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 13. GAMESS-UK under UNIX - Rungamess

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

Contents

1 Introduction

In the present section we consider execution of GAMESS–UK using the UNIX shell script *rungamess*. The jobs perform the same function as those in the previous Chapter, and the GAMESS–UK directives themselves are identical. However, the input files contain *only* GAMESS–UK directives, all reference to pathnames for executable, library, and scratch files is handled by the script.

The principle differences associated with this mode of usage of the program are as follows;

1. The users data files do not contain reference to specific pathnames for executables, scratch directories, etc. These are taken from user–set environment variables, or are supplied by the local support staff.
2. The job executes in a specially created directory, which will usually be on a scratch disk. Files are routed back to the directory of submission, or to longer–term temporary filestore as required.
3. A variety of batch queuing systems are provided (local configuration is required).
4. The decision regarding which files to retain beyond the end of the job may be set automatically, given the type of job
5. The job may execute on a remote networked machine, without the need to explicitly move input and output, or files (binary datasets are retained on the execution machine)

Note that this documentation describes version 3.x of rungamess, these versions differs substantially from the earlier versions. From a user point of view the main differences are

- only the short forms of the arguments are now accepted (–keep must be replaced by –k, *etc.*).
- the job name must be the last argument.

The next section summarises the syntax and the function of the arguments to the *rungamess* script, and subsequent sections contain the following illustrative examples.

1. Single point SCF calculation of water using the default basis.
2. Sequence of calculations on water, including starting and restarting geometry optimisation of the neutral molecule, an RHF calculation on the $^2B_1$ cation, boys localisation and 2-pair GVB calculation of the neutral molecule.
3. Closed-shell SCF calculations on C_4F_4, $C_6H_5NO_2$and $C_6H_2(NO_2)_3CH_3$.
4. STO-3G calculation of Na_7Mg^+, together with an RHF and UHF calculation of the triplet state.

5. Extended basis set calculation on Na_7Mg^+, with SCF calculation of the singlet state preceding an RHF calculation of the triplet state.
6. ECP calculation on Na_7Mg^+in a double-zeta valence basis, with SCF calculation of the singlet state preceding an RHF calculation of the triplet state.
7. 2-pair GVB calculation on 1-imino-2,4-pentadiene, $CH_2(CH)_4NH_2^+$
8. Graphical analysis of the X^1A_1 state of $Ni(CO)_4$.
9. STO-3G direct-SCF calculations on $C_{11}O_{10}NPH_{18}$ and the geometry optimisation of $Be(C_5H_5)_2$.
10. Determination of the transition state for the HCN/HNC isomerisation reaction. This example demonstrates usage of the three saddle point algorithms available, the default trust-region method plus the synchronous transit and Simons-Jorgensen algorithms.
11. Location of the transition state for the HPSi/HSiP isomerisation and subsequent force constant evaluation.
12. Specification of bond-centred functions, located at the midpoint of the C-N bond in the HCN/HNC transition state.
13. SCF geometry optimisation and analytic force constants of C_2H_4.
14. MP2 geometry optimisation and analytic force constants of C_2H_4.
15. MP2 geometry optimisation and polarisability of C_2H_4.
16. Direct-MP2 calculation of C_5H_5N in a 6-31G* basis.
17. CASSCF geometry optimisation of the X^1A_1 state of H_2O.
18. CASSCF + 2nd-order CI calculations of the $X^1\Sigma^+$ state of BeO.
19. MCSCF + 2nd-order CI calculations of the $X^1\Sigma^+$ state of BeO.
20. Table-CI calculation of the X^2A_1 and 1^2A_1 states of the ammonia cation, NH_3^+ cation.
21. ECP calculation on $NiCCH_2$ with CASSCF and Direct-CI calculations of the lowest triplet state wavefunction.
22. Table-CI calculations typical of those performed in the calculation of electronic spectra. In this case we are studying the disposition of the lowest $^1A_1$ and 1^1A_2 states of pyridine. The sequence of calculations involve the determination of the lowest 10 states of each category, performed in a DZ plus rydberg basis.
23. Full-CI calculations of the X^1A_1 state of H_2O.

2 Usage of the Rungamess Script

2.1 Installation

Please see the `README` file in the GAMESS-UK/rungamess directory for the installation and configuration instructions.

2.2 Running a Job

In its simplest usage, the script is invoked by typing *rungamess*, followed by a job name, for example:

```
rungamess myjob
```

This will cause GAMESS–UK to read the file myjob.in, and generate the listing file myjob.out, and (when appropriate) punch file myjob.pun in the directory of submission.

The exact pathnames used for the executable, and scratch files are determined from values of environment variables, as follows.

- The executable name is taken from the environment variable GAMESS_EXE.
- The environment variable GAMESS_SCR must be set to the name of a directory in which the user has permission to create files. A directory will be created under this directory for each job that is run, and all binary datasets that are not routed elsewhere will be written here for the duration of the job. The directory is deleted at the end of the job. If GAMESS_SCR does not exist, an attempt will be made to create it. The environment variable GAMESS_TMP should be set to a directory in which storage of files between jobs is possible (see below), although it is not used in this simple example.
- The environment variable GAMESS_LIB must be set to the name of a directory containing the GAMESS–UK Library files. (see Accessing Library Files below).

The values of these variables will usually be set as part of the login procedure. On a central computer facility, the values should be provided by the support staff, in other cases the user should edit values into the appropriate login scripts. For users of the C-Shell (/bin/csh) the entry should be made in the script ~/.login. The following is appropriate for user xyz on the DEC AXP 433AU at Daresbury.

```
#
setenv GAMESS_EXE /scr1/wab/GAMESS-UK/bin/gamess
setenv GAMESS_LIB /scr1/wab/GAMESS-UK/libs
setenv GAMESS_TMP /scr1/xyz
setenv GAMESS_SCR /scr1/xyz
```

Users of the Borne shell (/bin/sh) and Korn shell (/bin/ksh) should make the corresponding definitions in the file .profile in their home directory.

```
GAMESS_EXE=/scr1/wab/GAMESS-UK/bin/gamess
export GAMESS_EXE
GAMESS_LIB=/scr1/wab/GAMESS-UK/libs
export GAMESS_LIB
GAMESS_TMP=/scr1/xyz
export GAMESS_TMP
GAMESS_SCR=/scr1/xyz
export GAMESS_SCR
```

2.3 Saving Files

Command line options are used to request that datasets are to be retained beyond the end of the job. The option –k, followed by a GAMESS–UK logical filename (presented in lowercase) will cause the file to be retained in the directory from which the job is run, with a suffix derived from the logical filename. For example;

```
rungamess -k ed3 myjob
```

will save the dumpfile, with filename myjob.ed3.

If control of the filename is required, it may be provided, using the = symbol. The following will associate the file on LFN ed3 with the file oldjob.dump, again in the directory of job submission.

```
rungamess -k ed3=oldjob.dump myjob
```

If the filename supplied in this way is a UNIX relative pathname (ie it does not start with /), it is taken to be relative to the current directory at the time the job is submitted. Otherwise an absolute pathname is assumed, as in the following example.

```
rungamess -k ed3=/scr1/xyz/oldjob.dump myjob
```

This provides a completely general way of assigning the GAMESS–UK datasets, equivalent to the explicit *setenv* commands in the previous chapter. However, it becomes cumbersome if many datasets have to be named in this way. The remaining file options (–t, –r, –n and –l) serve to provide a more concise syntax.

The –t option is identical to –k, except that the default directory is the temporary directory $GAMESS_TMP. Since $GAMESS_TMP will usually refer to a scratch disk, this means that large files may be retained beyond the end of the job, without being written to the users own filestore, where space will usually be limited. On most installations, the variables GAMESS_TMP and GAMESS_SCR will actually be set to the same directory. The distinction is that GAMESS_TMP should point to a directory which will exist beyond the end of the job (since it will contain files required to restart calculations) whereas that referenced by GAMESS_SCR (a subdirectory of $GAMESS_SCR named after the job name) will be deleted after the program has finished.

As an example, consider an installation where GAMESS_SCR has been set to point to a workdirectory /scr2/xyz, and GAMESS_TMP to /scr1/xyz. The *rungamess* invocation

Table 1: Keywords accepted by the –r argument

keyword	function
hf	save ed2 and ed3
casscf	save ed1, ed2, ed3, ed4, ed6, ed9, ed10, ed11
mcscf	save ed2, ed3, ed4, ed6, ed13
ci	save ed2, ed3, ed4, ed5, ed6 (for direct CI calculations)
mrdci	save ed2, ed3, ftn031, ftn033, ftn034, ftn035, ftn036
fullci	save ed2, ed3, ed6, ftn008

```
rungamess -k ed3 -t ed2 myjob
```

will cause a temporary directory /scr2/xyz/myjob to be created, and will contain all datasets used by GAMESS–UK except that associated with LFN ed3, which will have pathname myjob.ed3, and ed2, which will have pathname /scr1/xyz/myjob.ed2.

The way in which saved files are reused is dependent on the preference of the user. If a job needs to be restarted, one option is simply to edit the input file (say to add a restart directive), and re–issue the command given above. This is the simplest method, but obviously results in the input and output files being overwritten.

If it is preferred that the restart job has a different name, it is necessary to ensure that the old files are correctly associated with the new job name. This is the model adopted in this Chapter, because of the need to include all the input files, with unique names, in the examples directory. This can simply be achieved by giving the filename as part of the argument to –k or –t, the following command will re-use the dumpfile ed3 (but not ed2) from the above job.

```
rungamess -k ed3=myjob.ed3 myjob_new
```

If many files are to be carried over between jobs of different names, it is simpler to rename them all using the –name (or –n) option. This sets the first part of the filename for all direct access and Fortran datasets, but leaves the names of the names input, output, punch, and NQS job files unchanged. The following sequence uses "water" as the root part of the filename for both jobs.

```
rungamess -t ed2 -t ed3 -n water water_scf
rungamess -t ed2 -t ed3 -n water water_opt
```

To simplify the file arguments required to allow restarting of a job as a result of time–out the –r argument has been provided. It is followed by a character string specifying the type of calculation for which the restart may be required. Valid key strings, and their effects, are given in Table 1. Clearly, the –r argument must be given on the start–up job, as well as the restart job.

The –r option is equivalent to using a series of –t options, one for each relevant file as given in Table 1. Files saved in this way thus reside on the directory GAMESS_TMP, with names of the form myjob.ed11. Since *rungamess* processes file arguments in the order they are presented, any of the file assignments implied by –r may be overridden if the appropriate option is placed *after* –r.

