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

Generalised Atomic and Molecular Electronic Structure System

GAMESS-UK

USER'S GUIDE and REFERENCE MANUAL

Version 8.0 June 2008

PART XX. THE VALENCE BOND PROGRAM TURTLE

J. Verbeek, J.H. Langenberg, C.P. Byrman, F. Dijkstra, J.J. Engelberts, M.L. Zielinski and J.H. van Lenthe

Contents

1 Introduction

```
           ----            ----         ----           ----   )))))
                      ----       ----         ----   _ _ _   { o o }
            ----                         ----      _(_)_(_)_  ( ^ )
     ----                   ----                _(_)_(_)_(_)_/ /
----              ----                         _(_)_(_)_(_)_(_)/
                        ----                  (_)_(_)_(_)_(_)_(_)
 ----      ----                  ----           //  //   //  //
```

TURTLE is an ab initio VB/VBSCF [1, 2] program written by J. Verbeek, J.H. Langenberg, C.P. Byrman, F. Dijkstra, J.J. Engelberts and J.H. van Lenthe (Utrecht)[3, 4, 5].

The wavefunction can be expressed as a linear combination of structures, where the coefficients are variationally optimised.

The program has the following properties :

- No restrictions are put on the number of configurations and structures.
- Orbitals may be kept localised on atoms or fragments and may have single or double occupancy.
- Both the orbitals and the coefficients of the structures are variationally optimised.
- Analytical gradients can be used to get insight into the structures of molecules and their reactions.
- It has been made parallel to let it run on multiple processors at the same time.

The program has two main parts, CRESTR (CREate STRuctures) and TURTLE (The VBSCF program)

- In this manual a *LIST* is a list of numbers possible including a TO to indicate a range and closed by an END.

  Example

  ```
  1 2 4 6 TO 10 5 END
  ```

2 CRESTR

2.1 Introduction

CRESTR is a support program for TURTLE, a program for Valence Bond calculations. Its name is derived from CREate STRuctures.

The task of CRESTR is to perform spin projections on chosen sets of configurations; the term configuration standing here for the distribution of electrons over space-orbitals. CRESTR defines symbolical Valence Bond structures. That is, numbers represent orbitals and the connection between these numbers and "real" orbitals is made by complementary programs.

CRESTR is able to construct both Rumer and branching diagram functions. For each configuration it generates all possible spin paths, but the user is able to select if required. For the theory of the construction of spin eigenfunctions see for example the paper by Raimondi et al. [6] The heart of the program consists of a spin projection procedure, based on leading terms [6], which generates Rumer functions. The construction of branching diagram functions is accomplished through Schmidt-orthogonalization of Rumer functions.

There are two ways in which configurations can be defined. The user can provide them directly or can define a set of reference configurations and virtuals, with which CRESTR automatically generates new configurations by performing single and double excitations.

Finally, CRESTR contains a symmetry option. If the user assigns irreps to orbitals, CRESTR removes all symmetry-forbidden configurations.

Note that memory requirements rise rapidly if the maximum number of open shells per configuration exceeds 6.

A configuration (or structure) is specified by a row of orbitals, which may include the string 'to'. A row of orbitals may always be spread out over several lines. If the definition of a configuration or Slater-determinant includes tags or integers other than orbitals and is scattered among several lines, then the first field in each line must be an orbital. Furthermore, a new configuration or determinant should begin on a new line. Doubly occupied orbitals are defined by the twofold occurrence of an orbital.

The CRESTR input starts with the keyword CRESTR and ends with an END.

```
CRESTR
........
........
........
END
```

2.2 Directives

All directives start with a keyword. The order in which the directives appear is irrelevant, except that the first ones (if specified) must be the NELEC, the MULT and/or the CORE-directives and the last one must be the END-directive.

2.2.1 The NELEC-directive

This directive consists of the keyword NELEC, followed by the number of electrons. GAMESS passes the number of electrons it used for the one and two electron-integrals to TURTLE. This directive overrides the number of electrons passed to CRESTR by GAMESS.

Example

```
NELEC  3
```

2.2.2 The MULT-directive

This directive consists of the keyword MULT, followed by the multiplicity. As with the NELEC-directive, this directive is used to override the value passed to CRESTR by GAMESS.

Example

```
MULT  3
```

The wave function is threefold spin degenerate (a triplet).

2.2.3 The CORE-directive

This directive consists of the keyword CORE, followed by the number of core orbitals. This number tells CRESTR and TURTLE how many of the specified orbitals in CONF/STRUC and later inside TURTLE are frozen.

