

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

Generalised Atomic and Molecular Electronic Structure System

G A M E S S - U K

USER'S GUIDE and REFERENCE MANUAL

Version 8.0 June 2008

PART 3. DATA INPUT - Pre-directives and CLASS 1 Directives

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

Copyright (c) 1993-2008 Computing for Science Ltd.

This document may be freely reproduced provided that it is reproduced
unaltered and in its entirety.

Contents

1 Introduction	2
2 The pre-directives	2
2.1 The MEMORY Pre-directive	2
2.2 The TIME Pre-directive	3
2.3 The FILE Pre-directive	3
3 Class I directives	4
4 The DUMPFILe Directive	4
5 The RESTART Directive	5

6	File specification: file identifiers, MAXBLOCK and MINBLOCK	7
6.1	file identifiers	7
6.2	MINBLOCK and MAXBLOCK	8
6.3	In-core SCF Calculations	9
7	Directives Controlling Printed Output	10
7.1	NOPRINT	10
7.2	IPRINT	10
7.3	FORMAT	11
8	Directives Controlling Integral Generation	12
8.1	INTEGRAL	12
8.2	ACCURACY	12
8.3	SUPER	12
8.4	BYPASS	16
8.5	ADAPT	18
9	Directives Defining the Molecular System	18
9.1	TITLE	18
9.2	MULTIPLICITY	19
9.3	CHARGE	19
9.4	ELECTRONS	20
9.5	ZMATRIX and GEOMETRY	20
10	The ZMATRIX Directive	20
10.1	Z-Matrix Definition Lines	21
10.2	VARIABLES and CONSTANTS Specification	25
10.2.1	VARIABLES Specification	26
10.2.2	CONSTANTS Specification	27
10.3	Specification and Role of DUMMY Centres	29
10.4	Specification of Ghost Centres	31
10.5	Specification of Point Charges	32
10.6	Examples of Z-matrix input	34
10.7	Controlling the Point Group Symmetry	38

11 The GEOMETRY Directive	41
11.1 Cartesian-based Optimisation	43
11.2 Z-Matrix Construction via GEOMETRY	44
11.3 Mixed Z-matrix and Cartesian Input	46
12 The WEIGHTS Directive	47
13 Basis Set Specification	50
13.1 The HARMONIC Directive	50
13.2 The DEPENDENCY Directive	51
13.3 The BASIS Directive	52
13.3.1 Minimal Basis Sets	52
13.3.2 Split-valence Basis Sets	53
13.3.3 Double-Zeta Basis Sets	53
13.3.4 Triple-zeta and Extended Basis Sets	54
13.3.5 ECP Basis Sets	54
13.3.6 Polarisation Basis Sets	55
13.3.7 Correlation consistent Basis Sets	56
13.3.8 DFT Basis Sets	56
13.4 Global Specification	57
13.5 Hybrid Specification	60
13.6 Scaling Basis Functions	61
13.7 General Basis Specification	62
13.8 NWChem and G94 Basis Specification	66
13.9 Shell-structured Basis Specification	69
13.10 PSEUDO	70
13.10.1 Non-local ECPs	71
13.10.2 Local ECPs	74
13.10.3 Compatibility with Version 6.2 of GAMESS-UK	76
14 Relativistic Calculations	76
14.1 ZORA	77
14.2 SPIN	78
14.3 MOLECULAR	78

14.4 SCALE	78
14.5 Directives controlling the calculation of the Coulomb matrix	79
14.5.1 GET	79
14.5.2 COULOMB	79
14.5.3 EXCHANGE	80
14.5.4 NITERATION	80
14.6 Directives controlling the internal basis	80
14.6.1 NOCOPY	80
14.6.2 INTERNAL	80
14.6.3 LIMIT	81
14.6.4 CONTRACTED	81
14.7 Miscellaneous	81
14.7.1 PRINT	81
14.7.2 PRI2	81
14.7.3 C	81
14.7.4 GAUGE	82
14.7.5 OFF	82

1 Introduction

In this and the following chapters we provide a fairly detailed description of the role and format of the directives within the GAMESS-UK data specification responsible for defining the molecular system under consideration and the required computation. In addition to defining the system geometry (through the ZMATRIX or GEOMETRY directive) and basis set (the BASIS directive), the *Class 1* directives provide control over file usage, printed output (IPRINT etc.) and the integral generation routines (SUPER, ACCURACY etc.).

Before the discussion of the *Class 1* directives, we firstly describe the "pre-directives" available within GAMESS-UK that should appear before any *Class 1* directives. The pre-directives may be required to set and/or modify the environment, defining for example memory requirements or job cpu time, or assisting in file assignments.

The following points should be noted:

- With certain exceptions noted below, the *Class 1* directives may be presented in any order.
- All *Class 1* directives must appear in the input stream before any *Class 2* directive (see Part 4).
- Any pre-directives must appear before any *Class 1* directives

2 The pre-directives

The program is capable of processing a set of 'pre-directives', each such directive extending over one data line, and appearing as input before the program specific data. These pre-directives allow the user to define or modify, through data input, certain characteristics of the job environment e.g., time allocations, routing of output, file allocations, memory requirements etc.

2.1 The MEMORY Pre-directive

This pre-directive provides a mechanism for specifying the dynamic core to be associated with the present run of the code. Memory requirements are in general a function of the RUNTYPE requested in the data input: allocation can be modified through MEMORY data specification. The pre-directive consists of a single data line, the first data field being set to the character string memory or core, the second to an integer defining the number of words of memory. Presenting the data line

```
memory 8000000
```

will yield an allocation of 64 MBytes. The default allocation of 4,000,000 words will prove adequate for most runs involving both SCF and CI wavefunctions.

2.2 The TIME Pre-directive

GAMESS–UK monitors the CPU time available at intervals, and if it is found that insufficient time remains to usefully continue, will send restart control information to the Dumpfile, and terminate execution. The time pre-directive is used to specify the time limit for the job in CPU minutes, e.g.,

```
time 120
```

will allocate 2 hours of CPU time to the job. In the absence of the time pre-directive, a default allocation of 600 minutes will be in effect.

2.3 The FILE Pre-directive

The FILE predirective may be used to override the default mode, in which each of the direct access data sets used by GAMESS–UK will be deleted on termination of the job. Such data sets are allocated to the program using logical file names (LFNs) in the range ED0, ED1, ..., ED19 and MT0, MT1, ..., MT19, and retaining such a data set on job completion requires the user to associate the LFN with a specified filename through the FILE directive. This is best shown through an example. Assume that we wish to retain the Dumpfile, which in default is routed to ED3 (and is crucial to the program, controlling all restart activities). This might involve the following pre-directive specification:

```
FILE ED3 DUMPFIL
```

in the startup job, and *all subsequent* jobs that make use of this file. In similar fashion, retaining both the Dumpfile and two-electron integral Mainfile between jobs might involve the following specifications:

```
FILE ED2 MAINFIL  
FILE ED3 DUMPFIL
```

There is one standard use of the file directive that the user need employ when accessing the various GAMESS–UK library data sets that reside in the directory:

```
GAMESS-UK/libs.
```

When using the library of ECPs (with the specification 'pseudo nonlocal'), the file pre-directive

```
FILE EDO ECPLIB
```

must be specified when using RUN_GAMESS.

In addition to the direct access data sets described above GAMESS–UK will, in certain phases of computation e.g., Table-Cl, make use of the more conventional FORTRAN unformatted

sequential data sets. In the majority of cases this usage is restricted to scratch activities, and the user need not be concerned with, for example, FORTRAN unit specification. In some instances, however, it may be necessary to 'keep' the files in question between jobs. This activity is again handled through the FILE pre-directives.

3 Class I directives

The following sections describe the CLASS 1 directives available within the programme.

4 The DUMPFILe Directive

The DUMPFILe directive may be used to redefine Dumpfile output from the default setting and consists of a single data line read to variables TEXT, LFNAME, IBLOCK using format (2A,I)

- TEXT should be set to the character string DUMPFILe
- LFNAME should be set to the LFN used to assign the data set for output of the Dumpfile. The LFN can be one of the files ED0 – ED19, with the exception of the Scratchfile, ED7.
- IBLOCK is the integer used to specify the starting block at which Dumpfile output will commence.

Example

Presenting the data line

```
DUMPFILe ED4 100
```

would route Dumpfile output to ED4, commencing at block 100. The following points should be noted:

- The DUMPFILe directive may be omitted, when Dumpfile information will be routed to block 1 of the data set assigned using the LFN ED3. This default thus corresponds to presenting the data line

```
DUMPFILe ED3 1
```

- If used, the DUMPFILe directive *MUST* be the first of the *Class 1* directives presented in the data stream, immediately following any pre-directives (see Parts 12-16 of the manual).

5 The RESTART Directive

Execution and control of the program is classified into various tasks, as nominated by the RUNTYPE directive (see Part 4, section 2). Once a calculation has been initiated, and the files saved, all subsequent restart jobs *must* specify the RESTART directive. The directive has various roles;

1. to indicate that the Dumpfile for the case in hand exists i.e., that a previous calculation is to be continued;
2. to restart the computation associated with a specific task;
3. to abort the computation currently in progress, prior to the definition of new activity under control of the RUNTYPE directive;
4. when instigating a new task under RUNTYPE control, the RESTART directive may be used to request that the input geometry be that currently resident on the Dumpfile (ie from the previous completed task) rather than that specified in the card input stream.

In general the format of the directive is dependent on the particular role above. The directive consists of a single data line read to the variables TEXT, TYPE, TREG using format (3A).

- TEXT should be set to the character string RESTART.
- There are two general specifications for TYPE;
 1. TYPE may be set to a character string identifying the task (RUNTYPE) to be restarted. The following strings are recognised when restarting some particular computation which had terminated successfully in a prior job:
 - RESTART OPTIMIZE - continue a geometry optimization as initiated by a RUNTYPE OPTIMIZE directive.
 - RESTART OPTXYZ - continue a geometry optimization as initiated by a RUNTYPE OPTXYZ directive.
 - RESTART SADDLE - continue a saddle point optimisation, as initiated by a RUNTYPE SADDLE directive.
 - RESTART SCF - continue an SCF, GVB, Møller Plesset, CASSCF or MCSCF calculation, as initiated by a RUNTYPE SCF directive.
 - RESTART FORCE - continue a numerical force constant calculation as initiated by a RUNTYPE FORCE directive.
 - RESTART HESSIAN - continue an analytical force constant calculation as initiated by a RUNTYPE HESSIAN directive.
 - RESTART POLARISABILITY - continue a polarisability calculation as initiated by a RUNTYPE POLARISABILITY directive.
 - RESTART MAGNET - continue a magnetisability calculation as initiated by a RUNTYPE MAGNET directive.

- RESTART HYPER - continue a hyperpolarisability calculation as initiated by a RUNTYPE HYPER directive.
 - RESTART INFRARED - continue an Infra-red intensity calculation as initiated by a RUNTYPE INFRARED directive.
 - RESTART RAMAN - continue a Raman intensity calculation as initiated by a RUNTYPE RAMAN directive.
 - RESTART CI - continue a CI calculation as initiated by a RUNTYPE CI directive.
 - RESTART GF or RESTART TDA - continue a Green's function calculation as initiated by the corresponding RUNTYPE directive.
 - RESTART ANALYSE - while the analysis modules within GAMESS-UK are not in general restartable, it is possible to restart the computation of a grid of electrostatic potentials, as initiated by the RUNTYPE ANALYSE, GRAPHICS and GTYPE directives (see Part 8).
2. TYPE may be set to the character string NEW or may be left unspecified. Let us assume that the computation associated with a given RUNTYPE has been completed, and the user wishes to begin another phase of the computation. This may be achieved by specifying the data line

```
RESTART NEW
```

where the NEW keyword indicates that the molecular geometry for the new calculation is to be taken from the data input stream. An alternative procedure is available on presenting simply the line

```
RESTART
```

in which case the molecular geometry used will be that currently resident on the Dumpfile.

- TREG may be set to the character string REGEN, in which case all integral files required to successfully continue the computation in progress will be recomputed. This option is provided to allow for instances when files have been inadvertently lost or corrupted. Thus presenting a data line of the form

```
RESTART OPTIMIZE REGEN
```

to restart a geometry optimisation would, if the startup job terminated in the course of the SCF iterations, lead to the two-electron integral list being recomputed.

The following points on use of the RESTART directive should be noted:

- Corruption or loss of the Dumpfile is viewed at present as a terminal error condition.
- In all cases except RESTART and RESTART NEW the character string specified on both the RESTART and RUNTYPE directives must agree.

Table 1: Specification of the file identifiers and the default associated logical file names

File	File Identifier	Default LFN
Mainfile	MFILE or MAINFILE	ED2
Symmetry Adapted Mainfile	AFILE	ED1
Secondary Mainfile	SFILE or SECOND	ED4
Direct-CI file	CIFILE	ED5
Transformed Integral File	FFILE or TFILE	ED6
Loop Formulae Tape	LOOP	ED9
Reordered Formulae Tape	RLOOP	ED10
Density Matrix File	DMFILE or TWOPDM	ED11
Hamiltonian File	HFILE	ED12

- **Note :-** At present the restriction is imposed in **RESTART** mode under control of **RUNTYPE OPTIMIZE** or **RUNTYPE SADDLE**, that use is made of the **RESTART NEW** specification ie it is not possible to read the optimised geometry for, say, one state of the molecule from the Dumpfile in instigating a second optimisation on, say, some other state.

6 File specification: file identifiers, MAXBLOCK and MINBLOCK

In default, each of the direct access files associated with the storage of intermediate quantities, such as two-electron integrals, transformed integrals, symbolic formula etc. is associated with a particular logical file name (LFN), in the range ED0–ED19. With the exception of the Dumpfile, output to each file will commence at block 1 and continue in uninterrupted fashion until the entire list is complete. The file identifiers, and the MAXBLOCK and MINBLOCK directives may be used to modify this default action by,

- changing the default LFN allocation;
- fragmenting the output over multiple files;
- commencing output to some specified position on a given file.

6.1 file identifiers

The file directives may be used to change the LFN associated with a given output file, and to fragment the output to multiple files. The directive consists of a single data line, the first data field of which comprises the file identifier, a character string identifying the particular output file whose attributes are to be changed. Valid file identifiers are shown in Table 1.

Subsequent data field(s) are read in free A-format and are used to specify output reassignment by nominating the direct access files involved. Thus simply reassigning output to a different LFN is accomplished by a data line of the form

```
MFILE ED4
```

whereby Mainfile output is routed to the data set assigned with the LFN ED4. Fragmentation of file output may be achieved by multiple LFN specification, so that splitting the Mainfile over three data sets, assigned with the LFNs ED0, ED1 and ED2 would require the specification

```
MFILE ED0 ED1 ED2
```

in conjunction with appropriate MINBLOCK and MAXBLOCK directives defining the starting and terminal blocks of each data set (see below).

6.2 MINBLOCK and MAXBLOCK

These directives may be used to reassign the starting and terminating blocks available for output on a given file, and are provided for use with the FILE directive(s) in fragmenting output. The MINBLOCK directive consists of a single data line read to the variables TEXT, LFNAME, MINBLOK using format (2A,I),

- TEXT should be set to the character string MINBLOCK;
- LFNAME should be set to the LFN of the file (ED0–ED19), output to which is to be modified;
- MINBLOK is an integer specifying the starting block at which output is to commence.

Thus the data line

```
MINBLOCK ED2 1000
```

would result in output to ED2 commencing at block 1000. The MAXBLOCK directive is of similar syntax, being read to the variables TEXT, LFNAME, MAXBLOK using format (2A,I).

- TEXT should be set to the character string MAXBLOCK;
- LFNAME should be set to the LFN of the file (ED0–ED19) output to which is to be modified;
- MAXBLOK is an integer specifying the block at which output is to terminate.

Thus the data line

```
MAXBLOCK ED2 10000
```

would result in output to ED2 terminating at block 10000.

Example

```

MFILE ED0 ED1 ED2
MAXBLOCK ED0 9000
MAXBLOCK ED1 9000
MINBLOCK ED2 2000

```

The above data sequence would result in Mainfile output being routed initially to ED0 (blocks 1-9000), then to ED1 (blocks 1-9000) and finally to ED2, commencing at block 2000.

6.3 In-core SCF Calculations

In default, the two-electron integral list is written to the data set on disk nominated by the MFILE directive (ED2 in default), with the format controlled by SUPER specification. It is possible to override this default, and avoid disk storage and the associated I/O overheads, by mapping the integral list directly to memory. This *modus operandi* is requested through a modification to the MFILE directive, thus

```
MFILE MEMORY
```

The following points should be noted;

1. The program employs a rather conservative algorithm in deciding whether sufficient memory is available to house the integral list prior to integral evaluation. If the host machine does not have the available memory, the program aborts. The estimated memory may not be sufficient, in which case one may override the default allocation in either blocks or in MBytes or GBytes. For parallel runs, this allocation is 'per processor'. **Example**

```
MFILE MEMORY 500MB
```

One may specify an overflow unit, in case the memory allocated is not sufficient. This may be any valid file with the exception of ED2 (in memory), ED3 or ED7 (for scf). Of course, one should be sure that this file is not in use in another stage of the calculation for another purpose. **Example**

```
MFILE MEMORY 500MB ED12
```

Now, if the memory (in this case 500 MByte) is exhausted the program switches to conventional integral storage, routing the remainder of the integral list to ED12.

2. This memory is independent of that specified under control of the CORE or MEMORY pre-directive (see Parts 12-16 of the manual), and the latter specification should be made independently of the present usage.
3. At present the memory resident integral list is not written to disk on job termination, so that it is not possible to RESTART in-core SCF computations.

4. This routing of integrals to memory is available for all SCFTYPEs and integral formats, regardless of the RUNTYPE in effect, and may thus be used, for example, in geometry optimisations.
5. In-core integral storage is targeted to those machines with large amounts of central memory (ie 128 MBytes and above), and is at present only available on those machines with a UNIX operating system. It is useful on large parallel machines with a lot of memory per processor, where the aggregated memory may amount to hundreds of Gigabytes.