2.4 Accessing Library Files

The –library argument (short form –l) can be used to associate a logical file name with the corresponding library file. The two LFNs which are accepted are ed0 (for the non-local pseudopotential library file, and table (for the Table–CI database used by the MRDCI module). The directory in which the library files are to be found is obtained from the environment variable GAMESS_LIB.

2.5 Batch Job Submission

Provided the local setup has been performed, the *rungamess* script will submit a job to a batch queuing system if the –q option is provided. A job file with suffix .script is generated and submitted. The exact behaviour of the script and any additional arguments that can be presented will depend on the queuing system in use. Template script files for submission to NQS, PBS and LoadLeveler are provided as part of the distribution. You will need to set the environment GAMESS_SUBMODE to determine which of the submission scripts are chosen. See the file README in the rungamess directory for up-to-date details of the setup required.

2.6 Parallel Job Execution

The –p argument specifies the number of processors to be used to run the job. It will only have an effect on a parallel machine, and relies on the appropriate environment variables having been made to *rungamess* to invoke the parallel code. Execution scripts for execution using TCGMSG/GA, IBM SP (under POE), LAM MPI, and the Cray T3E series are included in the distribution. You will need to set the environment variable GAMESS_PAREXE to the name of the parallel executable, and GAMESS_PARMODE to specify which of the execution scripts to use on the current platform. For interactive use on cluster systems GAMESS_HOSTS should be set to a list of hostnames to be use. See the file README in the rungamess directory for up-to-date details of the setup required.

2.7 Summary of Command Line Arguments

The command line arguments accepted by *rungamess* are tabulated in Table 2. A listing of this information may be obtained on–line by issuing the command

```
rungamess -i
```

which will indicate which options are supported on your local installation.

Table 2: Arguments to rungamess

Argument	Keyword	Function
-k	file	save file in home directory of job
-t	file	save file in temporary directory
-l	file	access library file (file must be ed0 or table)
-n	name	provide root part of name for files (default is the jobname)
-r	jobtype	request save of all files required for restart
-T	time	time for job queue
-q	(none)	place job into an NQS queue
-Q	queue args	place job into an NQS queue (non-default job settings)
-p	# nodes	specify number of nodes for parallel execution
-s	(none)	Provide listing on stdout
-i	(none)	print information

3 GAMESS–UK 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.

3.1 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.

3.2 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.

4 Single Point SCF Calculation

Let us consider a simple SCF run for H_2O in a 3-21G basis, and describe submission using the script *rungamess*.
Assume that the following lines reside in the file myjob.in

```
title\h2o 3-21g scf
zmat angstrom\o\h 1 roh\h 1 roh 2 theta
variables\roh 0.956 hess 0.7\theta 104.5 hess 0.2 \end
enter
```

The command *rungamess* may then be used to run the job in the background

```
rungamess myjob &
```

If batch submission were required, the option -q should be added.

```
rungamess myjob  -q >& myjob.err &
```

This will result in the files used by GAMESS–UK, in this case ED2, ED3 and ED7 being created in the scratch directory for the job, run, and deleted on job termination. The output from the job will be routed to the file myjob.out. System diagnostics will appear at the terminal *stderr*, unless redirected;

```
rungamess myjob >& myjob.err &
```

To retain the ed2 and ed3 files in the scratch directory,

```
rungamess myjob  -t ed2 -t ed3 >& myjob.err &
```

5 Sequence of Calculations on H_2O

In this example we include a sequence of input files and for performing various calculations based on the H_2O example given above. To simplify the commands required, "water" is used as the root of the filename of all binary datasets, using the –n argument.

1. The first job (as above) generates the starting vectors. In this and subsequent jobs, the mainfile (ed2) is saved in case a restart due to lack of time occurs.
   Contents of file water_scf.in:

```
title\h2o 3-21g scf
zmat angstrom\o\h 1 roh\h 1 roh 2 theta
variables\roh 0.956 hess 0.7\theta 104.5 hess 0.2 \end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n water water_scf
```

2. This run utilises the vectors made above. Both bond length and bond angle are to be optimised.
   Contents of file water_opt.in:

```
restart new
title
water optimisation at scf level 3-21g basis set
zmat angstrom
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
runtype optimise
enter
```

   to run:

```
rungamess -t ed2 -t ed3 -n water water_opt
```

3. This example would be used to complete, if necessary, the run started in 1.
   Contents of file water_restopt.in:

```
restart optimise
title
water optimisation at scf level 3-21g basis set
zmat angstrom
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
runtype optimise
enter
```

   to run:

```
rungamess -t ed2 -t ed3 -n water water_restopt
```

4. This example performs an open-shell RHF calculation at the optimised geometry from 2. (note the RESTART usage).
   Contents of file water_orhf.in:

```
restart
title
h2o+ doublet b1 state - ground state geometry
charge 1
mult 2
zmat angs
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
enter
```

   to run:

```
rungamess -t ed2 -t ed3 -n water water_orhf
```

5. The valence shell scf mos are to be localised. For the purpose of the example below it is assumed that orbitals 2 and 3 are the bond orbitals. The CHARGE and MULT directives are required to override the values set in the preceding job on the doublet cation.
   Contents of file water_loc.in:

```
restart
title
lmos for h2o--  ground state geometry
charge 0
mult 1
zmat angs
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
runtype analyse
local
2 to 5
end
vectors 1
enter 5
```

   to run:

```
rungamess -t ed2 -t ed3 -n water water_loc
```

6. The localised orbitals are restored from section 5 and the NOGEN facility used to generate the virtual pairs of the two bond orbitals. The SWAP directive has been used to move the localised bond orbitals to the top of the occupied orbital list. The NOGEN facility reorders the orbitals so that the GVB pairs occur together. Note the ADAPT OFF specification, now required when using localised orbitals as the input orbital set.
   Contents of file water_gvb.in:

```
restart
title
water gvb calculation using localised orbitals
adapt off
zmat ang
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
scftype gvb 2
vectors nogen 5
swap
2 5
3 4
end
enter
```

   to run:

```
rungamess -t ed2 -t ed3 -n water water_gvb
```

6 SCF Calculations on C_4F_4, $C_6H_5NO_2$ and $C_6H_2(NO_2)_3CH_3$

We show below the input files and job submission commands for three straightforward closed-shell SCF calculations, providing more examples of z-matrix specification and reliance on the default options in such calculations. In each case the direct access files will be deleted on job completion.

Closed shell SCF Job for C_4F_4

Contents of file cubane.in:

```
title
**** c4f4  3/21g ****
zmat angs
x
c 1  r1
c 1  r2  2  90.
c 1  r1  3  90.   2 180.
```

```
c 1  r2  4  90.   3 180.
x 2 1.   1 90.    3 0.
f 2 r3   6 90.    3 180.
x 4 1.   1 90.    3 0.
f 4 r3   8 90.    3 180.
x 3 1.   1 90.    4 0.
f 3 r3   10 90.   4 180.
x 5 1.   1 90.    4 0.
f 5 r3   12 90.   4 180.
variables
r1 1.2
r2  1.3
r3 1.313
end
enter
```

to run:

```
rungamess cubane
```

Closed shell SCF Job for $C_6H_5NO_2$

Contents of file nitrobenzene.in:

```
title
c6h5.no2  3-21g
accuracy 20 7
noprint
zmat angstrom
c
n 1 rcn
x 2 1.0 1 90.0
c 1 rcc1 2 t1 3 p1
c 1 rcc1 2 t1 3 -p1
c 4 rcc2 1 t2 2 p2
c 5 rcc2 1 t2 2 p2
c 7 rcc3 5 t3 1 p3
o 2 rno1 1 t5 3 -90.0
o 2 rno1 1 t5 3  90.0
h 4 rch1 1 t6 2 p5
h 5 rch1 1 t6 2 p5
h 6 rch2 4 t7 11 p6
h 7 rch2 5 t7 12 p6
h 8 rch3 7 t8 14 p7
variables
rcn 1.49
rcc1 1.37
rcc2 1.43
rcc3 1.37
rno1 1.21
rch1 1.084
rch2 1.084
rch3 1.084
t1 120.0
```

```
t2 120.0
t3 120.0
t5 120.0
t6 120.0
t7 120.0
t8 120.0
p1 90.0
p2 180.0
p3 0.0
p5 0.0
p6 0.0
p7 0.0
end
maxcyc 20
enter
```

to run:

```
rungamess nitrobenzene
```

Closed shell SCF Job for $C_6H_2(NO_2)_3CH_3$

Contents of file tnt.in:

```
title
2,4,6 tri-nitro-toluene. 3-21g basis.
noprint distance basis vectors hessian
zmat angstrom
x
c    1    r1
c    1    r2    2    a1
c    1    r2    2    a1    3    180.
c    1    r3    2    a2    3      0.
c    1    r3    2    a2    3    180.
x    1    1.    2    90.   3    180.
c    1    r4    7    90.   2    180.
n    3    r5    2    a3    1    180.
o    9    r6    3    a4    2      0.
o    9    r7    3    a5    2    180.
n    4    r5    2    a3    1    180.
o   12    r6    4    a4    2      0.
o   12    r7    4    a5    2    180.
x    8    1.    1    90.   6      0.
n    8    r8   15    90.   1    180.
o   16    r9    8    a6    5      0.
o   16    r9    8    a6    5    180.
x    2    1.    1    90.   4      0.
c    2   r10   19    90.   1    180.
h   20   r11    2    a7   19     90.
h   20   r12    2    a8   21    120.
h   20   r12    2    a8   21   -120.
h    5   r13    8    a9    6    180.
h    6   r13    8    a9    5    180.
variables
```

```
r1 1.431\r2 1.367\r3 1.397\r4 1.395
a1 60.38\a2 120.67
r5 1.521\r8 1.505\r6 1.277\r7 1.278\r9 1.277
a3 123.52\a4 121.19\a5 116.17\a6 117.53
r10 1.529\r11 1.087\r12 1.082
a7 110.07\a8 109.7
r13 1.086\a9 121.26
end
enter
```

to run:

```
rungamess tnt
```

7 STO-3G Calculations on Na_7Mg^+

In this calculation we perform an STO-3G calculation on Na_7Mg^+, followed by an RHF and then UHF calculation on the triplet state. Note the use of the SUPER directive to ensure an integral file format compatible with the use of BYPASS in the subsequent RHF and UHF calculations.