Example

```
CORE 5
```

The first five orbitals are frozen core.

2.2.4 The PRINT-directive

This directive consists of the keyword PRINT, followed by a string representing the output-mode. If ALL or H is given, a large amount of output will be generated, considering all phases of program-execution. Amongst it is, for instance, the step by step formation of leading terms from configurations. LIMI or M shows the defined structures and determinants. By default CRESTR gives sparse information.

The interpretation of the user-output CRESTR gives (if LIMI/M or ALL/H is requested) considering the definition of Valence Bond structures is explained in the following example:

```
 == list of configurations ==

 ** configuration     1  **
      1   1   2   2
     structure   1  determinants   1 leading term abab
       1 -1.00000

 ** configuration     2  **
```

```
      2   3   1   1
     structure   1  determinants   2 leading term abab
       1 -0.70711   2  0.70711
     ******** determinants ********
       1 : ab   2 : ba

 ** configuration     3  **
      1   2   3   4
     structure   1  determinants   4 leading term abab
       1  0.50000   2 -0.50000   3 -0.50000   4  0.50000
     structure   2  determinants   4 leading term aabb
       1 -0.50000   4 -0.50000   5  0.50000   6  0.50000
     ******** determinants ********
       1 : abab   2 : baab   3 : abba   4 : baba   5 : aabb   6 : bbaa

 ======== end of list  =========
```

The configurations are shown in ordered fashion: doubly occupied behind singly occupied orbitals. Doubly occupied orbitals are neglected in the spin-coupling procedure in CRESTR and are not shown in the Slater-determinants above. Consequently, for the first configuration in the example, which does not have any singly occupied orbital, no determinant is shown at all. The second configuration is projected into the following structure :

$$\Psi = -\sqrt{\frac{1}{2}}|1\overline{1}2\overline{3}| + \sqrt{\frac{1}{2}}|1\overline{12}3| \tag{1}$$

The third configuration is projected into two structures, which are interpreted as follows:

$$\Psi_1 = \frac{1}{2}|1\overline{2}3\overline{4}| - \frac{1}{2}|\overline{1}23\overline{4}| - \frac{1}{2}|1\overline{23}4| + \frac{1}{2}|\overline{1}2\overline{3}4| \tag{2}$$

$$\Psi_2 = -\frac{1}{2}|1\overline{2}3\overline{4}| - \frac{1}{2}|\overline{1}2\overline{3}4| + \frac{1}{2}|12\overline{34}| + \frac{1}{2}|\overline{12}34| \tag{3}$$

For each structure the leading term is printed as well.

2.2.5 The SPIN-directive

This directive consists of the keyword SPIN, followed by a string representing the spin projection. If RUMER or LT is given, Rumer functions will be defined. Any of the strings BD, YK, or GEN, cause branching diagram functions to be defined. OFF defines (all possible) Slater-determinants as independent entities, that is, not gathered in structures. By default CRESTR defines Rumer functions. (See also the CONF-directive, 2.2.6.)

Example

```
SPIN RUMER
```

Rumer functions will be used.

2.2.6 The SYM-directive

With this directive the irreducible representation of the wave function can be defined. It is used to exclude configurations of incorrect symmetry from the (symbolic) wave function. Note that the irreps of the orbitals must be determined by the user, as this is not taken care of by TURTLE.

Input-lines :

- The character string SYM.
- The number of orbitals participating in the wave function.
- The pointgroup of the molecule, followed by the irrep of the wave function.
- A number of lines, each one consisting of an irrep-symbol, followed by a series of orbitals.

The irreps of all present orbitals must be assigned. The pointgroups and corresponding irreps allowed for are the same as in DIRECT CI.

Example

```
SYM
10
D2H  AG
AG  1  TO  4
B1U  7 10
B2U  6   9
B3U  5   8
```

2.2.7 The CONF-directive

This directive is used to define one or more configurations.

Input-lines :

- The character string CONF.
- A number of lines consisting of orbitals, defining one or more configurations as a *LIST* First in each sequence CORE NCORE (A,I) may be specified, indicating that the first NCORE orbitals are doubly occupied. Each configuration may be accompanied by a spin tag and a selection of spin paths (see further).
- The character string END.

Each configuration must consist of NELEC orbitals.

Example

```
CONF
1 1 2 2
1 1 2 3
END
```

The first configuration consists of two closed shells, orbital 1 and 2. The second consists of one closed shell, orbital 1, and two open shells, orbital 2 and 3.