7 Directives Controlling Printed Output

7.1 NOPRINT

A single data line containing the keyword NOPRINT in the first data field will act to suppress all intermediate output. Greater selectivity in minimising the quantity of output is achieved by appending any combination of the following parameters to the NOPRINT command:

HISTORY	Suppress printing of the history file during optimisation
VECTORS	Suppress printing of the eigenvectors.
BASIS	Suppress printing of the basis set
ADAPT	Suppress adaptation output
DISTANCE	Prevent the distance matrix from being printed
HESS	Suppress printing of the force constant matrix during optimisation.
ANAL	Suppress printing of the wavefunction analysis

7.2 IPRINT

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

S	requests printing of the overlap matrix in the a.o. basis
T	requests printing of the kinetic energy matrix in the a.o. basis
T+V	requests printing of the 1-e hamiltonian matrix in the a.o. basis
X	requests printing of the x-dipole moments integrals
Y	requests printing of the y-dipole moments integrals
Z	requests printing of the z-dipole moments integrals
SCF	Full SCF output during optimisation
OPTIMISE	Increased diagnostics during optimisation
VECTORS	Only print vectors upon completion of optimisation.
MULLIK	Increased output from Mulliken analysis
LOWDIN	Increased output from Lowdin wavefunction analysis
GUESS	Requests printing of trial vectors
FOCK	requests printing of the Fock matrix at SCF convergence
ADAPT	printing of the symmetry adapted basis functions
DIIS	monitoring of the solution of the DIIS equations
SYMM	increased symmetry diagnostics
DIST	requests printing of the matrix of internuclear distances
DSCF	additional printing in the direct-SCF module
INERTIA	print moments of inertia etc. at optimised geometry
AOPR	additional printing of 1-e property integrals in a.o. basis
MOPR	additional printing of 1-e property integrals in m.o. basis

7.3 FORMAT

The FORMAT directive may be used to modify the format for printing eigenvectors and eigenvalues. The directive comprises a single data line read to the variables TEXT, FORM using format (2A)

- TEXT should be set to the character string FORMAT
- FORM may have the setting HIGH or LOW. In the former case the vectors will be printed to 7 significant figures, with all orbitals (occupied and virtual) output. In the latter case only the occupied and 5 lowest virtuals will be printed, and then to only 4 significant figures.

The default setting corresponds to presenting the data line

```
FORMAT LOW
```

8 Directives Controlling Integral Generation

8.1 INTEGRAL

This directive may be used to specify the mode for 2-electron integral and derivative integral evaluation. The program incorporates the rotated axes and Gauss-Rys algorithms for 2-electron integral evaluation, and the Schlegel algorithm [1] for s,p derivative integrals. In default the program will use the significantly faster s,p routines whenever possible, handling only those integrals involving d- and f- functions by the Gauss-Rys method. This default mode may be overridden by the INTEGRAL directive, which has the form:

```
INTEGRAL HIGH
```

requesting use of the Gauss-Rys routines for all integral evaluation.

8.2 ACCURACY

This directive may be used to control the accuracy of the integral routines, and comprises a single data line read to the variables TEXT, ITOL, ILOAD using format (A,2I).

- TEXT should be set to the character string ACCURACY.
- ITOL is an integer used in computing a threshold according to which integrals may be neglected. Specifically, integrals involving products of primitives whose pre-exponential factor is less than 10^{-ITOL} are skipped. Default value is 10^{-20} .
- ILOAD is an integer used in computing the threshold for loading two-electron integrals to the Mainfile. Specifically, integrals less than 10^{-ILOAD} are not loaded to the integral file. Default value is 10^{-9} .

Experience suggests that the following setting,

```
ACCURACY 20 7
```

can lead to a significant reduction in SCF cycle time without undue loss of accuracy.

8.3 SUPER

This directive may be used to control the format and structure of the two-electron integral file (Mainfile), and override the default format imposed by the program. This default is in general a function of both the SCFTYPE and RUNTYPE, and while the program will, in default, impose the appropriate format associated with the requested TYPES, the user need be aware of these when considering, for example, trying to re-use, rather than regenerate, the Mainfile produced under control of a given RUNTYPE/SCFTYPE combination in some subsequent calculation (under control of the BYPASS directive). First, let us consider the three possible Mainfile formats available, and how to request each type; these are,

1. P-supermatrix format (2J-K), for use in closed-shell SCF calculations, where it is the default format;
2. J and K supermatrix format (in fact 2J-K and K), the default in open-shell RHF, UHF and GVB calculations and usable in closed-shell SCF calculations;
3. conventional 2-electron integral list format, which may be used in all calculations, and as such is the most general.

The SUPER directive allows the user to override the format imposed by the program, and comprises a single data line with the character string SUPER in the first data field. The integral format required is controlled by the second keyword, with three possible settings;

- **SUPER ON:** This will request supermatrix format (the default setting), with the format chosen (P or J + K) dictated by the specified SCFTYPE i.e., P-matrix for closed-shell SCF calculations, J + K matrices for GVB and open shell calculations;
- **SUPER OFF:** Produces conventional two-electron integral format.
- **SUPER FORCE:** May be used to force J and K matrices to be used, even in closed-shell calculations. This would typically be introduced in a closed-shell SCF calculation, with the aim of re-using the Mainfile in subsequent open shell calculations (under control of the BYPASS directive).

Having nominated the integral format, we need next consider whether the Mainfile is to comprise just the 'skeletonised' list of integrals (i.e only the symmetry distinct elements), or whether the 'full' list is to be output. This choice may be controlled by the third parameter of the SUPER directive; presenting the optional keyword NOSYM will act to suppress the default skeletonisation, producing the complete list.

Examples

To produce a complete list of 2-electron integrals requires the data line

```
SUPER OFF NOSYM
```

while generation of a skeletonised list in J + K supermatrix format requires the data line

```
SUPER FORCE
```

Now let us consider the default formats imposed by the program (based on both computational efficiency and complexity). We have already given in Table 1 of Part 2 an initial list of these, together with the possible formats, as a function of SCFTYPE. Note however that the defaults specified in the Table are those appropriate to RUNTYPE SCF, and do not necessarily apply for all RUNTYPES; note also that Table 1 of Part 2 gives no consideration to the skeletonised nature of the list. We summarise below the various formats in common use;

1. While for small and medium sized molecules P–supermatrix usage is probably the most efficient, this may only be used for closed–shell SCF calculations. Thus if we wish to perform an initial closed–shell calculation, and then re-use the Mainfile in some subsequent open–shell calculation, the following data structure is typical of that required for the closed shell SCF;

```
TITLE
CLOSED SHELL SCF
SUPER FORCE
ZMATRIX
.
.
.
END
ENTER
```

with the open-shell data structure as follows;

```
RESTART
TITLE
OPEN-SHELL TRIPLET SCF
SUPER FORCE
BYPASS
MULT 3
ZMATRIX
.
.
.
END
ENTER
```

where the SUPER FORCE data line in the closed-shell data provides a usable Mainfile in the open–shell run, allowing BYPASS'ing of integral generation.

2. Conventional integral list format, with no skeletonisation, must be employed in all SCF calculations conducted under control of the TRANSFORM, CI, GF and TDA RUNTYPEs, a requirement that is also present in all CASSCF and MCSCF calculations. This corresponds to presenting the data line

```
SUPER OFF NOSYM
```

Note again that this format will be automatically employed in any run with the above SCFTYPEs and RUNTYPEs in the data stream; the user need only present the SUPER directive when changing these defaults, in anticipation of some future use of the generated Mainfile. Now let us consider performing the calculations above, followed by a CI calculation. The easiest way to proceed is to present the data sequence

```
RESTART
TITLE
OPEN-SHELL TRIPLET DIRECT-CI
```

```

MULT 3
ZMATRIX
.
.
.
END
RUNTYPE CI
DIRECT
.
.
ENTER

```

when the integral list will be regenerated as part of the RUNTYPE CI processing in the appropriate format. However, in order to avoid regenerating the integrals, we need to have requested the appropriate Mainfile format from the outset, with suitable changes to both the closed- and open-shell SCF data sets, thus;

Closed-shell SCF Data

```

TITLE
CLOSED SHELL SCF - SUPPRESS SKELETONISATION
SUPER OFF NOSYM
ZMATRIX
.
.
.
END
ENTER

```

Open-shell SCF Data

```

RESTART
TITLE
OPEN-SHELL TRIPLET SCF -- NO SKELETONISATION
SUPER OFF NOSYM
BYPASS
MULT 3
ZMATRIX
.
.
.
END
ENTER

```

Open-shell Direct-CI Data

```

RESTART
TITLE
OPEN-SHELL TRIPLET DIRECT-CI - BYPASS SCF
MULT 3
BYPASS SCF
ZMATRIX
.

```

Table 2: Parameters of the BYPASS Directive

PHASE	Excluded Computation
ONE	One-electron Integral Evaluation
TWO	Two-electron Integral Evaluation
ADAPT	Generation of Symmetry Adapted Integrals
SCF	Integral Evaluation plus SCF Computation
TRAN	Integral Transformation
OLD	See Text
HF	SCF computation

```

.
.
END
RUNTYPE CI
DIRECT
.
.
ENTER

```

- Conventional integral format, with skeletonisation of the integral list, must be employed in all SCF calculations conducted under control of the HESSIAN, POLARISABILITY, HYPER, MAGNET, RAMAN, INFRARED RUNTYPE's, and in all Møller Plesset calculations, corresponding to the explicit specification

```
SUPER OFF
```

8.4 BYPASS

The BYPASS directive may be used to exclude various phases of computation associated with a given RUNTYPE. For example, when performing multiple SCFs at a given nuclear geometry, it is clearly only necessary to evaluate the 1- and 2-electron list in the first SCF computation. If performing multiple direct-CI calculations under RUNTYPE CI control, it is again convenient to bypass the integral transformation step.

The directive consists of a single data line, read to the variables TEXT, PHASE using format (2A):

- TEXT should be set to the character string BYPASS;
- PHASE should be set to a character string identifying the particular phase of computation to be excluded from the present run of the program. Valid character strings, together with the associated computation are shown in Table 2).

The following points should be noted:

- Simply presenting the data line BYPASS will act to bypass both 1- and 2-electron integral evaluation i.e., is equivalent to the data line BYPASS ONE TWO.
- It is of course assumed that the integral files associated with the excluded steps are available from some previous run of the program. Thus when excluding integral generation, it is assumed that a Mainfile compatible with the particular SCFTYPE requested has been created by some previous job under control of the current Dumpfile. As such the directive is usually presented only in RESTART mode.
- An exception to this rule occurs in cases when the Dumpfile has been corrupted, but the user retains access to, for example, the Mainfile. Clearly there is then the need to associate the two-electron integral file with the computation instigated with a 'new' Dumpfile. Assuming the Mainfile is allocated to the program with the default ED2 LFN, this may be achieved through specifying the keyword OLD on the BYPASS directive.

Example

We illustrate this point by considering an SCF computation on H₂CO. Assuming the Dumpfile in Run I below had not been correctly saved on termination of the job, then Run II may recreate the Dumpfile, and use the Mainfile from the first job (assuming of course that this had been correctly saved).

Run I

```
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
```

Run II

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

8.5 ADAPT

In default the SCF, MCSCF and CI modules are performed internally in a symmetry adapted basis set. This involves the automatic generation of a symmetry adapted list of functions driven from the input molecular geometry and basis specified, and subsequent characterisation of the molecular orbitals in terms of the generated list. The ADAPT directive consists of a single data line comprising two data fields, the first of which should be set to the character string ADAPT. The second data field may be used to either,

- suppress symmetry adaptation by presenting the character string OFF;
- specify the section number on the Dumpfile where the adapted 1-electron integrals are to be placed. If omitted the integrals are routed to section 482. If specified, the section must lie between 1 and 350.

Example 1

```
ADAPT 200
```

This example routes the symmetry adapted 1-electron integrals to section 200 of the Dumpfile.

Example 2

```
ADAPT OFF
```

This example suppresses symmetry adaptation.

Note:-

There is one specific case where suppression of symmetry adaptation will be required, and this is in SCF or GVB calculations which use localised orbitals as the starting point. During the localisation process, under control of RUNTYPE ANALYSE, adaptation is automatically switched off to enable orbitals of different irreducible representation to mix (although the total wavefunction remains, of course, a unitary transformation of the SCF wavefunction). Assuming these orbitals are to be used in a subsequent SCF or GVB calculation, via the VECTORS directive, then the ADAPT OFF data line *must* be presented in any such job that utilises the LMOs.

9 Directives Defining the Molecular System

9.1 TITLE

This directive may be used to define an 80 character title for the run, and extends over two lines. The first line consists of the character string TITLE in the first data field, the second line comprises the title.

Example

```
TITLE
H2CO - 3-21G CALCULATION
```

9.2 MULTIPLICITY

This directive defines the spin multiplicity of the system, and consists of one line read to variables TEXT, MULT using format (A,I).

- TEXT should be set to the character string MULT.
- MULT is used to specify the spin degeneracy of the SCF wavefunction, using the values 1,2,3 etc. for singlet, doublet, triplet states etc. respectively. It is also possible to use one of the character strings SINGLET, DOUBLET, TRIPLET, QUARTET, QUINTET, SEXTET, SEPTET, OCTET and NONET to specify MULT.

The MULT directive may be omitted, when the program will set MULT to 1.

Example

```
MULT 4
MULT QUARTET
```

are equivalent; the wavefunction will be four-fold spin degenerate.

9.3 CHARGE

This directive defines the net charge of the system, and consists of one line read to variables TEXT, ICHA using format (A,I).

- TEXT should be set to the character string CHARGE.
- ICHA is used to specify the net system charge using the values 1,2,3 etc. for net single, double, triple etc. positive charges respectively.

The CHARGE directive may be omitted, when the program will set ICHA to 0, the default thus referencing a neutral system.

Example

```
CHARGE 2
```

would be required, for example, in calculations on a di-cationic species.

9.4 ELECTRONS

This directive defines the number of electrons in the system, and consists of one line read to variables TEXT, NELEC using format (A,I).

- TEXT should be set to the character string ELECTRONS.
- NELEC is used to specify the number of electrons required.

The ELECTRONS directive may be omitted, when the program will set NELEC to the sum of the nuclear charges, adjusted by the charge if specified by the CHARGE directive. The ELECTRONS directive will be required whenever additional nuclear charges are included in the system but should not affect the electron count, for example those used to create a model potential.

The ELECTRONS and CHARGE directives are incompatible.

9.5 ZMATRIX and GEOMETRY

Two directives may be used to define the molecular geometry, ZMATRIX and GEOMETRY. The ZMATRIX directive plays a far more general and versatile role in this specification, and may be used to,

- define a system of internal coordinates required by the geometry optimisation facilities in GAMESS-UK. This is achieved through definition of the so-called VARIABLES. Initial values are attributed to VARIABLES on the VARIABLE definition lines, and as such are used to define an initial geometry;
- define the starting Hessian or force-constant matrix to be used in geometry or transition-state optimisation;
- control the point group symmetry employed in subsequent integral evaluation.

The GEOMETRY directive may be used for direct input of cartesian coordinates, but offers far less control over subsequent optimisation studies. Given the fundamental importance of both directives, we present a detailed description of the associated data input below.

10 The ZMATRIX Directive

The first data line, the directive initiator, is read to variables TEXTA, TEXTB using format 2A.

- TEXTA should be set to the character string ZMATRIX or ZMAT
- TEXTB may be set to one of the strings A.U., AU, ANGSTROM or ANGS and is used to define the units in which the relative positions of the nuclei will be specified. If TEXTB is omitted atomic units will be assumed in default.

The last line of the ZMATRIX directive, the directive terminator, consists of the text END in the first data field. Lines appearing between the directive initiator and terminator are of the following type :

- Z-matrix definition lines, responsible for specifying the relative positions of the nuclei. Most of these will be real nuclei, although dummy nuclei may also be introduced to assist in the geometry specification (see section 8.3)
- VARIABLE and CONSTANT definition lines.
- COORDINATE definition lines.
- CHARGES definition lines.

10.1 Z-Matrix Definition Lines

Each nucleus (including dummies) is numbered sequentially and specified on a single data line, so that the (N+1)th line of the ZMATRIX directive is used to specify the nature and location of the Nth nucleus in terms of the positions of the (N-1) nuclei as determined by previous lines. The specification of the Nth nucleus is contained in up to eight items, read to variables

TAGN, N1, R1, N2, ANG12, N3, ANG123, ITYPE

using format (A,I,F,I,F,I,F,I)

- TAGN is used to give the nucleus a name by which it will be subsequently known, and to convey the chemical nature of the nucleus. Typically it consists of just the chemical symbol, such as 'H' or 'Si' for hydrogen or silicon. Alternatively it may be an alphanumeric string, commencing with the chemical symbol, and followed immediately by a secondary identifying integer to reflect, perhaps, a different environment of the atoms involved e.g., H1 and H2. TAGN may be up to 8 characters long, and must not include the space character. We describe below the role of the TAGN settings on the subsequent determination of molecular point group symmetry.
- N1 is an integer specifying a previously defined nucleus for which the internuclear length $R(N-N1)$ will be given.
- R1 is the internuclear length $R(N-N1)$ in the appropriate units.
- N2 is an integer specifying a second nucleus, N2, different from N1, for which the angle $(N,N1,N2)$ will be given.
- ANG12 is the value of $(N,N1,N2)$, the internuclear angle at N1 between N and N2, in degrees.
- N3 and ANG123 - the significance of N3 and ANG123 depends upon the value assigned to ITYPE, the last item in the list. We may identify the following two cases :

1. ITYPE is omitted (or set to zero), then
 - N3 is an integer specifying a nucleus for which the internuclear dihedral angle (N,N1,N2,N3) will be defined as ANG123.
 - ANG123 is the internuclear dihedral angle (N,N1,N2,N3) specified in degrees. It is the angle between the planes (N,N1,N2) and (N1,N2,N3), whose sign is governed by considering the movement of the vector $N1 \Rightarrow N$ towards the vector $N2 \Rightarrow N3$. The sign is positive if this movement involves a righthand screw motion, and negative if this movement involves a left hand motion.
2. ITYPE is set to 1 or -1, then
 - N3 specifies a nucleus for which a second internuclear angle, (N,N1,N3), will be defined by
 - ANG123 the value of the second internuclear angle, (N,N1,N3) in degrees.

Given the above specification of the two internuclear angles, ANG12 and ANG123, we must differentiate between the two possible positions arising for nucleus N: this is achieved through the sign specified for ITYPE. If the triple vector product

$$(N1 \Rightarrow N).(N1 \Rightarrow N2 \times N1 \Rightarrow N3) \quad (1)$$

is positive, then ITYPE is set to 1: if the vector product is negative, ITYPE should be set to -1.

Having dealt with the general format of a z-matrix definition line specifying the N'th nucleus, let us consider the specification of the first three nuclei, which requires only a subset of the data required in the general case.