Closed shell SCF Job

Contents of file na7mg_rhf.in:

```
title
* na7mg+ * sto-3g * closed shell * scf-energy= -1314.828516
mult 1
super force
charge 1
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
r1 3.0286740
r2 3.194799
end
basis sto3g
level 1.5   10 1.0
maxcyc 40
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_rhf
```

Open shell RHF Job

Contents of file na7mg_orhf.in

```
restart
title
* na7mg+ * sto-3g * triplet * scf  energy=-1314.900829
super force
bypass
mult 3
charge 1
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
r1 3.0286740
r2 3.194799
end
basis sto3g
maxcyc 40
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_orhf
```

UHF Job

Contents of file na7mg_uhf.in:

```
restart
super force
title
* na7mg+ * sto-3g * triplet * uhf  energy=-1314.901919
mult 3
charge 1
bypass
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
```

```
r1 3.0286740
r2 3.194799
end
basis sto3g
scftype uhf
vectors 5
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_uhf
```

Note that we are using the energy ordered open-shell RHF eigenvectors to initiate the UHF calculation (as written to the default section 5 by the Open shell RHF Job). Had this section not been specified using the VECTORS directive, then the closed-shell SCF MOs would been used in default.

8 Extended Basis Set Calculations of Na_7Mg^+

In this example we use the STO-3G calculation on Na_7Mg^+, performed above as a starting point for a more extensive basis set calculation. In particular the set of closed-shell vectors is restored under control of GETQ, with the STO-3G Dumpfile used as a 'foreign' Dumpfile. We then perform an RHF calculation on the singlet state of Na_7Mg^+, using the integrals calculated in the closed-shell case.

Closed shell SCF Job

Contents of file na7mg_ext.in:

```
dumpfile ed3 500
title
* na7mg+ (4s3p//4s3p1d)  scf energy=-1330.808718
mult 1
charge 1
super off
noprint vectors
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
r1 3.0286740
r2 3.194799
end
```

```
basis
s mg
  .005004   5609.67
  .037083   841.969
  .171495   191.263
  .444597   53.2621
  .480060   16.6003
s mg
  .352170   2.97082
  .692921   1.00728
s mg
  1.00000   .113641
s mg
  1.000000  .044678
p mg
  .039884   50.9665
  .223321   11.4364
  .514536   3.21935
p mg
  1.00000   0.914433
p mg
  1.00000   0.16
d mg
  1.00000   0.175
s na
    .003064  6902.67
    .022198  1059.04
    .095576  255.445
    .280448  77.3172
     .452587 26.8224
    .29313   10.0718
s na
    1.000000  2.17902
s na
    1.000000 .689482
s na
  1.0       .040274
p na
  .042422   38.9438
  .229433   8.71012
  .509774   2.42053
p na
  1.00000   .661896
p na
1.0        .065
end
maxcyc 40
vectors getq ed3 1 1
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_ext
```

Open shell RHF Job

Contents of file na7mg_ext_orhf.in:

```
dumpfile ed3 500
restart
# open shell scf using closed shell vectors from above   *
# bypass integral evaluation                             *
title
* na7mg+ (4s3p//4s3p1d)  triplet rhf *  energy=-1330.766950
super off
bypass
mult 1
charge 1
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
r1 3.0286740
r2 3.194799
end
basis
s mg
  .005004   5609.67
  .037083   841.969
  .171495   191.263
  .444597   53.2621
  .480060   16.6003
s mg
  .352170   2.97082
  .692921   1.00728
s mg
  1.00000   .113641
s mg
  1.000000  .044678
p mg
  .039884   50.9665
  .223321   11.4364
  .514536   3.21935
p mg
  1.00000   0.914433
p mg
  1.00000   0.16
d mg
  1.00000   0.175
s na
    .003064  6902.67
    .022198  1059.04
    .095576  255.445
    .280448  77.3172
```

```
     .452587 26.8224
    .29313   10.0718
s na
    1.000000  2.17902
s na
    1.000000 .689482
s na
  1.0       .040274
p na
  .042422   38.9438
  .229433   8.71012
  .509774   2.42053
p na
  1.00000   .661896
p na
1.0        .065
end
open 1 1 1 1
maxcyc 40
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_ext_orhf
```

9 ECP calculations of Na_7Mg^+

In this example on Na_7Mg^+, we perform a local ECP calculation, using the Hay-Wadt ECP's, together with the associated double zeta basis sets, augmented by a d-function on Mg. Having carried out the closed-shell SCF calculation, we perform an RHF calculation on the singlet state of Na_7Mg^+, using the integrals calculated in the closed-shell case. Note that we are overwriting the files from the previous example.

Closed shell SCF Job

Contents of file na7mg_ecp.in:

```
title
na7mg+ LANL ecp /closed shell singlet
charge 1
super off
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
```

```
variables
r1 3.0286740
r2 3.194799
end
basis
ecpdz na
ecpdz mg
d mg
1.0 0.175
end
ecp
na na
mg mg
level  1.0
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_ecp
```

Open shell RHF Job

Contents of file na7mg_ecp_orhf.in:

```
restart
title
na7mg+ ecp open-shell singlet rhf
mult 1
charge 1
super off
bypass
zmat angs
mg
na 1 r1
na 1 r2 2 90.
na 1 r2 2 90. 3 72.
na 1 r2 2 90. 4 72.
na 1 r2 2 90. 5 72.
na 1 r2 2 90. 6 72.
na 1 r1 3 90. 2 180.
variables
r1 3.0286740
r2 3.194799
end
basis
ecpdz na
ecpdz mg
d mg
1.0 0.175
end
ecp
na na
mg mg
runtype scf
```

```
open 1 1 1 1
level  0.3 1.0
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n na7mg na7mg_ecp_orhf
```

10 Graphical analysis of Ni(CO)$_4$

The following example illustrates features of the Graphical Analysis module, in analysing the ground state SCF wavefunction of Ni(CO)$_4$. Let us assume the following job has been used in constructing this wavefunction.

Contents of file nico4_scf.in:

```
title\ni(co)4 .. 3-21g / SCF total energy  -1947.864822 au
zmat angstrom
ni
c 1 nic
c 1 nic 2 109.471
c 1 nic 2 109.471 3 120.0
c 1 nic 2 109.471 4 120.0
x 2 1.0 1 90.0 3 180.0
o 2 co 6 90.0 1 180.0
x 3 1.0 1 90.0 2 180.0
o 3 co 8 90.0 1 180.0
x 4 1.0 1 90.0 5 180.0
o 4 co 10 90.0 1 180.0
x 5 1.0 1 90.0 4 180.0
o 5 co 12 90.0 1 180.0
variables
nic 1.831
co 1.131
end
level 1.5
enter
```

to run:

```
rungamess -t ed3=nico4.ed3 nico4_scf
```

Examination of the output reveals the following symmetry designation:

```
                                  ******************
                                  MOLECULAR SYMMETRY
                                  ******************

 MOLECULAR POINT GROUP    TD
 ORDER OF PRINCIPAL AXIS   0

 SYMMETRY POINTS :
```

```
POINT 1 :    0.0000000   0.0000000   0.0000000
POINT 2 :    0.0000000   0.0000000   1.0000000
POINT 3 :    0.0000000   1.0000000   0.0000000
```

and the following atomic coordinates:

```
 0.0000000      0.0000000      0.0000000       NI
-1.9976836      1.9976836     -1.9976836       C
 1.9976836     -1.9976836     -1.9976836       C
-1.9976836     -1.9976836      1.9976836       C
 1.9976836      1.9976836      1.9976836       C
-3.2316433      3.2316433     -3.2316433       O
 3.2316433     -3.2316433     -3.2316433       O
-3.2316433     -3.2316433      3.2316433       O
 3.2316433      3.2316433      3.2316433       O
```

The following job may be used to construct a total density plot of the SCF wavefunction in a plane containing the Ni atom and two carbonyl groups, with the Ni at the centre of the plot: a contour plot will be generated on line printer output.

Contents of file nico4_grid.in:

```
restart
punch grid 151
title\ni(co)4 .. 3-21g / SCF total energy  -1947.864822 au
zmat angstrom
ni
c 1 nic
c 1 nic 2 109.471
c 1 nic 2 109.471 3 120.0
c 1 nic 2 109.471 4 120.0
x 2 1.0 1 90.0 3 180.0
o 2 co 6 90.0 1 180.0
x 3 1.0 1 90.0 2 180.0
o 3 co 8 90.0 1 180.0
x 4 1.0 1 90.0 5 180.0
o 4 co 10 90.0 1 180.0
x 5 1.0 1 90.0 4 180.0
o 5 co 12 90.0 1 180.0
variables
nic 1.831
co 1.131
end
runtype analyse
graphics
gdef
type 2d
title
square 2d grid ni(co)4 - total density
calc
type dens
title
ni(co)4 - total density
section 151
```

```
plot
type line
title
ni(co)4 - total density
vectors 1
enter
```

to run:

```
rungamess -t ed3=nico4.ed3 nico4_grid
```

The grid data will appear in the punchfile nico4_grid.pun

11 2-Pair GVB Calculation on $CH_2(CH)_4NH_2^+$

Closed shell SCF Job

Contents of file imino_rhf.in:

```
# open shell rhf sto3g                                        *
# check super force                                           *
title
1-imino-2,4-pentadiene *  energy= -245.073475
super force nosym
mult 1
accuracy 20 7
charge 1
zmat angs
c
c 1 r1
c 2 r2 1 a1
c 3 r3 2 a2 1 cx
c 4 r4 3 a3 2 c2
n 1 r5 2 a4 3 c2
h 1 r6 2 a5 3 c1
h 2 r7 3 a6 7 c2
h 3 r8 4 a7 8 c2
h 4 r9 5 a8 9 c2
h 5 r10 4 a9 10 c1
h 5 r11 4 a10 11 c2
h 6 r12 1 a11 7 c1
h 6 r13 1 a12 13 c2
variables
r1     1.3463261
r2     1.4632685
r3     1.3961480
r4     1.3568062
r5     1.3416773
r6     1.0748794
r7     1.0669518
r8     1.0811641
r9     1.0707072
```

```
r10    1.0729928
r11    1.0747118
r12    0.9971261
r13    0.9985437
a1   118.2539432
a2   122.8419092
a3   119.3229511
a4   125.9857684
a5   119.9766875
a6   122.1120978
a7   116.3317717
a8   120.9014063
a9   121.5606571
a10  121.8124933
a11  120.9303349
a12  121.8324395
constants
c1 0.
c2 180.
cx  90.
end
basis sto3g
runtype scf
scftype rhf
open 1 1 1 1
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n imino imino_rhf
```

Two-Pair GVB Job

Contents of file imino_gvb.in:

```
# gvb-scf  with two pairs                                    *
# bypass integral evaluation                                 *
restart
bypass
title
1-imino-2,4-pentadiene *   energy=-245.114661
mult 1
super force nosym
accuracy 20 7
charge 1
zmat angs
c
c 1 r1
c 2 r2 1 a1
c 3 r3 2 a2 1 cx
c 4 r4 3 a3 2 c2
n 1 r5 2 a4 3 c2
h 1 r6 2 a5 3 c1
h 2 r7 3 a6 7 c2
```

```
h 3 r8 4 a7 8 c2
h 4 r9 5 a8 9 c2
h 5 r10 4 a9 10 c1
h 5 r11 4 a10 11 c2
h 6 r12 1 a11 7 c1
h 6 r13 1 a12 13 c2
variables
r1     1.3463261
r2     1.4632685
r3     1.3961480
r4     1.3568062
r5     1.3416773
r6     1.0748794
r7     1.0669518
r8     1.0811641
r9     1.0707072
r10    1.0729928
r11    1.0747118
r12    0.9971261
r13    0.9985437
a1   118.2539432
a2   122.8419092
a3   119.3229511
a4   125.9857684
a5   119.9766875
a6   122.1120978
a7   116.3317717
a8   120.9014063
a9   121.5606571
a10  121.8124933
a11  120.9303349
a12  121.8324395
constants
c1 0.
c2 180.
cx  90.
end
basis sto3g
runtype scf
scftype gvb 2
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n imino imino_gvb
```

12 Direct-SCF Calculations

In the first example below we show a direct-SCF calculation in which the input geometry in cartesian coordinates is converted to z–matrix representation; the second calculation features geometry optimisation of $Be(C_5H_5)_2$ using the direct-SCF module.