By default all structures corresponding to a configuration are formed by performance of the spin projection chosen with the SPIN-directive. However, the general approach will be overruled for a certain configuration if its definition is followed by a spin tag. For this tag, the same strings can be used as for the SPIN-directive (except OFF). If the tag is followed by integers, selected structures belonging to the particular configuration will be defined. In this case, the integers represent the selected spin paths. (By definition, the first spin path is the one that exhibits perfect pairing.) No more than 14 structures may be selected per configuration. This number can be raised easily by altering the statement DATA MAXSP/14/... in the main routine CRESTR.

Example

```
SPIN  BD
CONF
1 to 4
1 2 3 5  RUMER
1 2 4 5  BD  2
END
```

All (two) branching diagram functions are defined for the first configuration, and all Rumer functions for the second configuration. For the third configuration only the second branching diagram function is defined.

The user is protected against the multiple definition of one configuration. This protection will be overruled, for a particular configuration, if the user chooses to select structures as described above. The CONF-directive can not be combined with the MRSD-directive.

2.2.8 The ALL-subdirective for CONF

This subdirective is used to generate all possible configurations in which m electrons are distriuted over n orbitals. The $m \cdot 2$ can't be bigger than n.

Input-lines:

- The following input-characters should be provided on the same line.
- The character string ALL.

- The NORB number of orbitals over which we distribute the electrons.
- The NELEC number of electrons to distribute.
- The character string RESTR.
- The NRESTR number of restricted orbitals.
- A series of orbitals to restrict. The orbital numbers should include NCORE value in it.

If RESTR is specified after the NELEC value, only one of the provided orbitals will be doubly occupied per configuration. In all other cases, configurations will be disregarded.

Example

```
CONF
ALL 4 2
END
```

Generating all possible configurations by distributing two electrons over four orbitals. The resulting configurations are:

```
1 2
1 3
1 4
2 3
2 4
3 4
1 1
2 2
3 3
4 4
```

Example

```
CONF
ALL 5 4  RESTR 2  3 4
END
```

The resulting configurations are:

```
 1    1    2    3    4
 1    2    2    3    4
 1    2    3    3    4
 1    2    3    4    4
 1    1    2    2    3
 1    1    2    3    3
 1    2    2    3    3
 1    1    2    2    4
 1    1    2    4    4
 1    2    2    4    4
 1    1    3    3    4
 1    1    3    4    4
 2    2    3    3    4
 2    2    3    4    4
```

The skipped configurations have the orbitals number 3 and 4 doubly occupied.

2.2.9 The STRUC-directive

This directive is used to define structures manually. Structures may consist of any linear combination of Slater-determinants and need not necessarily be an eigenfunction of S^2. The structures may be used normalised by specifying NORM on the input line.

Input-lines:

- The character string STRUC If NORM is specified on this line the structure is normalised.
- A number of lines defining one or more determinants and their corresponding spin-coupling coefficients. The coefficients must precede the orbitals that define a determinant, which are specified as a *LIST*. First the a-orbitals must be specified, then the b-orbitals. First in each sequence CORE NCORE (A,I) may be specified, meaning that the first NCORE orbitals are doubly occupied.
- The character string END.

STRUC ..... END sequence specifies a structure, which may be normalised. If more structures needs to be introduced, they need to be specified in sequence STRUC [NORM] ... STRUC [NORM] ... STRUC [NORM] ... END.

Each determinant must consist of NELEC orbitals.

Example

```
STRUC
1.0  1 to  4
1.0  3 4 1 2
STRUC NORM
1.0  core 1   2 3
1.0  core 1   3 4
END
```

A 2 structure (spin-unrestricted) Valence Bond structure is defined which, in case of singlet multiplicity, consists of the following structures:

$1.0 * |12\overline{34}| + 1.0 * |34\overline{12}|.$

$1.0/\sqrt{2} * |1\overline{1}2\overline{3}| + 1.0/\sqrt{2} * |1\overline{1}3\overline{4}|$

The STRUC-directive may be applied as often as one likes but must not be used together with either the CONF- or the MRSD-directive.

2.2.10 The MRSD-directive

This directive is used to define a set of reference configurations and a group of virtual orbitals, with which single and double excitations will be performed. The excitations can be defined in two, distinct ways. In the first one (CONF), configurations serve as the reference space. After all single and double excitations to a certain virtual space have been carried out, spin projection is performed to all acquired configurations. In the second one (STRUC), the subject of the excitations is formed by Valence Bond structures, obtained by spin projection of reference configurations. The singles and doubles are generated by one- and twofold appliance of an excitation operator, which, in second quantization language, is defined as a $C_{a \rightarrow v} + C_{\overline{a} \rightarrow \overline{v}}$, where v represents a virtual orbital and a an active. Applied to a spin eigenfunction, this operator generates spin eigenfunctions, and no spin projection needs to be subsequently carried out.