1. For the *first* nucleus, the definition line comprises just a single data field, specifying the nucleus name and read to variable TAG1 using format (A).
2. For the *second* nucleus specified, the definition line comprises three data fields, read to variables TAG2, N1, R12 using format (A,I,F), and is used to specify the internuclear length with the first centre.
3. For the *third* nucleus specified, the definition line comprises five data fields, read to variables TAG3, N1, R1, N2, ANG12 using format (A,I,F,I,F) , and is used to specify an internuclear length and angle with the two previously specified nuclei.

Note that for consistency with data input of other codes, the Z-matrix connectivity may now also be defined using atom names directly. In this case the specification of the Nth nucleus is still contained in up to eight items, read to variables:

TAGN, AN1, R1, AN2, ANG12, AN3, ANG123, ITYPE

where AN1, AN2 and AN3 are now either integers or the names of previously defined centres. Clearly, to avoid ambiguities, each centre must be defined using a unique name. Failure to do so may result in the generation of an incorrect geometry: GAMESS-UK will print out a warning

when ambiguities are detected but it is up to the user to ensure that the geometry is correctly defined. Bearing in mind the discussion above detailing the use of the atom tags in defining symmetry, we see that the use of unique atom names required in this form of the Z-matrix definition lines will, in many cases, lead to a reduction in the molecular symmetry and should be avoided when defining the geometry of highly symmetric molecules.

Let us now consider a few examples to illustrate the specification so far

Example 1

A simple C_{2v} system, the water molecule, with $r(\text{O-H}) = 0.952 \text{ \AA}$ and angle $\text{HOH} = 104.5^\circ$.

```
ZMAT ANGS
O
H 1 0.951
H 1 0.951 2 104.5
END
```

Note that it is also possible to separate the individual data fields of each Z-matrix definition line by a 'comma', instead of a blank field, so that the sequence above is equivalent to presenting

```
ZMAT ANGS
O
H,1,0.951
H,1,0.951,2,104.5
END
```

The connectivity for the same structure could also be defined using atom names directly, as follows

```
ZMAT ANGS
O1
H1 O1 0.951
H2 O1 0.951 H1 104.5
END
```

However, in this case, the use of unique atom names will lead to lower symmetry (C_s).

Example 2

Specification of a simple D_{3h} system, the CH_3 radical ($r(\text{C-H}) = 1.120 \text{ \AA}$), either

- by definition of a dihedral angle

```
ZMAT ANGS
C
H 1 1.120
H 1 1.120 2 120.0
H 1 1.120 2 120.0 3 180.0
END
```

when the ITYPE parameter is omitted on the final specification line, or

- by specification of a second internuclear angle, with the ITYPE parameter specified, thus

```
ZMAT ANGS
C
H 1 1.120
H 1 1.120 2 120.0
H 1 1.120 2 120.0 3 120.0 1
END
```

Example 3

Specification of a Td system, the CH₄ molecule (C-H=1.083 Å). Again we may either make use of dihedral angle specification, as follows

```
ZMAT ANGS
C
H 1 1.083
H 1 1.083 2 109.471
H 1 1.083 2 109.471 3 120.0
H 1 1.083 2 109.471 4 120.0
END
```

or the specification of two internuclear angles, thus

```
ZMAT ANGS
C
H 1 1.083
H 1 1.083 2 109.471
H 1 1.083 2 109.471 3 109.471 1
H 1 1.083 2 109.471 3 109.471 -1
END
```

Note that it is not necessary to specify the tetrahedral angle to more than 3 decimal places - the program will replace the value specified with the full precision value. Again, we may use a 'comma' for the field separators on each data line, thus:

```
ZMAT ANGS
C
H,1,1.083
H,1,1.083,2,109.471
H,1,1.083,2,109.471,3,109.471,1
H,1,1.083,2,109.471,3,109.471,-1
END
```

Example 4

Specification of a system of D_{4h} symmetry, planar CH₄ (r(C-H) = 1.083 Å).

```
ZMAT ANGS
C
H 1 1.083
H 1 1.083 2 90.0
H 1 1.083 3 90.0 2 180.0
H 1 1.083 4 90.0 3 180.0
END
```

Example 5

Specification of a C_{3v} system, the methyl fluoride molecule, CH_3F , with $r(C-F)=1.384 \text{ \AA}$, $r(C-H)=1.097 \text{ \AA}$, and angle $HCF = 110.6^\circ$.

```
ZMAT ANGS
C
F 1 1.384
H 1 1.097 2 110.6
H 1 1.097 2 110.6 3 120.0
H 1 1.097 2 110.6 3 -120.0
END
```

Example 6

Specification of a D_{6h} system, the benzene molecule with $r(C-C) = 1.387 \text{ \AA}$ and $r(C-H) = 1.082 \text{ \AA}$.

```
ZMAT ANGS
C
H 1 1.082
C 1 1.387 2 120.0
H 3 1.082 1 120.0 2 0.0
C 3 1.387 1 120.0 4 180.0
H 5 1.082 3 120.0 4 0.0
C 5 1.387 3 120.0 6 180.0
H 7 1.082 5 120.0 6 0.0
C 7 1.387 5 120.0 8 180.0
H 9 1.082 7 120.0 8 0.0
C 9 1.387 7 120.0 10 180.0
H 11 1.082 9 120.0 10 0.0
END
```

10.2 VARIABLES and CONSTANTS Specification

In the examples and specification to date we have assumed that the relative positions of the nuclei are defined using actual values for the bond lengths and bond angles i.e., for R1, ANG12 and ANG123 on the z-matrix definition lines. An alternative specification allows for these quantities to be replaced by symbolic names on the definition lines, and to be subsequently assigned values on the so-called VARIABLE and CONSTANT definition lines.

While this mode of specification is optional in the context of a single point calculation, it *must* be adopted in the framework of geometry optimisations (under control of RUNTYPE OPTIMIZE) and transition state location (under control of RUNTYPE SADDLE). In such calculations distances and angles specified as VARIABLES will be allowed to vary throughout the optimisation, while those read as CONSTANTS will remain fixed at the nominated value, as will those actually specified by value in the z-matrix definition lines.

Note that optimisation studies require at least one variable to be nominated in the z-matrix. The revised format of a z-matrix definition line when using VARIABLES and/or CONSTANTS to specify the relative positions of the nuclei is as follows:

```
TAGN, N1, TEXT1, N2, TEXT2, N3, TEXT3, ITYPE
```

using format (A,I,A,I,A,I,A,I).

The definition of TAGN, N1, N2, N3 and ITYPE are as specified previously in section 8. TEXT1, TEXT2 and TEXT3 are now alphanumeric strings: each is read in A-format and may be up to 8 characters in length, but must not include the space character.

- TEXT1 is symbolic name for the internuclear length R(N-N1).
- TEXT2 is a symbolic name for the internuclear bond angle at N1 between N and N2.
- TEXT3 is a symbolic name for either the internuclear dihedral angle (N,N1,N2,N3) (when ITYPE is omitted), or for a second internuclear angle (N,N1,N3) (when ITYPE is set to 1 or -1).

Each of the symbolic names so defined must subsequently be assigned a value on a CONSTANT or VARIABLE definition line, to be presented after the z-matrix definition lines, prior to the ZMATRIX terminator.

10.2.1 VARIABLES Specification

The specification of all symbolic names to be assigned as VARIABLES during optimisation is initiated by presenting a data line containing the text VARIABLES in the first data line. Subsequent data lines may be used to,

- assign an initial value to the variables, one data line per name, which will subsequently be allowed to vary during optimisation;
- assist in the specification of the initial Hessian in geometry or transition state calculations. In default mode 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. While these defaults are, in most cases, perfectly adequate in equilibrium geometry optimisation, they will not be in transition state optimisations, and should be overridden by providing additional information on the corresponding VARIABLE definition lines.

The first two data fields on each definition line are read to the variables TEXT, VALUE using format (A,F).

- TEXT should be set to the character string of the symbolic variable to be assigned as a VARIABLE, as specified in the z-matrix.
- VALUE assigns an initial value to the symbolic name TEXT.

Additional data fields are used in Hessian specification, with two alternative formats possible. Assigning a value to the diagonal element(s) of the Hessian corresponding to the variable in question is achieved by presenting the character string HESSIAN followed by the value to be assigned. Thus in the H₂O examples above, the default value for the O-H bond variable may be replaced by the data line

```
OH 0.951 HESSIAN 1.1
```

whereby the diagonal force constant for the OH variable is set to 1.1. Note that the value specified is cumulative, and should be the sum of terms arising from *all* occurrences of the variable. Had the molecule in question been HOF, then the definition line might have read

```
OH 0.951 HESSIAN 0.55
```

The second specification requesting explicit computation of the Hessian may be 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.

10.2.2 CONSTANTS Specification

If a ZMATRIX parameter is to be assigned a CONSTANT value during optimisation, then a data line containing the text CONSTANTS in the first data field should be presented after the final VARIABLES definition line. Subsequent data lines assign a value to each symbolic name, one data line per name, which is to remain fixed throughout optimisation. Such a data line is read to the variables TEXT, VALUE using format (A,F).

- TEXT should be set to the character string of the symbolic name specified in the z-matrix.
- VALUE assigns a CONSTANT value to the symbolic name TEXT.

We illustrate the use of constant and variable specification below, drawing upon some of the examples given previously.

Example 1 A rather trivial example is bond length optimisation in a diatomic molecule. A single point calculation at a nominated bond length would be accomplished, in obvious fashion, by the following ZMATRIX

```
ZMAT ANGS
F
H 1 0.954
END
```

The corresponding data specifying the bond length as a VARIABLE is shown below, so that in a bond length optimisation the initial calculation would be performed at $r(\text{H-F}) = 0.954 \text{ \AA}$.

```
ZMAT ANGSTROM
F
H 1 HF
VARIABLES
HF 0.954
END
```

Example 2 The H_2O molecule, with the bond length and angle represented by the symbolic names OH and HOH.

```
ZMAT ANGS
O
H 1 OH
H 1 OH 2 HOH
VARIABLES
OH 0.951
HOH 104.5
END
```

Presenting the above data in a geometry optimisation would lead to optimisation of both the OH and HOH variables. The two ZMATRIX directives shown below are equivalent, and would, in an optimisation run, result in variation of the OH bond length with the HOH angle held fixed at 104.5° .

```
ZMATRIX ANGS
O
H 1 OH
H 1 OH 2 104.5
VARIABLES
OH 0.951
END
```

```
ZMATRIX ANGS
O
H 1 OH
H 1 OH 2 HOH
VARIABLES
OH 0.951
CONSTANTS
HOH 104.5
END
```

Example 3 CH₃ using the dihedral angle specification.

```
ZMAT ANGS
C
H 1 CH
H 1 CH 2 HCH
H 1 CH 2 HCH 3 180.0
VARIABLES
CH 1.120
HCH 120.0
END
```

10.3 Specification and Role of DUMMY Centres

Up to this point we have confined our attention to the specification of real nuclei within the z-matrix definition lines, with the appropriate TAGs commencing with the chemical symbol of the atom in question. It is frequently useful to introduce 'dummy nuclei' to assist in the geometry specification, but which are ignored subsequently throughout the calculation. The z-matrix definition lines for such DUMMY centres are identical in format to that specified in 3.8.1, with the TAGs set either to the symbol 'X' or to '-', the minus sign. The real value of z-matrix specification with dummy nuclei lies in the ability to avoid possible numerical instability that may arise in geometry optimisation studies. In particular, the present optimisation techniques exhibit such instabilities when dealing with internuclear angles close to 180°. It is always possible in such cases to locate a dummy atom on the bond angle bisector, and thus remove the problem.

Note that it is *not* possible to site basis functions on dummy centres; the so-called 'BQ' centres are provided for this purpose (see section 8.4)

Example 1: The water molecule

```
ZMAT ANGSTROM
O
X 1 1.0
H 1 OH 2 90.0
H 1 OH 2 90.0 3 HOH
VARIABLES
HOH 104.5
OH 0.951
END
```

In this case the specification of the HOH angle is as a dihedral angle, rather than as an internuclear angle as in Example 1 of section 8.1. Note that the setting for the O-X length is arbitrary, and is set to 1.0 in this and all subsequent examples.

Example 2 The CH₃ radical.

```
ZMAT ANGSTROM
C
X 1 1.0
```



```

H 1 1.120 2 90.0
H 1 1.120 2 90.0 3 120.0
H 1 1.120 2 90.0 3 -120.0
END

```

Again the internuclear HCH angle is specified as a dihedral angle through the use of the dummy centre.

Example 3 The linear HCN molecule.

```

ZMAT ANGSTROM
C
X 1 1.0
N 1 CN 2 90.0
H 1 CH 2 90.0 3 180.0
VARIABLES
CN 1.1506
CH 1.0532
END

```

Note that optimisation studies of linear species, or molecules containing potential linear fragments should always be conducted with a z-matrix of the type shown above. Let us consider an equivalent Z-MATRIX for the linear HNC isomer. Following the above example, we would write

```

ZMAT ANGSTROM
N
X 1 1.0
C 1 CN 2 90.0
H 1 NH 2 90.0 3 180.0
VARIABLES
CN 1.170
NH 1.011
END

```

This would be perfectly adequate for a calculation on this isomer. In some cases, however, we wish to be able to write a single Z-MATRIX connecting, as in this case, two minima on a potential surface to provide a suitable framework for locating the transition state connecting the 2 minima i.e., to have a single Z-MATRIX capable of describing all three points through the specification of common VARIABLES. We shall return to this concept in more detail in Part 4 when describing the synchronous transit data requirements, but note here that the required z-matrix would be specified thus

```

ZMAT ANGS
C
X 1 1.0
N 1 CN 2 90.0
H 1 CH 2 90.0 3 PHI

```

where the following variables would be specified for the 2 minima

	HCN	HNC
CN	1.151	1.170
CH	1.053	2.281 (ie CN + NH)
PHI	180.0	0.0

while a suitable starting point for the transition state location would be derived from the 'average' geometry, thus

```
ZMAT ANGSTROM
C
X 1 1.0
N 1 CN 2 90.0
H 1 CH 2 90.0 3 PHI
VARIABLES
CN 1.160
CH 1.617
PHI 90.0
END
```

10.4 Specification of Ghost Centres

In the discussion to date, TAG specification on the z-matrix definition lines has been used to convey the chemical nature of the nucleus. An additional requirement here is the ability to specify ghost centres i.e., centres with no electrons, but with basis functions, for use in either,

- basis set superposition estimates (Counterpoise calculations);
- specification of bond-centred functions for incorporating either polarisation effects, or for introducing Rydberg character through the addition of diffuse functions.

Such centres may be introduced in the ZMATRIX using the characters BQ as the first two characters of the TAG. We illustrate this specification below, showing also the citing of basis functions on such centres.

Example

```
TITLE
HCN DUNNING DZ + BOND(S,P)
ZMAT ANGSTROM
C
BQ 1 RCN2
X 2 1.0 1 90.0
N 2 RCN2 3 90.0 1 180.0
X 1 1.0 2 90.0 3 0.0
H 1 RCH 5 90.0 4 180.0
VARIABLES
RCN2 0.580
RCH 1.056
END
```

```

BASIS
DZ H
S BQ
1.0 1.0
P BQ
1.0 0.7
DZ C
DZ N
END
RUNTYPE OPTIMIZE
ENTER

```

For counterpoise calculations the ghosts may be specified by denoting the nuclei involved in a separate GHOST directive. They will retain their basis sets, but their charge will be set to 0.0 and the corresponding electrons will be deleted. e.g.

```
GHOST H1 O1 END
```

A * may serve as wild character, so specifying *1 will make all nuclei whose name ends with 1 into ghosts, as in the following example

```

TITLE
BSSE TEST
GHOST *1 END
GEOMETRY
0.00000000 -1.10092542 -1.43475395 1.0 H1
0.00000000 -1.10092542 1.43475395 1.0 H1
0.00000000 0.00000000 0.00000000 8.0 O1
3.24201636 2.02583666 0.00000000 1.0 H2
4.24693920 4.71362490 0.00000000 1.0 H2
4.77568401 2.98417857 0.00000000 8.0 O2
END
BASIS SV 4-31G
ENTER

```

10.5 Specification of Point Charges

As we have seen above, centres specified within the Z-matrix definition lines fall into three categories;

- atomic centres, with the element type characterised by the first, and when necessary, the second character of the TAG;
- dummy centres, characterised by the X specification;
- ghost centres, characterised by the BQ specification.

There remains the need to be able to specify point charges within the ZMATRIX framework; this capability is provided through the COORDINATES and CHARGES sub-directives. The first data line of the COORDINATES, the sub-directive initiator, contains the character string

COORDINATES in the first data field. Subsequent data lines are the 'point charge definition' lines, each line defining a given centre, and is read to variables TAG, X, Y, Z, and CHARGE using format (A,4F).

- TAG is used to give the centre a name by which it will be subsequently known. TAG may be up to 8 characters long, and should not include the 'space' character. Note that the TAG name *must* commence with the two characters 'BQ'.
- X, Y, Z are the cartesian co-ordinates of the given centre, in the appropriate units.
- CHARGE is the charge of a given nucleus, the units being such that the charge of the proton is unity. Negative, zero and fractional charges are allowed, as well as more usual positive integer values.

If specified, the COORDINATE definition lines must be presented *after* any CONSTANT and VARIABLES specification.

Alternatively, one may reassign the charges to previously defined centre's using the CHARGES sub-directive. The first data line of this subsection contains the character string CHARGES. Subsequent lines are the charge definition lines, each line defining the charge for a centre and is read to variables TAG CHARGE using format (A,F).

- TAG should be a previously defined centre name in either zmatrix (internal or cartesian) or coordinates. Note that TAG usually commences with the two characters 'BQ'.
- CHARGE is the charge which will be assigned to all centre's with that name, overwriting previously defined values. The same conditions as for the charge in coordinates and cartesian apply.

If specified, the CHARGES definition lines must be presented *last*.

It is important to realise that by default the number of electrons are estimated from the total overall charge (as specified by the CHARGE directive) and the sum of the atomic charges. If the sum of atom charges, including the 'BQ' centres, is not integral, or not equal to the sum of the overall charge plus the required number of electrons, then the number of electrons must be specified by the ELECTRONS directive and the CHARGE directive omitted.

When more than one point charge is included, the computed nuclear energy will exclude the interaction between the point charges. If this term is required, use the class 1 directive BQBQ.

