from section 1 of the Dumpfile.

3. While the subject of a later section, we note here that the initial hessian for MP2 optimisations may be computed analytically (under RUNTYPE HESSIAN control), and this will typically be far more efficient than using numerical evaluation through TYPE 3 specification on the VARIABLES data lines. The above example is still using numerical evaluation (we provide the data for analytic computation below). Note again that using the HF

hessian (through "RUNTYPE SADDLE ED3" specification) may not provide the optimal choice for post-HF calculations. With systems of fewer than 10 atoms, it is often more efficient, as above, to utilise the "TYPE 3" feature or "RUNTYPE HESSIAN" to compute the initial hessian, assuming the initial geometry is close to the final structure. Using the HF/6-31G hessian would be accomplished with the following data, where we are using the HF/6-31G TS geometry:

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' TS 6-31G / MP2 TOTAL ENERGY = -113.8779369833
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO             1.1565619
CHH            1.2935171
XH             0.6562584
ANG1          42.5942740
ANG2          57.8778292
END
BASIS 6-31G
RUNTYPE SADDLE ED3
XTOL 0.0005
SCFTYPE MP2
ENTER
```

Having computed the initial hessian, either numerically or analytically, a further 5 energy and gradient calculations are required in locating the MP2 transition structure, to be compared with 12 such calculations when using the initial SCF hessian.

4. For the present transition state, there is in fact nothing to be gained by first locating the HF transition structure; the following data illustrates this point. We first compute analytically the MP2 hessian at the initial trial geometry, then restore this hessian (using the "RUNTYPE SADDLE FCM" syntax in searching for the MP2 transition structure. This requires the same number of energy plus gradient calculations, five in total, as required when using the converged HF/6-31G structure as a starting point. We further illustrate HESSIAN usage in §16.2 below.

MP2 Hessian Construction

```
TITLE
H2 + CO <-> H2CO 1A' TS MP2/6-31G COMPUTE INITIAL HESSIAN
ZMAT ANGS
```

```
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134
ANG1 43.7
ANG2 57.8
CHH  1.292
XH 0.664
END
BASIS 6-31G
RUNTYPE HESSIAN
ENTER
```

MP2 Transition State Location

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' TS 6-31G BASIS / MP2 TOTAL ENERGY = -113.877936986
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134
ANG1 43.7
ANG2 57.8
CHH  1.292
XH 0.664
END
BASIS 6-31G
RUNTYPE SADDLE FCM
SCFTYPE MP2
XTOL 0.0005
ENTER
```

17 Force Constant Calculations

GAMESS–UK may now perform a force constant calculation, together with associated vibrational frequencies, either

- analytically, driven by the RUNTYPE HESSIAN directive (for SCF and MP2 wavefunctions), or

- numerically, through RUNTYPE FORCE specification.

Analytic second derivatives are, of course, considerably faster and more accurate than numerical differentiation, and it is strongly recommended that the analytic option be employed when applicable (*vide infra*). As a rough guide the time required for evaluation of SCF second derivatives is approximately three times that required for gradient evaluation, depending upon the number of atoms and the symmetry of the molecule. The MP2 second derivatives take some 4-5 times longer than the MP2 gradient, and 2-3 times longer than the SCF force constants. The following points should be noted;

1. In both numerical and analytic force constant calculations the program prints the cartesian second-derivative matrix, in atomic units, and the normal coordinates and vibrational frequencies. All 3N normal coordinates are given. The three translational modes will all have frequencies close to zero, the degree to which they approach zero depending upon the degree of convergence of the SCF stage, so that this SCF should be converged more stringently than usual (this is done automatically by the FORCE and HESSIAN directives). The three rotational modes will only be zero at a stationary point on the potential surface, and consequently their values will depend upon the convergence of any preceding geometry optimisations. Generally, to ensure that the rotational modes all have frequencies below 10 wavenumbers, the XTOL directive should be employed in any preceding geometry optimisation to reduce all elements of the gradient to about 10^{-5} a.u.
2. Imaginary frequencies are printed as negative values. The analysis at the end of both FORCE and HESSIAN runs will project out the rotations from the force constant matrix giving 6(5 for linear molecules) very small frequencies.

17.1 Numerical Force Constants

Numerical force constant evaluation proceeds by taking finite differences of gradients, using either a 1-point (forward difference) or 2-point (central difference) formula. The formula required (default 1-point), together with the step size to be used in differencing (default 0.001) may be specified on the associated RUNTYPE directive. Presenting the data line

```
RUNTYPE FORCE
```

will yield the default options. Overriding may be achieved in obvious fashion, thus

```
RUNTYPE FORCE 2 0.003
```

with the integer specifying the required difference formula, followed by the required step size. The following points should be noted:

1. A numerical force constant calculation is only meaningful when performed at an optimised equilibrium or transition state geometry.

2. Numerical force constant calculations may now be performed with DFT, CASSCF and MCSCF wavefunctions, but not with CI wavefunctions.

3. Specification of the optimised geometry in a FORCE run may be controlled by the form of the RESTART directive. Using

   ```
   RESTART NEW
   ```

   will require the user entering the optimised variable values on the variable definition lines of the ZMATRIX directive. Assuming, however, that geometry optimisation had been conducted immediately prior to the the FORCE run, it would be sufficient to use just

   ```
   RESTART
   ```

   when the optimised geometry will be read from the Dumpfile, and override the ZMATRIX data in the input stream.

Example 1: The (H_2CO to H_2 + CO) transition structure

Assuming the (H_2CO to H_2 + CO) transition structure optimisation of Example 2.15 had successfully converged, the following data file would be used in calculating the numerical force constants;

```
DUMPFILE ED3 350
RESTART
TITLE
H2 + CO <-> H2CO 1A' TS 3-21G BASIS
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.20061  TYPE 3
ANG1 36.981  TYPE 3
ANG2 68.824  TYPE 3
CHH  1.189447 TYPE 3
XH 0.563628 TYPE 3
END
RUNTYPE FORCE
ENTER
```

17.1.1 CASSCF Force Constants

In the examples below we present the data files for force constant determination based on both CASSCF and MCSCF wavefunctions for (i) the ground state of formaldehyde, and (ii) the (H_2CO to H_2 + CO) transition structure, both conducted in a 3-21G basis.

Example 1: Force Constants for H_2CO

Initially we show the CASSCF geometry optimisation data, followed by the force constant run.

CASSCF geometry optimisation

```
RESTART
TITLE
H2CO - CASSCF GEOM OPT. (10E IN 9 M.O.)/TOTAL ENERGY  -113.359134854
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
RUNTYPE OPTIMIZE
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
XTOL 0.0005
ENTER
```

Note the XTOL directive; this is used to converge the geometry optimisation more stringently, a typical tactic when subjecting the optimised geometry to a subsequent frequency analysis.

CASSCF force constants

```
RESTART
TITLE
H2CO - CASSCF FORCE CONSTANTS GEOM OPT. (10E IN 9 M.O.)
# FREQ. 1186.7 1291.8 1546.9 1710.5 2823.3 2871.9
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
RUNTYPE FORCE
```

```
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

Example 2: The (H_2CO to H_2 + CO) transition structure

We now assume that the Dumpfile used when performing the transition state optimisation is no longer available, and that the user will perform the calculation from scratch, using just the computed geometry from the optimisation. Initially the user should perform an SCF calculation to generate a trial set of vectors for input to a single point CASSCF run at the transition state geometry (shown below), followed by the force constant run.

CASSCF run at the transition state geometry

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' 3-21G CASSCF AT OPT. TS GEOMETRY
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO     1.2034714\CHH    1.3040587\XH     0.7415226
ANG1  41.4929919\ANG2  56.6341870
END
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

CASSCF force constants

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A' TS 3-21G CASSCF FORCE CONSTANTS
# FREQ. -1762.3  770.4  901.5 1252.8 1720.4 3184.8
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
```

```
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO     1.2034714\CHH    1.3040587\XH     0.7415226
ANG1  41.4929919\ANG2  56.6341870
END
RUNTYPE FORCE
SCFTYPE CASSCF
CONFIG
FZC 1 TO 3
DOC 4 TO 8
UOC 9 TO 12
END
ENTER
```

17.1.2 MCSCF Force Constants

Example 1: Force Constants for H_2CO

Initially we show the MCSCF geometry optimisation data, followed by the force constant run.

MCSCF geometry optimisation

```
RESTART
TITLE
H2CO - MCSCF GEOM OPT. (10E IN 9 M.O.)/TOTAL ENERGY  -113.359135534
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
RUNTYPE OPTIMIZE
XTOL 0.0005
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
ENTER
```

Note again use of the XTOL directive to converge the geometry optimisation more stringently than the default.

MCSCF force constants

```
RESTART
TITLE
H2CO - 3-21G - MCSCF FORCE CONSTANTS (10E IN 9 M.O.)
# FREQ. 1187.2 1290.9 1544.9 1709.4 2822.0 2866.0
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
RUNTYPE FORCE
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
ENTER
```

Example 2: The (H_2CO to H_2 + CO) transition structure

We now assume that the Dumpfile used when performing the transition state optimisation is no longer available, and that the user will perform the calculation from scratch, using just the computed geometry from the optimisation. Initially we show the data to re-generate a set of MCSCF vectors at the transition state geometry, followed by the force constant run.

MCSCF run at the transition state geometry

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A’ 3-21G MCSCF AT OPT. TS GEOMETRY
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO     1.2034717\CHH    1.3040659\XH     0.7415189
ANG1  41.4927811\ANG2  56.6325324
END
SCFTYPE MCSCF
MCSCF
ORBITAL
```

```
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
ENTER
```

MCSCF force constants

```
RESTART NEW
TITLE
H2 + CO <-> H2CO 1A’ TS 3-21G MCSCF FORCE CONSTANTS
# FREQ. -1825.4 767.8  900.3 1259.4 1720.5 3185.4
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO     1.2034717\CHH    1.3040659\XH     0.7415189
ANG1  41.4927811\ANG2  56.6325324
END
RUNTYPE FORCE
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
ENTER
```

17.2 Analytic Force Constants

Analytic derivatives are available for closed-shell SCF and RHF open-shell wavefunctions, together with MP2 closed-shell wavefunctions. At present, UHF or pair-GVB wavefunctions, together with DFT, CASSCF and MCSCF wavefunctions and ECP–based calculations, still have to employ finite differences of gradients under control of RUNTYPE FORCE. The following points should be noted;

1. RUNTYPE HESSIAN is in fact a combination of tasks, requesting integral generation, SCF, gradient evaluation (with additional evaluation of derivative Fock operators), integral transformation, solution of the coupled Hartree-Fock (CHF) equations, calculation of the two-electron second derivative contribution and, finally, determination of the projected harmonic frequencies. While in most cases it is feasible to perform all steps in a single calculation, it may be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

2. The degree to which the three translational modes of the cartesian second–derivative matrix approach zero will depend upon the degree of convergence of the both the SCF

and CHF stages, and the SCF particularly should be converged more stringently than usual (this is done automatically by the HESSIAN directive). The three rotational modes will only be zero at a stationary point on the potential surface, and consequently their values will depend upon the convergence of any preceding geometry optimisations. Generally, to ensure that the rotational modes all have frequencies below 10 wavenumbers, the XTOL directive should be employed in any preceding geometry optimisation to reduce all elements of the gradient to about 10^{-5} a.u.

3. Several files will be generated under RUNTYPE HESSIAN processing.

   - For SCF force constants, these include:
     - the Mainfile (ED2) and Dumpfile (ED3).
     - the Scratch file (ED7).
     - the semi-transformed (ED4) and transformed (ED6) integral files (note that ED4 is also used as a scratch file in the solution of the coupled Hartree-Fock equations).
     - the Hamiltonian file (ED12), which acts to store the derivative Fock operators.
     - temporary files for sorting both transformed integrals (the Sortfile) and intermediate matrices in the Hessian calculation.
   - The generation of MP2 force constants is significantly more complex; in addition to the files generated under SCF processing, additional temporary files will be required, including ED0, ED11, ED16, ED17, ED18, ED18, ED19, MT0 and MT1.

   Any restart jobs will require ED6 and ED12 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

The following examples demonstrate HESSIAN usage, where in each case we show data files for performing the appropriate geometry optimisation, together with data for determining the force constants under RUNTYPE HESSIAN processing;

1. Optimisation of the geometry and calculation of the vibrational Frequencies for H_2CO;
2. Locating the (H_2CO to t-HCOH) transition structure, with subsequent determination of the vibrational frequencies;
3. Using a HESSIAN calculation to obtain the starting hessian required in locating the (H_2CO to t-HCOH) transition structure;
4. Locating the (H_2CO to H_2 + CO) transition structure;
5. Open–shell RHF geometry optimisation and force constants for the $^3A''$ state of H_2CO.
6. MP2 geometry optimisation and force constants for H_2CO.

Example 1: Vibrational Frequencies for H_2CO

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - 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
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

Run II. Force Constant Evaluation

Note the form of the RESTART directive below; since the geometry optimisation has been conducted immediately prior to the HESSIAN run, it is sufficient to use just

```
RESTART
```

when the optimised geometry will be read from the Dumpfile, and override the ZMATRIX data in the input stream.

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - FREQUENCIES
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
RUNTYPE HESSIAN
ENTER
```

Example 2: The (H_2CO to t-HCOH) transition structure

Run I: Transition Structure Determination

Here we are using TYPE 3 specification on the variable definition lines of the ZMATRIX in numerically deriving the initial Hessian to be used in locating the required transition structure. The final force constant matrix (Run II) at the optimised geometry is derived analytically under RUNTYPE HESSIAN control.

```
TITLE
HCOH <-> H2CO 1A’ TS - 6-31G
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3  TYPE 3
CO   1.27  TYPE 3
CH1  1.22  TYPE 3
CH5  1.10  TYPE 3
H5CO 115.8 TYPE 3
END
BASIS 6-31G
RUNTYPE SADDLE
XTOL 0.001
ENTER
```

Run II: Force Constant Evaluation

```
RESTART
TITLE
HCOH <-> H2CO 1A’ TS - 6-31G OPT. GEOM
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3
CO   1.27
CH1  1.22
CH5  1.10
H5CO 115.8
END
BASIS 6-31G
RUNTYPE HESSIAN
ENTER
```

Example 3: The (H_2CO to t-HCOH) transition structure

As described in the Geometry and Transition state optimisation sections above, it is necessary to generate a trial force constant matrix for use in locating a transition structure. Use of the HESSIAN runtype, conducted at the starting geometry to be employed in locating the transition structure, provides a powerful addition to the mechanisms outlined before. This is illustrated in the example below, where the hessian generated in **Run I** is subsequently restored in **Run II** through specification of the FCM keyword on the RUNTYPE SADDLE data line.

Run I: Generating the Trial Hessian

```
TITLE
```

```
HCOH <-> H2CO 1A' TS - 6-31G
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3
CO   1.27
CH1  1.22
CH5  1.10
H5CO 115.8
END
BASIS 6-31G
RUNTYPE HESSIAN
ENTER
```

Run II: Restoring the Trial Hessian in Saddle Point Location

```
RESTART NEW
TITLE
HCOH <-> H2CO 1A' TS - 6-31G
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3
CO   1.27
CH1  1.22
CH5  1.10
H5CO 115.8
END
BASIS 6-31G
RUNTYPE SADDLE FCM
ENTER
```

Example 4: The (H_2CO to H_2 + CO) transition structure

Again we generate the initial hessian at the starting geometry to be employed in the saddle point location in Run I, and restore this hessian through FCM specification on the SADDLE directive in Run II.

Run I: Generating a Trial Hessian

```
TITLE
H2CO <-> H2 + CO 1A' TS - 6-31G
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
```

```
 X 4 1.0 2 90.0 3 0.0
 X 4 1.0 5 ANG2 3 0.0
 H 4 XH 6 90.0 2 180.0
 H 4 XH 6 90.0 2 0.0
 VARIABLES
 CO 1.134
 ANG1 43.7
 ANG2 57.8
 CHH  1.292
 XH 0.664
 END
 BASIS 6-31G
 RUNTYPE HESSIAN
 ENTER
```

Run II: Restoring the Trial Hessian in TS Location

```
RESTART NEW
TITLE
H2CO <-> H2 + CO 1A' TS - 6-31G
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO 1.134
ANG1 43.7
ANG2 57.8
CHH  1.292
XH 0.664
END
BASIS 6-31G
RUNTYPE SADDLE FCM
ENTER
```

Example 5: Open–shell RHF Force Constants

In this example we demonstrate HESSIAN usage in an open–shell RHF force constant calculation for the $^3A''$ state of H_2CO. The calculation is performed in three stages, (a) an initial closed–shell SCF calculation to provide trial eigen vectors, (b) the open–shell RHF geometry optimisation – note the use of the LEVEL directive in increasing the default level–shifters and XTOL in providing more stringent criteria for convergence of the geometry optimisation, and (c) the final force constant calculation at the converged geometry.

Run I: Initial Closed–shell SCF

```
TITLE
```

```
H2CO - 3-21G - CLOSED SHELL SCF AT 3A’’ GEOMETRY
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
ENTER
```

Run II: Geometry Optimisation

```
RESTART NEW
TITLE
H2CO - 3-21G BASIS - 3A’’ STATE OPTIMISATION
MULT 3
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
RUNTYPE OPTIMIZE
SCFTYPE GVB
OPEN 2 2
LEVEL .3 1.0
XTOL 0.0001
ENTER
```

Run III: Force Constant Evaluation

```
RESTART
TITLE
H2CO - 3-21G BASIS - 3A’’ STATE - HESSIAN
MULT 3
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
```

```
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
RUNTYPE HESSIAN
SCFTYPE GVB
OPEN 2 2
LEVEL .3 1.0
ENTER
```

Example 6: MP2 Force Constants

Performing an analogous MP2 computation to the SCF calculation of Example 1 follows straightforwardly by introduction of the SCFTYPE MP2 data line.

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF - 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
RUNTYPE OPTIMIZE
SCFTYPE MP2
XTOL 0.0001
ENTER
```

Run II: MP2 Vibrational Frequencies

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF - FREQUENCIES
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
RUNTYPE HESSIAN
SCFTYPE MP2
ENTER
```

18 Polarisability Calculations

Analytic calculations of molecular polarisabilities may be conducted at both the SCF and MP2 levels. In the former case the coupled Hartree-Fock calculations of polarisabilities may be extended to include both frequency dependence, and magnetisabilities; it is thus possible to evaluate static and frequency dependent polarisabilities for closed–shell SCF wavefunctions, and from the latter obtain dispersion coefficients. Static polarisabilities only are available for open shell SCF wavefunctions, and for closed-shell MP2 wavefunctions.

The following points should be noted;

1. Polarisability calculations are performed under control of the RUNTYPE POLARISABILITY directive, and are available for for closed-shell SCF and MP2 wavefunctions, and open-shell (RHF) wavefunctions, but not at present UHF, CASSCF or MCSCF, or ECP–based calculations.

2. For SCF wavefunctions, the dipole-dipole , dipole-quadrupole and quadrupole-quadrupole polarisabilities are calculated. For closed–shell MP2 wavefunctions only the former terms are calculated.

3. RUNTYPE POLARISABILITY is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and solution of the coupled Hartree-Fock equations. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

4. Several files will be generated under RUNTYPE POLARISABILITY processing. For SCF wavefunctions, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the Scratch file (ED7).
   - the semi-transformed (ED4) and transformed (ED6) integral files (note that ED4 is also used as a scratch file in the solution of the coupled Hartree-Fock equations).
   - temporary files for sorting both transformed integrals (the Sortfile) and intermediate matrices
   - MP2 polarisability calculations are more complex with more files required; these include ED0, ED16 and ED17.