Input-lines :

- The character string MRSD, followed by either one of the character strings CONF or STRUC.
- The character string REFER.
- A series of reference configurations. Spin path selection as with the CONF-directive may be applied only if the excitation type is STRUC.
- The character string VIRT.
- A series of virtual orbitals.
- The character string END.
- The virtuals may be defined before or after the reference configurations.

Example

```
MRSD  CONF
REFER
1 1 2 2 3 3 4 4 6
1 1 2 2 3 3 5 5 7
VIRT
8 9 10
END
```

Correlating configurations are generated by single and double excitations of the two reference configurations to the virtual orbitals 8, 9, and 10. Spin projection is performed after generation of the configurations.

2.2.11 The END-directive

This obligatory directive consists of the single keyword END and is used to close an input-session.

2.2.12 A complete example

```
CRESTR
NELEC 4
PRINT  LIMI
SPIN  BD
CONF
1 1 2 2
1 1 2 3
1 1 2 4
1 2 2 3
1 2 2 4
END
END
```

3 TURTLE

3.1 Introduction

TURTLE is a program designed to perform Valence Bond Self Consistent Field (VBSCF) and VBCI (VB) calculations. This part contains the directives that are for the general VB program.

3.2 General Directives

3.2.1 PASS

Pass may be used to specify the number of passes in the 2 sort stages of the 4-index transformation : PASS *ipas1 ipas2* (default 1 1).

Example

```
PASS 4 2
```

3.2.2 ACCURACY

Specify accuracy 4-index (smallest integral retained),using ACCUR *iacc* where the accuracy is set to $10^{(-iacc)}$ (default 10^{-10}).

Example

```
ACCU 12
```

The accuracy will be 10^{-12}.

3.2.3 ACTIVE

Defines set of active orbitals as a *LIST*. The number of active orbitals must exceed or equal the highest occupied orbital number. The active directive couples the orbitals defined with the CONF directive in CRESTR to the orbitals or a specified set of vectors as defined in TURTLE.

Example

```
ACTIVE
1 2 5 6 7
END
```

This statement couples the basis functions 1 2 5 6 and 7 to the functions 1 to 5 as defined in CRESTR.

3.2.4 SPLICE / ONELEC

Defines orbitals to be put in frozen core as a *LIST*. Cf. the ACTIVE directive.

3.2.5 NEW2

Turns on the new ordering scheme of 2-electron integrals. CURTAIL option switched off automatically.

3.2.6 VBVECTORS

Tell TURTLE to read vectors from dumpfile or input or a combination.

Syntax:

```
VBVECTORS isec
```

```
or
```

```
VBVECTORS MANUAL n s [PRINT]
```

or

```
VBVECTORS COMBINE [PRINT]
...
...
...
END
```

isec is the section of the dumpfile from which the vectors are to be restored.

MANUAL tells TURTLE to read *n* vectors from the input and to skip *s* columns per row. PRINT is optional. When it is used the vectors will be printed in the output. After the line with VBVECTORS the vectors have to be given.

COMBINE tells TURTLE to pick up orbitals from different locations, for example two or more different dump files. The VBVECTORS COMBINE recognises the following subdirectives

- FILE FILE FILENAME BLOCK SECTION *LIST* (2A,2I, *LIST*)

  The vectors specified in *LIST* are included from the section and dumpfile specified. If the dimension of vectors does not match (i.e. is less than) the current GAMESS dimension, the AO's to be filled in with zero's have to be specified using an line :

  ```
  EXTRA  \textit{LIST}  (A,\textit{LIST})
  ```

  Example

  ```
  FILE ED5  1  11     1 TO 5 8 9 END
  ```

- SECT SECT SECTION *LIST* (2A,2I, *LIST*)

  Vectors are added as in the FILE subdirective, except that they are taken from the current dumpfile.

- MANUAL MANUAL n s

  Like the VECTORS MANUAL directive.

- DUMP DUMP ISECV

- PERM (or MOPERM) PERM list

- ELIMINATE ELIMINATE list (A,nI) Eliminate the vectors that coefficients at the specified places.

- SCREEN (or CLEAN) SCREEN A I (A,F,I) Clean vectors up by making elements smaller than A^I exactly zero.

- END END COMBINE

-

Example 1

```
VBVECTORS 10
```

Vectors are picked up from section 10 of the dumpfile.