Example

In the data file below the COORDINATE data lines are being used to place a H₂O molecule above a grid of charged ions.

```
TITLE
H2O -- C2V -- SURROUNDING POINT CHARGES
ZMAT ANGSTROM
```

```

X
O 1 D
H 2 OH 1 HOX
H 2 OH 1 HOX 3 180.0
VARIABLES
OH 0.956 HESSIAN 1.4
HOX 133.05 HESSIAN 0.5
CONSTANTS
D 3.0
COORDINATES
BQ 0.0 0.0 0.0 2.0
BQ 0.0 -2.106 0.0 -2.0
BQ 0.0 2.106 0.0 -2.0
BQ 2.106 0.0 0.0 -2.0
BQ -2.106 0.0 0.0 -2.0
BQ 0.0 0.0 -2.106 -2.0
BQ 2.106 2.106 0.0 2.0
BQ -2.106 -2.106 0.0 2.0
BQ 2.106 -2.106 0.0 2.0
BQ -2.106 2.106 0.0 2.0
END
RUNTYPE OPTIMIZE
ENTER

```

Example

```

TITLE
CO ON SMALL CRYSTAL
ZMATRIX ANGSTROM
C
O 1 r
CARTESIANS
BQ1 -3.0 0.0 0.0
BQ1 +3.0 0.0 0.0
BQ2 0.0 +3.0 0.0
BQ2 0.0 -3.0 0.0
VARIABLES
R 1.08
CHARGES
BQ1 0.1
BQ2 -0.1
END
ENTER 1

```

10.6 Examples of Z-matrix input

1. Triatomic C_{2v}

- utilising a dummy:

```

ZMAT ANGSTROM
0

```

```

X 1 1.0
H 1 R 2 90.0
H 1 R 2 90.0 3 THETA
VARIABLES
THETA 104.5
R 0.951
END

```

- no dummy, no variables

```

ZMAT ANGSTROM
O
H 1 0.951
H 1 0.951 2 104.5
END

```

2. Polyatomic C_{2v} - $Fe(CO)_2(NO)_2$

```

ZMAT ANGSTROM
FE
X 1 1.0
C 1 FEC 2 CFEX
C 1 FEC 2 CFEX 3 180.0
X 3 1.0 1 90.0 2 180.0
O 3 CO 5 90.0 1 180.0
X 4 1.0 1 90.0 2 180.0
O 4 CO 7 90.0 1 180.0
X 1 1.0 2 90.0 4 180.0
X 1 1.0 9 90.0 2 180.0
N 1 FEN 10 NFEX 9 90.0
X 11 1.0 1 90.0 10 180.0
O 11 NO 12 90.0 1 180.0
N 1 FEN 10 NFEX 9 -90.0
X 14 1.0 1 90.0 10 180.0
O 14 NO 15 90.0 1 180.0
VARIABLES
FEC 1.84
CFEX 53.829
CO 1.15
FEN 1.77
NFEX 55.922
NO 1.12
END

```

3. Tetra-atomic H_3O C_{3v} , in terms of the out-of-plane bending angle

```

ZMAT ANGSTROM
O
X 1 1.0
H 1 R 2 ANG
H 1 R 2 ANG 3 120.0
H 1 R 2 ANG 3 -120.0
VARIABLES
ANG 70.0 HESSIAN 0.6
R 0.964 HESSIAN 1.5
END

```

4. Tetra-atomic PH_3 C_{3v} , in terms of the bond angles

```
ZMAT
P
H 1 RPH
H 1 RPH 2 THETA
H 1 RPH 2 THETA 3 THETA 1
VARIABLES
RPH 2.685 HESSIAN 0.7
THETA 93.83 HESSIAN 0.2
END
```

5. Tetra-atomic D_{3h}

```
ZMAT ANGSTROM
O
X 1 1.0
H 1 R 2 90.0
H 1 R 2 90.0 3 120.0
H 1 R 2 90.0 3 -120.0
VARIABLES
R 0.964 HESSIAN 1.5
END
```

6. NH_4 T_d

```
ZMAT ANGSTROM
N
H 1 R
H 1 R 2 109.471
H 1 R 2 109.471 3 120.0
H 1 R 2 109.471 4 120.0
VARIABLES
R 1.16 HESSIAN 2.0
END
```

7. $\text{Cr}(\text{NO})_4$ T_d

```
ZMAT ANGSTROM
CR
N 1 CRN
N 1 CRN 2 109.471
N 1 CRN 2 109.471 3 120.0
N 1 CRN 2 109.471 4 120.0
X 2 1.0 1 90.0 3 180.0
O 2 NO 6 90.0 1 180.0
X 3 1.0 1 90.0 2 180.0
O 3 NO 8 90.0 1 180.0
X 4 1.0 1 90.0 5 180.0
O 4 NO 10 90.0 1 180.0
X 5 1.0 1 90.0 4 180.0
O 5 NO 12 90.0 1 180.0
VARIABLES
CRN 1.79
NO 1.16
END
```

8. $\text{NH}_4 D_{4h}$

```

ZMAT ANGSTROM
N
H 1 R
H 1 R 2 90.0
H 1 R 3 90.0 2 THETA
H 1 R 4 90.0 3 THETA
VARIABLES
R 1.16 HESSIAN 2.0
THETA 180.0
END

```

9. $\text{NH}_4 D_{2h}$

```

ZMAT ANGSTROM
N
H 1 R
H 1 S 2 90.0
H 1 R 3 90.0 2 180.0
H 1 R 4 90.0 3 180.0
VARIABLES
R 1.339 HESSIAN 1.0
S 0.991 HESSIAN 1.0
END

```

10. $\text{NbCl}_5 D_{3h}$

```

ZMAT ANGSTROM
NB
CL 1 REQ
X 2 1.0 1 90
CL 1 REQ 2 120 3 180
CL 1 REQ 2 120 3 0
CL 1 RAX 2 90 3 90
CL 1 RAX 2 90 3 -90
CONSTANTS
REQ 2.338
RAX 2.362
END

```

11. $\text{Ni}(\text{PH}_3)_2 D_{3d}$

```

ZMAT ANGSTROM
X
NI 1 1.0
P 2 NIP 1 PNIX
P 2 NIP 1 PNIX 3 180.0
H 3 PH 2 HPNI 1 180.0
H 3 PH 2 HPNI 1 -60.0
H 3 PH 2 HPNI 1 60.0
H 4 PH 2 HPNI 1 0.
H 4 PH 2 HPNI 1 120.
H 4 PH 2 HPNI 1 240.

```



```
VARIABLES
NIP 2.1801684 HESSIAN 2.178
PNIX 90.0 HESSIAN 0.1
HPNI 120.6751816 HESSIAN 1.824
PH 1.4353437 HESSIAN 1.787
END
```

12. Fe(CO)₅ D_{3h}

```
ZMAT ANGSTROM
FE
C 1 RCEQ
X 2 1.00 1 90
O 2 RCO 3 90 1 180
C 1 RCEQ 2 120 3 180
X 5 1.00 1 90 2 180
O 5 RCO 6 90 1 180
C 1 RCEQ 2 120 3 0
X 8 1.00 1 90 2 180
O 8 RCO 9 90 1 180
C 1 RCAX 2 90 3 90
X 11 1.00 1 90 2 180
O 11 RCO 12 90 1 180
C 1 RCAX 2 90 3 -90
X 14 1.00 1 90 2 180
O 14 RCO 15 90 1 180
CONSTANTS
RCEQ 1.8273000
RCAX 1.8068000
RCO 1.1520
END
```

13. MoF₆ O_h

```
ZMAT ANGSTROM
MO
F 1 MOF
F 1 MOF 2 90.0
F 1 MOF 2 90.0 3 90.0
F 1 MOF 2 90.0 3 180.0
F 1 MOF 2 90.0 3 -90.0
F 1 MOF 3 90.0 2 180.0
VARIABLES
MOF 1.814
END
```

10.7 Controlling the Point Group Symmetry

In some instances the user need consider lowering the point group determined in default by the program, particularly in the case of degenerate point groups, which for some SCFTYPES and RUNTYPEs must be a subset of the D_{2h} group. Specifically the appearance of the message

```
*****
```

```

* The molecular point group prohibits use of either *
* the requested SCFTYPE or RUNTYPE. Reduce the *
* molecular symmetry by modifying the nuclear TAGs *
* (see chapter 2 of the User Manual) *
*****

```

requires remedial action, involving a simple modification of the TAGs used on the ZMATRIX definition lines. The symmetry handling routines within GAMESS-UK assume that *any centres with differing TAGs are not related by symmetry*. The user may thus control the point group actually adopted in the calculation through appropriate TAG specification. This is demonstrated below, where we consider the examples of section 8.6, and through TAG modification lower the point group symmetry to a subset of D_{2h} .

1. Tetra-atomic PH_3 , may be reduced from C_{3v} symmetry to C_s by modifying the first hydrogenic TAG, thus

```

ZMAT
P
H1 1 RPH
H 1 RPH 2 THETA
H 1 RPH 2 THETA 3 THETA 1
VARIABLES
RPH 2.685 HESSIAN 0.7
THETA 93.83 HESSIAN 0.2
END

```

2. Tetra-atomic OH_3 may be reduced from D_{3h} symmetry to C_{2v} by again modifying the first hydrogen, thus

```

ZMAT ANGSTROM
O
X 1 1.0
H1 1 R 2 90.0
H 1 R 2 90.0 3 120.0
H 1 R 2 90.0 3 -120.0
VARIABLES
R 0.964 HESSIAN 1.5
END

```

3. NH_4 Td; Note that simply changing the first hydrogenic TAG will not produce the desired effect, as this will still lead to a degenerate C_{3v} point group. Changes to the first two H TAGs will lead to C_{2v} symmetry, thus;

```

ZMAT ANGSTROM
N
H1 1 R
H1 1 R 2 109.471
H 1 R 2 109.471 3 120.0
H 1 R 2 109.471 4 120.0
VARIABLES
R 1.16 HESSIAN 2.0
END

```

4. NH_4 D_{4h} ; Changing the first hydrogenic TAG will yield a C_{2v} point group, while changes to the first and third TAG will yield a D_{2h} point group, the preferred option, thus;

```
ZMAT ANGSTROM
N
H1 1 R
H 1 R 2 90.0
H1 1 R 3 90.0 2 THETA
H 1 R 4 90.0 3 THETA
VARIABLES
R 1.16 HESSIAN 2.0
THETA 180.0
END
```

5. NbCl_5 D_{3h} ; changing the first equatorial chlorine TAG will yield a C_{2v} point group, thus;

```
ZMAT ANGSTROM
NB
CL1 1 REQ
X 2 1.0 1 90
CL 1 REQ 2 120 3 180
CL 1 REQ 2 120 3 0
CL 1 RAX 2 90 3 90
CL 1 RAX 2 90 3 -90
CONSTANTS
REQ 2.338
RAX 2.362
END
```

6. $\text{Ni}(\text{PH}_3)_2$ D_{3d} ; The following TAG modification will lower the symmetry to C_{2h} ;

```
ZMAT ANGSTROM
X
NI 1 1.0
P 2 NIP 1 PNIX
P 2 NIP 1 PNIX 3 180.0
H1 3 PH 2 HPNI 1 180.0
H 3 PH 2 HPNI 1 -60.0
H 3 PH 2 HPNI 1 60.0
H1 4 PH 2 HPNI 1 0.
H 4 PH 2 HPNI 1 120.
H 4 PH 2 HPNI 1 240.
VARIABLES
NIP 2.1801684 HESSIAN 2.178
PNIX 90.0 HESSIAN 0.1
HPNI 120.6751816 HESSIAN 1.824
PH 1.4353437 HESSIAN 1.787
END
```

7. MoF_6 O_h ; Simply changing the first two fluorine TAGs will produce a C_{2v} group; a more effective lowering may be achieved through introducing three fluorine TAGS, F, F1 and F2, which can lead to a D_{2h} lowering, thus

```

ZMAT  ANGSTROM
MO
F  1  MOF
F1  1  MOF  2  90.0
F2  1  MOF  2  90.0  3  90.0
F1  1  MOF  2  90.0  3  180.0
F2  1  MOF  2  90.0  3  -90.0
F  1  MOF  3  90.0  2  180.0
VARIABLES
MOF  1.814
END

```

11 The GEOMETRY Directive

It is possible to define the molecular geometry through simple specification of the coordinates, charge and 'tag' of the component atoms under control of the GEOMETRY directive. This mode of operation has been significantly extended in the present release of the program, so that it is now possible to either:

- perform a geometry optimisation in the cartesian space of the molecule under control of the RUNTYPE OPTXYZ specification. Note that automatic symmetry handling is now incorporated in GEOMETRY usage.
- generate the z-matrix and associated variables from the input list of cartesian coordinates, hence permitting use of the standard internal-coordinate based optimisation techniques within the program.

Let us first consider the format of the GEOMETRY directive when used simply as a mechanism for co-ordinate specification. The first data line, the directive initiator, is read to variables TEXTA, TEXTB using format (2A).

- TEXTA should be set to the string GEOMETRY.
- TEXTB may be set to one of the strings AU (or A.U.) or ANGSTROM, and is used to define the units in which the Cartesian co-ordinates will be specified. If TEXTB is omitted, the units will be assumed to be in atomic units.

The last line of the GEOMETRY directive, the directive terminator, consists of the text END in the first data field. Lines appearing between the initiator and terminator are the 'nucleus definition' lines. Each line defines a given nuclear centre, and is read to variables X, Y, Z, CHARGE and TAG using format (4F,A).

- X, Y, Z are the Cartesian co-ordinates of the given centre, in the appropriate units.
- CHARGE is the charge of a given nucleus, the units being such that the charge of the proton is unity. Negative, zero and fractional charges are allowed, as well as more usual positive integer values.

- TAG is used to give the centre a name by which it will be subsequently known. TAG may be up to 8 characters long, and should not include the 'space' character. Note that the TAG name is again used internally within the program to allocate, for example, basis functions to the centre in question. The user should follow certain conventions, as in the ZMATRIX directive, when specifying TAG. Typically the first, and where appropriate the second, character should be used to specify the chemical symbol of the centre. Subsequent characters may be used to provide more specific labelling information e.g., H1, SI2 etc. (see however the Examples below on the impact of TAG specification on subsequent symmetry handling by the program).

Example 1

```

GEOMETRY
  0.0000000    0.0000000    0.0000000  14.0 SI
-1.6165808    1.6165808   -1.6165808  1.0 H
  1.6165808   -1.6165808   -1.6165808  1.0 H
-1.6165808   -1.6165808    1.6165808  1.0 H
  1.6165808    1.6165808    1.6165808  1.0 H
END

```

Example 2

```

GEOMETRY
  0.0000000    0.0000000    0.0000000  14.0 SI
-1.6165808    1.6165808   -1.6165808  1.0 H1
  1.6165808   -1.6165808   -1.6165808  1.0 H2
-1.6165808   -1.6165808    1.6165808  1.0 H3
  1.6165808    1.6165808    1.6165808  1.0 H4
END

```

The difference between these two examples depicting the GEOMETRY data for SiH_4 is the TAG parameters. Thus the hydrogen centres which were user simply TAGged H in example 1, are TAGged H1,H2,H3 and H4 respectively by the user in example 2; this will have a drastic effect on the symmetry handling routines, which assume that *any centres with differing TAGs are not related by symmetry!* Thus in the first example the calculation will proceed in T_d symmetry while in the second C_1 i.e., (no symmetry) will be recognised, with a corresponding increase in integral evaluation and SCF time. The user may control the point group actually adopted in the calculation though appropriate TAG specification. This is demonstrated in the data below, which will lead to the calculation being performed in C_{2v} symmetry.

```

GEOMETRY
  0.0000000    0.0000000    0.0000000  14.0 SI
-1.6165808    1.6165808   -1.6165808  1.0 H1
  1.6165808   -1.6165808   -1.6165808  1.0 H2
-1.6165808   -1.6165808    1.6165808  1.0 H1
  1.6165808    1.6165808    1.6165808  1.0 H2
END

```

11.1 Cartesian-based Optimisation

We show below three examples of simple GEOMETRY specification, with single-point energy evaluation in Example 1, geometry optimisation requested under control of the OPTXYZ specification in Example 2, and the "freezing" of atomic positions when performing geometry optimisation under OPTXYZ control in examples 3.

Example 1

```
TITLE
H2O - EXPLICIT GEOMETRY SPECIFICATION - TZVP BASIS
GEOMETRY
0.0 0.0 -0.2212037 8.0 O
0.0 1.4284429 0.884815 1.0 H
0.0 -1.4284429 0.884815 1.0 H
END
BASIS TZVP
ENTER
```

Note that in this case the coordinates are specified in atomic units: the data line GEOMETRY ANGSTROM would be used for specification in angstrom.

Example 2

```
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
BASIS ST03G
RUNTYPE OPTXYZ
ENTER
```

Example 3

```
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
BASIS ST03G
RUNTYPE OPTXYZ
ENTER
```

A further variation of the 'nucleus definition' lines may be used to request the freezing of one or more atoms when performing geometry optimisation in cartesian space. In such cases

appending the character string NOOPT after the centre TAG will act to freeze the centre in question throughout; thus in the example above the H atoms will remain fixed throughout the geometry optimisation.

11.2 Z-Matrix Construction via GEOMETRY

Finally, it is possible to generate a z-matrix from the list of supplied x-, y- and z-coordinates under control of the GEOMETRY directive. This will then permit usage of the other optimisation procedures, in addition to OPTXYZ e.g., transition-state location, which rely on the definition of internal co-ordinate variables inherent in the z-matrix specification. This mode of operation is requested by the specification of additional keywords on the first data line of the GEOMETRY directive, with the remainder of the GEOMETRY data as defined above. The directive initiator, is now read to variables TEXTA, TEXTB, TEXTC, TEXTD and TEXTE and TEXTF SCALE using format (5A,A,F).

- TEXTA should be set to the string GEOMETRY.
- TEXTB may be set to one of the strings AU (or A.U.) or ANGSTROM, and is used to define the units in which the Cartesian co-ordinates will be specified. If TEXTB is omitted, the units will be assumed to be in atomic units.
- TEXTC may be set to the character string BOND, when all internuclear distances generated in the z-matrix will be assigned a variable name.
- TEXTD may be set to the character string ANGLE, when all bond angles generated in the z-matrix will be assigned a variable name.
- TEXTE may be set to the character string TORSION, when all dihedral angles generated in the z-matrix will be assigned a variable name.