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

5. It is possible to calculate frequency dependent polarisabilities at both real and imaginary frequencies, and to obtain dispersion coefficients. It is also possible to obtain excitation energies using the RPA method for closed–shell SCF wavefunctions.

6. SCF convergence should be as a general rule be tightened under control of the THRESH directive when proceeding to any of the coupled Hartree-Fock steps within the program. Note that this is implemented automatically under control of the POLARISABILITY runtype, and should only be considered when using the BYPASS directive to omit the associated SCF processing.

Example 1: Polarisability of H_2CO

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - POLARISABILITY
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
RUNTYPE POLARISABILITY
ENTER
```

Example 2: Open–shell RHF Polarisability Calculation

In the example below we consider an open–shell RHF polarisability calculation of the $^2B_2$ state of H_2CO^+, optimising the molecular geometry in Run I and calculating the polarisability at this geometry in Run II.

Run I: Geometry Optimisation

```
TITLE
H2CO+ - 2B2 - DZP BASIS - OPEN SHELL RHF
CHARGE 1
MULT 2
SUPER OFF
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 DZP
RUNTYPE OPTIMISE
XTOL 0.0005
ENTER
```

Run II. Polarisability calculation

Note the form of the RESTART directive below; since the geometry optimisation has been conducted immediately prior to the POLARISABILITY run, it is sufficient to use just

```
RESTART
```

when the optimised geometry will be read from the Dumpfile, and override the ZMATRIX data in the input stream.

```
RESTART
TITLE
H2CO+ - 2B2 - DZP BASIS - POLARISABILITY
CHARGE 1
MULT 2
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 DZP
RUNTYPE POLARISABILITY
ENTER
```

Example 3: MP2 Polarisabilities

The corresponding MP2 calculation to that of Example 1 above is given below, where we now optimise the molecular geometry at the MP2 level, and compute the MP2 polarisability at this optimised geometry.

Run I: Geometry Optimisation

```
TITLE
H2CO - DZ BASIS - MP2/RHF - 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 OPTIMISE
SCFTYPE MP2
XTOL 0.0001
ENTER
```

Run II: Polarisability Calculation

```
RESTART
TITLE
H2CO - DZ BASIS - MP2/RHF - POLARISABILITY
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 POLARISABILITY
SCFTYPE MP2
ENTER
```

19 Hyperpolarisability Calculations

Analytic calculations of molecular hyperpolarisabilities may be conducted for both closed–shell SCF and open–shell RHF wavefunctions; These are third derivatives of the energy, and require preceding calculations of the first derivative wavefunctions.

The following points should be noted;

1. Hyperpolarisability calculations are performed under control of the RUNTYPE HYPER directive;

2. RUNTYPE HYPER is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and solution of the coupled Hartree-Fock equations. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation.

Example: Hyperpolarisability of H_2CO

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - GEOMETRY 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
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

Run II. Hyperpolarisability calculation

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - HYPERPOLARIZABILITY
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
RUNTYPE HYPER
ENTER
```

20 Magnetisability Calculations

Analytic calculations of molecular magnetisabilities may be conducted for closed–shell SCF wavefunctions only. The following points should be noted;

1. Magnetisability calculations are performed under control of the RUNTYPE MAGNET directive;

2. RUNTYPE MAGNET is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and solution of the coupled Hartree-Fock equations. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation.

Example: Magnetisability of H_2CO

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - GEOMETRY 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
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

Run II. Magnetisability Calculation

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS MAGNETISABILITY
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
RUNTYPE MAGNET
VCD
ENTER
```

21 Infra–red Intensity Calculations

Analytic calculations of infra–red intensities may be carried out for both closed-shell SCF and RHF open-shell wavefunctions, together with MP2 closed-shell wavefunctions. The following points should be noted;

1. Infra–red intensity calculations are performed under control of the RUNTYPE INFRARED directive.

2. RUNTYPE INFRARED is in fact a combination of tasks, requesting integral generation, SCF, gradient evaluation (with additional evaluation of derivative Fock operators), integral transformation, solution of the coupled Hartree-Fock (CHF) equations, calculation of the dipole moment derivatives, calculation of the two-electron second derivative contribution and, finally, determination of the infra–red intensities. While in most cases it is feasible to perform all steps in a single calculation, it may be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

3. Several files will be generated under RUNTYPE INFRARED processing.
   - For SCF intensities, these include:
     - the Mainfile (ED2) and Dumpfile (ED3).
     - the Scratch file (ED7).
     - the semi-transformed (ED4) and transformed (ED6) integral files (note that ED4 is also used as a scratch file in the solution of the coupled Hartree-Fock equations).
     - the Hamiltonian file (ED12), which acts to store the derivative Fock operators.
     - temporary files for sorting both transformed integrals (the Sortfile) and intermediate matrices in the Hessian calculation.
   - The generation of MP2 intensities is significantly more complex; in addition to the files generated under SCF processing, additional temporary files will be required, including ED0, ED11, ED16, ED17, ED18, ED18, ED19, MT0 and MT1.

   Any restart jobs will require ED6 and ED12 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

The following examples demonstrate INFRARED usage, where in each case we show data files for performing the appropriate geometry optimisation, together with data for determining the intensities under RUNTYPE INFRARED processing;

1. Optimisation of the geometry and calculation of the SCF infra–red intensities for H_2CO;
2. Open–shell RHF geometry optimisation and intensities for the $^3A''$ state of H_2CO.
3. MP2 geometry optimisation and infra–red intensities for H_2CO.

Example 1: SCF Infra–red Intensities for H_2CO

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - 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
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

Run II. Calculation of Infra–red Intensities

Note the form of the RESTART directive below; since the geometry optimisation has been conducted immediately prior to the INFRARED run, it is sufficient to use just

```
RESTART
```

when the optimised geometry will be read from the Dumpfile, and override the ZMATRIX data in the input stream.

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF - INFRARED
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
RUNTYPE INFRARED
ENTER
```

Example 2: Open–shell RHF Intensities

Run I: Initial Closed–shell SCF

```
TITLE
H2CO - 3-21G - CLOSED SHELL SCF AT 3A'' GEOMETRY
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
ENTER
```

Run II: Geometry Optimisation

```
RESTART NEW
TITLE
H2CO - 3-21G BASIS - 3A'' STATE OPTIMISATION
```

```
MULT 3
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
RUNTYPE OPTIMIZE
SCFTYPE GVB
OPEN 2 2
LEVEL .3 1.0
XTOL 0.0001
ENTER
```

Run III: Calculation of Infra–red Intensities

```
RESTART
TITLE
H2CO - 3-21G BASIS - 3A'' STATE - INFRARED
MULT 3
ZMATRIX ANGSTROM
C
O 1 CO
X 1 1.0 2 90.0
H 1 CH 2 HCO 3 DI1
H 1 CH 2 HCO 3 DI2
VARIABLES
CO 1.203
CH 1.099
HCO 121.8
DI1 15.0
DI2 164.0
END
RUNTYPE INFRARED
SCFTYPE GVB
OPEN 2 2
LEVEL .3 1.0
ENTER
```

Example 3: MP2 Infra–red Intensities

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF - 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
RUNTYPE OPTIMIZE
SCFTYPE MP2
XTOL 0.0001
ENTER
```

Run II: MP2 Intensities

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - MP2/RHF - INFRARED
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
RUNTYPE INFRARED
SCFTYPE MP2
ENTER
```

22 Calculation of Raman Intensities

The calculation of Raman Intensities is only available for closed-shell SCF wavefunctions, and is performed under control of the RUNTYPE RAMAN specification. This involves the calculation of a force constant matrix, together with the analytic evaluation of polarizability derivatives .

1. RUNTYPE RAMAN is in fact a combination of tasks, requiring integral generation, SCF, gradient evaluation (with additional evaluation of derivative Fock operators), integral transformation, solution of the coupled Hartree-Fock equations, calculation of analytic second derivatives and finally, the calculation of the polarizability derivatives. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

2. Several files will be generated under RUNTYPE RAMAN processing. These include;

   - the Mainfile (ED2) and Dumpfile (ED3).

- the Scratch file (ED7).
- the semi-transformed (ED4) and transformed (ED6) integral files (note that ED4 is also used as a scratch file in the solution of the coupled Hartree-Fock equations).
- the Hamiltonian file (ED12), which acts to store the derivative Fock operators.
- temporary files for sorting both transformed integrals (the Sortfile) and intermediate matrices in the Hessian calculation

Any restart jobs will require ED6 and ED12 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

Example: Raman Intensities for H_2CO

Run I: Geometry Optimisation

```
TITLE
H2CO - 3-21G DEFAULT BASIS - GEOMETRY 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
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

Run II: Calculation of Raman Intensities

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - RAMAN INTENSITIES
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
RUNTYPE RAMAN
ENTER
```

23 Direct-CI Calculations

Direct-CI calculations are performed under control of the RUNTYPE CI specification, with data input characterising the nature of the CI introduced by a data line with the keyword DIRECT in the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS or ENTER. Before detailing example data files for performing direct-CI calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. RUNTYPE CI is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and, finally, the CI calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

2. Several files will be generated under RUNTYPE CI processing. For Direct-CI calculations, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Direct-CI file (ED5), which acts to 'carry' data between the various phases of the calculation, and must be preserved between separate jobs.
   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals (the Sortfile) and intermediate matrices in the CI calculation (the P-Sortfile).

   Any restart jobs will require ED6 and ED5 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

3. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
           SUPER OFF NOSYM
   ```

   in the SCF run.

4. In all direct-CI calculations the user *must* specify the division of the molecular orbital space into an internal and external space, where the internal space comprises all orbitals that appear in any of the nominated reference functions. The latter are defined by use of the CONF directive. The set of internal orbitals *must* appear first in the list of active orbitals, followed by the set of external orbitals.

5. An additional reordering is performed by the code whereby MOs of common irreducible representation are grouped together, both in the internal and external space. This reordering is driven off the symmetry characteristics of the input MOs, as reported by the

parent SCF calculation. If for any reason these orbitals are 'contaminated', the automatic treatment of symmetry may breakdown, leading to unreliable results in the subsequent CI (see Part 8).

6. In default all molecular orbitals will be deemed active in the CI calculation.

23.1 Direct-CI - Single-reference CISD Calculations

A direct-CI calculation is to performed on the formaldehyde molecule, using the SCF configuration as the sole reference function. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO - 3-21G  CISD DIRECT-CI CALCULATION
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
RUNTYPE CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
NATORB 11 0 PRINT
ENTER
```

The following points should be noted:

- The DIRECT data line carries three integers, defining in order
  - NELEC, the number of active electrons in the CI (16 in this case)
  - NINT, the number of orbitals in the *internal* space (8 in this example, the doubly occupied SCF MOs)
  - NEXT, the number of orbitals in the *external* space (14 in this example, the total number of MOs (the number of basis functions (22) as no orbitals have been frozen or discarded), minus the number of *internal* orbitals (8)).
- Each reference function in the CI is defined as a sequence of NINT integers specifying the orbital occupancy of each *internal* orbital in the function. In the present case this corresponds to a single data line, specified under the CONF directive, containing 8 data fields each set to the value 2 corresponding to the doubly occupied orbitals of the SCF reference function.
- The NATORB directive requests generation of the natural orbitals (NOs) of the CI wavefunction, with the spinfree NOs to be output to section 11 of the Dumpfile. The second integer on this line refers to routing of the spin natural orbitals (set to 0 here given a

closed-shell system), while the PRINT keyword requests printing of the NOs. Note that both spinfree and spin-natural orbitals are now generated in default (Version 6.3 onwards) so that the NATORB directive may now be omitted. The spinfree natural orbitals will then be written to section 11 and the spin natural orbitals (open-shell systems) to section 12 of the Dumpfile.

- The set of molecular orbitals to be used in the transformation and subsequent CI are restored from the section nominated on the ENTER directive. In this example such usage is clear, but the user need consider this usage in cases e.g., open-shell calculations where multiple section specification may arise. In such cases of multiple specification, the final ENTER section nominated will be used as the eigenvector source.

Let us now consider a direct-CI calculation on the $^2B_2$ state of H_2CO^+, again using the SCF configuration as the sole reference function. A valid data sequence for performing such a calculation is shown below, where we are still performing all the computation in a single job.

```
TITLE
H2CO+ - 2B2 - 3-21G  CISD DIRECT-CI CALCULATION
SUPER OFF NOSYM
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
RUNTYPE CI
OPEN 1 1
DIRECT 15 8 14
SPIN DOUBLET
CONF
2 2 2 2 2 2 2 1
NATORB 11 12 PRINT
ENTER
```

Considering the changes to the closed-shell run, the following points should be noted:

- Of the three integers on the DIRECT data line, NINT and NEXT remain unchanged, while NELEC, the number of active electrons, is now 15.

- The occupation number of the 8th SCF MO on the CONF data line is now 1, reflecting the open shell orbital occupancy.

- The OPEN directive is now present, specified prior to the direct-CI data.

- An additional directive is required in the direct-CI data, SPIN, defining the spin multiplicity of the CI wavefunction.

- NATORB now requests the spinfree and spin NOs to be routed to sections 11 and 12 respectively of the Dumpfile.

- The set of vectors used in the integral transformation will be restored in default from the Dumpfile section containing the energy-ordered SCF orbitals, as written to by the SCF process (section 5, see Table 1).

Now let us consider performing the two calculations above splitting each into a sequence of jobs, where the first job carries out the SCF, the second the transformation and CI. First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - 3-21G  SCF PRIOR TO DIRECT-CI CALCULATION
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
```

The only obvious point to note is the use of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and CI Job

```
RESTART
TITLE
H2CO - 3-21G  CISD DIRECT-CI CALCULATION
SUPER OFF NOSYM
BYPASS SCF
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 CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run are restored in default from Section 1 of the Dumpfile, and subsequently used in the transformation.

The calculation may be further subdivided by splitting Run II above into separate integral transformation and CI runs using the RUNTYPE TRANSFORM specification, with subsequent BYPASS'ing of the transformation in the CI job. Thus:

Run IIa: The Transformation Job

```
RESTART
TITLE
H2CO - 3-21G  CISD DIRECT-CI CALCULATION
SUPER OFF NOSYM
BYPASS SCF
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 TRANSFORM
ENTER
```

Run IIb: The Direct-CI Job

```
RESTART
TITLE
H2CO - 3-21G  CISD DIRECT-CI CALCULATION
SUPER OFF NOSYM
BYPASS TRANSFORM
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 CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
ENTER
```

23.2 Direct-CI - Default CISD Calculations

In order to simplify the process of configuration specification and data preparation, the Direct-CI module now provides a set of default options that require little or no data input. While these defaults are not expected to cover most in-depth requirements, they do provide a starting point for users, and a route to subsequent, more extensive calculations. To illustrate this default working of the module, we consider below a number of example calculations based on those described in the preceding sections.

23.2.1 Closed-shell Systems

A Direct-CI calculation is to performed on the formaldehyde molecule. Given the following data sequence:

```
TITLE
H2CO - 3-21G DEFAULT DIRECT-CI CISD OPTION
ZMAT 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 CI
ENTER
```

then the calculation undertaken will be based on the following;

1. The format of the 2e-integral file will be automatically set to the required "SUPER OFF NOSYM", triggered by the presence of the CI runtype.
2. Integral transformation will use the set of orbitals from section 1, the default section for output of the closed-shell SCF eigenvectors. All orbitals will be deemed ACTIVE in the transformation.
3. The Direct-CI module is the default module loaded under RUNTYPE CI control, so that the DIRECT directive is not required.
4. The division of the molecular orbital space into an internal and external space, typically specified by the DIRECT directive, is now handled automatically, with the internal space comprising all doubly occupied SCF MOs orbitals, the external space all SCF virtual MOs. All electrons will be deemed active in the CI.
5. The SYMMETRY and SPIN of the CI wavefunction are taken to be those of the SCF wavefunction.
6. A single reference configuration will be employed, just the SCF configuration; the final configuration space will include all single and double excitations from this SCF reference configuration.
7. The spinfree natural orbitals will be written to section 11 of the Dumpfile.

The full data specification corresponding to the defaults generated from the above data file is as shown before, namely:

```
TITLE
H2CO - 3-21G  CISD DIRECT-CI CALCULATION
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
RUNTYPE CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
NATORB 11 0 PRINT
ENTER
```

23.2.2 Open-shell Systems

Let us now consider a Direct-CI calculation on the $^2B_2$ state of H_2CO^+, again using default options available within the module. A valid data sequence for performing such a calculation is shown below:

```
TITLE
H2CO+ 2B2 3-21G - DEFAULT CISD DIRECT-CI OPTION
MULT 2
CHARGE 1
ZMAT 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 CI
ENTER
```

As with the closed-shell run above, no explicit data is required to define the nature of the CI calculation. In practice the defaults adopted correspond to the following:

1. The CI will be based on the high-spin open-shell RHF calculation.

2. The set of vectors used in the transformation will be the energy-ordered SCF orbitals from section 5 of the Dumpfile, the default section in the absence of section specification on the ENTER directive.

3. The symmetry and spin of the CI wavefunction will be deduced from the preceding SCF calculation i.e. a doublet CI wavefunction of B_2 symmetry (corresponding to SPIN 2).

4. The number of active electrons in the CI will be set to be those involved in the SCF calculation (i.e. 15).

5. The reference configuration to be employed will be just the open-shell SCF configuration. The internal space comprises the doubly plus singly occupied SCF orbitals, with the external space comprising the SCF virtual orbitals. All electrons will be deemed active in the CI.

6. The spinfree natural orbitals will be written to section 11 and the spin natural orbitals to section 12 of the Dumpfile.

The full data specification corresponding to the defaults generated from the above data file is as shown before, namely:

```
TITLE
H2CO+ - 2B2 - 3-21G  CISD DIRECT-CI CALCULATION
SUPER OFF NOSYM
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
RUNTYPE CI
OPEN 1 1
DIRECT 15 8 14
SPIN DOUBLET
CONF
2 2 2 2 2 2 2 1
NATORB 11 12 PRINT
ENTER
```

23.3 Direct-CI - Freezing and Discarding Orbitals

In the examples above we have assumed that all MOs, typically generated at SCF time, are active in the subsequent CI calculation. In many instances however this will not be the case, for the user may wish to

- 'freeze' inner-shell orbitals, performing a 'valence-only' CI calculation.
- discard certain virtual orbitals from the CI calculation, typically the high-energy inner-shell complement orbitals.

The CORE and ACTIVE directives are provided for controlling the final subset of orbitals for inclusion in the CI. The freezing of core, or inner-shell, orbitals is achieved by nominating the sequence nos. of those orbitals to be frozen under control of the CORE directive. The discarding of orbitals is performed under control of the ACTIVE directive, which specifies the sequence nos. of the active set of orbitals to appear in the CI. Consider the H_2CO calculation above. The following data sequence would be required to freeze both the O1s and C1s orbitals (with SCF sequence numbers 1 and 2 respectively) and to discard the corresponding inner-shell complement virtual orbitals (with SCF sequence numbers 21 and 22):

```
TITLE
H2CO - 3-21G  CISD VALENCE-ONLY DIRECT-CI
```

```
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
RUNTYPE CI
CORE
1 2 END
ACTIVE
3 TO 20 END
DIRECT 12 6 12
CONF
2 2 2 2 2 2
ENTER
```

The following points should be noted:

- both ACTIVE and CORE are control directives of the integral transformation module. As such they should be presented in the data stream prior to the specification of the direct-CI data i.e., *before* the DIRECT data line.
- the values of NELEC, NINT and NEXT specified on the DIRECT data line are modified to reflect the impact of CORE and ACTIVE. This involves in the present case reducing the values from the all-electron calculation (NELEC=16, NINT=8, NEXT=14) to NELEC=12 (only 12 electrons now explicitly considered), NINT=6 (two orbitals have been frozen) and NEXT=12 (two orbitals having been discarded).
- the CONF data line now comprises six integers specifying the double occupancy of the internal orbital set.
- the default settings of CORE and ACTIVE are, hopefully, self-evident.