Contents of file direct_scf.in:

```
# direct scf generate zmatrix sto3g-basis             *
title
* test2a * energy=-1550.28679356
charge -2
geometry  distance angles torsions       all
-2.9512196   -0.1547624    -2.3287565    8.0    o
-1.0815728   -1.9700376    -1.5385361    6.0    c
-1.1603816   -3.5363478    -2.8467927    1.0    h
 1.6367703   -0.8475562    -1.5624978    6.0    c
 2.1566895   -0.4797763    -3.4976155    1.0    h
 3.5011251   -2.6669663    -0.4827273    7.0    n
 2.9474874   -3.6433853     1.0782554    1.0    h
 1.7359097    1.7242063    -0.1090391    6.0    c
 1.3150929    1.4125382     1.8702785    1.0    h
 4.1957289    2.9182102    -0.4054688    8.0    o
-0.3417469    3.5022051    -1.1895255    6.0    c
 0.0532604    3.8272799    -3.1681491    1.0    h
-0.3292201    5.9011224     0.0783133    8.0    o
-1.4442971    7.1126500    -0.6802213    1.0    h
-2.9721287    2.1997957    -0.9534701    6.0    c
-3.3279629    1.8049798     1.0197182    1.0    h
-5.1750163    3.9122805    -1.8751357    6.0    c
-4.8159917    4.4971112    -3.8035825    1.0    h
-6.8815215    2.7942026    -1.8646838    1.0    h
-5.4889849    6.1460386    -0.3379417    8.0    o
-7.5612331    6.0739323     1.4363625    6.0    c
-7.4278090    4.3945431     2.5896385    1.0    h
 5.7357642    3.1285958     1.6502323    6.0    c
 5.5713092    1.7094915     3.4875662    8.0    o
 7.8017770    5.1679189     1.7463198    6.0    c
 7.0224823    6.8328410     2.6479587    1.0    h
 8.3530583    5.6784964    -0.1498644    1.0    h
 5.9321304   -2.9668945    -1.2970693    6.0    c
 6.7897779   -1.7121829    -3.0617741    8.0    o
 7.7021385   -4.8380024     0.0564767    6.0    c
 7.1302965   -6.7576698    -0.3499893    1.0    h
 7.6050313   -4.5640081     2.0812894    1.0    h
-1.5715523   -2.9124352     0.9658559    8.0    o
-3.9184431   -4.8124121     0.9101853   15.0    p
-3.1659952   -7.1913356    -0.3438675    8.0    o
-6.0342521   -3.6456571    -0.4943459    8.0    o
-4.4900093   -5.1460285     3.8664887    8.0    o
 9.4719556   -4.5440252    -0.5381892    1.0    h
 9.3293820    4.5891096     2.6957586    1.0    h
-9.1872029    6.0381561     0.4734880    1.0    h
-7.5522104    7.5798978     2.5757500    1.0    h
end
basis  sto3g
scftype direct
enter
```

to run:

```
rungamess direct_scf
```

Direct-SCF Geometry Optimisation of $Be(C_5H_5)_2$

Contents of file berylocene_opt.in:

```
time 60
title
be(c5h5)2 sto3g optimised total energy  = -394.279009 au
zmatrix angstrom
x
x  1  fxa
c  2  xc  1  xxc
c  2  xc  1  xxc  3  cxc
c  2  xc  1  xxc  4  cxc
c  2  xc  1  xxc  3  -cxc
c  2  xc  1  xxc  6  -cxc
x  2  xx  3  xxc  4  -xxc
h  2  hx  8  hxx  3  hcx
h  2  hx  8  hxx  4  hcx
h  2  hx  8  hxx  5  hcx
h  2  hx  8  hxx  6  hcx
h  2  hx  8  hxx  7  hcx
x  2  fxt 3  xxc  4  xxc
c 14  xc  1  xxc  3  cxxc
c 14  xc  1  xxc 15  cxc
c 14  xc  1  xxc 16  cxc
c 14  xc  1  xxc 15  -cxc
c 14  xc  1  xxc 18  -cxc
x 14  xx  15 xxc 16  -xxc
h 14  hx  20 hxx 15  hcx
h 14  hx  20 hxx 16  hcx
h 14  hx  20 hxx 17  hcx
h 14  hx  20 hxx 18  hcx
h 14  hx  20 hxx 19  hcx
be 2  fxa  3 xxc  4  xxc
variables
fxa   1.47
fxt   3.37
xc    1.22
hx    2.12
hxx  88.4
constants
cxc  72.0
cxxc  36.0
xxc  90.0
hcx   0.0
xx    1.0
end
basis sto3g
runtype optimize
scftype direct
level 2.0 10 1.4
enter
```

to run:

```
rungamess berylocene_opt
```

13 HCN/HNC Transition State Location

Transition state calculation for the HCN,HNC isomerisation process. The first job uses the default trust region algorithm, the second the synchronous transit algorithm, and the third the Jorgensen-Simons algorithm.

Contents of file hcn_tr.in:

```
title
hcn 4-31G saddle point
zmat angs
c
x 1 1.0
n 1 cn 2 90.0
h 1 ch 2 90.0 3 hcn
variables
cn 1.1484 type 3
ch 1.5960 type 3
hcn 90.0 type 3
end
basis 4-31g
runtype saddle
enter
```

to run:

```
rungamess hcn_tr
```

Saddle point for HCN using the synchronous transit algorithm - note the definition of the minima required (on the variable definition lines) and the LSEARCH directive. The default saddle point method does nor require minima definition (see above)

Contents of file hcn_st.in:

```
title
hcn saddle point - synchronous transit
zmat angs
c
x 1 1.0
n 1 cn 2 90.0
h 1 ch 2 90.0 3 hcn
variables
cn 1.1484 minima 1.1371 1.1597
ch 1.5960 minima 1.0502 2.1429
hcn 90.0   minima 180.0  0.0
end
basis 4-31g
runtype saddle
lsearch 0 4
enter
```

to run:

```
rungamess hcn_st
```

Saddle point for HCN using the Jorgensen-Simons algorithm.
Contents of file hcn_js.in:

```
title
hcn/hnc ts search . jorgensen-simons
zmat angs
c
x 1 1.0
n 1 cn 2 90.0
h 1 ch 2 90.0 3 hcn
variables
cn 1.1484 type 3
ch 1.5960 type 3
hcn 90.0   type 3
end
basis 4-31g
runtype saddle jorgensen
powell
maxjor 55
recalc off
rfo off
cutoffs
optprint on
xtol 0.0018
enter
```

to run:

```
rungamess hcn_js
```

14 HSiP/HPSi Transition State Location

This example is concerned with locating the transition state in the HPSi, HSiP isomerisation process, and calculating the associated vibrational frequencies. We provide sample jobs using both numerical and analytical techniques in the transition state location and subsequent force constant evaluation. Note that latter example is computationally the most efficient, and should certainly be adopted for small–medium sized molecules.

14.1 Numerical Force Constants

In the first step we perform an initial SCF for subsequent use in the saddle point calculation.

Closed shell SCF Job

Contents of file hsip_scf.in:

```
title
psih saddle point
```

```
zmat ang
p
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053 type 3
ph  2.44  type 3
hpsi  51.02 type 3
end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_scf
```

In the subsequent location of the transition state, note the use of TYPE 3 which causes the program to calculate the complete force constant matrix numerically before commencing the search for the saddle point, and the use of XTOL to provide more stringent optimisation criteria in view of the subsequent force constant evaluation. LOCK is used to retain the initial SCF configuration throughout the search.

Transition State Job

Contents of file hsip_ts.in:

```
restart new
title
psih <-> hpsi saddle point
bypass
zmat angs
p
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053 type 3
ph  2.44  type 3
hpsi  51.02 type 3
end
runtype saddle
lock
xtol 0.0005
enter 2
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_ts
```

Finally we present the job for numerical evaluation of the force constants at the optimised geometry. Note the use of restart in requesting usage of the geometry from the Dumpfile, rather

than from the data file.

Numerical Force Constant Job

Contents of file hsip_fc.in:

```
restart
title
psih <-> hpsi saddle point numerical fcm
zmat angs
p
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053 type 3
ph  2.44  type 3
hpsi  51.02 type 3
end
runtype force
vectors 2
lock
enter 3
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_fc
```

14.2 Analytic Force Constants

In the first step we perform the computation of the trial hessian under RUNTYPE HESSIAN control for subsequent use in the saddle point calculation.

Computing the trial Hessian

Contents of file hsip_fcm1.in:

```
title
psih trial hessian / SCF
zmat ang
p
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053
ph  2.44
hpsi  51.02
end
runtype hessian
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_fcm1
```

In the subsequent location of the transition state, note the use of the FCM keyword on the RUNTYPE data line to restore the trial hessian computed in the first job and the use of XTOL to provide more stringent optimisation criteria in view of the subsequent force constant evaluation. LOCK is used to retain the initial SCF configuration throughout the search.