Thus presenting the following data line as the GEOMETRY directive initiator,

```
GEOMETRY ANGSTROM BOND ANGLE TORSION
```

will assign variable status to all internal co-ordinates of the generated z-matrix. The following points should be noted

- Z-matrix construction is activated by the presence of any one of the three additional keywords, BOND, ANGLE or TORSION
- An abbreviated form of the data line requesting all three keywords is possible by specifying the character string ALL. Thus the data line above may be presented as

```
GEOMETRY ANGSTROM ALL
```

- The classification of the internal coordinates as BOND, ANGLE and TORSION provides a rather limited mechanism for constraining any geometry optimisation. Only those variables specifically requested through GEOMETRY will be treated as variables in subsequent optimisation.

- Sometimes the algorithm to assign bonds to specific atom-pairs may fail. Changing the criterion using for TEXTF SCALE e.g SCALE 0.9 may help.

Example 1

```
TITLE\H2O DZ GEOMETRY SPECIFICATION
GEOMETRY ALL
0.0 0.0 -0.2212037 8 0
0.0 1.4284429 0.8848150 1 H
0.0 -1.4284429 0.8848150 1 H
END
BASIS DZ
RUNTYPE OPTIMIZE
ENTER
```

Example 2

```
TITLE\H2CO DZ GEOMETRY SPECIFICATION
GEOMETRY BOND
0.0000000 0.0000000 0.9998722 6.0 C
0.0000000 0.0000000 -1.2734689 8.0 O
0.0000000 1.7650653 2.0942591 1.0 H
0.0000000 -1.7650653 2.0942591 1.0 H
END
RUNTYPE OPTIMIZE\ENTER
```

Example 3

```
TITLE
AZO-THIOPHENE DERIVATIVE
ACCURACY 20 7
NOPRINT VECTORS
GEOMETRY ANGSTROM BOND
-8.879100 -0.405200 -0.722100 8 O
-9.410100 1.638600 -1.156700 8 O
-3.144600 2.045000 -1.672400 7 N
-2.310300 1.077200 -1.875100 7 N
3.034600 2.388300 -2.473700 7 N
-8.598200 0.738400 -1.028100 7 N
-4.461900 1.616700 -1.528300 6 C
-4.925400 0.303600 -1.471300 6 C
-6.290900 0.011800 -1.326500 6 C
-7.176900 1.043200 -1.201600 6 C
-6.781200 2.356000 -1.242700 6 C
-5.434700 2.625100 -1.402300 6 C
-1.007400 1.465800 -2.038600 6 C
-0.563700 2.785400 -1.972400 6 C
0.753100 3.102500 -2.153400 6 C
1.719200 2.084000 -2.368300 6 C
1.266300 0.755700 -2.467100 6 C
-0.063300 0.431200 -2.279100 6 C
4.083900 1.366900 -2.686200 6 C
4.427300 1.253500 -4.104900 6 C
3.526400 3.769300 -2.331300 6 C
```



```

3.791900   4.129700  -0.898100   6   C
-6.572200  -0.941100  -1.297800   1   H
-7.415400   3.100900  -1.057700   1   H
 1.020200   3.997000  -2.073400   1   H
 1.832900   0.038400  -2.734900   1   H
 3.831500   0.343600  -2.299600   1   H
 4.959500   1.538200  -2.165800   1   H
 3.587800   1.020300  -4.637800   1   H
 5.259500   0.581700  -4.119400   1   H
 4.495300   2.043500  -4.561900   1   H
 2.933100   4.391300  -2.839600   1   H
 4.355900   3.794400  -2.887700   1   H
 2.914400   4.032800  -0.339900   1   H
 4.605000   3.512700  -0.603100   1   H
 4.221000   4.898400  -0.947400   1   H
-5.005500   4.009600  -1.441700   6   C
-4.662100   5.117100  -1.473200   7   N
-3.974500  -0.787200  -1.563900   6   C
-3.213800  -1.659800  -1.637900   7   N
-1.289600   3.580200  -1.885500   1   H
-0.512600  -1.020000  -2.328900   6   C
 0.348000  -1.660300  -2.522400   1   H
-1.245900  -1.146100  -3.125400   1   H
-0.962100  -1.294100  -1.374400   1   H
END
BASIS STO3G
RUNTYPE OPTIMIZE
THRESH 6
XTOL 0.005
ENTER

```

11.3 Mixed Z-matrix and Cartesian Input

Structures may also be input using a combination of cartesian coordinates and z-matrices using the ZMATRIX directive. The sub-directive, CARTESIANS, is used to define a set of cartesian atoms. Each line of the CARTESIANS directive comprises four data fields, with the atom name followed by the three cartesian coordinates. The latter may be input as either constants or variables (which may be optimised). If a fifth data-field is provided it's contents is taken to be the charge of the center, like in the COORDINATES section. The sub-directive INTERNALS terminates cartesian input and allows the user to define the remaining atoms using standard z-matrix notation. The example below illustrates how an initial structure for ethylene might be defined using mixed cartesian/zmatrix input.

```

TITLE
ETHYLENE - MIXED Z-MATRIX/CARTESIAN INPUT
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

```

Note the following restrictions; if the first atom in mixed z-matrix input/cartesian mode is a cartesian, then each coordinate (x,y,z) must be input as a constant. If the second atom is also specified as cartesian, then the x and y coordinates must also be constants. Similarly, the y coordinate of the third atom must be a constant. Specifying any of these coordinates as a variable will lead to an error.

Structures can also be input as alternating combinations of z-matrices and coordinates, e.g.

```

TITLE
ETHENE
ZMATRIX ANGSTROM
CARTESIANS
C 0.000 0.000 0.000
C 0.000 0.000 CC
H WIDTH 0.000 -DEPTH
INTERNALS
H 2 CH 1 CCH 3 TWIST
CARTESIANS
H -WIDTH 0.000 -DEPTH
INTERNALS
H 2 CH 1 CCH 4 180.0
VARIABLES
CC 1.4
CH 1.0
WIDTH 0.8
DEPTH 0.5
CCH 120.0
TWIST 10.0
END
RUNTYPE OPTIMISE
ENTER

```

12 The WEIGHTS Directive

A number of modules (runtypes HESSIAN, FORCE, INFRARED and 1-electron properties under RUNTYPE ANALYSE) are able to output results for a number of isotopic substitution patterns. The masses used for these calculations are defined using the WEIGHTS directive.

The user should note that the atomic mass tables are reset to their default values at the start

of each RUNTYPE, so the following data should be presented as part of the data for the RUNTYPE that will use it. By default, a single vector of atomic masses is defined, containing the mass of the most abundant isotope of the element. This mass vector may be modified using the VECTOR keyword within the WEIGHTS data block, as follows:

```
TITLE
H2O2 MODIFIED MASSES FOR PROPERTIES PACKAGE
ZMAT ANGS
0
X 1 1.
H 1 OH 2 90.0
O 1 OO 3 00H 2 0.0
H 4 OH 1 00H 2 HOOH
VARI
OH 1.0010897
OO 1.3962308
OOH 101.1195323
HOOH 124.9861586
END
BASIS ST03G
RUNTYPE SCF
ENTER
RUNTYPE ANALYSE
WEIGHTS
VECTOR 16 1 17 2
END
PROP
6 5
END
VECTORS 1
ENTER
```

will modify the isotopic substitution pattern of the system. One integer for each atom is required, corresponding to the number of nucleons in the required nucleus. The ordering of masses provided should correspond to the input order of the atoms.

The specification:

```
WEIGHTS
VECTOR CREATE M1 M2 M3 ...
END
```

is similar, but will create an additional vector of atomic masses, thus causing an extra set of frequencies and thermochemical data (in the case of RUNTYPE INFRARED) to be output.

Using a value of 0 in an isotope vector is shorthand for the most abundant isotope of the given element, and -1 can be used to request the average of natural abundances. If the integers are omitted, natural abundance for all atoms is assumed.

Alternatively, a single element of the atomic mass vector may be substituted using the SUBSTITUTE keyword:

```
WEIGHTS
```

```
SUBST 2 ISOTOPE 2
END
```

If atom 2 is an H atom, this will result on the mass of deuterium being assigned. For the (rarer) cases where an arbitrary mass is to be used, the form

```
WEIGHTS
SUBST 2 MASS 1.989
END
```

can be adopted. An alternative form of the substitute keyword allows the substitution to be performed for all centres with labels matching a specific tag:

```
WEIGHTS
SUBST C1 ISOTOPE 13
END
```

If the VECTOR CREATE and SUBSTITUTE keywords appear in the same input, substitute will act on the *most recently created* vector of masses. However, input of an explicit mass (rather than an integral isotope specification) is only possible for the first vector. These directives may be used to request a series of substitution patterns in a concise manner, by labelling the symmetry inequivalent atoms to be substituted. The following example will compute the frequencies for non-deuterated chloroethene and the 3 mono-deuterated forms:

```
TITLE
C2H3F - DEUTERATED FREQUENCIES
INTEG HIGH
ZMAT  ANGSTROM
C
C  1  CC
F  1  CF  2  FCC
H1 1  CH1 2  CCH1 3  180.0
H2 2  CH2 1  CCH2 3  0.0
H3 2  CH3 1  CCH3 3  180.0
VARI
CC  1.34
CF  1.6
CH1 1.0
CH2 1.0
CH3 1.0
FCC 120.0
CCH1 120.0
CCH2 120.0
CCH3 120.0
END
BASIS 6-31G
SCFTYPE RHF
RUNTYPE OPTIMISE
XTOL 0.00001
ENTER
```

```

RUNTYPE INFRARED
WEIGHTS
VECT CREATE
SUBST H1 ISOTOPE 2
VECT CREATE
SUBST H2 ISOTOPE 2
VECT CREATE
SUBST H3 ISOTOPE 2
END
ENTER

```

The internal data tables contain average masses and the abundant isotopes for $z < 54$. More comprehensive tables are included for $z < 18$. If other isotopes are needed it is possible to augment the internal tables, the isotope keyword may be used

```
ISOTOPE AG 108.904756
```

will define 109-Ag. If the nearest integer to the mass entered corresponds to an isotope stored within the program, the new mass will replace the stored one.

It is also possible to input the masses for each atom in the geometry block, as follows:

```

TITLE
H2O2 EXPLICIT MASSES WITH GEOMETRY
GEOMETRY
.7541095 -1.0813168 -.2371497 8 0 16.999131
-.7541095 1.0813168 -.2371497 8 0
-.2202484 -2.2328936 .9081618 1 H 2.0140
.2202484 2.2328936 .9081618 1 H
END
BASIS ST03G
RUNTYPE INFRARED
ENTER

```

In this case, only a single set of masses may be provided, and the user cannot make use of the internal table of nuclide masses.

13 Basis Set Specification

The program incorporates a variety of 'built-in' basis sets which may be invoked through the BASIS directive. Prior to outlining the variety of available basis sets, specified under control of BASIS, we consider the role of the HARMONIC directive in requesting usage of spherical-harmonic angular functions and the DEPENDENCY directive which allows control over projecting out nearly dependent parts of the basis.

13.1 The HARMONIC Directive

The default Cartesian angular functions (6 d, 10 f, 15 g) used throughout GAMESS-UK may be overridden under control of the HARMONIC directive. This provides the option of using

spherical-harmonic (5 d, 7 f, 9 g) angular functions. Note that such usage is implemented internally through appropriate transformations, and not by computing integrals or derivative integrals over the spherical functions.

The HARMONIC directive comprises a single data line read to variables TEXTA, TEXTB, TEXTC using format (3A).

- TEXTA should be set to the string HARMONIC
- TEXTB may be set to one of the strings ON or OFF to activate or de-activate the use of spherical harmonic functions.
- TEXTC may be set to the character string PRINT to provide additional diagnostic information.

Typical usage will involve just presenting the string HARMONIC. 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.
- The correlation-consistent basis sets together with a number of the ECP families of basis sets (see below) were designed using spherical harmonics and to use these the HARMONIC directive should be present.

13.2 The DEPENDENCY Directive

When linear dependency occur in the basis, as generally signified by ridiculous energies and wildly oscillating convergence behaviour, it may be necessary to project out the dependent parts of the basis set. The HARMONIC directive should have been tried first. The overlap matrix is diagonalised and any vector with a too small eigenvalue is removed from the calculation. This is implemented internally through appropriate transformations. By default all vectors with eigenvalues smaller than 10^{-7} are eliminated and warnings are printed if eigenvalues fall below 10^{-5} . The DEPENDENCY directive allows one to change this behaviour.

The directive comprises a single data line starting with the keyword DEPENDENCY followed by one or more of three subdirectives.

- CRIT - This directive allows one to specify the criterion used for eliminating vectors, by specifying CRIT II using format (A,I). The criterion is then set to 10^{-II} . Specifying CRIT 0 amounts to a criterion of 10^{-20} .
- NUMBER or QUANTITY - By specifying NUMBER NN using format (A,I) one requests the elimination of NN vectors with the lowest eigenvalues of the S-matrix.

- PRINT - If PRINT is specified using format(A), the eigenvalue analysis of the S-matrix is always printed.

If both NUMBER and CRIT are specified, the criterion that eliminates most vectors applies. Note that the eigenvalues of the S-matrix, correspond to a norm², so that 10^{-7} is close to the machine precision (double precision).

Example To eliminate all vectors with an eigenvalue under 10^{-6} and at least 4 of them, specify

```
DEPENDENCY CRIT 6 NUMBER 4
```

13.3 The BASIS Directive

Broadly speaking, the available basis sets may be classified into the following categories;

- Minimal Basis Sets
- Split-valence Basis Sets
- Double-Zeta Basis Sets
- Triple-zeta and Extended Basis Sets
- ECP Basis Sets
- Polarisation Basis Sets
- Correlation consistent Basis Sets
- Density Functional DFT Basis Sets

The specification of each category, referenced by a suitable codename, is outlined below.

13.3.1 Minimal Basis Sets

- The STO-nG minimal basis due to Pople et al [2]. Basis sets are stored for H-Xe, and are code-named STO3G, STO4G etc.
- The MINI basis sets due to Huzinaga [3]. Two sets are available. The MINI-1 set features three gaussian expansions of each atomic orbital: since the exponents and contraction coefficients are optimized for each element, and s and p exponents are not constrained to be equal, these bases give much lower energies than does STO-3G. This set represents the most extensive coverage of the elements within GAMESS, with the code-named MINI1 basis available for all elements from Hydrogen to Radon ($Z=86$). A second set, code-named MID14, features four gaussian orbitals for each inner shell function.

13.3.2 Split-valence Basis Sets

Code-named SV, these include:

- The split-valence set due to Dunning and Hay (code-named DUNNING) [4]; and Binning and Curtis [5], in the following contractions;

```

<4s/2s>   for hydrogen
<9s5p/3s2p> for first row atoms
<11s7p/6s4p> for second row atoms
<14s11p5d/6s4p1d> for 3rd row atoms

```

- The n-m1G split-valence basis sets due to Pople and co-workers [6] These include the 3-21G (H-Xe), 4-31G and 6-31G sets, and are code-named as such.
- The MIDI basis sets due to Huzinaga. These are derived from the MINI sets by floating the outermost primitive in each valence orbitals, and renormalizing the remaining 2 gaussians. MIDI bases are not scaled by GAMESS-UK. The transition metal basis sets are taken from s^1d^n states. The MIDI bases are code-named MIDI1 and MIDI4.
- the split-valence set due to Schafer, Horn and Ahlrichs [7] (code named AHLRICHS) in the following contractions

```

<4s/2s>   for hydrogen
<7s4p/3s2p> for first row atoms
<10s7p/4s3p> for second row atoms
<14s8p6d/5s2p2d> for 1st row transition metal atoms
<14s10p5d/5s4p2d> for 3rd row atoms

```

A summary of the split-valence basis sets available is given in Table 3.

13.3.3 Double-Zeta Basis Sets

Code-named DZ, these include:

- The Dunning-Huzinaga full double-zeta sets for H-Cl and Hay in the following contractions [8];

```

<4s/2s>   for hydrogen
<9s5p/4s2p> for first row atoms
<11s7p/6s4p> for second row atoms

```

- the double-zeta sets due to Schafer, Horn and Ahlrichs [7] in the following contractions

```

<4s/2s>   for hydrogen
<8s4p/4s2p> for first row atoms
<11s7p/6s4p> for second row atoms
<14s9p6d/8s5p3d> for 1st row transition metal atoms
<14s11p5d/8s6p2d> for 3rd row atoms

```


13.3.4 Triple-zeta and Extended Basis Sets

- The triple-zeta valence (code-named TZV) basis sets due to Dunning [9], McClean [10] and Curtis [5] in the following contractions;

```

<5s/3s>           for hydrogen
<10s6p/5s3p>      for first-row atoms
<12s9p/6s5p>      for second-row atoms
<14s9p5d/10s8p3d> for first-row transition metals [32]
<14s11p5d/9s6p2d> for 3rd row atoms

```

- the triple-zeta set due to Schafer, Horn and Ahlrichs [7] for H-Ar in the following contractions

```

<5s/3s>           for hydrogen
<10s6p/6s3p>      for first row atoms
<12s9p/7s5p>      for second row atoms

```

- The 6-311G basis set of Pople and co-workers, code-named 6-311G.

A summary of the double-zeta and extended basis sets available is given in Table 4.

13.3.5 ECP Basis Sets

Version 6.2 of GAMESS-UK contained just two of the standard literature basis sets for use in performing valence-only calculations, namely;

- that due to Hay and Wadt [11], now code-named LANL, covering the elements Na-Bi.
- the Compact Effective Potentials (code-name CEP or SBKJC) due to (i) Stevens et al [12] for the elements Li-Ar, (ii) Stevens et al [13] for the elements K-Rn, and (iii) Cundari et al [14] for the Lanthanides.

The variety of available ECP basis sets has now been extended through the addition of a further five basis set families, code-named as follows:

1. LANL2 – The double-zeta Hay and Wadt ECP basis sets, with the inner-valence forms used for transition metals etc. These are as provided in the Gaussian and NWChem suite of programs [11].
2. CRENBL – The large ECP orbital basis for use with the small core potentials due to Christiansen et al [15].
3. CRENBS – The small ECP orbital basis for use with the averaged relativistic, large core ECPs due to Ermler and co-workers [16].
4. STRLC – The Stuttgart relativistic, large core ECP basis sets due to Preuss et al. [17].

5. STRSC – The Stuttgart relativistic, small core ECP basis sets due to Preuss et al. [18].

A full list of the elements for which ECP basis sets are available for each of the seven library sets above is given in Table 5. Note that each of the five basis sets above are assumed to be spherical harmonic, and not cartesian (in contrast to the CEP/SBKJC basis). Their usage should thus be accompanied by the presentation of the HARMONIC directive, although this is not enforced by the code.