Example 2

```
VBVECTORS MANUAL 4 1
1  0.1000  0.3333
2  0.4444  0.1113
3  0.0000  0.0005

1  0.0000  0.0500
2  0.0000  0.1000
3  1.0000  0.6666
```

In this example four orbitals are manually fed to TURTLE. These molecular orbitals are linear combinations of three basis functions (atomic orbitals).

Example 3

```
VBVECTORS COMBINE PRINT
FILE ED13 1 1 1 to 3 END
EXTRA 7 TO 10 END
MANUAL 4 0
0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0
0.2   0.0   0.0   0.0
0.0   0.2   0.0   0.0
0.0   0.0   0.2   0.0
0.0   0.0   0.0   0.2
END
```

In this example orbitals are partly picked up from a dumpfile (ed13 in this case) and are partly defined manually. The vectors on ed13 consist of 6 coefficients only. The EXTRA directive increases the dimension of these vectors with 4 by adding zeroes for coefficients 7, 8, 9 and 10.

3.2.7 VBMO

Allows one to dump VB orbitals, in proper, understanable for molden format, to separate file. In order to picturize VB orbitals in molden, one has to copy those orbitals from this file, into

SCF orbitals in output file.

Example

```
VBMO
 local
```

Creates file called: out.vbmo.**local** in working directory (where all edX files are stored as well).

Given form of **VBMO** directive is the first step to new feature, allowing to dump all the necessary informations from VB calculations, for molden.

3.2.8 BYPASS

Allows one to bypass a stage of the VB calculation. Options are:

- 4INDEX
- HMATRIX
- DAVIDSON

3.2.9 DIPOLE

Defines the dipole integrals to mix in with the 1-electron integrals, for (e.g.) the calculation of a polarisability. One specifies on one line DIPOLE *fx fy fz*, where *fx*, *fy* and *fz* are the dipole mixing factors.

Example

```
   DIPOLE 0.0 0.0 0.05
```

3.2.10 BLKSIZE

Set the blocking factor for the sortfiles, by specifying BLKSIZE *ibl* (default 10, maximum 24).

3.2.11 CURTAIL

Specifies using CURTAIL *ncurt* that the 4-index transformation in TURTLE only produces all integrals for the first *ncurt* orbitals. For the remaining integrals at least 2 indices should be in the *ncurt* space. This directive is automatically set in VBSCF calculations. User intervention is not recommended.

3.2.12 TITLE

The following line is read in as a title of the job.

3.2.13 IPRINT

Sets printing level to *i*, using IPRINT *i*. A higher value of *i* yields more output. If no module is specified the print level *i* is set for all modules. One may specify individual print flags for different modules:

- DAVIDSON
- TRANSFORMATION
- HMATRIX
- SCF

Example

```
IPRINT 50 TRANS HMAT
```

3.2.14 WPRINT

Sets the printing level of branching or rumer diagrams with their corresponding weight at the end of a VBCI or VBSCF calculation. Weights above this value will be printed. The default value is 0.001.

Example

```
WPRINT 0.03
```

All weights with a value higher than 0.03 will printed with their corresponding branching or rumer diagrams.

3.2.15 SHIFT

Specify the Davidson levelshift (real), for the VBCI.

3.2.16 CRIT

Specify various criteria by give first the criterion as $RR.10^{(+II)}$, as format (F,I) and then as a string the intended module. The possibilities are

- DAVIDSON
- JACOBI
- ORTHOGONALISATION

If no module is specified, DAVIDSON is assumed.

3.2.17 MIX

Mix input orbitals by hand; If no orbitals are present a Unit matrix is used. On the keyword line the string 'PRINT' may be given, causing a print of the result. Each resulting orbital occupies one line; Specify for each orbital in the mix NO, FO (I,F), being the orbital number and the mixing factor. Finish the linew with the keyword 'END'

Example

```
mix
1 0.5 2 0.5
1 0.5 2 -0.5
end
```

3.2.18 MAX

Maximum number of cycles in davidson (default 50) or maximum number of expansion vectors (default 30), if 'EXPANSION' is specified.

3.2.19 CASES

Analyse the types of matrix elements if specified.

3.2.20 SELECT

SELECT NN

Number of States to select for starting vector. One has a choice between the NN first (default) and NN lowest, if 'LOWEST' is specified.

Example

```
select 5 lowest
```

3.2.21 MODE

Select Davidson diagonalisation mode, by giving MODE KEY. Key may be

- EMIN
- VMIN
- LOCK
- JACDAV

3.2.22 ALTER

Requests level shift alteration in Davidson. If 'OFF' is specified on the same card the level-shift alternation is switched off again.