Transition State Job

Contents of file hsip_tsfcm.in:

```
restart new
title
psih <-> hpsi saddle point location / using trial hessian
zmat angs
p
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053
ph  2.44
hpsi  51.02
end
runtype saddle fcm
xtol 0.0005
vectors 1
lock
enter 2
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_tsfcm
```

Finally we present the job for analytic computation of the force constants at the optimised geometry under control of runtype hessian. Note the use of restart in requesting usage of the geometry from the Dumpfile, rather than from the data file.

Analytic Force Constant Job

Contents of file hsip_fcm2.in:

```
restart
title
psih <-> hpsi saddle point / force constants
zmat angs
p
```

```
x 1 1.0
si 1 psi 2 90.0
h 1 ph 2 90.0 3 hpsi
variables
psi 2.053
ph  2.44
hpsi  51.02
end
runtype hessian
vectors 2
lock
enter 2
```

to run:

```
rungamess -t ed2 -t ed3 -n hsip hsip_fcm2
```

15 Use of Bond-centred Functions

In this example we demonstrate the use of bond-centred functions (s,p), cited at the mid-point of the C-N bond in the HCN, HNC transition state.

Contents of file hcn_bf.in:

```
title
hcn-hnc basis - dunning (9s5p-3s2p) bond(s,p) + p(h)
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 phi
variables
rcn2 0.5991
rch 1.2128
phi 71.2
end
basis
sv h
p h
1.0 0.7
s bq
1.0 1.0
p bq
1.0 0.7
sv c
sv n
end
enter
```

to run:

```
rungamess hcn_bf
```

16 SCF Analytic Force Constants for C_2H_4

We consider below computing the analytic force constants for C_2H_4, initially optimising the molecule at the SCF level, followed by the force constant calculation. Note the use of the XTOL directive in the optimisation job to ensure a higher degree of optimisation than that derived using the default XTOL.

Geometry Optimisation

Contents of file ethene_opt.in:

```
title
ethylene 6-31g**  geometry optimisation
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
hcc 118.0
end
basis 6-31g**
runtype optimize
xtol 0.0001
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_opt
```

Analytic Force Constants

Contents of file ethene_fcm.in:

```
restart
title
ethylene 6-31g** ground state vibrational frequencies
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
```

```
hcc 118.0
end
basis 6-31g**
runtype hessian
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_fcm
```

17 MP2 Analytic Force Constants for C_2H_4

We consider below computing the analytic force constants for C_2H_4, initially optimising the molecule at the MP2 level, followed by the force constant calculation. Note again the use of the XTOL directive in the optimisation job to ensure a higher degree of optimisation than that derived using the default XTOL.

MP2 Geometry Optimisation

Contents of file ethene_mp2opt.in:

```
title
ethylene 6-31g**  MP2/ optimised total energy  =  -78.3272309
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
hcc 118.0
end
basis 6-31g**
runtype optimize
scftype mp2
xtol 0.0001
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_mp2opt
```

MP2 Analytic Force Constants

Contents of file ethene_mp2fcm.in:

```
restart
title
ethylene MP2/6-31g** ground state vibrational frequencies
#freq    847.7,  941.6,  994.1, 1091.6, 1267.4,  1414.5,
#freq   1525.2, 1729.7, 3241.3, 3259.2, 3336.6,  3359.8
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
hcc 118.0
end
basis 6-31g**
runtype hessian
scftype mp2
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_mp2fcm
```

18 MP2 Polarisability for C_2H_4

We consider below computing the molecular polarisability of C_2H_4, initially optimising the molecule at the MP2 level, followed by the property calculation. Note again the use of the XTOL directive in the optimisation job to ensure a higher degree of optimisation than that derived using the default XTOL.

MP2 Geometry Optimisation

Contents of file ethene_mp2opt.in:

```
title
ethylene 6-31g**  MP2/ optimised total energy  =  -78.3272309
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
hcc 118.0
```

```
end
basis 6-31g**
runtype optimize
scftype mp2
xtol 0.0001
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_mp2opt
```

MP2 Polarisability

Contents of file ethene_mp2pol.in:

```
restart
title
ethylene MP2/6-31g** ground state polarisability
zmatrix angstrom
c
c 1 cc
h 1 ch 2 hcc
h 1 ch 2 hcc 3 180.0
h 2 ch 1 hcc 3 0.0
h 2 ch 1 hcc 3 180.0
variables
cc 1.40
ch 1.10
hcc 118.0
end
basis 6-31g**
runtype polarisability
scftype mp2
enter
```

to run:

```
rungamess -t ed3=ethene.ed3 ethene_mp2pol
```

19 Direct-MP2 Calculation of C_5H_5N

We show below the data for performing a direct–MP2 calculation on the C_5H_5N molecule, conducted in a 6–31G* basis set. Note the use of the MEMORY pre-directive in requesting a memory allocation of 4 MWords.

Contents of file pyridine.in:

```
title
pyridine 6-31g* direct-mp2
zmat angstrom
n
```

```
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
c2c3h 120.158516
end
basis 6-31g*
scftype direct mp2
enter
```

to run:

```
rungamess -t ed3 pyridine
```

20 CASSCF Geometry Optimisations

We consider below a CASSCF calculation on the X^1A_1 state of H_2O, using a full valence criterion in specifying the active space so that the formally vacant SCF virtual MOs, $4a_1$ and $2b_2$, are permitted variable occupancy. This example utilises the vectors from the closed shell SCF calculation of Example 1.

Contents of file water_cas.in

```
restart new
title
water  at casscf level 3-21g basis set
zmat angstrom
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
```

```
hoh 104.5
end
scftype casscf
config print
doc 1 to 5
uoc 6 7
end
superci 1 to 8
newton 9 to 20
hessian 9 to 20
simul 9 to 20
enter
```

to run:

```
rungamess -t ed3 -n water water_cas
```

The following points should be noted:

- It is not possible to use BYPASS in the above, given the data for the SCF job of Example 1. This would have resulted in generation of a P-supermatrix which is not usable in a CASSCF run (see Part 2, Table 1).
- CASSCF calculations require two scratch FORTRAN data sets, FT01 and FT02.

In the above we have assumed that the CASSCF calculation completes in the time allocated, with the associated direct-access files allocated in default scratch status. The following *rungamess* invocation is typical of that required if restarts of the CASSCF step are envisaged, here we are using the –r argument, which will cause all datasets required for a casscf (scf or optimise) restart calculation (specifically ed1 ed2 ed4 ed6 ed9 ed10, and ed11) to be saved.

```
rungamess -r casscf -n water water_cas
```

Assuming the above job terminated prior to convergence, the calculation might be restarted as follows:
Contents of file water_cas_rest.in:

```
restart scf
title
water at casscf level 3-21g basis set
zmat angstrom
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
scftype casscf
config bypass
```

```
doc 1 to 5
uoc 6 7
end
superci 1 to 2
newton 3 to 20
hessian 3 to 20
simul 3 to 20
vectors 6 7
enter 6 7
```

to run:

```
rungamess -r casscf -n water water_cas_rest
```

where the default sections housing the CASSCF vectors and ci coefficients (sections 6 and 7 respectively) created in the startup job are explicitly declared above, and CONFIG processing is bypass'ed. Having completed the single point calculation, the following might be used to perform a geometry optimisation at the CASSCF level.

Contents of file water_cas_opt.in:

```
restart new
title
water - geometry optimisation at casscf level
zmat angstrom
o
h 1 oh
h 1 oh 2 hoh
variables
oh 0.956
hoh 104.5
end
runtype optimise
scftype casscf
config bypass
doc 1 to 5
uoc 6 7
end
superci 1 to 5
newton 6 to 20
hessian 6 to 20
simul 6 to 20
enter
```

to run:

```
rungamess -r casscf -n water water_cas_opt
```

21 CASSCF + 2nd-order CI Calculations on BeO

First, we consider below a CASSCF calculation on the $X^1\Sigma^+$ state of BeO, characterised by the configuration

$$1\sigma^2 2\sigma^2 3\sigma^2 4\sigma^2 1\pi^4 \quad (1)$$

The initial closed–shell SCF using a DZP basis was conducted with the following job:
Contents of file beo_rhf.in:

```
super off nosym
title\beo .. dzp
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n beo beo_rhf
```

An examination of the closed-shell SCF output reveals the following symmetry adapted basis information

```
============================
IRREP  NO. OF SYMMETRY ADAPTED
       BASIS FUNCTIONS
============================
  1          17
  2           6
  3           6
  4           2
============================
```

and the SCF MO ordering shown below:

```
=============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
=============================================
  1     1    -20.45769692          2.0000000
  2     1     -4.72825831          2.0000000
  3     1     -1.15792230          2.0000000
  4     1     -0.46629250          2.0000000
  5     3     -0.39257378          2.0000000
  6     2     -0.39257378          2.0000000
  7     1     -0.05704423          0.0000000
  8     2      0.09936415          0.0000000
  9     3      0.09936415          0.0000000
 10     1      0.15573442          0.0000000
 11     1      0.25156032          0.0000000
 12     3      0.29518660          0.0000000
 13     2      0.29518660          0.0000000
 14     1      0.57290305          0.0000000
 15     4      0.66957865          0.0000000
 16     1      0.66957865          0.0000000
 17     2      0.84348942          0.0000000
 18     3      0.84348942          0.0000000
 19     1      1.01733643          0.0000000
 20     2      1.05237116          0.0000000
 21     3      1.05237116          0.0000000
 22     1      1.24301610          0.0000000
```

```
 23     1      1.45883277           0.0000000
 24     1      1.80402496           0.0000000
 25     4      2.36534068           0.0000000
 26     1      2.36534068           0.0000000
 27     2      2.62307878           0.0000000
 28     3      2.62307878           0.0000000
 29     1      3.16009549           0.0000000
 30     1      4.37857805           0.0000000
 31     1     45.42979631           0.0000000
================================================
```

We wish to perform a CASSCF calculation in which the inner shell and O2s orbitals (the 1σ–3σ) remain doubly occupied, with the active space including the formally vacant SCF virtual MOs, the 5σ and 2π. This example utilises the vectors from the closed shell SCF calculation. We wish to perform the CASSCF calculation under RUNTYPE CI specification where, having performed the 6 electrons in 6 orbital CASSCF, we use the natural orbitals in carrying out a second-order CI using the Direct-CI module. Specifically we aim to use a reference space in the CI consisting of all CSFs which can be generated by distributing 6 electrons in 6 MOs i.e. the CASSCF space. This may be achieved in a single run through the following job specification:

Contents of file beo_casscf_ci.in:

```
restart new
title\beo .. dzp casscf+ci (6 electrons in 6 mos)
bypass
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
runtype ci
active\4 to 31\end
core\1 to 3\end
scftype casscf
thresh 4
config print
fzc 1 to 3
doc 4 to 6
uoc 7 to 9
end
superci 1 to 8
newton 9 to 20
hessian 9 to 20
simul 9 to 20
direct 6 6 22
conf
2 2 2 0 0 0
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
```

```
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n beo beo_casscf_ci
```