Additional contractions of the two of the above sets remain available for the LANL and CEP basis sets (note these contractions are NOT available for the LANL2, CRENB, CRENB, STRLC or STRSC basis sets, and are retained at present for consistency with previous versions of the code). For the LANL basis the user may request minimal and double-zeta contractions: for the CEP set minimal, double-zeta plus extended (triple-zeta) contractions are available for Li-Ar, but only double-zeta for K-Rn and the Lanthanides.

The code names for these contractions are, in obvious notation, ECPMIN, ECPDZ and ECPTZV for minimal, double-zeta and triple-zeta respectively. The program will load the LANL set where possible, only loading the CEP set when

- the Hay set is not available e.g., for the 1st row elements, Li-Ne, or
- when requested by the user, through specification of the CEP keyword (see below)

13.3.6 Polarisation Basis Sets

Single first polarisation functions may be appended to several of the standard basis sets above through the use of suitably modified code-names. Specifically:

- For the split-valence Dunning, Ahlrichs and Pople basis sets, SVP will result in the split-valence set plus one set of d-functions on the heavy atoms and one set of p-functions on hydrogen.
- For Double-zeta basis sets, DZP will result in the DZ set plus one set of d-functions on the heavy atoms and one set of p-functions on hydrogen.
- For Triple-zeta basis sets, TZVP will result in the extended set plus one set of d-functions on the heavy atoms and one set of p-functions on hydrogen.
- Note that when appending polarisation functions to the standard GAUSSIAN basis sets e.g., 4-31G, 6-311G etc, the usual notation '*' (append to heavy-atoms only) and '**' (append to both heavy and hydrogen atoms) may be employed. Thus the code-name 6-311G** will yield the 6-311G set with polarisation functions on all heavy plus all hydrogen atoms.
- For the LANL and CEP ECP basis sets, ECPTZVP will provide one set of d-functions on the heavy atoms (only available for the CEP basis for Li-Ar).

Table 3: Library of Split-valence Basis Sets

Element	Basis Set									
	3-21G	4-21G	6-21G	4-31G	5-31G	6-31G	MIDI-1	MIDI-4	DUNNING	AHLRICHS
H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
He	✓	✓	✓							✓
Li	✓	✓	✓	5-21G	5-21G	✓	✓	✓	✓	✓
Be	✓	✓	✓	5-21G	5-21G	✓	✓	✓	✓	✓
B	✓	✓	✓	✓		✓	✓	✓	✓	✓
C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
N	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
O	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ne	✓	✓	✓	✓		✓	✓	✓	✓	✓
Na	✓		✓			✓	✓	✓		✓
Mg	✓		✓			✓	✓	✓		✓
Al	✓		✓			✓	✓	✓	✓	✓
Si	✓		✓			✓	✓	✓	✓	✓
P	✓		✓	✓		✓	✓	✓	✓	✓
S	✓		✓	✓		✓	✓	✓	✓	✓
Cl	✓		✓	✓		✓	✓	✓	✓	✓
Ar	✓		✓			✓	✓	✓		✓
K-Ca	✓					✓	✓	✓		✓
Sc-Cu	✓					✓	✓	✓	✓	✓
Zn	✓					✓	✓	✓		✓
Ga-Kr	✓						✓	✓	✓	✓
Cs-Xe	✓									✓

13.3.7 Correlation consistent Basis Sets

A subset of the correlation consistent basis sets due to Dunning and co-workers [19] are available. These include the double-zeta (code named CC-PVDZ), triple-zeta sets (code name CC-PVTZ), quadruple-zeta sets (code name CC-PVQZ), and quintuple-zeta sets (code name CC-PV5Z). With the exception of magnesium, all four basis sets available for all the H-Ar and Ga-Kr series.

13.3.8 DFT Basis Sets

The polarized DFT orbital basis sets due to Godbout et al. [20], originally developed for the DGauss and DeMon DFT packages. Three such sets are available, the DZVP, DZVP2 and TZVP sets.

Table 4: Library of Double-Zeta, Triple-Zeta and Extended Basis Sets

Element	Basis Set					
	DZ/DZP	DZ/DZP	TZV/TZVP	TZV/TZVP	6-311G	6-311G*
	Dunning	Ahlrichs	Dunning	Ahlrichs		
H	✓	✓	✓	✓	✓	✓
He		✓		✓	✓	✓
Li	✓	✓		✓	✓	✓
Be	✓	✓	✓	✓	✓	✓
B	✓	✓	✓	✓	✓	✓
C	✓	✓	✓	✓	✓	✓
N	✓	✓	✓	✓	✓	✓
O	✓	✓	✓	✓	✓	✓
F	✓	✓	✓	✓	✓	✓
Ne	✓	✓	✓	✓	✓	✓
Na		✓		✓	✓	✓
Mg		✓		✓	✓	✓
Al	✓	✓	✓	✓	✓	✓
Si	✓	✓	✓	✓	✓	✓
P	✓	✓	✓	✓	✓	✓
S	✓	✓	✓	✓	✓	✓
Cl	✓	✓	✓	✓	✓	✓
Ar	✓	✓	✓	✓	✓	✓
K-Ca		✓				
Sc-Zn	✓	✓	✓	✓		
Ga-Kr		✓	✓	✓		

13.4 Global Specification

At the simplest level, it is assumed that each element specified in the z-matrix is to be described by functions belonging to the same category of basis set. In this case the BASIS directive comprises a single data line read to variables TEXT, CODENAME using format (2A).

- TEXT should be set to the character string BASIS
- CODENAME should be set to the code-name of the required basis to be sited on each element specified in the ZMATRIX or GEOMETRY directive (not a DUMMY centre).

Note that in some cases two separate strings may be required in defining the CODENAME. Valid CODENAME strings include the following:

- **STO3G**, **STO4G** etc. - invokes the minimal STO-nG minimal basis due to Pople et al [2]

- **MINI1** - invokes the MINI-1 basis due to Huzinaga.
- **MINI4** - invokes the MINI-4 basis due to Huzinaga.
- **SV** - invokes the split-valence set due to Dunning and Hay.
- To invoke the split-valence basis sets due to Ahlrichs and co-workers [7] append the keyword **AHLRICHS** thus: **SV AHLRICHS**
- To invoke the split-valence basis sets due to Pople and co-workers [6] append the appropriate keyword thus: **SV 3-21G** , **SV 4-31G**, **SV 6-31G**
- **MIDI1** - invokes the MIDI-1 basis due to Huzinaga.
- **MIDI4** - invokes the MIDI-4 basis due to Huzinaga.
- **DZ** - invokes the double-zeta basis sets due to Dunning [9] and McClean [10].
- To invoke the double-zeta basis sets due to Ahlrichs and co-workers [7] append the keyword **AHLRICHS** thus: **DZ AHLRICHS**
- **CC-PVDZ** - invokes the double-zeta polarised correlation consistent basis sets due to Dunning [19].
- **TZV** - invokes the triple-zeta basis sets due to Dunning [9] and McClean [10].
- To invoke the triple-valence basis sets due to Ahlrichs and co-workers [7] append the keyword **AHLRICHS** thus: **TZV AHLRICHS**
- **TZVP** - the TZV basis, augmented with polarisation functions (p on H, d on first- and second-row atoms [21])
- To invoke the triple-zeta basis sets augmented with polarisation functions due to Ahlrichs and co-workers [7] append the keyword **AHLRICHS** thus: **TZVP AHLRICHS**
- **CC-PVTZ** - invokes the triple-zeta polarised correlation consistent basis sets due to Dunning [19].
- **CC-PVQZ** - invokes the quadruple-zeta polarised correlation consistent basis sets due to Dunning [19].
- **CC-PV5Z** - invokes the quintuple-zeta polarised correlation consistent basis sets due to Dunning [19].
- **ECP** - invokes one of the family of seven available ECP basis sets. An additional keyword is now required to specify the family code-name, using one of the character strings LANL, CEP or SBKJC, LANL2, CRENBL, CRENBS, STRLC or STRSC (see above). Thus to invoke the Stuttgart RLC ECP basis set, append the keyword **STRLC**, thus;
BASIS ECP STRLC

As in Version 6.2 of the code, minimal, double- and triple-zeta contractions of the LANL and CEP basis sets remain available. These are requested in different fashion to the above generic specification, involving one of the following keywords on the **BASIS** directive;

- **BASIS ECPMIN** - ECP valence-only basis in minimal contraction.
- **BASIS ECPDZ** - ECP valence-only basis in a double-zeta contraction.
- **BASIS ECPTZV** - ECP valence-only basis in a triple-zeta contraction.
- **BASIS ECPTZVP** - ECP valence-only basis in a triple-zeta contraction, augmented with polarisation functions.

Note again that these contractions are only available for the LANL and CEP basis sets, and not for the LANL2, CRENBL, CRENBS, STRLC or STRSC families.

Examples

1. The 4-31G basis of Pople and co-workers may be requested through either of the two data lines:

```
BASIS SV 4-31G      or      BASIS 4-31G
```

To append polarisation functions to just the heavy atoms, use

```
BASIS 4-31G*
```

and to both heavy and hydrogen atoms,

```
BASIS 4-31G**
```

with no blank characters preceding the '*'.

2. A minimal STO-3G basis is requested thus,

```
BASIS ST03G
```

3. The double-zeta plus polarisation basis of Dunning is requested thus,

```
BASIS DZP
```

4. The double-zeta plus polarisation basis of Ahlrichs is requested thus,

```
BASIS DZP AHLRICHS
```

5. The correlation consistent cc-pvdz basis of Dunning is requested thus,

```
BASIS CC-PVDZ
```

6. The correlation consistent cc-pvdz spherical harmonics basis of Dunning is requested through specification of the HARMONIC option, thus

```
HARMONIC
BASIS CC-PVDZ
```

7. The CEP ECP basis set due to Stevens et al [12] is requested thus,

```
BASIS ECP SBKJC
```

or

```
BASIS ECP CEP
```

8. The ECP spherical harmonic basis due to Hay [11] is requested through additional specification of the HARMONIC option, thus

```
HARMONIC
BASIS ECP LANL2
```

13.5 Hybrid Specification

In the more general case, it is possible to request basis functions from different categories of the available 'built-in' sets. In such cases the User must specify the basis to be sited on each element of the z-matrix which had been defined with a unique tag. A more general form of the BASIS directive is again used to define the basis set. The first line, the directive initiator, consists of the character string BASIS in the first data field. The last line of the directive, the directive terminator, consists of the text END in the first data field. Lines between the initiator and terminator define the basis set, and consist of 'basis definition' lines. Each basis definition line is read to variables TYPE, TAGA, TYPEE using format (3A).

- TYPE should be set to the code-name of the required set of basis functions.
- TAGA should be set to a TAG labelling one or more of one of the centres (not a DUMMY centre) defined in the ZMATRIX or GEOMETRY directive. The group of basis functions will be sited on the nominated centre(s).
- TYPEE should be used when necessary to uniquely define the requested set of basis functions.

Let us assume that in the formaldehyde examples of Part 2 we wished to include polarisation functions *only* on the oxygen atom. This could be achieved by the following data sequence:

```
BASIS
TZV H
TZV C
TZVP O
END
```

To nominate the 6-311G extended set of Pople would require use of the third field on each definition line, thus

```
BASIS
TZV H 6-311G
TZV C 6-311G
TZVP O 6-311G
END
```

while requesting the corresponding set of Ahlrichs would be achieved by the following data lines:

```
BASIS
TZV H AHLRICHS
TZV C AHLRICHS
TZVP O AHLRICHS
END
```

Similarly, locating a split-valence 3-21G basis on hydrogen and a 4-31G basis on carbon and oxygen would be achieved thus, where again the TYPEE field is used to uniquely define the required basis

```

BASIS
SV H 3-21G
SV O 4-31G
SV C 4-31G
END

```

One clear example where this hybrid specification will be required is when performing ECP calculations where the ECP basis is only to be deployed on a sub-set of the atoms specified in the z-matrix. Thus to locate an all electron DZP basis set on hydrogen and the CRENL ECP basis set on oxygen and carbon would be achieved as follows, where the TYPEE field is now used to define the required ECP basis,

```

BASIS
DZP H
ECP O CRENL
ECP C CRENL
END

```

13.6 Scaling Basis Functions

Under hybrid specification, it is possible to input scaling factors for the exponents of the built-in basis sets to override the standard values, so that

$$\zeta^i = \zeta_0^i \times scale^2 \quad (2)$$

where scale is the input value, and i indicates the i'th shell on the centre of interest. This is achieved by appending the scaling factors to the 'basis definition' lines. As such a line is in fact responsible for loading multiple shells on a given centre, the user need be aware of the loading order when attempting to override the default values. Considering, for example, the STO-nG functions, the loading order internally is 1s, 2sp, 3sp, 3d, 4sp, 4d and 5sp. Thus if one wishes to modify the 3d scaling factor for vanadium (default value of 2.55), the input factor needs point to the fourth shell. In specifying such values, the program recognises an input value of zero as requesting the default, so that in the present example, the data line

```
ST03G V 0.0 0.0 0.0 3.0
```

will modify the 3d scale factor from 2.55 to 3.0. To replace the default hydrogenic scaling factor of 1.24 by 1.30 would require the specification

```
ST03G H 1.30
```

For DZ, TZV and associated polarisation basis sets, this scaling is limited to hydrogen. For the DZ basis, the default scaling factors are 1.20 and 1.15 for the hydrogenic s-functions and 1.0 for the 2p (DZP). To override the s factors by unit scale would require the data line

```
DZ H 1.0 1.0
```


A full list of default factors and more details on the loading orders is available from the author on request. Note also that any request to load further basis sets to the built-in library will be handled as swiftly as possible.

13.7 General Basis Specification

Finally it is possible to define explicitly the basis set through the exponents and coefficients of the component primitive functions .

The first line, the directive initiator, again consists of the character string BASIS in the first data field. The last line of the directive, the directive terminator, consists of the text END in the first data field. Lines between the initiator and terminator define the basis set, and consist of 'group definition' and 'primitive definition' lines. The data for each group of contracted functions is introduced in turn, as follows:

- The group definition line is read to variables TYPE, TAGA using (2A).
 - TYPE should be set to one of the characters S,P,D or F. These symbols defines the type of group of basis functions being introduced.
 - TAGA should be set to a TAG of one or more of the centres defined in a GEOMETRY or ZMATRIX directive. The group of basis functions will be sited on the nominated centre(s).
- The primitive definition lines follow the group definition line, and are used to define the contraction coefficients and orbital exponents of the primitive associated with the group. If NTERM primitives are to be used ($NTERM \leq 25$), NTERM primitive definition lines are required, each read to CTRAN, ZETA using format (2F).
 - CTRAN the contraction coefficient of the primitive.
 - ZETA the orbital exponent of the primitive.

The following example defines a double zeta basis set for pyrimidine, with the addition of diffuse s,p and d basis basis functions at a bond centre. Note that basis functions are cited on those with unique TAGs in the z-matrix i.e C, N and H and the mid-point, BQ.

```
TITLE
PYRIMIDINE DZ AT EQUIL GEOMETRY
ZMAT ANGSTROM
C
H 1 CH2
X 1 1.0 2 90.
X 1 1.0 2 90. 3 90.
C 1 C1C4 3 90. 2 180.
X 5 1.0 1 90. 3 0.0
X 5 1.0 1 90. 4 0.0
H 5 CH4 6 90. 1 180.
N 1 CN1 2 NCH2 3 180.
N 1 CN1 2 NCH2 3 0.0
```

```
C 9 CN2 1 CNC 2 180.
C 10 CN2 1 CNC 2 180.
H 11 CH35 9 NCH3 1 180.
H 12 CH35 10 NCH3 1 180.
BQ 1 1.34249115 3 90. 6 0.0
VARIABLES
CH2 1.0789117
CH35 1.0810722
CH4 1.0793701
CN1 1.3350902
CN2 1.3385252
C1C4 2.6849823
NCH2 117.525610
CNC 117.642581
NCH3 116.792492
END
BASIS
S H
0.032828 13.3615
0.231208 2.0133
0.817238 0.4538
S H
1.0 0.1233
S N
0.0020040 5909.44
0.0153100 887.451
0.0742930 204.749
0.2533640 59.8376
0.6005760 19.9981
0.2451110 2.686
S N
1.0 7.1927
S N
1.0 0.7
S N
1.0 .2133
P N
0.018257 26.786
0.116407 5.9564
0.390111 1.7074
0.637221 0.5314
P N
1.0 0.1654
S C
0.002090 4232.61
0.015535 634.882
0.075411 146.097
0.257121 42.4974
0.596555 14.1892
0.242517 1.9666
S C
1.0 5.1477
S C
1.0 0.4962
S C
```

```

1.0 0.1533
P C
0.018534 18.1557
0.1154420 3.9864
0.3862060 1.1429
0.6400890 0.3594
P C
1.0 0.1146
S BQ
1.0 0.021
S BQ
1.0 0.008
S BQ
1.0 0.0025
P BQ
1.0 0.017
P BQ
1.0 0.009
D BQ
1.0 0.015
D BQ
1.0 0.008
END

```

Note that there is no restriction as to the ordering of the groups in the input stream. However the program will internally re-organise the data to produce a list of atom-ordered functions. Note that the User may mix such 'external' basis sets with the available 'built-in' functions. In the following example we wish to add a single diffuse s-function and polarisation f-function to the TZVP carbon basis in formaldehyde;

```

BASIS
TZVP H
TZVP C
TZVP O
S C
1.0 0.02
F C
1.0 1.0
END

```

In the pyrimidine example above, use of the built-in DZ basis set would reduce the BASIS data, thus

```

BASIS
DZ H
DZ N
DZ C
S BQ
1.0 0.021
S BQ
1.0 0.008
S BQ

```

```

1.0 0.0025
P BQ
1.0 0.017
P BQ
1.0 0.009
D BQ
1.0 0.015
D BQ
1.0 0.008
END

```

Finally, we note that it is possible to cite 'built-in' basis sets from the library onto BQ centres, intended for use in basis set superposition studies. Consider the HCN example presented in the previous discussion of ghost centres.

```

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

```

To cite the DZ basis functions at the hydrogenic position (denoted BQH below) while performing an open shell SCF computation on the $^2\Sigma^+$ state of the CN radical would be achieved thus, where the data line

```
DZ BQH H
```

is placing the H DZ functions on the BQH ghost centre:

```

TITLE
CN  DUNNING DZ + BOND(S,P)
MULT 2
ZMAT  ANGSTROM

```

```

C
BQ 1 RCN2
X 2 1.0 1 90.0
N 2 RCN2 3 90.0 1 180.0
X 1 1.0 2 90.0 3 0.0
BQH 1 RCH 5 90.0 4 180.0
VARIABLES
RCN2 0.5991
RCH 1.056
END
BASIS
DZ BQH H
S BQ
1.0 1.0
P BQ
1.0 0.7
DZ C
DZ N
END
ENTER

```

13.8 NWChem and G94 Basis Specification

Finally, in order to provide more compatibility with other ab initio packages, in particular NWChem and Gaussian, it is possible to invert the field ordering on both Group definition (Centre and Group tags) and Primitive definition lines (the exponents and coefficients of the component primitive functions).