23.4 Direct-CI - Multi-reference CI Calculations

In the simplest case, specification of additional reference functions in the Direct-CI input data is accomplished through the CONF directive, with each reference function characterised by an additional data line of NINT integers defining the orbital occupation pattern of the required function. There is the constraint imposed, however, that the NINT internal orbitals appear first in the final list of orbitals, preceding the list of external orbitals. In some cases this ordering will not be obeyed in the input set of MOs, and the user must, under control of the ACTIVE directive, re-order the input set to achieve the required ordering. This highlights the underlying requirement of orbital occupancy specification under control of CONF, namely that *occupancy specification refers to the ordering of MOs as specified by the ACTIVE directive*.

Consider initially an example where the ordering of the SCF MOs is consistent with the specification of the reference set, and where reordering ACTIVE will not be required (although it may still be required in the freezing/discarding of MOs). Assume that we wish to perform a

3-reference CI calculation for H_2CO, comprising the SCF configuration, that arising from the double excitation $1b_1$ to $2b_1$ and that from the double excitation $5a_1$ to $6a_1$. This leads to the following occupation patterns for the 3 reference functions:

Reference Function	$1a_1$	$2a_1$	$3a_1$	$4a_1$	$1b_2$	$5a_1$	$1b_1$	$2b_2$	$2b_1$	$6a_1$
1	2	2	2	2	2	2	2	2	0	0
2	2	2	2	2	2	2	0	2	2	0
3	2	2	2	2	2	0	2	2	0	2

Then the internal space must , in addition to the ground state doubly occupied SCF MOs, include the $2b_1$ and $5a_1$ virtual orbitals i.e., NINT=10. Then each data line of the CONF directive will comprise ten integers reflecting the occupation pattern above:

```
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 0 2 2 2 0
2 2 2 2 2 0 2 2 0 2
```

The full data input for the job would be as follows:

```
TITLE
H2CO - 3-21G  CISD 3-REFERENCE CI
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
RUNTYPE CI
DIRECT 16 10 10
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
ENTER
```

23.5 Direct-CI - Multi-reference MP2 and MP3 Calculations

As an alternative to the multi-reference CI calculations described in the previous section, the user may also perform multi-reference MP2 and MP3 calculations [49, 50, 51, 52]. The basic principle is that the program computes the wavefunction in the basis of those reference functions nominated by the user. This wavefunction is then used as the reference function for generating the single and double excitations, and the subsequent solving of the perturbation equations.

Data input when performing multi-reference MP calculation is very similar to that required for a multi-reference CI calculation, e.g. the following job is the multi-reference MP2 equivalent of the multi-reference CI example of the previous section:

```
TITLE
H2CO - 3-21G  CISD 3-REFERENCE MP2
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
RUNTYPE CI
DIRECT 16 10 10
MP 2
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
ENTER
```

where the `MP 2` directive switches the program to do the multi-reference MP2 calculation. The output shows a block like:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$             Multi-reference MP2                 $$
$$                 model    -1                     $$
$$    |ref>F<ref|+|sin>F<sin|+|doub>F<doub|        $$
$$          Nref      3   Spin      1              $$
$$                                                 $$
$$      E-reference      -113.2222609769           $$
$$      E-correlation      -0.1660233176           $$
$$      --- total ---    -113.3882842945           $$
$$                                                 $$
$$      <psi1|psi1>          0.63871E-01           $$
$$        residue            0.46973E-07           $$
$$                                                 $$
$$             singles        doubles              $$
$$  vacuum -0.240888E-03   -0.240306E-01           $$
$$  n-1    -0.250548E-03   -0.567533E-01           $$
$$  n-2     0.000000E+00   -0.847480E-01           $$
$$                                                 $$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

where

@E-reference@ gives the energy of the multi-configuration reference wavefunction,

@E-correlation@ is the amount of correlation energy obtained from the MP2 calculation,

@¡psi1—psi1¿@ is the norm of the first order correction to the wavefunction as obtained by the MP2 calculation. This number may be used as a diagnostic to judge whether the perturbation approach is valid.

The multi-reference MP3 job corresponding to the MP2 example above is as follows:

```
TITLE
H2CO - 3-21G  CISD 3-REFERENCE MP3
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
RUNTYPE CI
DIRECT 16 10 10
MP 3
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
ENTER
```

Because the MP3 energy can be obtained from the first order corrected wavefunction, the second order correction to the wavefunction is not computed. Therefore the only information obtained additional to the MP2 calculation is the MP3 correlation energy and the MP3 total energy which is printed as:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$             Multi-reference MP3                 $$
$$              Ecorr      -0.1741698388           $$
$$              E        -113.3964308158           $$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

However, for a consistent treatment of the perturbation theory one should optimise the orbitals for the state of interest. In general that will involve an MCSCF calculation of some sort. Suppose that we want to calculate an excited state of formaldehyde with the same symmetry as the ground state. We start by building the guess orbitals with a simple RHF calculation

```
TITLE
H2CO - 3-21G  CLOSED SHELL SCF
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
```

Next we perform an MCSCF calculation for the excited of interest, immediately followed by the MRMP calculation:

```
RESTART NEW
TITLE
H2CO - 3-21G  CISD 12-REFERENCE MP2 EXCITED STATE
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 CI
SCFTYPE MCSCF
MCSCF
ORBITALS
4COR1 2COR3 DOC1 DOC2 UOC1 UOC2
END
STATE 2
WEIGHT 0 1
COPT 0.1 -1.0
CANONICAL 2 FOCK DENSITY FOCK
PRINT CIVECTOR
DIRECT 16 10 10
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 2 0 2 0
2 2 2 2 2 2 2 0 0 2
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 2 0 2 0 2
2 2 2 2 2 2 0 0 2 2
2 2 2 2 2 2 2 1 0 1
2 2 2 2 2 2 0 1 2 1
2 2 2 2 2 2 1 2 1 0
2 2 2 2 2 2 1 0 1 2
2 2 2 2 2 2 1 1 1 1
END
TRIAL DIAG REF STATE 2 PRINT
MP 3
ENTER 2
```

In this case the MCSCF ORBITALS directive asks the program to construct a CAS space. The directives STATE and WEIGHT make the program optimise 2 states where the ground state isn't taken into account in the optimisation. This way we are guaranteed to find the first excited state.

From the MCSCF run we only obtain the orbitals; due to differences in the representation of the CI vector we cannot transfer that information from the MCSCF to the CI module. Therefore we have to reconstruct the reference wavefunction in the CI module itself. First we specify all the reference configurations that were present in the MCSCF calculation, for example with the CONF directive. Next we ask for the first excited state in that reference space in the TRIAL DIAG directive. Having constructed the correct reference wavefunction we can simply apply perturbation theory to it using the MP directive.

23.6 Direct-CI - Restarting Calculations

In the examples considered above, we have assumed that the Direct-CI job completes in the time allocated. This may not be the case and we need consider restarting the computation in a controlled fashion. Such a requirement may be met in RUNTYPE CI processing when:

- the associated integral evaluation or SCF has not completed, due either to lack of time or to convergence problems in the SCF
- the integral transformation has not completed due to lack of time
- CI processing has not completed, due invariably to lack of time or to convergence problems in the iterative processing associated with the Davidson diagonalisation

Restarting the computation is achieved under control of the RESTART directive, which nominates the CI task for restarting. Consider the multi–reference CI job described above, and let us assume the job dumped during the Davidson diagonalisation. The full data input for the restart job would be as follows:

```
RESTART CI
TITLE
H2CO - 3-21G  CISD 3-REFERENCE CI
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
RUNTYPE CI
DIRECT 16 10 10
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
ENTER
```

The only change to the startup–job is the RESTART directive. Note that the default VECTORS and ENTER data will still apply in the RESTART job, with the default VECTORS section for closed-shell SCF orbitals that is written in the startup–job now being used as the source of eigenvectors. The following points should be noted:

- Within the Direct-CI module itself, restarts are only possible if the iterative diagonalisation has been initiated i.e. if the startup–job runs out of time in the CI–Hamiltonian builder or in the post–sort, the user must repeat that processing with an increased allocation of time.
- Any restart–jobs will require some, if not all, of the files associated with RUNTYPE CI processing. The safest course of action is to save all the files i.e. ED2, ED4, ED5, ED6

and, of course, the Dumpfile ED3. In some environments it may be essential to minimize the amount of disk space retained between jobs, in which case the user should be aware of the crucial files involved in restarting each of the sub-tasks associated with CI processing. Together with the Dumpfile, these are

– the SCF – ED2

– the Transformation – ED2, ED4 and ED6

– the Direct-CI – ED5 and ED6

Thus if the user is confident that all SCF and Transformation processing will complete in the time allocated, with any possible restart localised to the CI phase, then ED4 and even ED2 may be relegated to scratch status.

23.7 Direct-CI - Property Calculations

Computing the default set of one-electron properties at completion of CI processing may be readily accomplished through the addition of the PROPERTY ATOMS data line. Note that any such calculation will retrieve the spinfree, and where relevant, the spin NOS from the Dumpfile, such orbitals having been written to the default sections (11 for the spinfree and 12 for the spin NOs), or to those sections nominated on the NATORB directive.

```
TITLE
H2CO - 3-21G  CISD DCI + PROPERTIES CALCULATION
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
RUNTYPE CI
PROPERTY ATOMS
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
ENTER
```

To further illustrate property evaluation for CI wavefunctions, let us consider a CI calculation on the $^3A^2$ state of H_2CO. First, the data for the open-shell SCF calculation;

```
TITLE
H2CO - DZ BASIS - 3A2 GRHF  TOTAL ENERGY = -113.73954029 AU
MULT 3
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
BASIS DZ
ENTER
```

Having generated the SCF wavefunction, the following data sequence would be used for a single reference CI calculation; with the spinfree and spin natural orbitals routed to sections 11 and 12 of the Dumpfile by default, subsequent property generation may be requested by presenting the data line PROPERTY ATOMS.

```
RESTART NEW
TITLE
H2CO - DZ BASIS - 3A2 CISD DIRECT-CI -113.934177537 AU
MULT 3
SUPER OFF NOSYM
BYPASS SCF
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 DZ
RUNTYPE CI
PROPERTY ATOMS
DIRECT 16 9 15
SPIN 3
CONF
2 2 2 2 2 2 2 1 1
ENTER
```

Note that properties could also have been calculated after the CI job by specifying the appropriate natural orbitals under RUNTYPE ANALYSE. The data below would compute the isotropic ESR coupling constants (property index 19) at carbon, oxygen and hydrogen, where the spin NOS are nominated on the VECTORS line.

```
RESTART NEW
TITLE
H2CO - DZ - 3A2 UHF SPIN DENSITIES
MULT 3
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 DZ
RUNTYPE ANALYSIS
PROPERTY
19 C
19 O
19 H
END
```

```
VECTORS 12
ENTER
```

24 Table-CI Calculations

GAMESS-UK now contains two separate modules for performing Table-CI calculations, the original Conventional module that involves explicit storage of the CI hamiltonian on disk, and a new, semi-direct module that avoids explicit storage of the hamiltonian, and is capable of handling significantly larger secular problems. While the Conventional module will ultimately be phased out, our intention at this stage is to support both, so that the data requirements and file handling characteristics of both are described below.

Table-CI calculations are performed under control of the RUNTYPE CI specification, with data input characterising the nature of the CI introduced by a data line with the keyword MRDCI in the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS or ENTER.

Before detailing example data files for performing both Conventional and Semi-direct Table-CI calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. The data requirements, computational strategy and overall philosophy of the Table-CI modules are quite distinct from the Direct-CI module described above.

2. The aim of both modules is to calculate one or more roots of a given symmetry from a Multi-Reference CI calculation. Both modules can also calculate transition moments (TM) between states of the same symmetry or states of different symmetry, in addition to CI-Dipole and Quadrupole moments. The modules are based on the Table-CI algorithm of R.J.Buenker [39], the main practical difference between this and the Direct-CI module being the use of configuration selection and energy extrapolation.

3. The final list of selected configurations is derived from an initial list of configurations generated by single plus double excitations from a user–specified list of reference functions. Note that there is effectively no limit on this number of initial configurations. The selection and extrapolation procedure may be applied to a number of roots of a given secular problem. The Direct-CI module, of course, considers explicitly all configurations which are single and double excitations of a given set of reference configurations: this module is typically limited to the lowest few roots of a given symmetry.

4. RUNTYPE CI is in fact a combination of tasks, requesting integral generation, SCF, and finally, the various sub-tasks associated with the Table-CI calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS keyword on the data lines initiating each of the sub-modules. We illustrate this point in the sections below.

24.1 Table-CI and Molecular Symmetry

A crucial requirement in running the Table-CI modules is an understanding of the treatment of symmetry. Unlike the SCF and direct-CI modules, the molecular orbitals are automatically reordered at the outset of the CI into groups belonging to the same irreducible representation, with the ordering within each group dictated by the ordering encountered at orbital generation time (i.e. at SCF time). Note that each representation has an associated index number e.g. in a system of C_{2v} symmetry the four representations a_1, b_1, b_2 and a_2 have associated index numbers of 1,2 3 and 4 respectively. Groups of orbitals of common representation are ordered by virtue of increasing representation sequence number, so that in a C_{2v} system all molecular orbitals of a_1 symmetry would occur first in the list (with the occupied orbitals preceding the virtual orbitals in the subset), followed by the orbitals of b_1 symmetry (again with the DOMOS preceding the VMOS), followed by orbitals of b_2 symmetry (DOMOS before VMOS) and finally, orbitals of a_2 symmetry. Any subsequent reference to the orbitals, for example when specifying the reference functions, *must* be in this revised numbering scheme. Let us consider an example to try and clarify this point. Consider again the output from the closed shell SCF calculation on H_2CO, in particular the symmetry adapted basis set information,

```
=============================
IRREP  NO. OF SYMMETRY ADAPTED
       BASIS FUNCTIONS
=============================
  1          12
  2           4
  3           6
=============================
```

and the list of MOs printed at convergence:

```
===============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
===============================================
   1     1    -20.48275080           2.0000000
   2     1    -11.28286952           2.0000000
   3     1     -1.40833443           2.0000000
   4     1     -0.86648626           2.0000000
   5     3     -0.69818828           2.0000000
   6     1     -0.63034883           2.0000000
   7     2     -0.52027278           2.0000000
   8     3     -0.43433094           2.0000000
   9     2      0.14397469           0.0000000
  10     1      0.27419771           0.0000000
  11     3      0.36740523           0.0000000
  12     1      0.45123743           0.0000000
  13     2      0.93266602           0.0000000
  14     3      1.02032602           0.0000000
  15     1      1.02498516           0.0000000
  16     1      1.14613786           0.0000000
  17     3      1.27971217           0.0000000
  18     1      1.57176247           0.0000000
  19     2      1.86744709           0.0000000
```

```
   20      1      1.91087974          0.0000000
   21      3      1.98262324          0.0000000
   22      1      3.31460342          0.0000000
===============================================
```

Based on the reordering scheme outlined above, the table below outlines the sequence numbers of the MOs both prior to and after reordering. Note that a list of irreducible representations (IRreps) and their associated indices for each of the abelian point groups are given in Table 7. With the molecular orbitals reordered thus, the user must apply the revised numbering scheme in specification of, for example, the reference configurations. Thus consider the SCF configuration for H_2CO in terms of the doubly occupied SCF m.o.s:

m.o.	$1a_1$	$2a_1$	$3a_1$	$4a_1$	$1b_2$	$5a_1$	$1b_1$	$2b_2$
SCF ordering	1	2	3	4	5	6	7	8
Table ordering	1	2	3	4	17	5	13	18

Each reference function in the CI is defined in terms of the reordered MOs under control of the CONF directive, with the m.o.s in each representation presented in turn, in order of increasing representation number. Thus the following sequence:

```
1    2    3    4    5    13    17    18
```

would define the SCF configuration for H_2CO. Note that an additional integer is required in specifying the number of open-shell orbitals (NONO, non-identically coupled orbitals) in each function. This value is specified first in the CONF data sequence, and would typically be followed by a sequence of NONO integers defining the orbitals in question. In the present case NONO is zero, as all m.o.s are doubly occupied, so that the full CONF data line would be:

```
0    1    2    3    4    5    13    17    18
```

IRrep	IRrep No.	SCF Sequence No.	Table-CI Sequence No.	Occupation No.
a_1	1	1	1	2.0
		2	2	2.0
		3	3	2.0
		4	4	2.0
		6	5	2.0
		10	6	0.0
		12	7	0.0
		15	8	0.0
		16	9	0.0
		18	10	0.0
		20	11	0.0
		22	12	0.0
b_1	2	7	13	2.0
		9	14	0.0
		13	15	0.0
		19	16	0.0
b_2	3	5	17	2.0
		8	18	2.0
		11	19	0.0
		14	20	0.0
		17	21	0.0
		21	22	0.0

Let us consider the specification for the following configuration:

$$1a_1^2 2a_1^2 3a_1^2 4a_1^2 1b_2^2 (5a_1 6a_1)(1b_1 2b_1) 2b_2^2 \tag{1}$$

In this case there are two non-identically spin-coupled pairs i.e., 4 orbitals, which must be specified first in the CONF data line. This would then be:

```
4  5  6  13  14    1   2   3   4   17   18
```

where the four orbitals, 5(5a_1), 6(6a_1), 13(1b_1) and 14(2b_1) precede the doubly occupied orbitals in the list.

24.2 Conventional Table-CI Calculations

There is a formal limit of 200,000 selected configurations derived from an initial list of configurations generated by single plus double excitations from a user–specified list of reference functions. The selection and extrapolation procedure may be applied on up to twenty roots of a given secular problem.

Table 7: Irreducible Representations and Associated Indexing used in the Table-CI Module

Point Group	IRrep	Sequence No.
C_s	a'	1
	a''	2
C_2	a	1
	b	2
C_i	a_g	1
	a_u	2
C_{2v}	a_1	1
	b_1	2
	b_2	3
	a_2	4
C_{2h}	a_g	1
	a_u	2
	b_u	3
	b_g	4
D_{2h}	a_g	1
	b_{3u}	2
	b_{2u}	3
	b_{1g}	4
	b_{1u}	5
	b_{2g}	6
	b_{3g}	7
	a_u	8

1. The Conventional Table-CI module comprises a set of 9 sub-modules, which must be user–driven (either implicitly or explicitly, see below) through data input. These sub modules are as follows:

   - ADAPT: generation of a symmetry adapted list of integrals, derived by a pseudo-transformation from the list of 'raw' integrals.
   - TRAN: integral transformation, using the list of adapted integrals generated above together with a molecular orbital coefficient array nominated by the user. Note that in contrast to the Direct-CI module, transformation is an integral part of the Conventional Table-CI module. required by both the SELECT and CI sub-modules (see below – this data base will usually be available on a given machine, but may be generated by the user).
   - SELECT: performs configuration generation and subsequent selection based on a user-specified set of reference configurations and appropriate thresholds.
   - CI: generates the CI-hamiltonian based on the set of selected configurations from SELECT and integrals from TRAN.
   - DIAG: calculates one or more CI eigenfunctions of the hamiltonian generated under CI.