3.2.23 MULLIKEN

Atom definition in terms of orbital numbers. Defines which ao's belong to which atom and which basis functions belong to that atom. This directive must be terminated by 'end'. Usually the hybrid definition is more efficient at this.

Example

```
mulliken
 c1
 1 3 5 7 end
 1 to 12 end
 c2
 2 4 6 8 end
 13 to 24 end
end
```

3.2.24 HYBRIDS

Try to make bonding hybrids according to rumer-bonds. The hybrids are guessed by maximising overlap.

If this directive is followed by 'AO' then per atom the AO's will be used for it.

3.2.25 CLEAR

Clear ao's. That is remove coefficients on alien atoms. Atom definitions have to be given at the 'mulliken' option.

3.2.26 NMOS

Reduce the number of MO's defined in the active statement. This may be useful when mixing mo's by hand

3.2.27 SCHMIDT

SCHMIDT (a string of numbers (possibly using 'TO')

Orthogonalise vectors in the specified set (on 1 card) using Schmidt-orthogonalisation. More then 1 set may be specified. Now SERVEC may be more flexible and well defined.

Example

```
schmidt
1 2 6 to 8
```

3.2.28 LOWDIN

LOWDIN (a string of numbers (possibly using 'TO')

Orthogonalise vectors using Lowdin-orthogonalisation. Cf. SCHMIDT

3.2.29 ENERGY

Read sum of atomic energies, using ENERGY E , (A,F). Probably nothing is done with this; If we really want something like this the atomscf might be used .

3.2.30 EIGEN

By giving EIGEN NV (A,I), one requests the print of NV CI eigenvalues and eigenvectors. Instead of a number the text-string ALL may be given.

3.2.31 MRSD

This option is meant to steer a MR-VB option;

3.2.32 DETS

DETS requests output of matrix-elements between determinants

3.2.33 PARALLEL

Sets parallel modes and checks

Allowed keywords are:

- SYNCH : synchronous communication
- ASYNCH : asynchronous communication
- CHECK : use checksum do check communication
- NOCHECK : don't use checksum do check communication

3.2.34 hcpu

This directive is meant for really huge (in number of determinants) VB calculations. It causes the program to print out each row of the H- and S-matrices it calculates in high precision. If an integer is supplied on the same card, the calculation starts at that group, offering a primitive restart mechanism. So to skip calculation of the matrix-elements for the first 8 groups one specifies

Example

```
hcpu 9
```

3.2.35 END or FINISH

END or FINISH conclude the total TURTLE input an extra string 'VB" is allowed. In addition a section number to store the VB orbitals may be specified

Example

```
END VB 23
    ..or..
END 23 VB
```

3.3 SCF-directives

3.3.1 Introduction

The SCF part of the TURTLE input is started by a SCF keyword and concluded with an END. On this card the word SCF and a section number for the VBSCF orbitals may be given.

Example of global TURTLE input

```
    turtle
    ......
    ......
    scf
    .....
    ....
    end 33 scf
    .....
    end vb
```

The SCF section contains quite a few directives, that also occur in the 'normal' TURTLE input (especially directives referring to diagonalisation). These directives then refer to the diagonalisation in the Brillouin state part of the code.

3.3.2 EXCIT / MIX

EXCIT (or MIX) controls the orbital mixings in the SCF. In general the program is is able to determine the mixing options by itself (using the HYBRID definitions, if specified). This directive is meant to allow more direct control of the optimisation process. This is quite cumbersome and error-prone.

On the same line the word ASIS may be given, forcing the program to use the atomic orbitals as they are (as is). This may well cause dependency problems

After that equivalence restrictions and orbital excitation patterns may be specified; See the program text for details.......

3.3.3 CRIT

Criterion for Jacobi, Davidson or SCF (default SCF) as CRIT FFF III (A,F,I), which specifies the criterion as FFF*$10^{(}+_{i})$. By default the SCF criterion is set. On the same line the following keywords (one or more) may be set.

- JACOBI .. set jacobi criterion
- DAVIDSON .. set davidson criterion
- SCF .. set scf criterion
- OVERLAP .. overlap criterion is used for convergence

Example

```
crit 2.3 -6 jacobi davidson
```

3.3.4 MAX

Set the maximum number of iterations for and in the SCF, using MAX IMAX KEY (A,I,A), where key may be one of the following :

- SCF .. set maximum number of SCF iterations (default 25)
- DAVIDSON .. set maximum no. davidson iterations (in SCF) (default 50)
- EXPANSION .. set max. no. expansion vectors in davidson (in SCF) (def. 30)

3.3.5 MODE

Set diagonalisation mode using MODE KEYWORD (A,A), where Keyword is one of:

- EMIN

- VMIN
- LOCK
- JACDAV

3.3.6 ALTER

Request level shift alternation in the Davidson

3.3.7 SHIFT

Perform level shift in SCF or SCF-Davidson, using either SHIFT ESHIFT SCF or SHIFT ESHIFT DAVIDSON (A,F,A).