The first line, the directive initiator, again consists of the character string BASIS in the first data field, with either the character string NWCHEM or G94 now presented as the second data field. The last line of the directive, the directive terminator, consists of the text END in the first data field. Lines between the initiator and terminator define the basis set, consisting of 'group definition' and 'primitive definition' lines. The data for each group of contracted functions is introduced in turn, with the data fields in reverse ordering to the General Basis lines above, as follows:

- The group definition line is now read to variables TAGA, TYPE using (2A).
 - TAGA should be set to a TAG of one or more of the centres defined in a GEOMETRY or ZMATRIX directive. The group of basis functions will be sited on the nominated centre(s).
 - TYPE should be set to one of the characters S,P,D or F. These symbols defines the type of group of basis functions being introduced.
- The primitive definition lines follow the group definition line, and are used to define the contraction coefficients and orbital exponents of the primitive associated with the group. If NTERM primitives are to be used ($NTERM \leq 25$), NTERM primitive definition lines are required, each read to ZETA, CTRAN using format (2F).
 - ZETA the orbital exponent of the primitive.

– CTRAN the contraction coefficient of the primitive.

The following example is that given above, defining a double zeta basis set for pyrimidine, with the addition of diffuse s,p and d basis basis functions at a bond centre.

```
TITLE
PYRIMIDINE DZ AT EQUIL GEOMETRY
ZMAT ANGSTROM
C
H 1 CH2
X 1 1.0 2 90.
X 1 1.0 2 90. 3 90.
C 1 C1C4 3 90. 2 180.
X 5 1.0 1 90. 3 0.0
X 5 1.0 1 90. 4 0.0
H 5 CH4 6 90. 1 180.
N 1 CN1 2 NCH2 3 180.
N 1 CN1 2 NCH2 3 0.0
C 9 CN2 1 CNC 2 180.
C 10 CN2 1 CNC 2 180.
H 11 CH35 9 NCH3 1 180.
H 12 CH35 10 NCH3 1 180.
BQ 1 1.34249115 3 90. 6 0.0
VARIABLES
CH2 1.0789117
CH35 1.0810722
CH4 1.0793701
CN1 1.3350902
CN2 1.3385252
C1C4 2.6849823
NCH2 117.525610
CNC 117.642581
NCH3 116.792492
END
BASIS G94
H S
13.3615 0.032828
 2.0133 0.231208
 0.4538 0.817238
H S
0.1233 1.0
N S
5909.44 0.0020040
887.451 0.0153100
204.749 0.0742930
59.8376 0.2533640
19.9981 0.6005760
 2.686 0.2451110
N S
7.1927 1.0
N S
0.7 1.0
N S
0.2133 1.0
```

```

N P
26.786 0.018257
5.9564 0.116407
1.7074 0.390111
0.5314 0.637221
N P
0.1654 1.0
C S
4232.61 0.002090
634.882 0.015535
146.097 0.075411
42.4974 0.257121
14.1892 0.596555
1.9666 0.242517
C S
5.1477 1.0
C S
0.4962 1.0
C S
0.1533 1.0
C P
18.1557 0.018534
3.9864 0.1154420
1.1429 0.3862060
0.3594 0.6400890
C P
0.1146 1.0
BQ S
0.021 1.0
BQ S
0.008 1.0
BQ S
0.0025 1.0
BQ P
0.017 1.0
BQ P
0.009 1.0
BQ D
0.015 1.0
BQ D
0.008 1.0
END

```

Note that the User may still mix 'external' basis sets with the available 'built-in' functions. In the following example we wish to add a single diffuse s-function and polarisation f-function to the TZVP carbon basis in formaldehyde;

```

BASIS NWCHEM
H TZVP
C TZVP
O TZVP
C S
0.02 1.0
C F

```

```
1.0 1.0
END
```

In the pyrimidine example above, use of the built-in DZ basis set would reduce the BASIS data, thus

```
BASIS NWCHEM
H DZ
N DZ
C DZ
BQ S
0.021 1.0
BQ S
0.008 1.0
BQ S
0.0025 1.0
BQ P
0.017 1.0
BQ P
0.009 1.0
BQ D
0.015 1.0
BQ D
0.008 1.0
END
```

13.9 Shell-structured Basis Specification

A modification to the data input specification above is required if the user wishes to explicitly input sp-type shells i.e., one s-type function and three p functions, while retaining the computational advantages associated with such basis sets. Two extensions have been made to the group definition and primitive definition lines, namely

1. The TYPE variable on the group definition line should be set to the character L.
2. Each primitive definition line is now read to the variables CTRANS, ZETA, CTRANP using format (3F)
 - CTRANS is the contraction coefficient of the s-primitive
 - ZETA is the common orbital exponent
 - CTRANP is the contraction coefficient of the p-primitive.

Example

The following data contains explicit specification of the 3-21G basis for the H₂CO molecule in a closed-shell SCF calculation.

```
TITLE
```



```

H2CO - 3-21G BASIS - EXPLICIT BASIS SPECIFICATION
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
S C
0.061767 172.256000
0.358794 25.910900
0.700713 5.533350
L C
-0.395897 3.664980 0.236460
1.215840 0.770545 0.860619
L C
1.000 0.195857 1.000
S O
0.059239 322.037000
0.351500 48.430800
0.707658 10.420600
L O
-0.404453 7.402940 0.244586
1.221560 1.576200 0.853955
L O
1.000 0.373684 1.000
S H
0.156285 5.447178
0.904691 0.824547
S H
1.000 0.183192
END
ENTER

```

13.10 PSEUDO

This directive is used to request a valence-only rather than all-electron treatment of the molecular system in hand, to specify the nature of this valence-only treatment and to define those centres within the system to be treated by the pseudo-orbital method. The first data line is read to the variables TEXT, TYPE using format (2A).

- TEXT is set the character string PSEUDO
- TYPE is used to define the nature of the pseudopotential treatment. When requesting the non-local formalism, TYPE should be set to the character string NONLOCAL. If TYPE is set to the character string ECP, or is left blank, one of the available semi-local ECPs will be employed. Seven such semi-local 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 [12] for the elements Li-Ar, (ii) Stevens et al [13] for the elements K-Rn, and (iii) Cundari et al [14] for the Lanthanides.

2. LANL – available in version 6.2 of the code, the LANL ECPs are due to Hay and Wadt [11] 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 [15]. 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 [16].
6. STRLC – The Stuttgart relativistic, large core ECP due to Preuss et al. [17].
7. STRSC – The Stuttgart relativistic, small core ECP due to Preuss et al. [18].

A full list of the elements for which ECPs are available for each of the seven library sets above is given in Table 5 and Table 6.

The subsequent data input is, in general a function of whether the non-local or semi-local ECPs are requested. We consider the data input for each type below.

13.10.1 Non-local ECPs

When performing non-local treatments (TYPE=NONLOCAL) two additional data fields may be presented to override the default location of the Library File. These are read to the variables LIBNAME, IBLLIB using format (A,I). LIBNAME should be set to the LFN used to assign the Library File (ED0-ED19) and IBLLIB to the starting block of the Library. If these two fields are omitted, pseudo-orbital information will be input from block 1 of the data set assigned using the LFN ED0.

Subsequent data lines, the 'ECP definition' lines, are used to assign a specific ECP to centres specified in the ZMATRIX or GEOMETRY directive. The first data field of such a line is read to the variable ECPTAG in A-format, specifying the name or TAG of an ECP within the Library. Subsequent data fields are also read in A-format, and specify the unique TAGs of all centres (as specified by the ZMATRIX or GEOMETRY) to be allocated ECPTAG. The definition lines are continued until all centres to be treated as valence-only have been allocated an appropriate ECP.

The following points should be noted.

- The standard ECPs as input from the library file are labelled with just the element name. Thus the TAG for the ECP for nickel is simply NI.
- Presenting the data line

```
PSEUDO NONLOCAL EDO 1
```

corresponds to the default when requesting a non-local treatment.

Table 5: Library of Available ECPs with Number of Frozen-core Electrons (Li-Xe)

Element	Z	ECP Library						
		SBKJC	LANL	LANL2	CRENBL	CRENBS	STRLC	STRSC
Li- Ne	3 - 10	2	2	-	2	-	2	-
Na	11	10	10	10	10	-	10	-
Mg	12	10	10	10	10	-	10	-
Al	13	10	10	10	10	-	10	-
Si	14	10	10	10	10	-	10	-
P	15	10	10	10	10	-	10	-
S	16	10	10	10	10	-	10	-
Cl	17	10	10	10	10	-	10	-
Ar	18	10	10	10	10	-	10	-
K	19	18	18	18	10	-	18	10
Ca	20	18	18	18	10	-	18	10
Sc	21	10	18	10	10	18	-	10
Ti	22	10	18	10	10	18	-	10
V	23	10	18	10	10	18	-	10
Cr	24	10	18	10	10	18	-	10
Mn	25	10	18	10	10	18	-	10
Fe	26	10	18	10	10	18	-	10
Co	27	10	18	10	10	18	-	10
Ni	28	10	18	10	10	18	-	10
Cu	29	10	18	10	10	18	-	10
Zn	30	10	18	18	10	18	28	10
Ga	31	10	28	28	18	-	28	-
Ge	32	28	28	28	18	-	28	-
As	33	28	28	28	18	-	28	-
Se	34	28	28	28	18	-	28	-
Br	35	28	28	28	18	-	28	-
Kr	36	28	28	28	18	-	28	-
Rb	37	36	36	28	28	-	36	28
Sr	38	36	36	28	28	-	36	28
Y	39	28	36	28	28	36	-	28
Zr	40	28	36	28	28	36	-	28
Nb	41	28	36	28	28	36	-	28
Mo	42	28	36	28	28	36	-	28
Tc	43	28	36	28	28	36	-	28
Ru	44	28	36	28	28	36	-	28
Rh	45	28	36	28	28	36	-	28
Pd	46	28	36	28	28	36	-	28
Ag	47	28	36	28	28	36	-	28
Cd	48	28	36	36	28	36	-	28
In	49	28	46	46	36	-	46	-
Sn - Xe	50 - 54	46	46	46	36	-	46	-

Table 6: Library of Available ECPs with Number of Frozen-core Electrons (Cs-Lw)

Element	Z	ECP Library						
		SBKJC	LANL	LANL2	CRENBL	CRENBS	STRLC	STRSC
Cs - Ba	55 - 56	54	54	46	46	-	54	46
La	57	46	54	46	46	54	-	-
Ce	58	46	-	-	54	-	-	28
Pr	59	46	-	-	54	-	-	28
Nd	60	46	-	-	54	-	-	28
Pm	61	46	-	-	54	-	-	28
Sm	62	46	-	-	54	-	-	28
Eu	63	46	-	-	54	-	-	28
Gd	64	46	-	-	54	-	-	28
Tb	65	46	-	-	54	-	-	28
Dy	66	46	-	-	54	-	-	28
Ho	67	46	-	-	54	-	-	28
Er	68	46	-	-	54	-	-	28
Tm	69	46	-	-	54	-	-	28
Yb	70	46	-	-	54	-	-	28
Lu	71	46	-	-	54	-	-	-
Hf- Au	72 - 79	60	68	60	60	68	-	60
Hg	80	60	68	-	-	-	78	60
Ll	81	60	68	-	-	-	78	-
Pb	82	78	78	-	68	78	78	-
Bi	83	78	78	-	68	78	78	-
Po	84	78	-	-	68	78	78	-
At	85	78	-	-	68	78	78	-
Rn	86	78	-	-	68	78	78	-
Fr	87	-	-	-	78	-	-	-
Ra	88	-	-	-	78	-	-	-
Ac	89	-	-	-	78	-	78	60
Th	90	-	-	-	78	-	78	60
Pa	91	-	-	-	78	-	78	60
U	92	-	-	78	78	-	78	60
Np	93	-	-	78	78	-	78	60
Pu	94	-	-	-	78	-	78	60
Am	95	-	-	-	78	-	78	60
Cm	96	-	-	-	78	-	78	60
Bk	97	-	-	-	78	-	78	60
Cf	98	-	-	-	78	-	78	60
Es	99	-	-	-	78	-	78	60
Fm	100	-	-	-	78	-	78	60
Md	101	-	-	-	78	-	78	60
No	102	-	-	-	78	-	78	60
Lw	103	-	-	-	78	-	78	60

Example

The following data sequence would be required in performing a non-local ECP calculation on the H₂CO molecule:

```
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 with the library ECPs for carbon and oxygen, tagged C and O respectively, allocated 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 [12, 13, 14].

13.10.2 Local ECPs

Data lines after the PSEUDO directive, the 'ECP definition' lines, are used to assign a specific ECP to centres specified in the ZMATRIX or GEOMETRY directive. The first data field of such a line is read to the variable ECPTAG in A-format, specifying the name or TAG of an ECP within the Library. The second data field is the code name to identify the required family of ECP, and should comprise one of the tags CEP (or SBKJC), LANL, LANL2, CRENBL, CRENBS, STRLC or STRSC (see above). Subsequent data fields are also read in A-format, and specify the unique TAGs of all centres (as specified by the ZMATRIX or GEOMETRY) to be allocated ECPTAG. The definition lines are continued until all centres to be treated as valence-only have been allocated an appropriate ECP.

Note that the standard ECPs within the nominated program-resident library are labelled with just the element name. Thus the TAG for the ECP for nickel is simply NI.

Example

The following data sequence would be required in performing a local ECP calculation on the H₂CO molecule using the CRENBL ECP for both oxygen and carbon:

```
TITLE
```

```

H2CO - 1A1 - CRENBL 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 C CRENBL
ECP O CRENBL
DZ H
END
PSEUDO
O CRENBL O
C CRENBL C
ENTER

```

The following points should be noted:

- The CRENBL library ECPs for carbon and oxygen, tagged C and O respectively, are allocated to the unique nuclei tags specified in the z-matrix.
- The BASIS directive defines the CRENBL basis set for C and O, and a DZ basis on H.

Example 2

The following data sequence would be required in performing a local ECP geometry optimisation calculation on SiO using the SBKJC/CEP potentials of [12]. The basis sets have been augmented with d-functions.

```

TITLE
SIO SBKJC-31G+D
MULT 1
ZMAT ANGSTROM\O\SI 1 SIO\VARIABLES\SIO 1.582\END
BASIS
ECP SI SBKJC
D SI
1.0 0.35
ECP O SBKJC
D O
1.0 0.90
END
PSEUDO ECP
SI SBKJC SI
O SBKJC O
RUNTYPE OPTIMIZE
LEVEL 2.0
MAXCYC 30
ENTER

```

The local ECP is requested through the ECP field on the PSEUDO directive: the subsequent data lines of this directive again allocate the appropriate SBKJC ECP from the program resident library to the atoms specified in the z-matrix through TAG specification.

13.10.3 Compatibility with Version 6.2 of GAMESS-UK

Users familiar with previous versions of GAMESS-UK will have noted that differences in the above format for presenting the local ECP data. Note that the previous format for the CEP and LANL basis sets is, however, still supported. The ECP definition lines for these two cases may be presented as previously; the following points should be noted.

- The standard ECPs within the nominated program-resident library are labelled with just the element name. Thus the TAG for the ECP for nickel is simply NI. The CEP pseudopotentials for the elements, Na–Rn, may be TAGged with the three characters CEP appended to the element name, so that the CEP TAG for the sulphur ECP is SCEP.

Example

The following data sequence is equivalent to Example 2 above, and may be used in performing a local ECP geometry optimisation calculation on SiO using the CEP potentials of [12]. The basis sets have been augmented with d-functions.

```
TITLE
SIO CEP-31G+D
MULT 1
ZMAT ANGSTROM\0\SI 1 SIO\VARIABLES\SIO 1.582\END
BASIS
ECPDZ SI CEP
D SI
1.0 0.35
ECPDZ O CEP
D O
1.0 0.90
END
PSEUDO ECP
SICEP SI
O O
RUNTYPE OPTIMIZE
LEVEL 2.0
MAXCYC 30
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 from the program resident library to the atoms specified in the z-matrix through TAG specification. The library ECP for silicon is tagged SICEP.

14 Relativistic Calculations

Relativistic calculations with the Zeroth Order Regular Approximation (ZORA) [23, 24, 25, 26, 28] are available. The implemented formalism can be used to treat both the effects due to electron mass changes and the average spin-orbit interaction (scalar effects) and that due to spin-orbital coupling.

The formalism is effected by incorporating the relativistic components in the 1-electron integrals. This means that scalar relativistic calculations can be performed within every formalism available in the program [30].

14.1 ZORA

The relativistic modifications of the 1-electron integrals are controlled through the ZORA directive. The ZORA directive may be followed by one or more subdirectives. The ZORA directive may be repeated on various lines each time with a selection the subdirectives. In this case the subdirectives are applied in the sequence provided and accumulated settings will take effect in the calculation.

An important aspect of ZORA calculations is the requirement for an internal basis set to represent the relativistic kinetic energy [28]. This internal basis set has to be complete to the extent that all couplings between it and the AO basis set over the impulse operator are represented accurately, i.e.

$$\sum_{\nu} \langle \chi_{\mu}^{AO} | \vec{p} | \chi_{\nu}^{internal} \rangle \langle \chi_{\nu}^{internal} | \approx \langle \chi_{\mu}^{AO} | \vec{p} \quad (3)$$

The internal basis should also be able to represent the density accurately. The program automatically generates this internal basis set by copying the AO basis set and adding functions generated by applying the momentum operator to primitive Gaussians of the AO basis set. This way a primitive function with angular momentum l and exponent e contributes a function of angular momentum $l + 1$ and exponent e to the internal basis. To avoid the internal basis becoming linearly dependent the new $l + 1$ function is kept only if e is higher than any exponent of a $l + 1$ primitive function already present in the AO basis. However, the generation of this internal basis set may be modified by some of the ZORA subdirectives described below.