If restarts are envisaged, the following invocation would be suitable:

```
rungamess -r ci -r casscf -n beo beo_casscf_ci
```

22 MCSCF + 2nd-order CI Calculations on BeO

First, we consider below a MCSCF calculation on the $X^1\Sigma^+$ state of BeO, characterised by the configuration

$$1\sigma^2 2\sigma^2 3\sigma^2 4\sigma^2 1\pi^4 \tag{2}$$

The initial closed–shell SCF using a DZP basis was conducted with the following job:
Contents of file beo_rhf.in:

```
super off nosym
title\beo .. dzp
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n beo beo_rhf
```

An examination of the closed-shell SCF output reveals the following symmetry adapted basis information

```
============================
IRREP  NO. OF SYMMETRY ADAPTED
       BASIS FUNCTIONS
============================
  1          17
  2           6
  3           6
  4           2
============================
```

and the SCF MO ordering shown below:

```
=============================================
M.O.  IRREP  ORBITAL ENERGY   ORBITAL OCCUPANCY
=============================================
  1     1    -20.45769692           2.0000000
  2     1     -4.72825831           2.0000000
  3     1     -1.15792230           2.0000000
  4     1     -0.46629250           2.0000000
  5     3     -0.39257378           2.0000000
  6     2     -0.39257378           2.0000000
  7     1     -0.05704423           0.0000000
  8     2      0.09936415           0.0000000
  9     3      0.09936415           0.0000000
 10     1      0.15573442           0.0000000
 11     1      0.25156032           0.0000000
 12     3      0.29518660           0.0000000
 13     2      0.29518660           0.0000000
 14     1      0.57290305           0.0000000
 15     4      0.66957865           0.0000000
 16     1      0.66957865           0.0000000
 17     2      0.84348942           0.0000000
 18     3      0.84348942           0.0000000
 19     1      1.01733643           0.0000000
 20     2      1.05237116           0.0000000
 21     3      1.05237116           0.0000000
 22     1      1.24301610           0.0000000
 23     1      1.45883277           0.0000000
 24     1      1.80402496           0.0000000
 25     4      2.36534068           0.0000000
 26     1      2.36534068           0.0000000
 27     2      2.62307878           0.0000000
 28     3      2.62307878           0.0000000
 29     1      3.16009549           0.0000000
 30     1      4.37857805           0.0000000
 31     1     45.42979631           0.0000000
=============================================
```

We wish to perform a CASSCF calculation in which the inner shell and O2s orbitals (the 1σ–3σ) remain doubly occupied, with the active space including the the formally vacant SCF virtual MOs, the 5σ and 2π. This example utilises the vectors from the closed shell SCF calculation.

Contents of file beo_mcscf.in:

```
restart new
super off nosym
title\beo .. dzp
bypass
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
scftype mcscf
thresh 4
mcscf
orbital\3cor1 doc1 doc3 doc2  uoc1 uoc2 uoc3 \end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n beo beo_mcscf
```

The following points should be noted:

- The MCSCF Natural orbitals are routed to section 10 of the Dumpfile on convergence, the default section used for NO output.
- Integral evaluation has been bypassed as the initial SCF job specified the necessary integral format for the subsequent SCF.

In the above we have assumed that the MCSCF calculation completes in the time allocated, with the associated direct-access files allocated in default scratch status. The following *rungamess* invocation is typical of that required if restarts of the MCSCF step are envisaged:

```
rungamess -r mcscf -n beo beo_mcscf
```

Note that the symmetry adapted list of integrals are sorted at the outset of processing to ED13, and this file should be preserved across restart jobs, given that the DONT SORT data line is presented to the restart job. Assuming the above job terminated prior to convergence, the calculation might be restarted as follows:

Contents of file beo_mcscf_rest.in

```
restart scf
super off nosym
title\beo .. dzp
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
scftype mcscf
thresh 4
mcscf
orbital\3cor1 doc1 doc3 doc2  uoc1 uoc2 uoc3 \end
dont sort
enter
```

to run:

```
rungamess -r mcscf -n beo beo_mcscf_rest
```

Now let us consider performing the MCSCF calculation under RUNTYPE CI specification where, having performed the 6 electrons in 6 orbital CASSCF, we use the natural orbitals in carrying out a second-order CI using the Direct-CI module. Specifically we aim to use a reference space in the CI consisting of all CSFs which can be generated by distributing 6 electrons in 6 MOs i.e. the CASSCF space. This may be achieved in a single run through the following job specification:

Contents of file beo_mcscf_ci.in

```
restart new
super off nosym
```

```
title\beo .. dzp mcscf+2nd-order ci (6 electrons in 6 mos)
zmat angstrom\be\o 1 beo\
variables\beo 1.300 hessian 0.7\end
basis dzp
runtype ci
active\4 to 31\end
core\1 to 3\end
scftype mcscf
thresh 4
mcscf
orbital\3cor1 doc1 doc3 doc2  uoc1 uoc2 uoc3 \end
direct 6 6 22
conf
2 2 2 0 0 0
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
refgen
1 4 1 5 1 6  2 4 2 5 2 6  3 4 3 5 3 6
vectors 1\enter 20 21
```

to run:

```
rungamess -t ed3 -n beo beo_mcscf_ci
```

Note that the vectors specification is requesting that the closed-shell SCF eigenvectors be used to initiate the MCSCF calculation. The MCSCF natural orbitals, routed to section 20 of the Dumpfile, will be used as the orbitals for the Direct-CI calculation.

23 Table-CI calculations on the Ammonia Cation

We consider below a Table-CI calculation of the X^2A_1 state and 1^2A_1 state of the ammonia cation. In the first instance we consider performing the calculation in two steps, initially the open-shell SCF calculation followed by the MRD-CI 2-reference state calculation. We then subdivide the CI calculation into 5 separate steps, performing the symmetry adaptation + integral transformation, followed by configuration selection, construction of the CI Hamiltonian, the diagonalisation and, finally, the subsequent analysis of the CI wavefunctions.

Open-shell SCF Job

Contents of file nh3_rohf:

```
title
```

```
nh3+ * 3-21g * scf-energy=-55.53325817 hartree
super off nosym
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n nh3 nh3_rohf
```

Table-CI Job

Contents of file nh3_mrdci.in:

```
restart
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt
tran
table
select
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci
diag
```

```
extrap 3
dthr 0.0001 0.0001
natorb
cive 1 2
prop
cive 1 2
1 4  1 2 3  12
1 3  1 2 4  12
moment
36 1 36 2 1
enter
```

to run:

```
rungamess -t ed2 -t ed3 -t table -n nh3 nh3_mrdci
```

The orbitals employed in the CI calculation will be taken from the default section associated with the open-shell RHF module, section 5, that containing the energy-ordered canonicalised open-shell vectors written on termination of the SCF process. Note that the table keyword will activate generation of the Table-CI data base, to be used in the subsequent steps below. In this case the data base will be written to the file nh3.table in the temporary directory for subsequent use. Note that on most installations a copy of the data base will already be available and some time is saved if recomputation is avoided. It is accessed with the –library (or –l) option on the *rungamess* command line. In this case the table keyword must be removed.

```
rungamess -t ed2 -t ed3 -l table -n nh3 nh3_mrdci
```

Now let us consider dividing the above CI calculation. The following points should be noted in this division:

- the "-r mrdci" option has been used to save all files that must be retained between separate runs of the program. The option "-t ed2" is now implicit and so may be omitted. We assume below that the Table-CI data-base is available from the previous job, thus omitting the TABLE step.
- Note the use of the BYPASS keyword on the various steps comprising the Table-CI procedure. Such a keyword is required on *both* those steps already completed and those steps to be handled in a subsequent run of the program.

Table-CI Data I. Symmetry Adaption and Integral Transformation

Contents of file nh3_sa_tran.in:

```
restart
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
```

```
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt
tran
select bypass
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci bypass
diag bypass
extrap 3
dthr 0.0001 0.0001
enter
```

to run:

```
rungamess -r mrdci -t table -n nh3 nh3_sa_tran
```

Table-CI Job II. Configuration Selection

Contents of file nh3_select.in:

```
restart ci
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt bypass
```

```
tran bypass
select
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci bypass
diag bypass
extrap 3
dthr 0.0001 0.0001
enter
```

to run:

```
rungamess -r mrdci -t table -n nh3 nh3_select
```

Table-CI Job III. CI Hamiltonian Construction

Contents of file nh3_hamil.in:

```
restart ci
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt bypass
tran bypass
select  bypass
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci
diag bypass
```

```
extrap 3
dthr 0.0001 0.0001
enter
```

to run:

```
rungamess -r mrdci -t table -n nh3 nh3_hamil
```

Table-CI Job IV. Diagonalisation

Contents of file nh3_diag.in:

```
restart ci
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt bypass
tran bypass
select bypass
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci bypass
diag
extrap 3
dthr 0.0001 0.0001
enter
```

to run:

```
rungamess -r mrdci -t table -n nh3 nh3_diag
```

Table-CI Job V. CI Wavefunction Analysis

Contents of file nh3_analy.in:

```
restart ci
title
* nh3+ * 3-21g * mrdci-energies 1r -55.6393336 2r -55.4116210
bypass scf
charge 1
mult 2
zmat angstrom
n
h 1 roh
h 1 roh 2 theta
h 1 roh 2 theta 3 theta  1
variables
roh 1.03   hessian 0.7
theta 104.2  hessian 0.2
end
runtype ci
mrdci
adapt bypass
tran  bypass
select bypass
symmetry 1
spin 2
cntrl 9
conf
1 4 1 2 3  12
1 3 1 2 4  12
roots 2
thresh 5 5
ci bypass
diag  bypass
extrap 3
dthr 0.0001 0.0001
natorb
cive 1 2
prop
cive 1 2
1 4  1 2 3  12
1 3  1 2 4  12
moment
36 1 36 2 1
enter
```

to run:

```
rungamess -r mrdci -t table -n nh3 nh3_analy
```

24 ECP, CASSCF and Direct-CI Calculations on $NiCCH_2$

This example illustrates the use of CASSCF and Direct-CI calculations in the framework of ECP studies. The molecular system under investigation is $NiCCH_2$, with a 5-reference direct-CI calculation performed using a CASSCF wavefunction for the lowest triplet state. Five data files are presented below:

1. Start–up closed–shell SCF calculation for the $^1A_1$ state. Note the SUPER directive for compatibility with the subsequent open–shell calculation. The ECP library file /scr1/gamess/GAMESS-UK/libs/ecplib (on the HP735 at Daresbury) is allocated to LFN ed0, with the NIHAY and C non-local ECPs requested under control of the PSEUDO directive.
2. Restart SCF job, with appropriate use of the SWAP directive to converge the closed–shell SCF.
3. RHF calculation for the $^3A_1$ state.
4. CASSCF calculation for the $^3A_1$ state.
5. 5-reference Direct-CI calculation for the $^3A_1$ state.