   The remaining modules are optional, and may be used to analyse one or more of the CI eigenvectors:

   - NATORB: generate the spin-free natural orbitals for one or more of the calculated CI eigenvectors.
   - PROP: compute various 1-electron properties of the CI wavefunctions. Note that the natural orbitals generated above may be routed to the Dumpfile and examined by the other analysis modules of GAMESS–UK in a subsequent job.
   - TM: compute the transition moments between nominated CI eigenvectors.

   Note at this point that there may be additional data input associated with each of the sub-modules e.g., for defining the reference configurations and selection attributes in SELECT.

2. In the interests of efficiency the Table-CI module requires as input a 'data-base' of pattern symbolic matrix elements for use in both the selection process and in construction of the final CI Hamiltonian over the selected configurations. These pattern elements are assumed to reside on a data set with LFN TABLE. The data base may be constructed in a given run of the Table-CI module by entering the TABLE sub-module prior to SELECT and CI. Thus the following data-driven loading of sub-modules:

```
  .
  .
MRDCI
ADAPT
TRAN
TABLE
```

```
SELECT
CI
DIAG
 .
 .
```

would be typical of that required when the user is explicitly constructing the TABLE data set in a given run of the program. Since TABLE generation is somewhat expensive, it will be more usual for the user to allocate a pre-generated version of the data set prior to executing the Table-CI modules. This allocation process and detailed locations of TABLE are, of course, machine specific, and will be outlined at the appropriate points in Parts 12–16 of the Manual. In this case the TABLE data line is simply omitted from the data sequence shown above, thus:

```
 .
 .
MRDCI
ADAPT
TRAN
SELECT
CI
DIAG
 .
 .
```

Note that failure to correctly allocate TABLE when using the above sequence will lead to an error condition.

3. Several direct-access files will be generated under RUNTYPE CI processing. For Conventional Table-CI calculations, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals and intermediate matrices in the CI calculation (the Sortfile).
   - in addition to the standard direct access files listed above, the Table-CI module makes extensive use of FORTRAN data sets (hereafter referred to as *interfaces*).

   Any restart jobs will require a subset of *interfaces* to be saved (see Table 8) in addition to the Dumpfile (ED3) and Mainfile (ED2). Extensive use is also made of scratch FORTRAN data sets, with LFNs FTN001, 002, 003, 004, 008, 009 and FTN010.

4. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

Table 8: FORTRAN Interfaces Used by the Conventional Table-CI Module

File	Contents	Generated by	Required by
		Sub-Module	Sub-Module
FTN022	Symmetry Adapted Integrals	ADAPT	TRAN
FTN031	Transformed Integrals	TRAN	SELECT, CI
FTN033	Partial Matrix Elements	SELECT	CI
FTN034	Partial Matrix Elements	SELECT	CI
FTN035	CI Hamiltonian	CI	DIAG
FTN036	CI Vectors	DIAG	NATORB PROP, TM

24.3 Conventional Table-CI - Single-reference CISD Calculations

A Conventional Table-CI calculation is to performed on the formaldehyde molecule, using the SCF configuration as the sole reference function. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 1M/1R
ZMAT 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 CI
MRDCI
ADAPT
TRAN
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
ROOTS 1
THRESH 30 10
CI
DIAG
EXTRAP 2
ENTER
```

The following points should be noted:

1. The ADAPT data line specifies generation of a symmetry adapted list of 1- and 2-electron integrals.

2. Integral transformation, requested through the TRAN directive, will use the converged closed shell SCF vectors resident in section 1 (see Table 1), the default closed-shell section. If section specification is made on the TRAN directive, it should point to this section i.e. TRAN 1.

3. In this example we are generating the TABLE data base (as requested by the presence of the TABLE data line) rather than restoring from the library file.

4. The majority of the data input characterising the CI calculation is presented under the SELECT keyword. In the present case we are:

   - requesting CI wavefunctions of A_1 symmetry (SYMMETRY 1)
   - requesting singlet CI wavefunctions (SPIN 1)
   - defining the number of active electrons in the CI through the CNTRL directive
   - requesting the inclusion of all singly excited configurations with respect to the reference configuration (SINGLES 1) regardless of the computed energy lowerings.
   - defining the reference configuration under control of the CONF directive
   - controlling the selection process through the ROOTS and THRESH directives. In the Table-CI procedure this selection process involves construction of an explicit zero-order Hamiltonian H_0 (over the nominated reference functions) followed by perturbative selection of configurations with respect to the user-nominated roots of H_0. The ROOTS directive specifies these roots - typically the nominated roots will exhibit a strong overlap with the final CI wavefunctions, with the dominant configurations in the final CI wavefunction present in the vectors of the zero-order Hamiltonian (indeed the process of extrapolation based on selection assumes this to be true). In the present case this specification is trivial - the zero-order Hamiltonian is simply a unit matrix comprising the SCF configuration.
   - The thresholds to be used in selection are specified on the THRESH data line. The first integer specified is the minimum threshold to to be used (T_{min}, in units of micro-Hartree), the second integer the increment to be used in defining higher-threshold cases to be solved in the process of extrapolation [39].

5. The CI data line requests construction of the CI Hamiltonian over the set of selected configurations .

6. The DIAG directive requests diagonalisation of the CI Hamiltonian, with EXTRAP requesting two extrapolation passes to be performed in the process of extrapolation to the zero-threshold limit (T=0).

The sequence of data lines defining the Conventional Table-CI calculation is terminated by the VECTORS directive. Let us now consider a Conventional Table-CI calculation on the $^2B_2$ state of H_2CO^+, again using the SCF configuration as the sole reference function. A valid data sequence for performing such a calculation is shown below, where we are still performing all the computation in a single job.

```
TITLE
H2CO+ - 2B2 - 3-21G  CISD TABLE-CI CALCULATION
CHARGE 1
MULT 2
ZMAT 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 CI
MRDCI
ADAPT
TRAN 5
SELECT
SYMMETRY 3
SPIN 2
CNTRL 15
SINGLES 1
CONF
1 18   1 2 3 4 5  13  17
ROOTS 1
THRESH 30 10
CI
DIAG
EXTRAP 2
ENTER
```

Considering the changes to the closed-shell run, the following points should be noted:

- The OPEN directive is not explicitly required since it we are performing a default high-spin RHF calculation. If presented, it should be specified prior to the Table-CI data.
- The set of vectors used in the Table-CI transformation will be restored from section 5 of the Dumpfile (specified by the TRAN 5 data line), having been placed in that Section by the SCF process. Note again that this corresponds to the default section number of the energy-ordered SCF eigenvectors generated by the open-shell SCF module (see Table 1).
- The majority of data changes appear within the SELECT data. Thus the SPIN directive now defines the spin multiplicity of the doublet CI wavefunction: SYMMETRY specifies the IRREP of the $^2B_2$ state (sequence number 3) while CNTRL defines the number of active electrons, now 15.
- The first integer of the CONF data line indicates a single open–shell orbital, the second the sequence number of that orbital (the $2b_2$, no. 16 in the reordered sequence) with the remaining integers the sequence numbers of the doubly occupied MOs.
- The TABLE directive is omitted, for we assume the data-base generated in the closed-shell run has been retained and allocated to the open-shell calculation.

Now let us consider performing the closed–shell calculation above in a sequence of jobs, where the first job carries out the SCF, the second the Table-CI calculation. Valid data sequences for performing the calculation are shown below.

Run I: The SCF Job

```
TITLE
H2CO - 3-21G  SCF PRIOR TO TABLE-CI CALCULATION
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
```

The only obvious point to note is the use of the SUPER directive in requesting full integral list generation required in the subsequent symmetry adaption and integral transformation.

Run II: The Table–CI Job

```
RESTART
TITLE
H2CO - 3-21G  TABLE-CI 1M/1R
SUPER OFF NOSYM
BYPASS SCF
ZMAT 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 CI
MRDCI
ADAPT
TRAN
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
ROOTS 1
THRESH 30 10
CI
DIAG
EXTRAP 2
ENTER
```

Considering the changes to the complete run, the following points should be noted:

- The SCF computation is BYPASS'ed.
- Note that the default VECTORS and ENTER data will still apply in the RESTART job, with the default VECTORS section for closed-shell SCF orbitals that is written in the startup–job now being used as the source of eigenvectors. Use of this default section will be carried through into the Table-CI module, so that explicit specification on the TRAN directive (i.e. TRAN 1 in the above) is not in fact required.

The calculation may be further subdivided by splitting Run II above into separate integral transformation and CI runs using the BYPASS keyword on the data lines of the appropriate Table-CI sub-modules to deactivate the computation accordingly. Thus:

Run IIa: The Transformation Job

```
RESTART
TITLE
H2CO - 3-21G  TABLE-CI 1M/1R -TRANSFORMATION
SUPER OFF NOSYM
BYPASS SCF
ZMAT 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 CI
MRDCI
ADAPT
TRAN
SELECT BYPASS
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
ROOTS 1
THRESH 30 10
CI BYPASS
DIAG BYPASS
EXTRAP 2
ENTER
```

Thus BYPASS is appended to the data lines requesting those Table-CI sub-modules (SELECT, CI and DIAG) to deactivate the associated processing.

Run IIb: The Table-CI Job

```
RESTART
TITLE
H2CO - 3-21G  TABLE-CI 1M/1R - SELECTION AND CI
```

```
SUPER OFF NOSYM
BYPASS SCF
ZMAT 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 CI
MRDCI
ADAPT BYPASS
TRAN BYPASS
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
ROOTS 1
THRESH 30 10
CI
DIAG
EXTRAP 2
ENTER
```

Now BYPASS is appended to both the ADAPT and TRAN data lines, since the associated processing has been completed in the previous job.

24.4 Conventional Table-CI - Freezing and Discarding Orbitals

In the examples above we have assumed that all MOs, typically generated at SCF time, are active in the subsequent CI calculation. In many instances however this will not be the case, for the user may wish to

- 'freeze' inner-shell orbitals, performing a 'valence-only' CI calculation.
- discard certain virtual orbitals from the CI calculation, typically the high-energy inner-shell complement orbitals.

The final subset of orbitals to be included in the CI is controlled by the specification of additional data for the TRAN sub-module. The freezing of core, or inner-shell, orbitals and the discarding of virtual orbitals is signaled by appropriate keywords on the TRAN directive (CORE and DISCARD respectively), with subsequent data lines nominating the number and sequence nos. of those orbitals *within* each IRrep to be frozen or discarded. Note that the sequence numbers to be specified refer to the *Table reordered orbital* set defined above. Consider the previous H_2CO calculation. Suppose we wish to freeze both the O1s and C1s orbitals (with SCF sequence numbers 1 and 2 respectively) and to discard the two highest–energy virtual orbitals (with SCF sequence numbers 21 and 22): The core orbitals are both of a_1 symmetry, and have sequence numbers 1 and 2. The virtual orbitals are of b_2 (SCF sequence no. 21) and a_1 (SCF sequence no. 22) symmetry, and as the highest orbital of each IRrep, correspond to the 6th orbital of b_2 symmetry and the 12th orbital of a_1 symmetry respectively. The TRAN data will then appear as follows

```
TRAN CORE DISCARD
2 0 0 0               ... core MOs
1 2
1 0 1 0               ... discarded MOs
12 6
```

where two additional data lines are associated with each category, the first specifying the *number* of orbitals within each IRrep, the second the sequence number of the orbitals in question. Note again that the sequence numbers refer to the numbering *within* each IRrep. Thus if we were to also freeze the $1b_2$ orbital, the revised TRAN data would appear as follows:

```
TRAN CORE DISCARD
2 0 1 0               ... core MOs
1 2 1
1 0 1 0               ... discarded MOs
12 6
```

Before detailing the Table-CI data, we should mention that the revised numbering scheme used in the specification of, for example, the reference configurations is that in effect *after* the freezing and discarding of orbitals. Having effectively removed three orbitals of a_1 symmetry and one of b_2 from the subsequent CI, the table below presents the final orbital numbering to be used in CONF specification:

IRrep	IRrep No.	SCF Sequence No.	Table-CI Sequence No.	Occupation No.
a_1	1	3	1	2.0
		4	2	2.0
		6	3	2.0
		10	4	0.0
		12	5	0.0
		15	6	0.0
		16	7	0.0
		18	8	0.0
		20	9	0.0
b_1	2	7	10	2.0
		9	11	0.0
		13	12	0.0
		19	13	0.0
b_2	3	5	14	2.0
		8	15	2.0
		11	16	0.0
		14	17	0.0
		17	18	0.0

The data for performing the Table-CI calculation is shown below:

```
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 1M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
ADAPT
TRAN CORE DISCARD
2 0 0 0
1 2
1 0 1 0
12 6
SELECT
SYMMETRY 1
SPIN 1
CNTRL 12
SINGLES 1
CONF
0 1 2 3  10  14 15
ROOTS 1
THRESH 30 10
```

```
CI
DIAG
EXTRAP 2
ENTER
```

The following points should be noted:

- the number of active electrons in the CI specified on the CNTRL data line is now 12.
- the integers specified on the CONF data line now label the six doubly-occupied orbitals in the revised numbering scheme outlined above.

24.5 Conventional Table-CI - Multi-reference CI Calculations

Specification of additional reference functions in the Table-CI input data is accomplished through the CONF directive, with each reference function characterised by an additional data line of integers defining

- the number, and sequence numbers, of any open shell orbitals, and
- the sequence numbers of the doubly-occupied orbitals.

Consider initially a 4-reference CI calculation for H_2CO, comprising the SCF configuration, that arising from the double excitation $1b_1$ to $2b_1$, that from the double excitation $2b_2$ to $3b_2$ and that from the excitation ($1b_12b_1$) to ($2b_23b_2$). This leads to the following occupation patterns for the 4 reference functions:

Reference Function	$1a_1$	$2a_1$	$3a_1$	$4a_1$	$1b_2$	$5a_1$	$1b_1$	$2b_2$	$2b_1$	$3b_2$
1	2	2	2	2	2	2	2	2	0	0
2	2	2	2	2	2	2	0	2	2	0
3	2	2	2	2	2	2	2	0	0	2
4	2	2	2	2	2	2	1	1	1	1

Then each data line of the CONF directive will reflect the occupation patterns above:

```
CONF
0 1 2 3 4 5  13  17 18        ..  Ref.1
0 1 2 3 4 5  14  17 18        ..  Ref.2
0 1 2 3 4 5  13  17 19        ..  Ref.3
4 13 14 18 19  1 2 3 4 5 17   ..  Ref.4
```

The full data input for the job would be as follows:

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
```

```
ZMAT 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 CI
MRDCI
ADAPT
TRAN
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
ROOTS 1
THRESH 30 10
CI
DIAG
EXTRAP 2
ENTER
```

24.6 Conventional Table-CI - Default Sub-module Attributes

To simplify the data-driven loading of sub-modules, the program assumes a default loading order so that, *assuming no additional data input* is required by a given sub-module i.e., the default attributes of that sub-module are in effect, the user may omit explicit specification of the module from the data input. The assumed default is shown below:

```
       .
       .
      MRDCI
      ADAPT
      TRAN
      SELECT
      CI
      DIAG
.
.
```

In practice the SELECT module will always require input, characterising for example the nature of the reference configurations, selection attributes etc, but in many instances the defaults of the other sub-modules will hold so that the associated data input may be omitted. Clearly this omission of data requires a firm understanding of the defaults in effect, which will only be apparent after the detailed description of directives presented in section 5. For the moment we illustrate this by considering the simplified data file for the multi-reference calculation on H_2CO above:

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
SELECT
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
ENTER
```

The following points should be noted:

- It is assumed that the complete sequence of sub-tasks is to be carried out. If any task is to be 'BYPASS'ed, then the associated data line must be present.
- In the absence of the TRAN directive, the transformation module will use the default section number corresponding to use of "ENTER" (i.e. section 1) directive as the location of the molecular orbital coefficient array.
- The SYMMETRY, SPIN, CNTRL, ROOTS and THRESH directives of the SELECT submodule may all be omitted, since the required specification corresponds in each case to the default values.
- Both CI and DIAG may be omitted, the EXTRAP specification of DIAG corresponding to the default value.

24.7 Conventional Table-CI - Restarting Calculations

In the examples considered above, we have assumed that the Table-CI job completes in the time allocated. This may not be the case and we need consider restarting the computation in a controlled fashion. Such a requirement may be met in RUNTYPE CI processing when:

- the associated integral evaluation or SCF has not completed, due either to lack of time or to convergence problems in the SCF;
- Table-CI processing itself has not completed.

In the present implementation it is not possible to restart Table-CI processing within a given submodule in the event of job termination due to lack of time. It is possible however to fragment the calculation into separate sub-module runs, through the use of the BYPASS directive on the sub-module data lines. In such usage restarting the computation is achieved under control of the RESTART directive, which nominates the CI task for restarting. Consider the Table-CI job of §16.5; we show below the data files for fragmenting this CI into,

- symmetry adaptation and integral transformation;
- configuration selection and hamiltonian construction;
- Davidson diagonalisation.

The subset of *interfaces* to be saved between the various steps is given in Table 8.

Adaptation and Transformation

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
SELECT BYPASS
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
CI BYPASS
DIAG BYPASS
ENTER
```

Selection and Hamiltonian Construction

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
ADAPT BYPASS
```

```
TRAN BYPASS
SELECT
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
DIAG BYPASS
ENTER
```

Diagonalisation

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
ADAPT BYPASS
TRAN BYPASS
SELECT BYPASS
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
CI BYPASS
ENTER
```

24.8 Semi-direct Table-CI Calculations

The formal limits that apply to conventional calculations are significantly extended in the semi-direct module. There is now a limit of 800,000 selected configurations derived from an initial list of configurations generated by single plus double excitations from a user–specified list of reference functions, the number of which may not exceed 256. The selection and extrapolation procedure may now be applied on up to thirty roots of a given secular problem.

Semi-direct Table-CI calculations are performed under control of the RUNTYPE CI specification, with data input characterising the nature of the CI introduced by a data line with the keyword MRDCI in the first data field and the keyword DIRECT in the second field. Termination of this data is again accomplished by presenting a valid *Class 2* directive, such as VECTORS or ENTER.

Note that while the Conventional and Semi-direct modules are based on the same Table-CI algorithm, there are significant differences in both file utilisation and data requirements. The most significant of these are highlighted below:

1. In contrast to the conventional module, the integral transformation is now performed under control of the conventional 4-index module of GAMESS, rather than the ADAPT and TRAN Table CI modules.

2. The memory requirements of the semi-direct module may be significantly greater than those associated with the conventional algorithm. While the default memory allocations will prove adequate for "small-medium" cases, the user should use the MEMORY pre-directive to request at least 8 MWords in calculations with, say, more than 20 active electrons.

3. The Semi-direct Table-CI module comprises a reduced set of 6 sub-modules, which may be user–driven (either implicitly or explicitly, see below) through data input. These sub modules are as follows:

   - TABLE: generates an input a 'data-base' of pattern symbolic matrix elements for use in both the selection process and in solving the secular problem.
   - SELECT: performs configuration generation and subsequent selection based on a user-specified set of reference configurations and appropriate thresholds. Note that the semi-direct module requires *at least two reference configurations*.
   - CI: provides pre-processing prior to the semi-direct evaluation of the CI eigenfunctions, followed by calculation, in semi-direct fashion, of one or more CI eigenfunctions of the secular problem. In contrast to the conventional module, just two secular problems are solved as part of the extrapolation process, one at the lowest threshold (T_{min}) and one at the threshold (T_{min} + T_{inc}).
   - NATORB: generate the spin-free natural orbitals for one or more of the calculated CI eigenvectors. Note that this module is now executed in default.