Another important aspect is that the ZORA kinetic matrix involves the inverse of the Coulomb matrix represented in the internal basis [28]. In principle this means that this Coulomb matrix should be computed and inverted in every SCF cycle. In practice various approximations are available, the selection of which may be controlled by some of the ZORA subdirectives.

If the ZORA directive appears without any subdirectives then

```
ZORA LIMIT G SCALE ON GET ATOM COULOMB ATOM EXCHANGE OFF
ZORA NITERATION 100 INTERNAL AUTO C 137.0359895
```

is used. This invokes a strictly atomic ZORA approach [29]. This is in general a superior approximation, although it might be less suitable in cases, where the molecular charge distribution deviates significantly (near the nuclei) from the atomic charge densities, because it

- is free from the problems with gauge invariance, that plague normal ZORA.
- yields exact gradients and second derivatives.

The atomic scaling is realised through an effective one-electron operator (like the ZORA corrections themselves).

The available subdirectives are described below.

14.2 SPIN

The subdirective SPIN should be used to request inclusion of spin-orbit coupling terms in the ZORA formalism [31]. This involves a switch to a complex 2-component formalism. It is currently only available at the Hartree-Fock level and gradients have not been tested.

14.3 MOLECULAR

Switches from the strictly atomic approximation (the default) to a molecular ZORA. A normal scalar non-approximate scaled ZORA is thus obtained by using

```
ZORA MOLECULAR
```

The subdirective MOLECULAR is equivalent to

```
GET NONE COULOMB FULL SCALE ON
```

noting that the scaling is now molecular in nature.

14.4 SCALE

The ZORA equation differs from the Hartree-Fock equation in the form of the kinetic energy operator but also in the presence of a scale factor for the equation for each orbital (see [28] Eq.(7)). The SCALE subdirective can be used to choose between different choices for calculating the scale factors in the ZORA 1-electron equations. It is read to the variables TEXT, TEXTA using the format (2A), where

- TEXT should be set to the string SCALE
- TEXTA can be set to
 - ON (default) in which case the scale factors will be evaluated ; The scaling will be either strictly atomic (if GET ATOM, the default, is in effect), or applied to the molecular SCF orbital energies. (ZORA MOLE)
 - OFF in which case the scale factors are assumed to be 1.
 - IORA in which case the metric will be changed to perform infinite order scaling as proposed by Kenneth Dyall and Erik van Lenthe (in an experimental and not the published form).
 - MOLECULAR in which case scaling is applied to the converged molecular orbital energies even if GET ATOM (the default) is in effect. This is also the default behaviour for molecular scaled ZORA calculations. *Note* that this scaling spoils the exactness of gradients, even for GET ATOM.

14.5 Directives controlling the calculation of the Coulomb matrix

14.5.1 GET

The GET subdirective is used to determine the density, which is used to calculate the Coulomb matrix needed in the ZORA formalism. This density may be read from the dumpfile (DENSITY) or calculated from natural orbitals (VECTORS), or the densities of the constituent atoms may be used in an by itself iterative ZORA procedure as is done in the default "GET ATOM" behaviour. If GET is not specified in a molecular ZORA calculation (ZORA MOLECULAR), the density of the Hartree-Fock calculation in that iteration is used (cf NITERATION).

The directive is read to the variables TEXT, TEXTA, ISECT using the format (2A,I), where

- TEXT should be set to the string GET
- TEXTA should be set to either
 - VECTORS in which case ISECT should be set to a vectors dumpfile section
 - DENSITY in which case ISECT should be set to a density dumpfile section
 - ATOM (which shouldn't be followed by an integer)
 - ATOI in which case NIT should be provided, giving the number of atomic zora scfs to be performed (default (ATOM) results in NIT=3)
 - NONE (which shouldn't be followed by an integer and resets get)

14.5.2 COULOMB

The COULOMB subdirective should be used to choose among the various possible approximations for the Coulomb matrix in the ZORA kinetic energy expression [28]. In the default case only ATOMIC and NONE have a meaning. The subdirective is read to the variables TEXT, TEXTA using the format (2A), where

- TEXT should be set to the string COULOMB
- TEXTA can be set to
 - FULL in which case the complete Coulomb matrix in the internal basis will be used without any approximations.
 - ATOMIC in which case only intra-atomic components of the Coulomb matrix are taken into account.
 - NONE in which case all 2-electron contributions to the Coulomb matrix will be neglected (bare nuclei).
 - SMALL in which case the Coulomb matrix will be evaluated in the AO basis and subsequently projected onto the internal basis before inverting it.

14.5.3 EXCHANGE

It is read to the variables TEXT, TEXTA using the format (2A), where

- TEXT should be set to the string EXCHANGE
- TEXTA can be set to
 - OFF (default): No exchange is included in the Coulomb potential
 - ON : Exchange is included in the Coulomb potential (not justifiable)

14.5.4 NITERATION

In principle the Coulomb matrix in the internal basis should be computed every SCF cycle. In practice it turns out to be sufficient to compute it far less often and assume it to be constant for a number of SCF cycles. The NITERATION subdirective can be used to set the maximum number of SCF cycles between recomputing the Coulomb matrix in the internal basis. The directive is read to the variables TEXT, NITER using the format (A,I), where

- TEXT should be set to the string NITERATION
- NITER should be set to the number of iterations required

14.6 Directives controlling the internal basis

14.6.1 NOCOPY

The subdirective NOCOPY is read to variable TEXT using the format (A). If given it signifies, that the internal basis is to be generated, without copying the external basis first.

14.6.2 INTERNAL

The INTERNAL subdirective should be used to select a mechanism for defining the internal basis. The subdirective INTERNAL is read to the variables TEXT, TEXTA using format (2A).

- TEXT should be set to the string INTERNAL
- TEXTA should be set to either
 - AUTO in which case the internal basis will be constructed automatically.
 - NONE or OFF in which case no internal basis will be used.
 - MANUAL in which case the internal basis should be entered explicitly.

14.6.3 LIMIT

The LIMIT subdirective can be used to limit the maximum angular momentum of the functions included in the internal basis in addition to the AO basis functions.

The subdirective is read to the variables TEXT, TEXTA using the format (2A). Where

- TEXT should be set to the string LIMIT
- TEXTA should be one of the strings S, P, D, or F.

Example: Consider an AO basis containing S and P functions, then by default the internal basis set would consist of the AO basis functions plus newly generated P and D functions. In this case one could use LIMIT P to restrict the additional functions to P functions only and suppress the generation of the D functions.

14.6.4 CONTRACTED

The CONTRACTED subdirective attempts to create contracted $l + 1$ angular momentum basis functions to extend the internal basis set instead of creating uncontracted ones. This should help to limit the size of the internal basis.

14.7 Miscellaneous

14.7.1 PRINT

The PRINT subdirective can be used to turn on additional printing. Its probably most important effect is to print the internal ZORA basis which used to represent the inverse of the Coulomb matrix [28].

14.7.2 PRI2

The PRI2 subdirective can be used to invoke a debug level of output for ZORA related entities.

14.7.3 C

The C subdirective can be used to explicitly specify the speed of light in atomic units. It is read to the variables TEXT, LIGHTSPEED using the format (A,F), where

- TEXT should be set to the character C
- LIGHTSPEED is the light speed in atomic units

By default the light speed will be set to 137.0359895 a.u.

14.7.4 GAUGE

The subdirective GAUGE allows one to investigate gauge-invariance errors in the ZORA formalism. Using the subdirective results in adding IVAL times the overlap matrix to the potential. If ZORA was exactly gauge-invariant this would result in a change of the total energy by IVAL times the number of electrons. The deviation from this gives an indication of the gauge-variation error in ZORA. The subdirective GAUGE is read to the variables TEXT, IVAL using the format (A,I), where

- TEXT should be set to the string GAUGE
- IVAL should be set to an integer, which is used to add IVAL/NEL time the S-matrix to the 1-electron (h) matrix, where NEL is the number of electrons. If a method is GAUGE-invariant, this should result in an energy shift of exactly IVAL Hartree.

This directive might be handy for non-ZORA calculations also, to test the Gauge-invariance.

14.7.5 OFF

The subdirective OFF may be used to switch the use of the ZORA formalism off explicitly.

References

- [1] H.B. Schlegel, *J. Chem. Phys.* **77** (1982) 3676, doi:[10.1063/1.444270](https://doi.org/10.1063/1.444270).
- [2] W.J. Hehre, R.F. Stewart and J.A. Pople, *J. Chem. Phys.* **51** (1969) 2657, doi:[10.1063/1.1672392](https://doi.org/10.1063/1.1672392); W.J. Hehre, R. Ditchfield, R.F. Stewart and J.A. Pople, *J. Chem. Phys.* **52** (1970) 2769, doi:[10.1063/1.1673374](https://doi.org/10.1063/1.1673374); W.J. Pietro and W.J. Hehre, *J. Comp. Chem.* **4** (1983) 241, doi:[10.1002/jcc.540040215](https://doi.org/10.1002/jcc.540040215); J.B. Collins, P.van R. Schleyer, J.S. Binkley and J.A. Pople, *J. Chem. Phys.* **64** (1976) 5142, doi:[10.1063/1.432189](https://doi.org/10.1063/1.432189); H. Tatewaki and S. Huzinaga, *J. Chem. Phys.* **72** (1980) 399, doi:[10.1063/1.438863](https://doi.org/10.1063/1.438863).
- [3] S. Huzinaga, J. Andzelm, M. Klobukowski, E. Radzio-Andzelm, Y. Sakai and H. Tatewaki, 'Gaussian Basis Sets for Molecular Calculations', Elsevier, Amsterdam, 1984.
- [4] T.H. Dunning Jr. and P.J. Hay in 'Modern Theoretical Chemistry', Vol. 3, ed. H.F. Schaefer, Plenum, New York (1977) 1.; T.H. Dunning Jr., *J. Chem. Phys.* **66** (1977) 1382, doi:[10.1063/1.434039](https://doi.org/10.1063/1.434039).
- [5] R.C. Binning and L.A. Curtiss, *J. Comp. Chem.* **11** (1990) 1206–1216, doi:[10.1002/jcc.540111013](https://doi.org/10.1002/jcc.540111013).
- [6] J.S. Binkley, J.A. Pople and W.J. Hehre, *J. Am. Chem. Soc.*, **102** (1980) 939, doi:[10.1021/ja00523a008](https://doi.org/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](https://doi.org/10.1021/ja00374a017); M.J. Frisch, J.A. Pople and J.S. Binkley, *J. Chem. Phys.* **80** (1984) 3265, doi:[10.1063/1.447079](https://doi.org/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](https://doi.org/10.1002/jcc.540070313); *ibid* **8** (1987) 861–879, doi:[10.1002/jcc.540080614](https://doi.org/10.1002/jcc.540080614), 880–893, doi:[10.1002/jcc.540080615](https://doi.org/10.1002/jcc.540080615).
- [7] A. Schafer, H. Horn and R. Ahlrichs, *J. Chem. Phys.* **97** (1992) 2571, doi:[10.1063/1.463096](https://doi.org/10.1063/1.463096).
- [8] T.H. Dunning Jr., *J. Chem. Phys.* **53** (1970) 2823, doi:[10.1063/1.1674408](https://doi.org/10.1063/1.1674408).
- [9] T.H. Dunning Jr., *J. Chem. Phys.* **55** (1971) 716, doi:[10.1063/1.1676139](https://doi.org/10.1063/1.1676139).
- [10] A.D. McLean and G.S. Chandler, *J. Chem. Phys.* **72** (1980) 5639, doi:[10.1063/1.438980](https://doi.org/10.1063/1.438980).
- [11] P.J. Hay and W.R. Wadt, *J. Chem. Phys.* **82** (1985) 270 doi:[10.1063/1.448799](https://doi.org/10.1063/1.448799), 284 doi:[10.1063/1.448800](https://doi.org/10.1063/1.448800), 299 doi:[10.1063/1.448975](https://doi.org/10.1063/1.448975).
- [12] W.J. Stevens, H. Basch and M. Krauss, *J. Chem. Phys.* **81** (1984) 6026, doi:[10.1063/1.447604](https://doi.org/10.1063/1.447604).
- [13] W.J. Stevens, P.G. Jasien, M. Krauss, and H. Basch, *Can. J. Chem.* **70** (1992) 612, doi:[10.1139/v92-085](https://doi.org/10.1139/v92-085).
- [14] T.R. Cundari and W.J. Stevens, *J. Chem. Phys.* **98** (1993) 5555, doi:[10.1063/1.464902](https://doi.org/10.1063/1.464902).

- [15] (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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1063/1.453069); (Xe) - M. M. Hurley et al., *J. Chem. Phys.* **84** (1986) 6840, doi:[10.1063/1.450689](https://doi.org/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](https://doi.org/10.1063/1.458934), erratum: doi:[10.1063/1.468517](https://doi.org/10.1063/1.468517); (Ba, Ce-Lu) - R.B. Ross, W.C. Ermler, S. Das, Unpublished; (Fr-Ra, Ac-Pu) - W.C. Ermler, R.B. Ross, P.A. Christiansen, *Int. J. Quant. Chem* **40** (1991) 829, doi:[10.1002/qua.560400611](https://doi.org/10.1002/qua.560400611).
- [16] (Sc - Co, Cu-Zn) - M.M. Hurley et al., *J. Chem. Phys.*, **84** (1986) 6840, doi:[10.1063/1.450689](https://doi.org/10.1063/1.450689); (Ni, Y-Cd) - L.A. LaJohn et al., *J. Chem. Phys.*, **87** (1987) 2812, doi:[10.1063/1.453069](https://doi.org/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](https://doi.org/10.1063/1.458934), erratum: doi:[10.1063/1.468517](https://doi.org/10.1063/1.468517).
- [17] (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](https://doi.org/10.1016/0009-2614(82)80012-2); (B-Ne) - A. Bergner, M. Dolg, W. Küchle, H. Stoll, H. Preuss, *Mol. Phys.* **80** (1993) 1431, doi:[10.1080/00268979400100024](https://doi.org/10.1080/00268979400100024); (Mg) - P. Fuentealba, L. v. Szentpaly, H. Preuss, H. Stoll, *J. Phys. B* **18** (1985) 1287, doi:[10.1088/0022-3700/18/7/010](https://doi.org/10.1088/0022-3700/18/7/010); (Al) - G. Igel-Mann, H. Stoll, H. Preuss, *Mol. Phys.* **65** (1988) 1321, doi:[10.1080/00268978800101811](https://doi.org/10.1080/00268978800101811); (Hg-Rn) - W. Küchle, M. Dolg, H. Stoll, H. Preuss, *Mol. Phys.* **74** 1245 (1991), doi:[10.1080/00268979100102941](https://doi.org/10.1080/00268979100102941); (Ac-Lr) W. Küchle, M. Dolg, H. Stoll, H. Preuss, *J. Chem. Phys.* **100** (1994) 7535, doi:[10.1063/1.466847](https://doi.org/10.1063/1.466847).
- [18] (K) - A. Bergner, M. Dolg, W. Küchle, H. Stoll, H. Preuss, *Mol. Phys.* **80** (1993) 1431, doi:[10.1080/00268979400100024](https://doi.org/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](https://doi.org/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](https://doi.org/10.1021/j100124a012).
- [19] T.H. Dunning, Jr., *J. Chem. Phys.* **90** (1989) 1007, doi:[10.1063/1.456153](https://doi.org/10.1063/1.456153); D.E. Woon and T.H. Dunning, Jr., *J. Chem. Phys.* **98** (1993) 1358, doi:[10.1063/1.464303](https://doi.org/10.1063/1.464303); D.E. Woon and T.H. Dunning, Jr., *J. Chem. Phys.* **100** (1994) 2975, doi:[10.1063/1.466439](https://doi.org/10.1063/1.466439); A.K. Wilson, D.E. Woon, K.A. Peterson, T.H. Dunning, Jr., *J. Chem. Phys.* **110** (1999) 7667, doi:[10.1063/1.478678](https://doi.org/10.1063/1.478678).
- [20] N. Godbout, D. R. Salahub, J. Andzelm and E. Wimmer, *Can. J. Chem.* **70** (1992) 560, doi:[10.1139/v92-079](https://doi.org/10.1139/v92-079).
- [21] R. Ahlrichs and P.R. Taylor, *J. Chim. Phys.*, **78** (1981) 315.
- [22] Ph. Durand and J.-C. Berthelat, *Theoret. Chim. acta*, **38** (1975) 283, doi:[10.1007/BF00963468](https://doi.org/10.1007/BF00963468).
- [23] Ch. Chang, M. Pelissier, and Ph. Durand, *Phys. Scr.* **34** (1986) 394, doi:[10.1088/0031-8949/34/5/007](https://doi.org/10.1088/0031-8949/34/5/007).

- [24] J.-L. Heully, I. Lindgren, E. Lindroth, S. Lundqvist, and A.-M. Mårtensson-Pendrill, *J. Phys.* **B19** (1986) 2799, doi:[10.1088/0022-3700/19/18/011](https://doi.org/10.1088/0022-3700/19/18/011).
- [25] E. van Lenthe, E.J. Baerends, and J.G. Snijders, *J. Chem. Phys.* **99** (1993) 4597, doi:[10.1063/1.466059](https://doi.org/10.1063/1.466059).
- [26] 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](https://doi.org/10.1016/0009-2614(95)01156-0).
- [27] Kenneth G. Dyall, and Erik van Lenthe, *J. Chem. Phys.* **111** (1999) 1366, doi:[10.1063/1.479395](https://doi.org/10.1063/1.479395).
- [28] S. Faas, J.G. Snijders, and J.H. van Lenthe, *Scalar Relativistic Ab Initio ZORA Calculations in: Quantum Systems in Chemistry and Physics Vol. 1: Basic Problems and Model Systems*, page 251 Kluwer Academic Publishers (2000).
- [29] J.H. van Lenthe, S. Faas and J.G. Snijders, *Chem. Phys. Lett.* **328** (2000) 107, doi:[10.1016/S0009-2614\(00\)00832-0](https://doi.org/10.1016/S0009-2614(00)00832-0).
- [30] S. Faas, J.H. van Lenthe and J.G. Snijders, *Mol. Phys.* **98** (2000) 1467–1472, doi:[10.1080/002689700417574](https://doi.org/10.1080/002689700417574).
- [31] S.Faas, J.H. van Lenthe, A.C. Hennem and J.G. Snijders, *J. Chem. Phys.* **113** (2000) 4052, doi:[10.1063/1.1288387](https://doi.org/10.1063/1.1288387).