Closed-shell SCF Start-up Job

Contents of file nicch2_rhf.in

```
title
ni(cch2) 1a1 rhf ,hay's ni, bar's nm ecp's
mult 1
super force
zmat angstrom
ni
c 1 nica
x 2 1.0 1 90.0
c 2 cacb 3 90.0 1 180.0
x 4 1.0 2 90.0 3 0.0
h 4 hcb 2 hcc 5 90.0
h 4 hcb 2 hcc 5 -90.0
variables
nica 2.0895
cacb 1.3604
hcb 1.1047
hcc 122.646
end
basis
s h
0.032828     13.3615
0.231208      2.0133
0.817238      0.4538
s h
1.000000      0.1233
s c
1.000000      0.4962
s c
1.000000      0.1533
p c
0.018534     18.1557
0.115442      3.9864
0.386206      1.1429
0.640089      0.3594
p c
```

```
1.000000      0.1146
s ni
-0.4372528 0.6778
1.1889453 0.1116
s ni
1.0000000 0.0387
p ni
1.0000000 0.0840
p ni
1.0000000 0.0240
d ni
0.0360414 42.7200
0.1938645 11.7600
0.4596238 3.8170
0.5599305 1.1690
d ni
1.0000000 0.2836
end
pseudo nonlocal
nihay  ni
c c
runtype scf
maxcyc 30
level 2.5 15 1.0
vectors hcore
enter
```

to run:

```
rungamess -t ed2 -t ed3 -l ed0 -n nicch2 nicch2_rhf
```

Closed-shell SCF Restart Data

Contents of file nicch2_swap.in

```
restart new
title
ni(cch2) 1a1 rhf restart ,hay's ni, bar's nm ecp's
mult 1
bypass
super force
zmat angstrom
ni
c 1 nica
x 2 1.0 1 90.0
c 2 cacb 3 90.0 1 180.0
x 4 1.0 2 90.0 3 0.0
h 4 hcb 2 hcc 5 90.0
h 4 hcb 2 hcc 5 -90.0
variables
nica 2.0895
cacb 1.3604
hcb 1.1047
hcc 122.646
```

```
end
basis
s h
0.032828     13.3615
0.231208      2.0133
0.817238      0.4538
s h
1.000000      0.1233
s c
1.000000      0.4962
s c
1.000000      0.1533
p c
0.018534     18.1557
0.115442      3.9864
0.386206      1.1429
0.640089      0.3594
p c
1.000000      0.1146
s ni
-0.4372528 0.6778
1.1889453 0.1116
s ni
1.0000000 0.0387
p ni
1.0000000 0.0840
p ni
1.0000000 0.0240
d ni
0.0360414 42.7200
0.1938645 11.7600
0.4596238 3.8170
0.5599305 1.1690
d ni
1.0000000 0.2836
end
pseudo nonlocal
nihay  ni
c c
runtype scf
maxcyc 40
level 3.0 10 1.0
swap
4 5
6 8
8 10
10 11
end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -l ed0 -n nicch2 nicch2_swap
```

Open-shell SCF Job

Contents of file nicch2_orhf.in

```
restart new
title
ni(cch2) 3a1 rhf ,hay's ni, bar's nm ecp's
mult 3
super force
bypass
zmat angstrom
ni
c 1 nica
x 2 1.0 1 90.0
c 2 cacb 3 90.0 1 180.0
x 4 1.0 2 90.0 3 0.0
h 4 hcb 2 hcc 5 90.0
h 4 hcb 2 hcc 5 -90.0
variables
nica 2.0895
cacb 1.3604
hcb 1.1047
hcc 122.646
end
basis
s h
0.032828     13.3615
0.231208      2.0133
0.817238      0.4538
s h
1.000000      0.1233
s c
1.000000      0.4962
s c
1.000000      0.1533
p c
0.018534     18.1557
0.115442      3.9864
0.386206      1.1429
0.640089      0.3594
p c
1.000000      0.1146
s ni
-0.4372528 0.6778
1.1889453 0.1116
s ni
1.0000000 0.0387
p ni
1.0000000 0.0840
p ni
1.0000000 0.0240
d ni
0.0360414 42.7200
0.1938645 11.7600
0.4596238 3.8170
```

```
0.5599305 1.1690
d ni
1.0000000 0.2836
end
pseudo nonlocal
nihay ni
c c
runtype scf
scftype gvb
maxcyc 50
level 2.0 3.0 15 1.0 1.0
open 2 2
swap
11 12
14 15
end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -l ed0 -n nicch2 nicch2_orhf
```

Eigenvector utilisation in the above job will drive off the default sections of the Dumpfile, with the open-shell SCF module using the closed-shell SCF vectors from section 1 to initiate the SCF process, and writing the SCF open-shell orbitals to sections 4 and 5 (the energy-ordered SCF MOs). These latter orbitals will be used below to instigate the CASSCF processing.

CASSCF Job

Contents of file nicch2_cas.in

```
restart new
title
ni(cch2) 3a1 c2v cas at opt (3 2 2 0), hay's ni, bar's nm ecp's
mult 3
super off nosym
zmat angstrom
ni
c 1 nica
x 2 1.0 1 90.0
c 2 cacb 3 90.0 1 180.0
x 4 1.0 2 90.0 3 0.0
h 4 hcb 2 hcc 5 90.0
h 4 hcb 2 hcc 5 -90.0
variables
nica 2.0895
cacb 1.3604
hcb 1.1047
hcc 122.646
end
basis
s h
```

```
0.032828     13.3615
0.231208      2.0133
0.817238      0.4538
s h
1.000000      0.1233
s c
1.000000      0.4962
s c
1.000000      0.1533
p c
0.018534     18.1557
0.115442      3.9864
0.386206      1.1429
0.640089      0.3594
p c
1.000000      0.1146
s ni
-0.4372528 0.6778
1.1889453 0.1116
s ni
1.0000000 0.0387
p ni
1.0000000 0.0840
p ni
1.0000000 0.0240
d ni
0.0360414 42.7200
0.1938645 11.7600
0.4596238 3.8170
0.5599305 1.1690
d ni
1.0000000 0.2836
end
pseudo nonlocal
nihay  ni
c c
runtype scf
scftype casscf
config print
fzc 1 to 7
doc 8 to 9
alp 10 to 11
uoc 12 to 14
end
superci 1 to 12
newton 13 to 20
hessian 13  to 20
simul 15 to 20
vectors 5
swap
14 15
end
enter
```

to run:

```
rungamess -t ed2 -t ed3 -t ed4 -t ed6 -l ed0 -n nicch2 nicch2_cas
```

Direct-CI Job

Contents of file nicch2_ci.in

```
restart new
title
ni(cch2) 3a1 mrsdci (3 2 2 0) ,hay's ni, bar's nm ecp's
mult 3
super off nosym
bypass scf
zmat angstrom
ni
c 1 nica
x 2 1.0 1 90.0
c 2 cacb 3 90.0 1 180.0
x 4 1.0 2 90.0 3 0.0
h 4 hcb 2 hcc 5 90.0
h 4 hcb 2 hcc 5 -90.0
variables
nica 2.0895
cacb 1.3604
hcb 1.1047
hcc 122.646
end
basis
s h
0.032828     13.3615
0.231208      2.0133
0.817238      0.4538
s h
1.000000      0.1233
s c
1.000000      0.4962
s c
1.000000      0.1533
p c
0.018534     18.1557
0.115442      3.9864
0.386206      1.1429
0.640089      0.3594
p c
1.000000      0.1146
s ni
-0.4372528 0.6778
1.1889453 0.1116
s ni
1.0000000 0.0387
p ni
1.0000000 0.0840
p ni
1.0000000 0.0240
d ni
```

```
0.0360414 42.7200
0.1938645 11.7600
0.4596238 3.8170
0.5599305 1.1690
d ni
1.0000000 0.2836
end
pseudo nonlocal
nihay  ni
c c
runtype ci
core
end
active
1 to 40
end
direct 20 14 26
spin triplet
conf
2 2 2 2 2 2 2 2 2 1 1 0 0 0
2 2 2 2 2 2 2 2 2 1 2 0 0 1 0
2 2 2 2 2 2 2 2 2 1 1 1 0 1 0
2 2 2 2 2 2 2 2 2 0 1 1 0 2 0
2 2 2 2 2 2 2 2 1 2 2 0 1 0 0
2 2 2 2 2 2 2 2 1 2 1 1 1 0 0
vprint 100 0.01
maxcyc 10
enter
```

to run:

```
rungamess -t ed2 -t ed3 -t ed5 -t ed6 -l ed0 -n nicch2 nicch2_ci
```

25 Table-CI Calculations of the Electronic Spectra of Pyridine

This example demonstrates the use of the Table-CI module in the calculation of the low-lying states of Pyridine. Specifically, we are involved in determining the disposition of the first ten $^1A_1$ and $^1A_2$ states, using a common set of orbitals (the X^1A_1 SCF-MOs) in a DZ plus Rydberg basis set of 91 functions. Five job files are presented below:

1. Start–up closed–shell SCF calculation for the X^1A_1 state. Note the SUPER directive for compatibility with the subsequent CI calculation. Note also the particular syntax for siting the DZ basis on H: the third and fourth data fields are to provide an unscaled hydrogen basis, since the default specification will scale the two components by 1.2 (the more contracted) and 1.15 (the more diffuse component)

2. 1M/1R Table-CI calculation of the X^1A_1 state.

3. 6M/1R Table-CI calculation of the X^1A_1 state.

4. 21M/10R Table-CI calculation of the ten lowest $^1A_1$ states.
5. 19M/10R Table-CI calculation of the ten lowest $^1A_2$ states.