   The remaining analysis modules remain optional, and may be used to further analyse one or more of the CI eigenvectors:

   - PROP: compute various 1-electron properties of the CI wavefunctions. Note that the natural orbitals generated above may be routed to the Dumpfile and examined by the other analysis modules of GAMESS–UK in a subsequent job.
   - TM: compute the transition moments between nominated CI eigenvectors.

   Note at this point that there may be additional data input associated with each of the sub-modules e.g., for defining the reference configurations and selection attributes in SELECT.

4. In the interests of efficiency the Table-CI module again requires as input a 'data-base' of pattern symbolic matrix elements for use in both the selection process and in construction of the final CI Hamiltonian over the selected configurations. These pattern elements are assumed to reside on a data set with LFN "table-ci". The data base may be constructed in a given run of the Table-CI module by entering the TABLE sub-module prior to SELECT and CI. Thus the following data-driven loading of sub-modules:

```
  .
  .
MRDCI DIRECT
TABLE
SELECT
CI
NATORB
  .
  .
```

would be typical of that required when the user is explicitly constructing the TABLE data set in a given run of the program. This is now the recommended route in semi-direct calculations, rather than the user allocating a pre-generated version of the data set prior to executing the Table-CI modules. Note that failure to correctly allocate table-ci when using the above sequence will lead to an error condition.

5. Several direct-access files will be generated under RUNTYPE CI processing. For Semi-direct Table-CI calculations, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the Transformed Integral file (ED6).
   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals and intermediate matrices in the CI calculation (the Sortfile).
   - in addition to the standard direct access files listed above, the Table-CI module again makes extensive use of FORTRAN data sets (hereafter referred to as *interfaces*).

   Any restart jobs will require a subset of *interfaces* to be saved (see Table 9) in addition to the Dumpfile (ED3), Mainfile (ED2) and Transformed Integral File (ED6). Extensive use is also made of scratch FORTRAN data sets, with LFNs FTN001, 002, 003, 004, 008, 009, 010, 011, 022, 041, 043, and FTN044.

6. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

24.9 Semi-direct Table-CI - Multi-reference CI Calculations

We would again point out that all semi-direct Table-CI calculations require at least two reference configurations i.e. CISD calculations based on a single reference configuration are not possible with this module. However we do not consider this to be a major disadvantage given that the process of configuration choice and specification has been simplified through the use of automated configuration generation (see below).

Table 9: FORTRAN Interfaces Used by the Semi-direct Table-CI Module

File	Contents	Generated by Sub-Module	Required by Sub-Module
FTN031	Transformed Integrals	Transformation module	SELECT, CI
FTN033	Partial Matrix Elements	SELECT	CI
FTN034	Partial Matrix Elements	SELECT	CI
FTN042	Zero-order vectors	SELECT	CI
FTN012	Configuration data	SELECT	CI
FTN036	CI Vectors	CI	NATORB PROP, TM

Note again that specification of additional reference functions in the Table-CI input data is again accomplished through the CONF directive; in contrast to CONF specification in the conventional module, however, the data lines specifying the configurations are now terminated by a single data line containing the character string END in the first data field. Each reference function is characterised by an additional data line of integers defining

- the number, and sequence numbers, of any open shell orbitals, and
- the sequence numbers of the doubly-occupied orbitals.

Consider initially a 4-reference CI calculation for H_2CO, comprising the SCF configuration, that arising from the double excitation $1b_1$ to $2b_1$, that from the double excitation $2b_2$ to $3b_2$ and that from the excitation $(1b_1 2b_1)$ to $(2b_2 3b_2)$. This leads to the following occupation patterns for the 4 reference functions:

Reference Function	$1a_1$	$2a_1$	$3a_1$	$4a_1$	$1b_2$	$5a_1$	$1b_1$	$2b_2$	$2b_1$	$3b_2$
1	2	2	2	2	2	2	2	2	0	0
2	2	2	2	2	2	2	0	2	2	0
3	2	2	2	2	2	2	2	0	0	2
4	2	2	2	2	2	2	1	1	1	1

Then each data line of the CONF directive will reflect the occupation patterns above:

```
CONF
0 1 2 3 4 5  13  17 18        ..  Ref.1
0 1 2 3 4 5  14  17 18        ..  Ref.2
0 1 2 3 4 5  13  17 19        ..  Ref.3
4 13 14 18 19  1 2 3 4 5 17   ..  Ref.4
END                           ..  the directive terminator
```

The full data input for the job would be as follows:

```
TITLE
H2CO - 3-21G BASIS - semi-direct MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI DIRECT
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES ALL
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
ROOTS 1
THRESH 2 2
CI
NATORB
ENTER
```

Now let us consider performing the closed–shell calculation above in a sequence of jobs, where the first job carries out the SCF, the second the Table-CI calculation. Valid data sequences for performing the calculation are shown below.

Run I: The SCF Job

```
TITLE
H2CO - 3-21G SCF PRIOR TO TABLE-CI CALCULATION
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
```

The only obvious point to note is the use of the SUPER directive in requesting full integral list generation required in the subsequent symmetry adaption and integral transformation.

Run II: The Table–CI Job

```
RESTART
```

```
TITLE
H2CO - 3-21G BASIS - semi-direct MRDCI 4M/1R
SUPER OFF NOSYM
BYPASS SCF
ZMAT 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 CI
MRDCI DIRECT
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES ALL
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
ROOTS 1
THRESH 2 2
CI
NATORB
ENTER
```

Considering the changes to the complete run, the following points should be noted:

- The SCF computation is BYPASS'ed
- Note that the default VECTORS and ENTER data will still apply in the RESTART job, with the default VECTORS section for closed-shell SCF orbitals that is written in the startup–job now being used as the source of eigenvectors.

The calculation may be further subdivided by splitting Run II above into separate integral transformation and CI runs using the BYPASS keyword on the data lines of the appropriate Table-CI sub-modules to deactivate the computation accordingly. Thus:

Run IIa: The Transformation Job

```
RESTART
TITLE
H2CO - 3-21G  TABLE-CI 4M/1R -TRANSFORMATION
SUPER OFF NOSYM
BYPASS SCF
ZMAT 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 CI
MRDCI DIRECT
TABLE BYPASS
SELECT BYPASS
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES ALL
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
ROOTS 1
THRESH 2 2
CI BYPASS
NATORB BYPASS
ENTER
```

Thus BYPASS is appended to the data lines requesting those Table-CI sub-modules (SELECT, CI and NATORB) to deactivate the associated processing.

Run IIb: The Table-CI Job

```
RESTART
TITLE
H2CO - 3-21G  TABLE-CI 4M/1R - SELECTION AND CI
SUPER OFF NOSYM
BYPASS SCF TRAN
ZMAT 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 CI
MRDCI DIRECT
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES ALL
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
ROOTS 1
```

```
THRESH 2 2
CI
NATORB
ENTER
```

Now that TRAN is now appended to the BYPASS directive since the associated processing has been completed in the previous job.

24.10 Semi-direct Table-CI - Default MRDCI Calculations

In order to simplify the process of configuration specification and data preparation, the semi-direct module now provides a set of default options that require little or no data input. While these defaults are not expected to cover most in-depth requirements, they do provide a starting point for users, and a route to subsequent, more extensive calculations. To illustrate this default working of the module, we consider below a number of example calculations.

A Semi-direct Table-CI calculation is to performed on the formaldehyde molecule. Given the following data sequence:

```
TITLE
H2CO - 3-21G DEFAULT TABLE-CI OPTIONS
ZMAT 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 CI
MRDCI DIRECT
ENTER
```

then the calculation undertaken will be based on the following;

1. Integral transformation will use the set of orbitals from section 1, the default section for output of the closed-shell SCF eigenvectors.

2. The table-ci data base will be generated rather than restored from the library file.

3. A CI wavefunction of A_1 symmetry (i.e. SYMMETRY 1).

4. A singlet CI wavefunction (i.e. SPIN 1).

5. The number of active electrons in the CI will be set to be those involved in the SCF calculation (i.e. CNTRL 16).

6. Singly excited configurations with respect to each of the default reference configurations (SINGLES ALL) will be included, regardless of their computed energy lowerings.

7. The set of reference configurations to be employed will include the SCF configuration, plus those generated from this configuration by including (i) for each symmetry IRREP,

the doubly excited configuration arising from excitation of the highest occupied DOMO of that symmetry to the lowest virtual orbital (VMO) of the same symmetry, and (ii) the lowest singly excited configuration, again arising from the highest occupied DOMO to the lowest VMO of the same symmetry. In the present example, this will correspond to the SCF configuration, the double and single excitation arising from the DOMO $5a_1$ to VMO $6a_1$, the double and single excitation arising from the DOMO $1b_1$ to VMO $2b_1$, and the double and single excitation arising from the DOMO $2b_2$ to VMO $3b_2$. No reference configurations will be included involving orbitals of a_2 symmetry given the absence of such orbitals involved in the occupied manifold. This results in a total reference set of 7 functions, as shown thus in the job output:

```
   numbers of open shells and corresponding main configurations

  0            1   2   3   4   5  13  17  18       ..   SCF configuration
  0            1   2   3   4   6  13  17  18       ..   5a1 -> 6a1 double
  2            5   6   1   2   3   4  13  17  18   ..   5a1 -> 6a1 single
  0            1   2   3   4   5  14  17  18       ..   1b1 -> 2b1 double
  2           13  14   1   2   3   4   5  17  18   ..   1b1 -> 2b1 single
  0            1   2   3   4   5  13  17  19       ..   2b2 -> 3b2 double
  2           18  19   1   2   3   4   5  13  17   ..   2b2 -> 3b2 single
```

8. The default selection process subsequently undertaken is equivalent to the following ROOTS and THRESH directives.

   ```
   THRESH 10 10
   ROOTS 1
   ```

   Thus this default selection process involves construction of an explicit zero-order Hamiltonian H_0 (over the reference functions described above) followed by perturbative selection of configurations with respect to the lowest root of H_0. The minimum threshold to be used in selection (T_{min}) is 10 micro-Hartree, with an increment of 10 uH to be used in defining the higher-threshold case to be solved in the process of extrapolation [39].

9. In default the module will, having solved the secular problem for the lowest root of the CI secular problem, generate the spinfree natural orbitals from the associated CI eigenfunction.

The sequence of data lines defining the Semi-direct Table-CI calculation is terminated by the VECTORS directive. Note at this stage that the full data specification corresponding to the defaults generated from the above data file is as follows:

```
TITLE
H2CO - 3-21G - EXPLICIT DATA FOR DEFAULT MRDCI SETTINGS  -113.43885803
SUPER OFF NOSYM
ZMAT 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 CI
ACTIVE
1 TO 22 END
MRDCI DIRECT
TABLE
SELECT
CNTRL 16
SPIN 1
SYMM 1
SINGLES ALL
CONF
  0     1   2   3   4   5  13  17  18
  0     1   2   3   4   6  13  17  18
  2     5   6   1   2   3   4  13  17  18
  0     1   2   3   4   5  14  17  18
  2    13  14   1   2   3   4   5  17  18
  0     1   2   3   4   5  13  17  19
  2    18  19   1   2   3   4   5  13  17
END
THRESH 10 10
ROOTS 1
CI
NATORB
CIVEC 1
ENTER
```

Let us now consider a Semi-direct Table-CI calculation on the $^2B_2$ state of H_2CO^+, again using default options available within the module. A valid data sequence for performing such a calculation is shown below, where we are still performing all the computation in a single job.

```
TITLE
H2CO+ 2B2 3-21G - DEFAULT MRDCI SETTINGS  -113.06446075
MULT 2
CHARGE 1
ZMAT 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 CI
MRDCI DIRECT
ENTER
```

Considering the changes to the closed-shell run, the following points should be noted:

- The set of vectors used in the Table-CI transformation will be the energy-ordered SCF orbitals from section 5 of the Dumpfile, the default section in the absence of section specification on the ENTER directive.

- The symmetry and spin of the CI wavefunction will be deduced from the preceding SCF calculation i.e. a CI wavefunction of B_2 symmetry (corresponding to SYMMETRY 3). and a doublet CI wavefunction (corresponding to SPIN 2).

- The number of active electrons in the CI will be set to be those involved in the SCF calculation (i.e. CNTRL 15).

- Singly excited configurations with respect to each of the default reference configurations (SINGLES ALL) will be included, regardless of their computed energy lowerings.

- The set of reference configurations to be employed will follow the same algorithm used in the closed shell case above i.e. the SCF configuration, plus those generated from this configuration by including (i) for each symmetry IRREP, the doubly excited configuration arising from excitation of the highest occupied DOMO of that symmetry to the lowest virtual orbital (VMO) of the same symmetry, and (ii) the lowest singly excited configuration, again arising from the highest occupied DOMO to the lowest VMO of the same symmetry. In the present example, this will correspond to the SCF configuration, the double and single excitation arising from the DOMO $5a_1$ to VMO $6a_1$, the double and single excitation arising from the DOMO $1b_1$ to VMO $2b_1$, and the double and single excitation arising from the DOMO $1b_2$ to VMO $3b_2$. Note that the DOMO involved in the latter configurations is now the $1b_2$ given that the $2b_2$ is now singly occupied, and again the absence of excitations involving a_2 MOs given the absence of such orbitals involved in the occupied manifold. This again results in a total reference set of 7 functions, as shown thus in the job output:

```
  numbers of open shells and corresponding main configurations

 1          18   1   2   3   4   5  13  17       ..   SCF configuration
 1          18   1   2   3   4   6  13  17       ..   5a1 -> 6a1 double
 3           5   6  18   1   2   3   4  13  17  ..   5a1 -> 6a1 single
 1          18   1   2   3   4   5  14  17       ..   1b1 -> 2b1 double
 3          13  14  18   1   2   3   4   5  17  ..   1b1 -> 2b1 single
 1          18   1   2   3   4   5  13  19       ..   1b2 -> 3b2 double
 3          17  18  19   1   2   3   4   5  13  ..   1b2 -> 3b2 single
```

The sequence of data lines defining the Semi-direct Table-CI calculation is again terminated by the ENTER directive. Note at this stage that the full data specification corresponding to the defaults generated from the above data file is as follows

```
TITLE
H2CO+ 2B2 3-21G - EXPLICIT DATA FOR DEFAULTS  -113.06446075
MULT 2
CHARGE 1
SUPER OFF NOSYM
ZMAT 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 CI
OPEN 1 1
ACTIVE
1 TO 22 END
MRDCI DIRECT
TABLE
SELECT
CNTRL 15
SPIN 2
SYMM 3
SINGLES ALL
CONF
 1    18           1   2   3   4   5  13  17
 1    18           1   2   3   4   6  13  17
 3     5   6  18   1   2   3   4  13  17
 1    18           1   2   3   4   5  14  17
 3    13  14  18   1   2   3   4   5  17
 1    18           1   2   3   4   5  13  19
 3    17  18  19   1   2   3   4   5  13
END
THRESH 10 10
ROOTS 1
CI
NATORB
CIVEC 1
ENTER
```

24.11 Semi-direct Table-CI - Freezing and Discarding Orbitals

In the examples above we have assumed that all MOs, typically generated at SCF time, are active in the subsequent CI calculation. In many instances however this will not be the case, for the user may wish to

- ‘freeze’ inner-shell orbitals, performing a ‘valence-only’ CI calculation.
- discard certain virtual orbitals from the CI calculation, typically the high-energy inner-shell complement orbitals.

In contrast to the Conventional Table-CI module, the final subset of orbitals to be included in the Table-CI calculation is now controlled by the CORE and ACTIVE directives of the integral transformation module.

The freezing of core, or inner-shell, orbitals is achieved by nominating the sequence nos. of those orbitals to be frozen under control of the CORE directive. The discarding of orbitals is performed under control of the ACTIVE directive, which specifies the sequence nos. of the active set of orbitals to appear in the CI. Note that the sequence numbers to be specified refer to the *input orbitals*, typically those produced by the SCF code, and not the *Table reordered orbital* as in the conventional module.

Consider the previous H_2CO calculation. Suppose we wish to freeze both the O1s and C1s orbitals (with SCF sequence numbers 1 and 2 respectively) and to discard the two highest–

energy virtual orbitals (with SCF sequence numbers 21 and 22): The CORE and ACTIVE data will then appear as follows

```
CORE
1 2 END
ACTIVE
3 TO 20 END
```

The core orbitals are both of a_1 symmetry, and have sequence numbers 1 and 2. The virtual orbitals are of b_2 (SCF sequence no. 21) and a_1 (SCF sequence no. 22) symmetry, and as the highest orbital of each IRrep, correspond to the 6th orbital of b_2 symmetry and the 12th orbital of a_1 symmetry respectively. Before detailing the Table-CI data, we should mention that the revised numbering scheme used in the specification of, for example, the reference configurations is, as in the conventional case, that in effect *after* the freezing and discarding of orbitals. Having effectively removed three orbitals of a_1 symmetry and one of b_2 from the subsequent CI, the table below presents the final orbital numbering to be used in CONF specification:

IRrep	IRrep No.	SCF Sequence No.	Table-CI Sequence No.	Occupation No.
a_1	1	3	1	2.0
		4	2	2.0
		6	3	2.0
		10	4	0.0
		12	5	0.0
		15	6	0.0
		16	7	0.0
		18	8	0.0
		20	9	0.0
b_1	2	7	10	2.0
		9	11	0.0
		13	12	0.0
		19	13	0.0
b_2	3	5	14	2.0
		8	15	2.0
		11	16	0.0
		14	17	0.0
		17	18	0.0

The data for performing the semi-direct Table-CI calculation is shown below:

```
TITLE
H2CO - 3-21G BASIS - valence direct-MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
CORE
1 2 END
ACTIVE
3 TO 20 END
MRDCI DIRECT
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 12
SINGLES ALL
CONF
0 1 2 3  10  14 15
0 1 2 3  11  14 15
0 1 2 3  10  14 16
4 10 11 15 16 1 2 3 14
END
ROOTS 1
THRESH 2 2
CI
NATORB
ENTER
```

The following points should be noted:

- the number of active electrons in the CI specified on the CNTRL data line is now 12.
- the integers specified on the CONF data line now label the six doubly-occupied orbitals in the revised numbering scheme outlined above.

24.12 Semi-direct Table-CI - Default Sub-module Attributes

To simplify the data-driven loading of sub-modules, the program assumes a default loading order so that, *assuming no additional data input* is required by a given sub-module i.e., the default attributes of that sub-module are in effect, the user may omit explicit specification of the module from the data input. The assumed default is shown below:

```
  .
  .
MRDCI DIRECT
TABLE
SELECT
CI
NATORB
  .
  .
```

In practice the SELECT module will require input (except in cases where the default configuration generation described above is used), characterising for example the nature of the reference configurations, selection attributes etc, but in many instances the defaults of the other submodules will hold so that the associated data input may be omitted. Clearly this omission of data requires a firm understanding of the defaults in effect, which will only be apparent after the detailed description of directives presented in Part 6. For the moment we illustrate this by considering the simplified data file for the multi-reference calculation on H_2CO above:

```
RESTART
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI
SELECT
CNTRL 16
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
THRESH 2 2
ENTER
```

The following points should be noted:

- It is assumed that the complete sequence of sub-tasks is to be carried out. If any task is to be 'BYPASS'ed, then the associated data line must be present.
- The transformation module will again use the default section number corresponding to the closed-shell SCF vectors (section 1) as the location of the molecular orbital coefficient array.
- The SYMMETRY, SPIN, SINGLES and ROOTS directives of the SELECT sub-module may all be omitted, since the required specification corresponds in each case to the default values. Note however that the CNTRL directive *must* be specified in any SELECT data in which the CONF directive is also present (in contrast to the conventional module).
- To provide compatibility with the DIRECT-CI module, an alternative form of the CONF directive may also be used, where the configurations are defined in terms of occupation patterns rather than by the orbital numbering. In this case a second keyword, OCCUPATION, is specified on the CONF data line; thus in the present example, we may also use the following CONF data;

```
CONF OCCUPATION
 2   2   2   2   2   2   2   2   0   0  0
 2   2   2   2   2   2   0   2   2   0  0
 2   2   2   2   2   2   2   0   0   0  2
 2   2   2   2   2   2   1   1   1   0  1
END
```

where the occupations specified correspond to the occupancies of the input SCF MOs. At this stage we leave it to the user to confirm that this data is equivalent to the CONF specification in the example above.

