HAVMOL

Luis A. Montero
Universidad de La Habana, Havana, Cuba, and CSIC, Madrid, Spain, April 1993
2nd. Edition of Manual, Universidad de La Habana, Havana, Cuba, 2010

***INDEX*:**

1.INTRODUCTION

1.1.Origins

HAVMOL is a simple computer program designed to perform calculations of molecular structure and properties by a quantum mechanical approach, according to the *ab initio* SCF MO theory of Roothaan[1] and Hall[2].

The pedigree of this program is the following: **HAVMOL** is a version of MICROMOL, an adaption made by Dr. Susan Colwell of the *Cambridge Analytic Derivatives Package* (CADPAC). That is a suite of programs developed in Cambridge, United Kingdom, between 1980 and 1983 by Dr. Roger D. Amos. Those, in turn, originated from former versions of Dupuis and King's HONDO program.

HAVMOL includes an intensive use of memory and resources of personal computers to increase portability and calculations in very modest CPU's, input simplicity and easiness of use by both initiated and non experienced scientists, anywhere. It allows to perform top level calculations of molecular structure and other properties in any place where personal computers exist, including home and experimental laboratories without mainframe or work station terminals. Substantial enhancements have been made to code and integral management in certain critical routines and improvements have been introduced to the input and output of information with respect to MICROMOL. A very easing feature is added to this version which allow a simple input from either Cartesian or internal coordinate files obtained by **TC-HABANA** program outputs, like semiempirical SCF-MO **HAVPAC**, molecular mechanics. Internal coordinates in the widespread MOPAC-AMPAC, or Gaussian's Z matrix formats are also allowed as input. A contracted two electron integral external file is built to save disk space, respect to former versions.

1.2.Main capabilities and features.

This program is able to calculate at the Roothaan's SCF-MO level both close and open shell systems, analytic first derivatives of the energy, and hence it can perform geometry optimizations and calculate numerical force constants. Thermodynamic quantities on the grounds of the statistical approach for ideal gases at standard and other desired temperatures, rotational constants, dipole and quadrupole moments and isotopic effects are also possible to be calculated for either input and optimized molecular structures. It can take into account isotopic substitution effects.

Briefly, program abilities are the following:

- Evaluation of 1 and 2-electron integrals over contracted cartesian gaussian basis functions of type *s, p* and *d*.
- SCF iterative procedure for closed and high spin open shell wave functions. Ghost terms can be used to perform *counterpoise* basis set superposition effect corrections and floating function calculations.
- Calculation of the energy gradients, and derivatives of both one and two electron integrals.
- Use of the gradients for automatic geometry optimization by Murtagh-Sargent minimiza-

tion routine.

- Use of the gradients for the calculation of force constants by numerical differentiation.
- Force constants matrix is used to calculate molecular vibrational energies and partition function derived quantities. It can include isotopic effects on vibrational terms.
- Calculation of rotational constants and partition function derived quantities of the optimized and other given geometries of the system. It can include isotopic effects.
- Calculation of some electrical properties of the desired system, including dipole and quadrupole moments, Mulliken gross charges and electric field at nuclei.
- Calculation of translational, rotational, and vibrational thermodynamic constants at 298.16 K and other desired temperatures as ideal gases.
- When geometry optimization is performed, an atomic cartesian coordinate output file is produced, to be compatible with other TC-HABANA personal computer program inputs.

1.3. Executable files

- **HAVMOL.EXE** which has been optimized to be run in any personal computer (minimum 640 KB RAM) with or without the i-8087/287/387 arithmetic coprocessors. It is dimensioned for:
  (a) a maximum of 63 basis functions.
  (b) a maximum of 30 shells.
  (c) a maximum of 12 atoms.
  (d) a maximum of 10 primitive gaussians in any contracted function.
  (e) a maximum of 110 unique primitives in total.

This is the fastest and most portable version.

- **HAVMOLE.EXE** which has been optimized to be run in any personal computer (with a minimum of 640 KB RAM) with an installed i-8087/287/387 arithmetic coprocessor, 486 DX processors or compatible higher in the series. Dimensioned for:
  (a) a maximum of 100 basis functions.
  (b) a maximum of 60 shells.
  (c) a maximum of 24 atoms.
  (d) a maximum of 10 primitive gaussians in any contracted function.
  (e) a maximum of 200 unique primitives in total.

This version is dimensioned for full use of the available DOS memory which can calculate larger molecular systems with larger basis sets. It uses an optimized overlay structure to save code room in memory.

- **HAVMOLW.EXE** which has been optimized to be run in any 386 or higher personal computer under MS WINDOWS environment. Dimensioned for:
  (a) a maximum of 127 basis functions.
  (b) a maximum of 100 shells.
  (c) a maximum of 30 atoms.
  (d) a maximum of 10 primitive gaussians in any contracted function.
  (e) a maximum of 508 unique primitives in total.