1. Closed-shell SCF Job

Contents of file pyridine2_rhf.in

```
title
pyridine dz+bond-centred functions
super off nosym
zmat angstrom
n
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
bq 1 1.39 3 90. 2 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
c2c3h 120.158516
end
basis
dz h 1.0 1.0
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
enter
```

to run:

```
rungamess -t ed2 -t ed3 -n pyridine2 pyridine2_rhf
```

2. 1M/1R Table-CI Job for the X^1A_1 State

An examination of the SCF output reveals the following symmetry adapted basis functions, given the C_{2v} geometry and DZ plus Rydberg basis set:

```
============================
IRREP  NO. OF SYMMETRY ADAPTED
       BASIS FUNCTIONS
============================
  1          45
  2          12
  3          28
  4           6
============================
```

Thus the orbital reordering performed by the Table-CI module will yield the sequence numbers 1–45 for the a_1 MOs, 46–57 for the b_1 MOs, 58–85 for the b_2 MOs and 86–91 for the a_2 MOs. We are both freezing and discarding orbitals in the subsequent CI calculations. Confining this to both a_1 and b_2 MOs, the first six orbitals of a_1 symmetry and first four orbitals of b_2 symmetry are to be frozen, and the eight highest energy orbitals of a_1 symmetry and six highest of b_2 are to be discarded. Thus the final sequence numbers for the active orbitals in the CI are 1–31 for the a_1 MOs, 32–43 for the b_1 MOs, 44–61 for the b_2 MOs and 62–68 for the a_2 MOs. The CONF data line specifying the reference configuration is based on the associated sequence numbers of these active orbitals.

Contents of file pyridine2_1m1r.in

```
restart
title
pyridine dz+bond-centred functions
super off nosym
bypass scf
zmat angstrom
n
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
bq 1 1.39 3 90. 2 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
c2c3h 120.158516
end
basis
dz h 1.0 1.0
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
runtype ci
mrdci
adapt
tran freeze discard
6 0 4 0
1 to 6 1 to 4
8 0 6 0
38 to 45 23 to 28
select
cntrl 22
spin singlet
symmetry 1
conf
0 1 2 3 4 5 32 33 44 45 46 62
```

```
roots 1 1
thresh 30 10
ci
diag
title
pyridine 1m1r ground state   dz + 3s2p2d rydberg basis
extrap 3
natorb
civec 1
putq aos 2
enter
```

to run:

```
rungamess -r mrdci -l table -n pyridine2 pyridine2_1m1r
```

3. 6M/1R Table-CI Job for the X^1A_1 State

Contents of file pyridine2_6m1r.in

```
restart
title
pyridine dz+bond-centred functions
super off nosym
bypass scf
zmat angstrom
n
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
bq 1 1.39 3 90. 2 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
```

```
c2c3h 120.158516
end
basis
dz h 1.0 1.0
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
runtype ci
mrdci
adapt bypass
tran freeze discard bypass
6 0 4 0
1 to 6 1 to 4
8 0 6 0
38 to 45 23 to 28
select
cntrl 22
spin singlet
symmetry 1
conf
0 1 2 3 4 5 32 33 44 45 46 62
0 1 2 3 4 5 32 38 44 45 46 62
0 1 2 3 4 5 32 33 44 45 46 65
4 32 33 38 39  1 2 3 4 5  44 45 46 62
4 32 39 62 65  1 2 3 4 5  33 44 45 46
4 33 38 62 65  1 2 3 4 5  32 44 45 46
roots 1 1
thresh 30 10
ci
diag
title
pyridine 6m1r ground state   dz + 3s2p2d rydberg basis
extrap 3
natorb
civec 1
putq aos 2
enter
```

to run:

```
rungamess -r mrdci -l table -n pyridine2 pyridine2_6m1r
```

4. 21M/10R Table-CI Job for the $^1A_1$ States

Contents of file pyridine2_21m10r.in

```
restart
title
pyridine   dz+rydberg basis
super off nosym
bypass scf
zmat angstrom
n
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
bq 1 1.39 3 90. 2 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
c2c3h 120.158516
end
basis
dz h 1.0 1.0
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
runtype ci
mrdci
adapt bypass
tran freeze discard bypass
6 0 4 0
1 to 6 1 to 4
8 0 6 0
38 to 45 23 to 28
select
cntrl 22
spin singlet
symmetry 1
conf
2 62 65 1 2 3 4 5 32 33 44 45 46
2 33 34 1 2 3 4 5 32 44 45 46 62
2 5 8 1 2 3 4 32 33 44 45 46 62
2 62 63 1 2 3 4 5 32 33 44 45 46
2 33 38 1 2 3 4 5 32 44 45 46 62
2 33 35 1 2 3 4 5 32 44 45 46 62
2 5 7 1 2 3 4 32 33 44 45 46 62
2 33 37 1 2 3 4 5 32 44 45 46 62
2 5 9 1 2 3 4 32 33 44 45 46 62
2 62 64 1 2 3 4 5 32 33 44 45 46
2 33 36 1 2 3 4 5 32 44 45 46 62
2 5 11 1 2 3 4 32 33 44 45 46 62
2 33 39 1 2 3 4 5 32 44 45 46 62
2 32 34 1 2 3 4 5 33 44 45 46 62
2 32 38 1 2 3 4 5 33 44 45 46 62
4 5 8 33 38 1 2 3 4 32 44 45 46 62
4 33 38 62 65 1 2 3 4 5 32 44 45 46
4 32 33 34 38 1 2 3 4 5 44 45 46 62
4 33 34 62 65 1 2 3 4 5 32 44 45 46
4 32 38 62 63 1 2 3 4 5 33 44 45 46
4 5 7 33 38 1 2 3 4 32 44 45 46 62
roots 10 1 2 3 4 5 6 7 8 9 10
thresh 30 10
ci
diag
title
pyridine 21m10r 1a1    dz + 3s2p2d rydberg basis
extrap 3
enter
```

to run:

```
rungamess -r mrdci -l table -n pyridine2 pyridine2_21m10r
```

5. 19M/10R Table-CI Job for the $^1A_2$ States

Contents of file pyridine2_19m10r.in

```
restart
```

```
title
pyridine  dz+rydberg basis
super off nosym
bypass scf
zmat angstrom
n
x 1 1.0
x 1 1.0 2 90.
x 1 1.0 2 90. 3 90.
c 1 c4n 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.
c 1 c2n 2 c2nz 3 180.
c 1 c2n 2 c2nz 3 0.0
c 9 c2c3 1 ccn 2 180.
c 10 c2c3 1 ccn 2 180.
h 9 c2h6 1 nch2 2 0.0
h 10 c2h6 1 nch2 2 0.0
h 11 c3h5 9 c2c3h 1 180.
h 12 c3h5 10 c2c3h 1 180.
bq 1 1.39 3 90. 2 180.
variables
c4n 2.7845546
ch4 1.0823078
c2n 1.3372389
c2nz 120.641858
c2c3 1.3944571
ccn 122.662269
c2h6 1.0814291
c3h5 1.0809550
nch2 116.400433
c2c3h 120.158516
end
basis
dz h 1.0 1.0
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
runtype ci
mrdci
```

```
adapt bypass
tran freeze discard bypass
6 0 4 0
1 to 6 1 to 4
8 0 6 0
38 to 45 23 to 28
select
cntrl 22
spin singlet
symmetry 4
conf
2 5 65 1 2 3 4 32 33 44 45 46 62
2 33 47 1 2 3 4 5 32 44 45 46 62
2 5 63 1 2 3 4 32 33 44 45 46 62
2 33 48 1 2 3 4 5 32 44 45 46 62
2 8 62 1 2 3 4 5 32 33 44 45 46
2 7 62 1 2 3 4 5 32 33 44 45 46
2 33 50 1 2 3 4 5 32 44 45 46 62
2 5 64 1 2 3 4 32 33 44 45 46 62
2 33 49 1 2 3 4 5 32 44 45 46 62
2 6 62 1 2 3 4 5 32 33 44 45 46
4 5 33 38 65 1 2 3 4 32 44 45 46 62
4 5 33 38 63 1 2 3 4 32 44 45 46 62
4 33 47 62 65 1 2 3 4 5 32 44 45 46
4 5 33 34 65 1 2 3 4 32 44 45 46 62
4 7 33 38 62 1 2 3 4 5 32 44 45 46
4 33 47 62 63 1 2 3 4 5 32 44 45 46
4 8 33 38 62 1 2 3 4 5 32 44 45 46
4 5 33 34 63 1 2 3 4 32 44 45 46 62
4 33 48 62 63 1 2 3 4 5 32 44 45 46
roots 10 1 2 3 4 5 6 7 8 9 10
thresh 30 10
ci
diag
title
pyridine 19m10r 1a2  dz + 3s2p2d rydberg basis
extrap 3
enter
```

to run:

```
rungamess -r mrdci -l table -n pyridine2 pyridine2_19m10r
```

26 Full-CI calculations

We consider below Full-CI calculations of the X^1A_1 state of the H_2O molecule. In the first instance we consider correlating all electrons. We then perform a valence-only calculation, freezing the O1s orbital through the ACTIVE and CORE directives, specifying a total of 8 electrons on the FULLCI data line.

All-electron Job

Contents of file fullci_all.in

```
core 8000000
title
h2o - DZ basis - full-ci
super off nosym
zmat angstrom\o\h 1 roh\h 1 roh 2 theta
variables\roh 0.956 hess 0.7\theta 104.5 hess 0.2 \end
basis dz
runtype ci\fullci 14 5 5
enter
```

to run:

```
rungamess -r fullci -n h2o_fullci fullci_all
```

Assuming the above job did not complete in the time allocated, and dumped to disk in a controlled fashion, the following job would act to continue the processing, assuming that the FORTRAN file from stream ftn008 had been saved, along with the Mainfile (ed2) Dumpfile (ed3) and Transformed integral file (ed6). All these files should be stored on temporary disk space, given the –r fullci argument.

Restarting the Full-CI job

Contents of file fullci_rest.in:

```
core 8000000
restart ci
title
h2o - DZ basis - restart full-ci
super off nosym
zmat angstrom\o\h 1 roh\h 1 roh 2 theta
variables\roh 0.956 hess 0.7\theta 104.5 hess 0.2 \end
basis dz
runtype ci\fullci 14 5 5
enter
```

to run:

```
rungamess -r fullci -n h2o_fullci fullci_rest
```

Valence-electron Job

Contents of file fullci_val.in:

```
core 8000000
restart new
title
h2o - DZ basis - valence full-ci
super off nosym
```

```
bypass
zmat angstrom\o\h 1 roh\h 1 roh 2 theta
variables\roh 0.956 hess 0.7\theta 104.5 hess 0.2 \end
basis dz
runtype ci
active\2 to 14 end\core\1\end
fullci 13 4 4
enter
```

to run:

```
rungamess -r fullci -n h2o_fullci fullci_val
```

The following points should be noted:

- the use of –r, which causes the FORTRAN stream ftn008 to be saved;
- the use of the core pre-directive to specify memory requirements.