- Both CI and NATORB may be omitted, the default values being required.

24.13 Semi-direct Table-CI - Restarting Calculations

In the examples considered above, we have assumed that the Table-CI job completes in the time allocated. This may not be the case and we need consider restarting the computation in a controlled fashion. Such a requirement may be met in RUNTYPE CI processing when:

- the associated integral evaluation, SCF or integral transformation has not completed, due either to lack of time or to convergence problems in the SCF;
- Table-CI processing itself has not completed.

In the present implementation it is not possible to restart Table-CI processing within a given sub-module in the event of job termination due to lack of time. It is possible however to fragment the calculation into separate sub-module runs, through the use of the BYPASS directive on the sub-module data lines. In such usage restarting the computation is achieved under control of the RESTART directive, which nominates the CI task for restarting. Consider the Table-CI job of §16.5; we show below the data files for fragmenting this CI into,

- integral transformation;
- configuration selection;
- hamiltonian pre-processing and Davidson diagonalisation.

The subset of *interfaces* to be saved between the various steps is given in Table 7.

Integral Transformation

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI DIRECT
TABLE BYPASS
SELECT BYPASS
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
CI BYPASS
NATORB BYPASS
ENTER
```

Configuration Selection

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI DIRECT
SELECT
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
CI BYPASS
NATORB BYPASS
ENTER
```

Diagonalisation and Natural Orbital Generation

```
RESTART CI
TITLE
H2CO - 3-21G DEFAULT BASIS - MRDCI 4M/1R
SUPER OFF NOSYM
ZMAT 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 CI
MRDCI DIRECT
SELECT BYPASS
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
CI
NATORB
ENTER
```

25 Full-CI Calculations

Full-CI calculations are performed under control of the RUNTYPE CI specification, with data input characterising the nature of the CI introduced by a data line with the keyword FULLCI in the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS. Before detailing example data files for performing full-CI calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. RUNTYPE CI represents a combination of tasks, requesting integral generation, SCF, integral transformation and, finally, the CI calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

2. The Full-CI procedure is, of course, exceedingly demanding in its memory requirements [40]. Of the machines discussed in Parts 12-16 of the manual.

3. Several direct-access files will be generated under RUNTYPE CI processing. For Full-CI calculations, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Scratch file (ED7).
   - temporary files for sorting transformed integrals (the Sortfile)

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

4. In addition to the direct-access files above, the full CI module uses conventional unformatted FORTRAN data sets. Four such files will be used, with LFNs FTN002, FTN003,FTN004 and FTN008. Note that any restart jobs will rely on the availability of the latter data set.

5. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

A Full-CI calculation is to performed on the H_2CO molecule. Before detailing the data requirements, let us again consider the mechanisms for restricting the scale of the all-electron computation, since this will often be required in full-CI treatments. The user will typically wish to

- 'freeze' inner-shell orbitals, performing a 'valence-only' full-CI calculation.
- discard certain virtual orbitals from the CI calculation, typically the high-energy inner-shell complement orbitals.

The CORE and ACTIVE directives of the transformation module are provided for controlling the final subset of orbitals for inclusion in the CI. The freezing of core , or inner-shell, orbitals is achieved by nominating the sequence nos. of those orbitals to be frozen under control of the CORE directive. The discarding of orbitals is performed under control of the ACTIVE directive, which specifies the sequence nos. of the active set of orbitals to appear in the CI. Turning to the H_2CO calculation, the following data sequence would be required to freeze the two inner shell and two lowest valence SCF-MOs while retaining all virtual orbitals in the subsequent full-CI treatment:

```
TITLE
H2CO - 3-21G BASIS - VALENCE FULL-CI
SUPER OFF NOSYM
NOPRINT
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
ACTIVE\5 TO 22 END\CORE\1 TO 4\END
RUNTYPE CI\FULLCI 18 4 4
ENTER
```

The following points should be noted:

- both ACTIVE and CORE are control directives of the integral transformation module. As such they should be presented in the data stream prior to the specification of the full-CI data i.e., *before* the FULLCI data line.
- The FULLCI data line carries three integers, defining in order
  - NACT, the number of active orbitals in the CI (18 in this case)
  - NALPHA, the number of α-electrons (4 in this example, the four highest doubly occupied SCF MOs).
  - NBETA, the number of β-electrons (again 4 in the closed shell molecule) example

  Note that the values of NACT, NALPHA and NBETA reflect the impact of CORE and ACTIVE – had we been considering an all-electron calculation, the integer values would have been 22, 8 and 8 respectively.
- The set of molecular orbitals to be used in the transformation and subsequent CI are restored from the section associated with the ENTER directive, either the default section (here section 1, that for the closed-shell SCF module) or that explicitly nominated nominated. In this example such usage is clear, but the user need consider this usage in cases e.g., open-shell calculations where multiple section specification may arise. In such cases of multiple specification, the final ENTER section nominated will be used as the eigenvector source.

Assuming the fortran file FTN008 had been saved, the following job would continue processing the above example, assuming this had terminated cleanly during the iterative Davidson process;

```
RESTART CI
TITLE
H2CO - 3-21G BASIS - VALENCE FULL-CI
SUPER OFF NOSYM
NOPRINT
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
ACTIVE\5 TO 22 END\CORE\1 TO 4\END
RUNTYPE CI\FULLCI 18 4 4
ENTER
```

Let us now consider the corresponding calculation on the $^2B_2$ state of H_2CO^+, now using the open–shell SCF orbitals. A valid data sequence for performing such a calculation is shown below, where we are still performing all the computation in a single job.

```
TITLE
H2CO+ - 2B2 - 3-21G  VALENCE FULL-CI
SUPER OFF NOSYM
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
RUNTYPE CI
OPEN 1 1
FULLCI 18 4 3
SYMMETRY 3
ENTER
```

Considering the changes to the closed-shell run, the following points should be noted:

- Of the three integers on the FULLCI data line, NACT and NALPHA remain unchanged, while NBETA, the number of β-electrons, is now 3.
- The OPEN directive is now present, specified prior to the full-CI data.
- An additional directive is required in the full-CI data, SYMMETRY, defining the spatial symmetry index of the CI wavefunction.
- The set of vectors used in the integral transformation will be restored from section 5 of the Dumpfile, the default section associated with the energy-ordered open-shell SCF MOS, having been placed in that Section by the SCF process.

Now let us consider performing the closed–shell calculation above in a sequence of jobs, where the first job carries out the SCF, the second the transformation and CI. First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - 3-21G  SCF PRIOR TO FULL-CI CALCULATION
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
```

The only obvious point to note is the use of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and CI Job

```
RESTART
TITLE
H2CO - 3-21G BASIS - VALENCE FULL-CI
SUPER OFF NOSYM
BYPASS SCF
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
ACTIVE\5 TO 22 END\CORE\1 TO 4\END
RUNTYPE CI\FULLCI 18 4 4
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run will be restored from Section 1 of the Dumpfile, the default section associated with the closed-shell SCF MOs.

The calculation may be further subdivided by splitting Run II above into separate integral transformation and CI runs using the RUNTYPE TRANSFORM specification, with subsequent BYPASS'ing of the transformation in the CI job. Thus:

Run IIa: The Transformation Job

```
RESTART
TITLE
H2CO - 3-21G  INTEGRAL TRANSFORMATION
SUPER OFF NOSYM
BYPASS SCF
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 TRANSFORM
ACTIVE\5 TO 22 END\CORE\1 TO 4\END
ENTER
```

Run IIb: The Full-Ci Job

```
RESTART
TITLE
H2CO - 3-21G  VALENCE FULL-CI CALCULATION
SUPER OFF NOSYM
BYPASS TRANSFORM
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 CI
ACTIVE\5 TO 22 END\CORE\1 TO 4\END
RUNTYPE CI\FULLCI 18 4 4
ENTER
```

26 Coupled-Cluster Calculations

Coupled-cluster (CC) calculations [41] are performed under control of the RUNTYPE CI specification, with data input characterising the nature of the CI introduced by a data line with the character string CCSD as the first four characters of the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS. Before detailing example data files for performing CC calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. RUNTYPE CI represents a combination of tasks, requesting integral generation, SCF, integral transformation and, finally, the coupled-cluster calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.

2. Several direct-access files will be generated under RUNTYPE CI processing. For coupled-cluster calculations, these include:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Scratch file (ED7).
   - temporary files for sorting transformed integrals (the Sortfile)

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

3. In addition to the direct-access files above, the coupled cluster module uses a variety of conventional unformatted FORTRAN data sets.

4. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

A CC calculation is to to be performed on the H_2CO molecule. Before detailing the data requirements, let us again consider the mechanisms for restricting the scale of the all-electron computation, since this will often be required in coupled-cluster treatments. The user will typically wish to

- 'freeze' inner-shell orbitals, performing a 'valence-only' coupled-cluster calculation.
- discard certain virtual orbitals from the coupled cluster calculation, typically the high-energy inner-shell complement orbitals.

The CORE and ACTIVE directives of the transformation module are provided for controlling the final subset of orbitals for inclusion in the CC. The freezing of core , or inner-shell, orbitals is achieved by nominating the sequence nos. of those orbitals to be frozen under control of the CORE directive. The discarding of orbitals is performed under control of the ACTIVE directive, which specifies the sequence nos. of the active set of orbitals to appear in the CC. Turning to the H_2CO calculation, the following data sequence would be required to freeze the two inner shell and two lowest valence SCF-MOs while retaining all virtual orbitals in the subsequent coupled-cluster treatment:

```
TITLE
H2CO - TZVP - VALENCE CCSD / CCSD  ENERGY = -114.2600151982
SUPER OFF NOSYM
NOPRINT
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 TZVP
ACTIVE\3 TO 50 END\CORE\1 TO 2\END
RUNTYPE CI\CCSD 48 6 6
CCTH 10
CCIT 30
ENTER
```

The following points should be noted:

- both ACTIVE and CORE are control directives of the integral transformation module. As such they should be presented in the data stream prior to the specification of the coupled-cluster data i.e., *before* the CCSD data line.
- The nature of the coupled cluster calculation is determined by the data line commencing with the character string CCSD. In this case a singles+doubles calculation is performed [42, 43]. Inclusion of the triples component to the correlation energy [44, 45] is requested by presenting the character string CCSD(T) (see below). The additional three integers, define in order
  - NACT, the number of active orbitals in the calculation (48 in this case)

  - NALPHA, the number of α-electrons (6 in this example, the four highest doubly occupied SCF MOs).
  - NBETA, the number of β-electrons (again 6 in the closed shell molecule) example

  Note that the values of NACT, NALPHA and NBETA reflect the impact of CORE and ACTIVE – had we been considering an all-electron calculation, the integer values would have been 52, 8 and 8 respectively.

- The set of molecular orbitals to be used in the transformation and subsequent CC calculation are restored from the default section (section 1) associated with the closed-shell SCF module.

- Two additional directives, CCIT and CCTH, may be used to control convergence of the iterative CCSD procedure:

  - CCIT: This directive consists of a single data line, read to variables TEXT, MXCCIT using format (A,I), where TEXT should be set to the character string CCIT. MXCCIT is an integer used to specify the maximum number of cycles required in the iterative CCSD procedure.

    Note that this directive may be omitted when MXCCIT will be set to the default value of 20.
  - CCTH: This directive may be used to define a convergence threshold for CCSD iterations, and comprises a single data line read to the variables TEXT, IAMP using format (A,I). With TEXT set to the character string CCTH, IAMP is an integer parameter used in defining the threshold. At convergence, the magnitude of the CCSD T_1 and T_2 amplitudes will be converged to within an absolute error 10^{-IAMP}. Again this directive may be omitted, when the default value 10^{-10} will be used.

Let us now consider the corresponding calculation with inclusion of the triples (T) component to the correlation energy. A valid data sequence for performing such a calculation is shown below, where we are still performing all the computation in a single job.

```
TITLE
H2CO - TZVP - VALENCE CCSD(T) / CCSD(T)  ENERGY = -114.2714886289
SUPER OFF NOSYM
NOPRINT
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 TZVP
ACTIVE\3 TO 50 END\CORE\1 TO 2\END
RUNTYPE CI\CCSD(T) 48 6 6
CCTH 10
CCIT 30
ENTER
```

Now let us consider performing the CC calculation above in a sequence of jobs, where the first job carries out the SCF, the second the transformation and CCSD(T). First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - TZVP SCF PRIOR TO CCSD(T) CALCULATION
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
BASIS TZVP
ENTER
```

The only obvious point to note is the use of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and CCSD(T) Job

```
RESTART
TITLE
H2CO - TZVP - VALENCE CCSD(T) / CCSD(T)  ENERGY = -114.2714886289
SUPER OFF NOSYM
BYPASS SCF
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 TZVP
ACTIVE\3 TO 50 END\CORE\1 TO 2\END
RUNTYPE CI\CCSD(T) 48 6 6
CCTH 10
CCIT 30
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run will be restored from Section 1 of the Dumpfile, the default section associated with the closed-shell SCF MOs.

The calculation may be further subdivided by splitting Run II above into separate integral transformation and CCSD runs using the RUNTYPE TRANSFORM specification, with subsequent

BYPASS'ing of the transformation in the CC job. Thus:

Run IIa: The Transformation Job

```
RESTART
TITLE
H2CO - TZVP  INTEGRAL TRANSFORMATION
SUPER OFF NOSYM
BYPASS SCF
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 TZVP
RUNTYPE TRANSFORM
ACTIVE\3 TO 50 END\CORE\1 TO 2\END
ENTER
```

Run IIb: The CCSD(T) Job

```
RESTART
TITLE
H2CO - TZVP - VALENCE CCSD(T) / CCSD(T) ENERGY = -114.2714886289
SUPER OFF NOSYM
BYPASS TRANSFORM
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 TZVP
ACTIVE\3 TO 50 END\CORE\1 TO 2\END
RUNTYPE CI\CCSD(T) 48 6 6
CCTH 10
CCIT 30
ENTER
```

27 CI Geometry Optimisation

Energy-only geometry optimisation for direct-CI, full-CI and CCSD wavefunctions may be performed using a variant of the RUNTYPE OPTIMIZE directive. In each case the data line

```
RUNTYPE OPTIMIZE CI
```

requests use of the Fletcher Powell optimiser, with subsequent data used to identify the underlying CI wavefunction to be employed in the energy calculation. We illustrate such usage below for the cases of direct-CI, Full-CI and CCSD calculations. Note that FP optimisation may also be performed for Table-CI wavefunctions (see below), although experience to date suggests that the such optimisations are unlikely to lead to converged geometries given the implicit lack of rigour associated with configuration selection.

27.1 Direct-CI Geometry Optimisation

I. CISD calculations

```
TITLE
H2CO - 3-21G - FP GEOMETRY OPT. / TOTAL CI ENERGY = -113.43777426
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
RUNTYPE OPTIMIZE CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
ENTER
```

Here DIRECT requests a direct-CI wavefunction, with CONF quantifying attributes of the CI calculation.

II. MRDCI calculations

```
TITLE
H2CO - TZVP MRDCI DIRECT-CI / CI TOTAL ENERGY  -114.26950815
SUPER OFF NOSYM
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
H 1 CH 2 HCO 3 180.0
VARIABLES
CO 1.2359174\CH  1.0991666\HCO 122.7868963
END
BASIS TZVP
RUNTYPE OPTIMIZE CI
DIRECT 16 10 42
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
ENTER
```

Here the CONF directive is used to specify a 3 reference function multi-root direct-CI wavefunction. Note that in practice the above calculation would typically be performed in two steps; an initial SCF calculation would be required to identify the MOS to be specified when quantifying the reference functions under control of the CONF data (in the second job).

III. Open Shell calculations

We consider performing the calculation in several steps, where the first two carry out an RHF open shell geometry optimisation, and the third the corresponding CI optimisation.

Runs I and II: The SCF Optimisation

```
TITLE
H2CO - 3-21G - CLOSED SHELL STARTUP
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
ENTER
```

The first step is merely used to generate a suitable set of MOS for initiating the SCF geometry optimisation on the ion below.

```
RESTART NEW
TITLE
H2CO+ - 3-21G - GEOMETRY OPTIMISATION SCF
MULT 2
CHARGE 1
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
RUNTYPE OPTIMIZE
ENTER
```

Run III: The Direct-CI Calculation

```
RESTART NEW
TITLE
H2CO+ - 3-21G - FP GEOMETRY OPTIMISATION FROZEN CORE + DISCARDED VMOS
MULT 2
CHARGE 1
ZMATRIX ANGSTROM
C
O 1 CO
H 1 CH 2 HCO
```

```
H 1 CH 2 HCO 3 180.0
VARIABLES
CO     1.2408256 HESS            .691212
CH     1.0818145 HESS            .751037
HCO 118.2135930 HESS            .674146
END
RUNTYPE OPTIMIZE CI
ACTIVE
3 TO 20 END
CORE
1 2 END
DIRECT 11 6 12
CONF
2 2 2 2 2 1
ENTER
```

In this example we illustrate the freezing and discarding of MOs under control of the ACTIVE and CORE directives. Note that some care must be taken when reducing the orbital space in FP CI optimisations. In open shell calculations, the CI step will derive the orbital set at each point from the second section specified on the ENTER directive i.e., the energy ordered MOs. If this ordering varies from point to point in the FP optimisation, and symmetry is used in minimising the configuration space, it is quite likely that this space will vary during successive points, with disastrous consequences on the optimisation pathway. As a general rule, the user should only consider freezing or discarding orbitals that are well separated from those MOs included in the CI space i.e. inner shell or inner shell complement MOs.

27.2 Table-CI Geometry Optimisation

The examples below are provided to demonstrate the data requirements when performing FP optimisations with the both the conventional and direct-Table-CI modules. In these cases the optimisation converges to give identical results to the first Direct-CI example presented above. Note that this arises from the use of zero threshold specification; the inherent lack of accuracy associated with finite threshold specification is likely to case convergence problems when trying to optimize the geometry, as noted above.

Conventional Table-CI Data

```
TITLE
H2CO - MRDCI CISD FP OPTIMIZATION : CI ENERGY  -113.43777426
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
RUNTYPE OPTIMIZE CI
MRDCI
ADAPT
```

```
TRAN 1
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES 1
CONF
0 1 2 3 4 5  13  17 18
ROOTS 1
THRESH 0 0
CI
DIAG
EXTRAP OFF
ENTER
```

Note that the example above corresponds to a CISD calculation.

Semi-direct Table-CI Data

```
TITLE
H2CO - TABLE-CI 4M/1R FP OPTIMIZATION
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
RUNTYPE OPTIMIZE CI
MRDCI DIRECT
TABLE
SELECT
SYMMETRY 1
SPIN 1
CNTRL 16
SINGLES ALL
CONF
0 1 2 3 4 5  13  17 18
0 1 2 3 4 5  14  17 18
0 1 2 3 4 5  13  17 19
4 13 14 18 19  1 2 3 4 5 17
END
ROOTS 1
THRESH 0 0
CI
ENTER
```

This semi-direct optimisation is now using a reference set of 4 functions; remember again that it is not possible to conduct CISD calculations with this module.