3.3.8 SELECT

Select start of Brillouin Interaction Davidson, specifying either

- SELECT FIRST
- SELECT LOWEST

The SELECT FIRST is the default, and I would be really concerned, if that was not the right choice.

3.3.9 ORTHO

The ortho directive allows one to switch off the allowed orthogonalisations; By default for instance singly occupied are orthogonalised to the doubly occupied ones. Note that the VBCI does not do these orthogonalizations

```
example :
  ortho off
```

This directive may also be used to specify the allowed orthogonalisations explicitly. This a a very complicated and possibly "dangerous" procedure. Look in the code to see how it's done.

3.3.10 REMOVE

Specify the criterion to ignore excitations based on the nuclear attraction matrix as FFF*$10^I II$, which is read as REMOVE FFF III (A,F,I). An excitation corresponding to a nuclear attraction integral smaller then this criterion is omitted in the automatic SCF procedure. (Default 1.0d-8)

3.3.11 HYBRID

Define atoms for automatic atom-scf procedure. If a mulliken as been specified, those definitions are used. On the hybrid card the words CLEAN (Default) or DIRTY may be specified. CLEAN means that all ao's in a hybrid-mo not belonging to that hybrid are zeroed (cleaned). DIRTY leaves them alone.

On a new line the word GUESS may be specified, which means that the program tries to guess spin-bonded hybrids based on the spinfunctions and the hybrid definitions, using singular value decomposition.

Next lines specify the hybrids.

Syntax:

```
hybrids
 atom-name
 functions (mo's) end
 basisfunctions end
end
```

Instead of specifying the basis-functions one can give the keyword ATOM and specify the numbers of the atoms for this hybrid, in which case atom-definitions and the mo's just specified are used to determine the ao's in this hybrid. This makes the hybrid definition basis-set independent.

Example

```
hybrids
 c1
 1 3 5 7 end
 1 to 12 end
 c2
 2 4 6 8 end
 13 to 24 end
end
```

Example 2

```
hybrids
 c1
 1 3 5 7 end
 atom 1 end
 c2
 2 4 6 8 end
 atom 2 end
end
```

3.3.12 FORCE

FORCE IGROUP (A,I) forces the orbitals in group IGROUP to be equivalent.

3.3.13 NOSYMM

Do not use spatial symmetry. If CALC is specified, do not use symmetry in the calculation of matrix elements

3.3.14 NOSPIN

Do not use symmetry due to spin in matrix element evaluation.

3.3.15 SUPER

Do not allow internal excitations. These are replaced by excitations to ao's. This may be very beneficial in cases, where the active orbitals are nearly (or exactly) identical. When convergence is troublesome in e.g. a Breathing Orbital [7] calculation try SUPER. In addition some super options/approaches may be specified. Not all of these are well tested or understood. They are :

- HYBRID : determine mixing completely by HYBRID definition. One can ask for the last IGNO AO's to be ignored, by giving IGNORE IGNO (A,I).
- CAS : allow no active-active mixing (CASSCF assumed)
- ACTIVE : allow no excitations to virtual orbitals

ATTENTION!! SUPER HYBRID ¡large/first¿ IGNORE definition HAVE TO appear AFTER HYBRIDS definition

3.3.16 FORBI

FORBID IPAIR1 IPAIR2 ... (a,I,I,I,I,...)

This directive forbids (usual internal) excitations at an orbital level. It is meant for eliminations of undetected redundancies.

Example

```
FORBID 1 2 1 3
```

3.3.17 OPTIMISE

OPTIMISE TYPE [IDELP IFIRSP SHIFTP]

This directive is used to specify the way orbitals are optimised. When this directive is not given the orbitals are optimised using all possible Brillouin states, which is called Super CI. Building the Hamiltonian in the Brillouin basis is the rate determining step in most VBSCF calculation. With this directive it is possible to specify certain parts of the Brillouin matrix which will be approximated, or not calculated at all. Several example are given to show how it works.