27.3 CCSD Geometry Optimisation

Date requirements follow in straightforward fashion from the examples provided above for direct-CI and Table-CI calculations.

```
TITLE
H2CO - FP GEOMETRY OPT. / CCSD  ENERGY =  -114.08113594505
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 CI
CCSD
ENTER
```

28 Green's Function Calculations I. The OVGF Method

GAMESS–UK incorporates two modules designed to incorporate the effects of electron correlation in the computation of molecular ionisation potentials. The first of these methods, the OVGF or outer-valence green's function method provides a quantitative account of ionisation phenomena when the independent-particle picture of ionisation holds, and as such is most applicable in the treatment of outer-valence orbitals [46].

OVGF calculations are performed under control of the RUNTYPE GF specification, with data input characterising the nature of the calculation introduced by a data line with the character string I.P. in the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as ENTER. Before detailing example data files for performing OVGF calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. OVGF calculations are limited to the treatment of ionization in closed shell molecules.
2. RUNTYPE GF is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and, finally, the Green's function calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.
3. Several files will be generated under RUNTYPE GF processing. These include the following:
   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.

   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals (the Sortfile).

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).

4. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

5. In OVGF calculations the user *must* specify those valence shell molecular orbitals to be included in the valence ionisation computation. These orbitals are defined by use of the I.P. directive. Any core orbitals should be removed from computation using the CORE directive of the integral transformation module.

An OVGF calculation is to performed on the formaldehyde molecule, with estimates required of the ionisation energies of the six valence orbitals, the $3a_1$ to $2b_2$. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO - DZ BASIS - OVGF VALENCE I.E.S
SUPER OFF NOSYM
NOPRINT
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 GF
ACTIVE\3 TO 22 END\CORE\1 TO 2\END
I.P.
SECOND 3 TO 8 END
THIRD 3 TO 8  END
ENTER
```

The following points should be noted:

- The ACTIVE and CORE directives are used to discard the two inner shell functions from the OVGF calculation. Note that it is also possible to truncate the virtual manifold employed, although this has not been done in the present case.
- The data lines associated with the I.P. directive are used to nominate the required valence orbital ionisation energies and the particular level of perturbation theory (SECOND and THIRD) to be employed in this computation.

- The set of molecular orbitals to be used in the transformation and subsequent GF calculation will be restored from the default section associated with closed-shell SCF module, or from the section explicitly nominated on the ENTER directive.

Now let us consider performing the above calculation in two separate jobs, where the first carries out the SCF, the second the transformation and OVGF calculation. First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - 3-21G  SCF PRIOR TO OVGF CALCULATION
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
```

The only obvious point to note is the user of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and OVGF Job

```
RESTART
TITLE
H2CO - 3-21G  OVGF CALCULATION
SUPER OFF NOSYM
BYPASS SCF
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 GF
ACTIVE\3 TO 22 END\CORE\1 TO 2\END
I.P.
SECOND 3 TO 8 END
THIRD 3 TO 8  END
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run will be restored from Section 1 of the Dumpfile, the default section associated with the closed-shell SCF MOs.

29 Green's Function Calculations II. The TDA Method

The second module designed to incorporate the effects of electron correlation in the computation of molecular ionisation potentials employs the so-called two particle-hole Tamm-Dancoff approximation (2ph-TDA) for the one-particle green's function [46, 47]. The TDA method provides at least a qualitative account of ionisation phenomena when the independent-particle picture of ionisation no longer holds, and as such is applicable to the treatment of ionisation processes throughout the whole energy scale. It has been widely employed in the study of phenomena associated with the breakdown of the molecular orbital picture of ionisation e.g., satellite bands.

TDA calculations are performed under control of the RUNTYPE TDA specification, with data input characterising the nature of the calculation introduced by a data line with the character string I.P. in the first data field. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS ore ENTER. Before detailing example data files for performing TDA calculations on the X^1A_1 state of formaldehyde, we mention some general points on conducting such calculations.

1. TDA calculations are limited to the treatment of ionization in closed shell molecules.
2. RUNTYPE TDA is in fact a combination of tasks, requesting integral generation, SCF, integral transformation and, finally, the Green's function calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.
3. Several files will be generated under RUNTYPE GF processing. These include the following:
   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals (the Sortfile).
   - In addition to the direct-access files above, use is made of the more conventional FORTRAN unformatted data sets. These files, allocated with the LFNs FTN002, FTN004, FTN008 and FTN009, can become large.

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).
4. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines requires the data line

```
SUPER OFF NOSYM
```

in the SCF run.

5. In TDA calculations the user *must* specify those valence shell molecular orbitals to be included in the valence ionisation computation. These orbitals are defined by use of the BAND directive. Any core orbitals may be removed from computation using the CORE directive of the integral transformation module.

6. The user should note that the rate-determining step in TDA calculations involves the diagonalisation of matrices whose order is a function of the square of the number of virtual orbitals included in the computation. Although spatial symmetry considerations are used to reduce these large diagonalisation problems, use of the ACTIVE directive in reducing the virtual manifold will often be required when performing these calculations.

A 2ph-TDA calculation is to performed on the formaldehyde molecule, with estimates required of the ionisation energies of the six valence orbitals, the $3a_1$ to $2b_2$. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO - DZ BASIS - TDA VALENCE I.E.S
SUPER OFF NOSYM
NOPRINT
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 TDA
ACTIVE\3 TO 22 END\CORE\1 TO 2\END
I.P.
BAND 3 TO 8 END
ENTER
```

The following points should be noted:

- The ACTIVE and CORE directives are used to discard the two inner shell functions from the TDA calculation. Note that it is also possible to truncate the virtual manifold employed, although this has not been done in the present case.

- The data lines associated with the I.P. directive are used to nominate the required valence orbital ionisation energies through the BAND data line.

- The set of molecular orbitals to be used in the transformation and subsequent TDA calculation are restored from the default section associated with the closed-shell SCF module, section 1, or from the section explicitly nominated on the ENTER directive.

Now let us consider performing the above calculation in two separate jobs, where the first carries out the SCF, the second the transformation and TDA calculation. First the closed shell case:

valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - 3-21G  SCF PRIOR TO TDA CALCULATION
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
```

The only obvious point to note is the user of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and TDA Job

```
RESTART
TITLE
H2CO - 3-21G  TDA CALCULATION
SUPER OFF NOSYM
BYPASS SCF
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 TDA
ACTIVE\3 TO 22 END\CORE\1 TO 2\END
I.P.
BAND 3 TO 8 END
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run will be restored from Section 1 of the Dumpfile, the default section associated with the closed-shell SCF MOs.

30 Linear Response Calculations I. The RPA Method

GAMESS–UK incorporates two modules designed to perform calculations of electronic transition energies and corresponding oscillator strengths, using either the Random Phase Approximation (RPA) method or the Multiconfigurational Linear Response (MCLR) procedure. The RPA calculations may be performed either within the conventional approach where the two–electron integrals are transformed or with a "direct" implementation. The first of these methods, the RPA or Random Phase Approximation is applicable in the treatment of excited states that are dominated by single excitations from a zero order closed shell SCF wavefunction.

Both RPA and MCLR calculations are performed under control of the RUNTYPE RESPONSE specification, with subsequent keyword specification detailing the method to be employed. RPA calculations of excitation energies and oscillator strengths are thus initiated by specifying the data line

```
RUNTYPE RESPONSE RPA
```

in the input file. Data input characterising the details of the calculation is presented immediately after the RUNTYPE data line. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS. Before detailing example data files for performing RPA calculations on the singly excited states of formaldehyde, we mention some general points on conducting such calculations.

1. RPA calculations are limited to the treatment of excited states of closed shell molecules.
2. RUNTYPE RESPONSE is in fact a combination of tasks, requesting integral generation, SCF, integral transformation (in conventional RPA calculations) and, finally, the response calculation itself. While in simple cases it may be feasible to perform all steps in a single calculation, it will often be necessary to break up the calculation into multiple jobs, driving through each of the tasks under control of the appropriate RUNTYPE directive, with use made of the BYPASS directive in the latter stages of the computation. We illustrate this point below.
3. Several files will be generated under conventional RPA processing. These include the following:

   - the Mainfile (ED2) and Dumpfile (ED3).
   - the semi-transformed (ED4) and transformed (ED6) integral files.
   - the Scratch file (ED7).
   - temporary files for sorting both transformed integrals (the Sortfile).

   Any restart jobs will require ED6 being saved, in addition to the Dumpfile (ED3) and Mainfile (ED2).
4. As mentioned above, generation of a valid Mainfile for subsequent use in the integral transformation routines associated with conventional RPA processing requires the data line

   ```
   SUPER OFF NOSYM
   ```

   in the SCF run.

5. In RPA calculations the user *must* specify the number of states of each irreducible representation for which excitation energies and corresponding oscillator are to be computed. Such states are defined by use of the SYMM directive, with a separate data line required for for each irreducible representation. Thus calculation of the excitation energies for the lowest five states of each of the optically allowed symmetries B_{1u}, B_{2u}, B_{3u} of a molecule with D_{2h} symmetry requires the data lines

```
SYMM 2 1 TO 5
SYMM 3 1 TO 5
SYMM 5 1 TO 5
```

   where the first integer nominates the irreducible representation.

An RPA calculation is to performed on the formaldehyde molecule, with estimates required of the excitation energies for the lowest 5 states of each irreducible representation. A valid data sequence for performing such a calculation is shown below.

```
TITLE
H2CO - TZVP + R(SP) BASIS - RPA EXCITATION ENERGIES
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
BASIS
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
RUNTYPE RESPONSE RPA
TDA
SYMM 1 1 TO 5
SYMM 2 1 TO 5
SYMM 3 1 TO 5
SYMM 4 1 TO 5
ANALYSE
ENTER
```

The following points should be noted:

- By presenting the data line, TDA, an additional Tamm-Dancoff (TDA) calculation may be requested for specified irreps and roots, corresponding to a CI in the space of single excitations. Note that the line, TDA ONLY, can be used to suppress the RPA calculation, performing a TDA calculation only.

- The result table printed after successful completion of the iterative TDA/RPA procedure contains the most important one-electron excitations of the corresponding states. If $(\boldsymbol{y}, \boldsymbol{z})$ denotes an RPA eigenvector, then all components of the vector $\boldsymbol{y} - \boldsymbol{z}$ with modulus larger than a certain threshold which may be specified in the THRESH directive (see Part 7) are listed in this table. With the help of the ANALYSE directive, the user may in addition, examine smaller components, without having them listed in the result table. The additional output generated by the ANALYSE directive also contains the dipole integrals, useful for an investigation which mono-excitations contribute to a large oscillator strength, and the weights of the vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ in the RPA eigenvectors $(\boldsymbol{y}, \boldsymbol{z})$.
- Note that files `tda_table.tex` and `rpa_table.tex` are generated, containing the LATEX input for a list of the excited states computed by both methods, comprising excitation energies, oscillator strengths and most important single excitations.
- The set of molecular orbitals to be used in the transformation and subsequent RPA calculation are restored from the default section associated with the ENTER directive, in this case section 1.

Now let us consider performing the above calculation in two separate jobs, where the first carries out the SCF, the second the transformation and RPA calculation. First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The Scf Job

```
TITLE
H2CO - TZVP + R(SP) BASIS - SCF PRIOR TO RPA CALCULATION
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
BASIS
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
ENTER
```

The only obvious point to note is the user of the SUPER directive in requesting full integral list generation required in the subsequent transformation.

Run II: The Transformation and RPA Job

```
RESTART
```

```
TITLE
H2CO - TZVP + R(SP) BASIS - CONVENTIONAL RPA CALCULATION
SUPER OFF NOSYM
BYPASS SCF
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
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
RUNTYPE RESPONSE RPA
TDA
SYMM 1 1 TO 5
SYMM 2 1 TO 5
SYMM 3 1 TO 5
SYMM 4 1 TO 5
ANALYSE
ENTER
```

The following points should be noted:

- The SCF computation is BYPASS'ed
- The SCF vectors from the first run will be restored from Section 1 of the Dumpfile, the default section associated with the closed-shell SCF MOs.

30.1 Direct RPA calculations

For large atomic orbital basis sets, the integral transformation step in conventional calculations can become prohibitive. In this case it is possible to resort to a direct implementation of the RPA procedure which breaks up the four–index transformation into two two–index transformation whenever the RPA matrix acts on a trial vector. The direct RPA module is requested by the RUNTYPE directive

```
RUNTYPE RESPONSE RPA DIRECT
```

In this case the only preparatory run is a closed shell SCF calculation which may be direct or conventional, when the integrals may be generated in either supermatrix or 2E format. Only the Dumpfile of the SCF calculation must be kept. All directives that are available for conventional RPA calculations can also be used for the direct RPA case, with two exceptions (see Part 7). Since direct RPA calculations on larger systems are rather time-consuming, it is desirable to have the possibility to interrupt a calculation and restart it at a subsequent point. This feature, performed under control of the DUMP and RESTORE directives, is also described in Part 7.

Let us consider the corresponding data files to those above for performing a direct RPA calculation on the formaldehyde molecule, with estimates again required of the excitation energies for the lowest 5 states of each irreducible representation. A valid data sequence for performing such a calculation is shown below. We assume that the SCF calculation is also performed in direct fashion.

```
TITLE
H2CO - TZVP + R(SP) BASIS - DIRECT-RPA EXCITATION ENERGIES
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
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
SCFTYPE DIRECT
RUNTYPE RESPONSE RPA DIRECT
TDA
SYMM 1 1 TO 5
SYMM 2 1 TO 5
SYMM 3 1 TO 5
SYMM 4 1 TO 5
ANALYSE
ENTER
```

Now let us consider performing the above calculation in two separate jobs, where the first carries out the direct SCF, the second the direct RPA calculation. First the direct SCF: valid data sequences for performing the calculation are shown below.

Run I: The direct-Scf Job

```
TITLE
H2CO - TZVP + R(SP) BASIS - DIRECT SCF PRIOR TO RPA
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
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
SCFTYPE DIRECT
ENTER
```

The second restart job, requiring only the Dumpfiles from Run I, may be driven as follows.

Run II: The direct-RPA Job

```
RESTART
TITLE
H2CO - TZVP + R(SP) BASIS - DIRECT RPA CALCULATION
BYPASS SCF
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
TZVP O
TZVP C
TZVP H
S O
1.0 0.02
P O
1.0 0.02
END
SCFTYPE DIRECT
RUNTYPE RESPONSE RPA DIRECT
TDA
SYMM 1 1 TO 5
SYMM 2 1 TO 5
SYMM 3 1 TO 5
SYMM 4 1 TO 5
ANALYSE
ENTER
```

31 Linear Response Calculations II. The MCLR Method

The second module for performing calculations of electronic transition energies and corresponding oscillator strengths is the Multiconfigurational Linear Response (MCLR) procedure [48]. MCLR calculations are also performed under control of the RUNTYPE RESPONSE specification, with subsequent keyword specification specifying the method, thus

```
RUNTYPE RESPONSE MCLR
```

Data input characterising the details of the calculation is presented immediately after the RUNTYPE data line. Termination of this data is accomplished by presenting a valid *Class 2* directive, such as VECTORS.

A necessary condition for performing an MCLR calculation is the successful completion of a corresponding multiconfigurational SCF calculation with the MCSCF module, from which the Dumpfile (ED3) and the transformed integral file (ED6) must be saved. The MCLR calculation may be performed in-situ, or as a restart job. In the following we perform a similar MCLR calculation to that described in the RPA section above.

An MCLR calculation is to performed on the formaldehyde molecule, in a DZ + s,p Rydberg basis, with estimates required of the excitation energies for the lowest 5 states of A_1, B_1 and B_2 symmetry. Let us consider performing this calculation in two separate jobs, where the first carries out the SCF, the second the MCSCF and MCLR calculation. First the closed shell case: valid data sequences for performing the calculation are shown below.

Run I: The closed shell Scf Job

```
TITLE
H2CO  CLOSED SHELL DZ+R(SP) /TOTAL ENERGY -113.8308839
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
BASIS
DZ O
DZ C
DZ H
S O
1.0 0.02
P O
1.0 0.02
END
ENTER
```

The second restart job, requiring both the Mainfile and Dumpfile from Run I, may be driven as follows.

```
RESTART
TITLE
H2CO  - DZ+R(SP) MCSCF (10E/9 M.O.)/TOTAL ENERGY = -113.9547201
SUPER OFF NOSYM
NOPRINT
BYPASS
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
DZ O
DZ C
DZ H
S O
1.0 0.02
P O
1.0 0.02
END
RUNTYPE RESPONSE MCLR
ORBITAL
FZC1 FZC1 FZC1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
SECTIONS
SCF 1
MCSCF 8
CANONICAL 10
CIVEC 9
SYMM 1 5
SYMM 2 5
SYMM 3 5
SPLIT 0
MAXIT 50
END
SCFTYPE MCSCF
THRESH 4
MCSCF
ORBITAL
FZC1 FZC1 FZC1 DOC1 DOC3 DOC1 DOC2 DOC3 UOC2 UOC1 UOC3 UOC1
END
ENTER
```

The following points should be noted:

1. The ORBITALS, SECTIONS, and SYMM directives are obligatory, the MCLR data being terminated by the END keyword.

2. The set of active orbitals must be specified by means of the ORBITAL directive. The individual lines of this directive are identical to those presented in the preceding MCSCF calculation (see §10).

3. In order to perform an MCLR calculation, several vectors have to be retrieved from the Dumpfile. The SECTIONS directive specifies in which sections of the Dumpfile the corresponding vectors are stored. Thus the data lines

   ```
   SECTIONS
   SCF        1
   MCSCF      8
   CANONICAL 10
   CIVEC      9
   ```

   instruct the program to read the SCF eigenvectors from section 1, the MCSCF MOs from section 8, the pseudocanonical MCSCF orbitals from section 10 and the MCSCF CI vector from section 9 of the Dumpfile. Note that these section numbers correspond to the default ENTER sections associated with the MCSCF module (see Table 1), and with the default section for the MCSCF natural orbitals (this may be overwritten using the CANONICAL directive).

4. As with RPA calculations, the SYMM directive controls the calculation of excited states. Note that the syntax of this directive is different from the corresponding directive in the RPA module. In particular, it is not possible to calculate an interval [ILOW,IHIGH] of roots with ILOW different from 1.

A more through description of the role of the various MCLR directives is given in Part 7.

32 ZORA relativistic effects

Relativistic effects can be included through the ZORA formalism (Zero Order Regular Approximation) [53]. In its simplest form this changes only the 1-electron terms in the Hamiltonian, so that the relativistic effects once included can be carried over to all formalisms available. This will be demonstrated based on a Hartree-Fock calculation and a multi-reference CI calculation.

The relativistic equivalent of the very first closed shell Hartree-Fock example is:

```
TITLE
H2CO - 3-21G DEFAULT BASIS - CLOSED SHELL SCF
ZORA
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
```

where the directive ZORA enables the relativistic effects. Although it may seem overkill to use a relativistic approach with such light atoms, the total energy of the relativistic calculation is about 0.0531500 au (1.4462 eV) lower than that of the non-relativistic calculation. The relativistic HOMO-LUMO gap is 0.00001615 au (0.0004395 eV) smaller than its non-relativistic counter part.

The relativistic version of the multi-reference CI example is

```
TITLE
H2CO - 3-21G  ZORA CISD 3-REFERENCE CI
SUPER OFF NOSYM
ZORA
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 CI
DIRECT 16 10 10
CONF
2 2 2 2 2 2 2 2 0 0
2 2 2 2 2 2 0 2 2 0
2 2 2 2 2 0 2 2 0 2
NATORB 10 0 PRINT
ENTER
```

Because the ZORA directive includes the relativistic contributions in the 1-electron integrals the CI part of the input need not be changed.

33 Multiple RUNTYPE Calculations

In previous releases of the program, and indeed in all discussion of the available options so far, we have assumed that each invocation of the RUNTYPE directive (i.e. each task to be performed) is carried out in a single run of the program. It is also possible to simplify this *modus operandi* through the issuing of multiple RUNTYPE's within a single run of the program. This is still, however, subject to certain constraints which are summarised below.

As will become clearer from Parts 3 and 4 of the manual, the directive data structure of GAMESS–UK involves two categories of directives, *Class1* and *Class 2*, with the former preceding the latter in the data stream. A schematic representation of the structure of a typical data file is shown below;

```
*  Class 1 Directives
     TITLE
     ......
     ZMATRIX ANGSTROM
     ......
     END
     BASIS

*  Class 2 Directives

     RUNTYPE ....
     SCFTYPE RHF
     ENTER 1
```

Broadly speaking, multiple RUNTYPE's may be issued subject to the constraint that the *Class 1* directives defining the molecular geometry (ZMATRIX, GEOMETRY), the spin multiplicity and charge (MULT, CHARGE) and the basis set (BASIS) appear only once in the job. Thus a schematic representation of a multiple RUNTYPE job is given below;

```
*  Class 1 Directives
     TITLE
     ......
     ZMATRIX ANGSTROM
     ......
     END
     BASIS

*  Class 2 Directives (1st Task)

     RUNTYPE ....
     SCFTYPE RHF
     ENTER 1

*  Class 2 Directives (2nd Task)

     RUNTYPE ....
     SCFTYPE RHF
     VECTORS 1
```

```
          ENTER 2

     *  Class 2 Directives (3rd Task)

          RUNTYPE ....
          SCFTYPE RHF
          VECTORS 1
          ENTER 3
```

We delay a fuller discussion of multiple RUNTYPE's until a later Part of the manual, but note the following points prior to illustrating this usage through a number of examples.

- The molecular geometry used in the n'th RUNTYPE invocation is that determined in the preceding RUNTYPE;
- The ENTER directive terminates data for a given RUNTYPE. It follows that the habit of stacking 'old' data cards after the ENTER directive is now to be avoided!

The following examples illustrate multiple RUNTYPE usage;

1. Optimisation of the geometry of H_2CO and subsequent direct-CI calculation at this optimised geometry;
2. An SCF calculation of H_2CO and subsequent wave function analysis;
3. Using a HESSIAN calculation to obtain the starting hessian, and subsequent location of the (H_2CO to t-HCOH) transition structure;
4. Optimisation of the geometry of H_2CO and subsequent evaluation of the Raman intensities at this optimised geometry.

33.1 Geometry Optimisation and Direct-CI Calculation

In this example we initially perform a geometry optimisation of H_2CO, followed by a direct-CI calculation at this optimised geometry. The first occurrence of ENTER in the data stream terminates the input for the geometry optimisation step, while the RUNTYPE CI data line initiates data input for the direct-CI calculation, with the VECTORS directive pointing to section 1 as the source of molecular orbitals. The CI calculation will be performed at the geometry determined in the first step, being restored from the Dumpfile.

```
TITLE
H2CO - DZ BASIS - CLOSED SHELL SCF - OPTIMISATION + DIRECT-CI
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
ENTER
# CI calculation
RUNTYPE CI
DIRECT 16 8 14
CONF
2 2 2 2 2 2 2 2
NATORB 10 0 PRINT
ENTER
```

33.2 SCF Calculation and Property Evaluation

In this example we perform an initial SCF calculation on H_2CO, then analyse the associated SCF wavefunction through repeated RUNTYPE ANALYSE directives. The first occurrence of ENTER in the data stream terminates the input for the SCF step, while each RUNTYPE ANALYSE data line initiates data input for some different analysis option, with the VECTORS directive pointing to section 1 as the source of molecular orbitals in each case. Note that the ENTER directive should be specified in those cases where a new set of orbitals will be generated as a result of the requested analysis (the localised orbital case below). Use of the default ENTER directive would result in the LMO over-writing the closed shell SCF vectors.

```
TITLE
H2CO - 3-21G  CLOSED SHELL SCF  -  WITH ANALYSIS
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
RUNTYPE ANALYSE
# Molecular Properties
PROPERTY
4 C
4 O
END
VECTORS 1
ENTER
RUNTYPE ANALYSE
# Localised Orbitals
LOCAL
3 TO 8 END
VECTORS 1
ENTER 2
RUNTYPE ANALYSE
# Distributed Multipole Analysis
DMA
VECTORS 1
ENTER
RUNTYPE ANALYSE
```

```
# Graphical Analysis
GRAPHICS
GDEF
TYPE 2D
POINTS 99
TITLE
SQUARE 2D GRID (99*99)
CALC
TYPE ATOM
TITLE
H2CO -ATOM DIFFERENCE
SECTION 150
PLOT
TYPE LINE
TITLE
ATOM DIFFERENCE DENSITY LINEPRINTER PLOT
CALC
TYPE DENS
SECTION 151
TITLE
H2CO - TOTAL DENSITY
PLOT
TYPE LINE
TITLE
DENSITY LINEPRINTER PLOT
CALC
TYPE MO 2
TITLE
H2CO MO 2 AMPLITUDE
SECTION 152
PLOT
TYPE LINE
TITLE
MO 2 LINEPRINTER PLOT
GDEF
TYPE 2D
POINTS 25
TITLE
SQUARE 2D GRID (25*25)
CALC
TYPE POTE
TITLE
H2CO - POTENTIAL
SECTION 153
PLOT
TYPE LINE
TITLE
POTENTIAL LINEPRINTER PLOT
VECTORS 1
ENTER
RUNTYPE ANALYSE
# Mulliken Analysis
MULLIKEN ATOM ORBITAL 3 TO 8 END
VECTORS 1
ENTER
```

33.3 Initial Hessian and Transition State Location

In this example we initially compute a trial hessian to be used in locating the required transition state, followed by the saddle point location itself, where the FCM keyword on the RUNTYPE SADDLE line requests utilisation of the pre-computed hessian. Finally we derive the vibrational frequencies at the optimised geometry, again under control of RUNTYPE HESSIAN. Again note that this final step will be performed at the geometry determined in the preceding SADDLE run.

```
TITLE
HCOH <-> H2CO 1A' TS - 6-31G OPTIMIZE + FREQUENCIES
ZMAT ANGS
C
O 1 CO
H 1 CH1 2 OCH1
H 1 CH5 2 H5CO 3 180.0
VARIABLES
OCH1 56.3\CO 1.27\CH1 1.22\CH5 1.10\H5CO 115.8
END
BASIS 6-31G
# Calculate initial hessian
RUNTYPE HESSIAN
ENTER
# Locate transition State, restoring above hessian
RUNTYPE SADDLE FCM
MINMAX 10 10
XTOL 0.0001
ENTER
# Calculate Vibrational frequencies at TS geometry
RUNTYPE HESSIAN
ENTER
```

33.4 Geometry Optimisation and Raman Intensities

In this example we initially optimise the geometry of H_2CO, followed by a computation of the Raman intensities at the optimised geometry.

```
TITLE
H2CO - 3-21G DEFAULT BASIS -  OPTIMISATION + RAMAN INTENSITIES
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
# Optimise molecular geometry
RUNTYPE OPTIMIZE
XTOL 0.0001
ENTER
```

```
 # Raman intensities at optimised geometry
 RUNTYPE RAMAN
 ENTER
```

33.5 MCSCF Force Constant Calculation

In this example we perform an initial SCF and MCSCF at a previously optimised transition state geometry, then compute the numerical force constants at this geometry.

```
#
# GENERATE INITIAL GUESS ORBITALS FOR MCSCF
TITLE
H2 + CO <-> H2CO 1A’ 3-21G BASIS SCF//MCSCF
ZMAT ANGS
O
C 1 CO
X 2 1.0 1 90.0
X 2 CHH 3 ANG1 1 180.0
X 4 1.0 2 90.0 3 0.0
X 4 1.0 5 ANG2 3 0.0
H 4 XH 6 90.0 2 180.0
H 4 XH 6 90.0 2 0.0
VARIABLES
CO  1.2034717\CHH 1.3040659\XH 0.7415189
ANG1 41.4927811\ANG2 56.6325324
END
ENTER
# NOW PERFORM SINGLE POINT MCSCF CALCULATION, USING SCF MOS
TITLE
H2 + CO <-> H2CO 1A’ 3-21G MCSCF AT OPT. TS GEOMETRY
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
ENTER
# NOW COMPUTE NUMERICAL FORCE CONSTANTS
TITLE
H2 + CO <-> H2CO 1A’ TS 3-21G MCSCF FORCE CONSTANTS
# FREQ. -1825.4 767.8  900.3 1259.4 1720.5 3185.4
RUNTYPE FORCE
SCFTYPE MCSCF
MCSCF
ORBITAL
COR1 COR1 COR1 DOC1 DOC1 DOC1 DOC2 DOC1 UOC2 UOC1 UOC1 UOC1
END
PRINT ORBITALS VIRTUALS NATORB
CANONICAL 12 FOCK DENSITY FOCK
ENTER
```

References

[1] M. Dupuis and H.F. King, Int. J. Quant. 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.

[2] W.J. Hehre, W.A. Lathan, R. Ditchfield, M.D. Newton and J.A. Pople, GAUSSIAN-70, Prog. No. 236, QCPE, Indiana University (1973); J.S. Binkley, R.A. Whiteside, R. Krishnan, R.Seeger, D.J. DeFrees, H.B. Schlegel, S.Topiol, L.R. Kahn and J.A. Pople, GAUSSIAN80, QCPE **13** (1980) 406.

[3] See for example, Carnegie-Mellon Quantum Chemistry Archive, 2nd edn., R.A. Whiteside, M.J. Frisch, J.S. Binkley, D.J. DeFrees, H.B. Schlegel, K. Raghavachari and J.A. Pople (1981), which contains approximately 3500 SCF structures.

[4] J.S. Binkley, J.A. Pople and W.J. Hehre, J. Am. Chem. Soc., **102** (1980) 939, doi:10.1021/ja00523a008; M.S. Gordon, J.S. Binkley, J.A. Pople, W.J. Pietro and W.J. Hehre, J. Am. Chem. Soc., **104** (1982) 2797, doi:10.1021/ja00374a017; M.J. Frisch, J.A. Pople and J.S. Binkley, J. Chem. Phys. **80** (1984) 3265, doi:10.1063/1.447079, and references cited therein; K.D. Dobbs and W.J. Hehre, J. Comp. Chem. **7** (1986) 359–378, doi:10.1002/jcc.540070313; *ibid* **8** (1987) 861–879, doi:10.1002/jcc.540080614, 880–893, doi:10.1002/jcc.540080615.

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

[6] P.J. Hay and W.R. Wadt, J. Chem. Phys. **82** (1985) 270 doi:10.1063/1.448799, 284 doi:10.1063/1.448800, 299 doi:10.1063/1.448975.

[7] Ph. Durand and J.-C. Berthelat, Theoret. Chim. Acta, **38** (1975) 283, doi:10.1007/BF00963468.

[8] W.J. Stevens, H. Basch and M. Krauss, J. Chem. Phys. **81** (1984) 6026, doi:10.1063/1.447604.

[9] W.J. Stevens, P.G. Jasien, M. Krauss, and H. Basch, Can. J. Chem. **70** (1992) 612, doi:10.1139/v92-085.

[10] T.R. Cundari and W.J. Stevens, J. Chem. Phys. **98** (1993) 5555, doi:10.1063/1.464902.

[11] (H) - T.H.Dunning, Jr. and P.J. Hay, Methods of Electronic Structure Theory, Vol. 3, H. F. Schaefer III, Ed. Plenum Press (1977); (Li-Ne, Na-Ar) - L. F. Pacios and P. A. Christiansen, J. Chem. Phys. **82** (1985) 2664, doi:10.1063/1.448263; (K-Ca, Sc-Zn, Ga-Kr) - M. M. Hurley et al. J. Chem. Phys. **84** (1986) 6840, doi:10.1063/1.450689; (Rb-Sr,Y-Cd, In) - L. A. LaJohn et al. J. Chem. Phys., **87** (1987) 2812, doi:10.1063/1.453069; (Xe) - M. M. Hurley et al., J. Chem. Phys. **84** (1986) 6840, doi:10.1063/1.450689; (Cs, La, Hf-Hg, Tl-Rn) - R.B. Ross, W.C. Ermler, P.A. Christiansen et al. J. Chem. Phys. **93** (1990)

6654, doi:10.1063/1.458934, erratum: doi:10.1063/1.468517; (Ba, Ce-Lu) - R.B. Ross, W.C. Ermler, S. Das, To be published; (Fr-Ra, Ac-Pu) - W.C. Ermler, R.B. Ross, P.A. Christiansen, Int. J. Quant. Chem **40** (1991) 829.

[12] (Sc - Co, Cu-Zn) - M.M. Hurley et al., J. Chem. Phys., **84** (1986) 6840, doi:10.1063/1.450689; (Ni, Y-Cd) - L.A. LaJohn et al., J. Chem. Phys., **87** (1987) 2812, doi:10.1063/1.453069; (La, Hf-Hg, Tl-Tn) - R.B. Ross, W.C. Ermler, P.A. Christiansen et al. J. Chem. Phys. **93** (1990) 6654, doi:10.1063/1.458934, erratum: doi:10.1063/1.468517.

[13] (Li - Be, Na) - P. Fuentealba, H. Preuss, H. Stoll, L. v. Szentpaly, Chem. Phys. Lett. **89** (1982) 418, doi:10.1016/0009-2614(82)80012-2; (B-Ne) - A. Bergner, M. Dolg, W. Kuechle, H. Stoll, H. Preuss, Mol. Phys. **80** (1993) 1431, doi:10.1080/00268979400100024; (Mg) - P. Fuentealba, L. v. Szentpaly, H. Preuss, H. Stoll, J. Phys. B **18** 1287 (1985); (Al) - G. Igel-Mann, H. Stoll, H. Preuss, Mol. Phys. **65** (1988) 1321, doi:10.1080/00268978800101811; (Hg-Rn) - W. Küchle, M. Dolg, H. Stoll, H. Preuss, Mol. Phys. **74** 1245 (1991), doi:10.1080/00268979100102941; (Ac-Lr) W. Küchle, to be published

[14] (K) - A. Bergner, M. Dolg, W. Küchle, H. Stoll, H. Preuss, Mol. Phys. **80** (1993) 1431, doi:10.1080/00268979400100024; (Ca) - M. Kaupp, P. v. R. Schleyer, H. Stoll, H. Preuss, J. Chem. Phys. **94** (1991) 1360, doi:10.1063/1.459993; (Rf - Db) - M. Dolg, H. Stoll, H. Preuss, R.M. Pitzer, J. Phys. Chem. **97** (1993) 5852, doi:10.1021/j100124a012.

[15] P.J. Knowles and H.J. Werner, Chem. Phys. Lett. **115** (1985) 259, doi:10.1016/0009-2614(85)80025-7.

[16] 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.

[17] A.D. Becke, Physical Reviews **A38** (1988) 3098.

[18] C. Lee, W. Yang, and R.G. Parr, Physical Reviews **B37** (1988) 785-789.

[19] M. E.Mura, and P.J. Knowles, J. Chem. Phys. **104** (1996) 9848-9858, doi:10.1063/1.471749.

[20] C.W. Murray, N.C. Handy, and G.J. Laming, Mol. Phys. **78** (1993) 997, doi10.1080/00268979300100651.

[21] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257** (1996) 213–223, doi:10.1016/0009-2614(96)00600-8.

[22] A. D. Becke, J. Chem. Phys. **98** (1993) 5648, doi:10.1063/1.464913.

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

[24] J.P. Perdew, Physical Review **B33** (1986) 8822-8824.

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

[26] F.A. Hamprecht, A.J. Cohen, D.J. Tozer, and N.C. Handy, J. Chem. Phys. **109** (1998) 6264-6271, doi:10.1063/1.477267.

[27] P.J. Wilson, T.J. Bradley, and D.J. Tozer, J. Chem. Phys. **115** (2001) 9233-9242, doi:10.1063/1.1412605.

[28] 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.

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

[30] 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.

[31] J.M. Foster and S.F. Boys, Rev. Mod. Phys. **32** (1960) 300, doi:10.1103/RevModPhys.32.300; J. Pipek and P.G. Mezey, J. Chem. Phys. **90** (1989) 4916, doi:10.1063/1.456588.

[32] A.J. Stone, Chem. Phys. Lett. **83** (1983) 233, doi:10.1016/0009-2614(81)85452-8.

[33] T. Amos and L.C. Snyder, J. Chem. Phys. **41** (1964) 1773, doi:10.1063/1.1726157.

[34] 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.

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

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

[37] 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.

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

[39] R.J. Buenker in 'Proc. of the Workshop on Quantum Chemistry and Molecular Physics', Wollongong, Australia (1980); R.J. Buenker in 'Studies in Physical and Theoretical Chemistry', **21** (1982) 17.

[40] S. Zarrabian and R.J. Harrison, Chem. Phys. Lett. **81** (1989) 393, doi:10.1016/0009-2614(89)87358-0.

[41] T.J. Lee, J.E. Rice and A.P. Rendell, The TITAN Set of Electronic Structure Programs, 1991,

[42] T.J. Lee and J.E. Rice, Chem. Phys. Lett. **150** (1988) 406, doi:10.1016/0009-2614(88)80427-5.

[43] G.E. Scuseria, A.C. Scheiner, T.J. Lee, J.E. Rice and H.F. Schaefer, J. Chem. Phys. **86** (1987) 2881, doi:10.1063/1.452039.

[44] T.J. Lee, A.P. Rendell and P.R. Taylor, J. Phys. Chem.**94** (1990) 5463, doi:10.1021/j100377a008.

[45] A.P. Rendell, T.J. Lee and A. Komornicki, Chem. Phys. Lett, **178** (1991) 462, doi:10.1016/0009-2614(91)87003-T.

[46] L.S. Cederbaum and W. Domcke, Adv. Chem. Phys. **36** (1977) 205.

[47] J. Schirmer and L.S. Cederbaum, J. Phys. **B11** (1978) 1889, doi:10.1088/0022-3700/11/11/006.

[48] C. Fuchs, V. Bonačić-Koutecký and J. Koutecký, J. Chem. Phys. **98** (1993) 3121, doi:10.1063/1.464086.

[49] H.J.J. van Dam, J.H. van Lenthe, and P. Pulay, Mol. Phys. **93** (1998) 431, doi:10.1080/002689798169122.

[50] H.-J. Werner, Mol. Phys. **89** (1996) 645, doi:10.1080/002689796173967.

[51] K. Andersson, P.-Å. Malmqvist, and B.O. Roos, J. Chem. Phys. **96** (1992) 1218, doi:10.1063/1.462209.

[52] K. Wolinski, H.L. Sellers, and P. Pulay, Chem. Phys. Lett. **140** (1987) 225, doi:10.1016/0009-2614(87)80448-7.

[53] S. Faas, J.G. Snijders, J.H. van Lenthe, E. van Lenthe, and E.J. Baerends, Chem. Phys. Lett. **246** (1995) 632, doi:10.1016/0009-2614(95)01156-0.