Example 1

```
OPTIMISE KIND
 PERT DOC UOC
 PERT DOC VOC
 PERT VOC VOC
 PERT VOC UOC
END OPTIMISE
```

In this example the orbitals are grouped by their occupation in the wave function (KIND). Perturbation theory is used for excitations from doubly to unoccupied orbitals (PERT DOC UOC), for excitations from doubly to variably occupied orbitals (PERT DOC VOC), for excitations from variably to variably occupied orbitals (PERT VOC VOC) and for excitations from variably to unoccupied orbitals (PERT VOC UOC). Note that the parameters IDELP IFIRSP and SHIFTP are optional. In this case perturbation theory is switched on from the beginning.

Example 2

```
OPTIMISE ORBITALS -3
 FOCK 1 TO 7 END
 PERT 8 12 END
END OPTIMISE
```

In this case the orbitals are not grouped, but directly specified (ORBITALS). The optimise option starts working after the DEL-value has dropped below -3. A Fock matrix instead of Brillouin elements is used for excitations from and to orbital 1 to 7 (FOCK 1 TO 7 END). Perturbation theory is used for excitations from and to orbital 8 and 12 (PERT 8 12 END). Other orbitals are optimised using the normal Super CI method. An excitation from orbital 1 to 9 for example will be done by Super CI, because orbital 9 is not explicitly mentioned after the FOCK directive.

Example 3

```
OPTIMISE BRILLOUIN -1 5 0
 PERT 4 5 10 23 END
END OPTIMISE
```

In this example one can select certain Brillouin states. To use this option it is useful to know the way Brillouin states are numbered. An earlier run of Turtle can provide this information. The perturbation option is switched on after the DEL-value has dropped below -1, but not sooner than iteration 5 for Brillouin states 4, 5, 10 and 23.

Example 4

```
OPTIMISE CANONICAL 1 0 -1.0
 PERC DOC 15.0
 PERC UOC 20.0
END OPTIMISE
```

Here orbitals are sorted on their Eigenvalue. Excitations from the lowest 15% of the doubly occupied to the highest 20% of the virtual orbitals are treated by Fock matrix element. For the rest normal Super CI is used. Furthermore, a level shifter of -1.0 is used. This option is switched on from the beginning.

Example 5

```
OPTIMISE CANONICAL
 ENER DOC -5.1234
 ENER UOC  0.1234
END OPTIMISE
```

In this last example excitations with from doubly occupied orbitals with Eigenvalues below -5.1234 Hartree to unoccupied orbitals with Eigenvalues above 0.1234 are treated by Fock matrix elements. Other excitations are treated using normal Super CI. The option is switched on directly from the beginning.

Nota Bene: The canonical option has not thoroughly been tested yet.

3.3.18 EVEC

EVEC IVEC (A,I)

Optimize orbitals for excited state IVEC. If the state order changes during optimisation, you're out of luck

3.3.19 FREEZE

FREEZE ISTRING

Freeze (i.e. do not optimise) the orbitals in ISTRING.

3.3.20 VIRTUAL

VIRTUAL specifies the way to determine the virtuals in the SCF procedure. Options are :

- CANONICALISE : Canonicalise over the nuclear attraction matrix
- LOCALISE : Use localised virtuals ; using NIT or ITER a maximum number of iterations may be specified.
- IDEMPOTENT : The projection operator used is idempotent.

- AOS : Use AO's as virtuals.

If the word NOT is specified before an option, the option is negated.

3.3.21 END

END concludes the SCF-TURTLE input. An extra string 'SCF" is allowed. In addition a section number to store the VBSCF orbitals may be specified

References

[1] J.H. van Lenthe and G.G. Balint-Kurti, Chem Phys. Lett. **76**, 138 (1980)

[2] J.H. van Lenthe and G.G. Balint-Kurti, J. Chem. Phys. **78**, 5699 (1983), doi:10.1063/1.445451.

[3] J. Verbeek, Nonorthogonal orbitals in *ab initio* many-electron wavefunctions, Ph.D. Thesis, Utrecht University (1990)

[4] C.P. Byrman, Nonorthogonal orbitals in Chemistry, Ph.D. Thesis, Utrecht University (1995)

[5] F. Dijkstra, Valence Bond theory, implementation and use of analytical gradients, Ph.D. Thesis, Utrecht University (2000)

[6] M.Raimondi, M. Simonetta and G.F. Tantardini, Ab initio Valence Bond Theory, Computer Physics Reports, Vol **2**, e 171 (1985)

[7] P.C. Hiberty, S.Humbel, C.P. Byrman, J.H. van Lenthe, J. Chem. Phys. **101**, 5969 (1994), doi:10.1063/1.468459.