This version is dimensioned for full use of the available RAM, and eventually, virtual memory from hard disk. It can calculate the largest molecular systems with largest basis sets in this series of programs.

The programs were developed for IBM PC/XT/AT and PS/n, or compatibles, and they run under patched DOS v. 3.2 or higher versions. Eventually, they has been compiled and linked with overlays, using a MICROSOFT optimizing FORTRAN compiler version 5.1. In the case of **HAVMOLE**, the overlay structure forces to set the program in the default directory or in DOS path and not to change the executable file name. See in your operating system manuals the DOS environment related instructions, such as PATH and SET for further information. Typical HAVMOL.PIF and HAVMOLE.PIF command programs are provided in the package for running the corresponding DOS programs in MS-WINDOWS environment, taking advantage of multitasking abilities of this i80386 protected mode operating system.

The DOS configuration must allow at least 10 files to be opened simultaneously (see in your DOS manual reference to the CONFIG.SYS file). An approx. 512 kb virtual disk (see in your DOS manual reference to the VDISK.SYS or RAMDRIVE.SYS device drivers) is recommended to be used for allocating an scratch file (see *Section 10*) and saving much computing time in long calculations. Larger virtual disk capacities could be required in certain cases.

1.4.Structure of the manual.

This manual gives instructions for running the program once it has been installed on the personal computer. Instructions for a very simple setting up of the program are given in the appendix.

Section 2 gives the instructions to prepare inputs for all purposes, detailing several additional abilities, and it is recommended to be the first to be read. *Section 3* indicates how to specify the default input molecular geometry format and basis sets. *Section 4* describes the way to specify the molecular symmetry, which is very useful for saving time and performing more efficient calculations. *Section 5* describes how to run the program, provides procedures and hints to start and restart calculations, and using the maximum computer capabilities. It provides descriptions of methods to continue interrupted calculations, with different options. *Section 6* describes console input and special starting features by command line in DOS versions, which allows batch processing for series of input data. *Section 7* gives instructions to use alternative input file formats, as standard cartesian and internal coordinates. *Section 8* shows a compendium of representative input examples for significant different options included in this version. *Section 9* is a list of coded HONDO and **HAVMOL** error messages. *Section 10* is a description of file managing during the calculation.

2.OPTIONS FOR A JOB.

2.1.Files.

The standard input file is named <fn>.IMM, where <fn> is an user selected DOS file name and .IMM is a TC-HABANA system extension for file names, which is the default one. It must be written in ASCII text code to the current drive. Directives described in this manual must be contained in such input file. All the second and subsequent run input files of a restarting calculation are named as <fn>.IM1. Such input file names are asked to the user by the program from the console, or provided in the command line (only in DOS versions). This option is described below in *Section 6*.

Alternatives to .IMM files are provided to simplify input. They consist in a standard TC-HABANA cartesian (with a .CAR extension), or internal (with the .IMC extension) coordinate file. Moreover, MOPAC-AMPAC and Gaussian's Z matrix internal coordinate input formats are also accepted with the respective default .AMI and .ZMT file name extensions. All of them are described below in *Section 7*. However, here in this section are discussed almost all options for a job which are limited and simplified in such alternative input file formats.

The standard output file is named <ofn>.OMM, where <ofn> is an user selected DOS file name, which could be the same as input <fn>. The program sets <ofn> = <fn> as the default name if none is entered after the appropriate call from the console. OMM is a TC-HABANA system file extension, which is the default. It is written in ASCII code by the program to the drive where the input file is, or to the default one. Comprehensive results of the desired calculation are contained in such output file. The second and all subsequent run output files of a restarting calculation are named as *<fn>.nnn*, where *nnn* corresponds to the number of the run. Such output file names are assigned by the program from the main input file name.

A secondary output file is obtained when the option **OPTIMIZE** is used (see this run type below) and the geometry optimization process is finished. It is named <fn>.CAR, where <fn> is the input file name and .CAR is an standard TC-HABANA system DOS file name extension. It contains the number of real atoms in the molecule, the **TITLE** of the run (see this section below) and the cartesian coordinates (in Angstrom units) of the optimized polyatomic system. Each line of x, y, and z coordinates is followed by an integer which is the atomic number of the current center. This file is written in ASCII code, formatted, and ready to be used as input by other programs in the TC-HABANA system. *If a cartesian coordinate .CAR file is used as alternative input format, this output file with the optimized geometry is appended to it.*

Two direct access files are allocated by the program when a given run begins: i) the *dumpfile* used to hold general data about the run, geometry and basis set information, one-electron integrals, etc, and ii) the *mainfile*, used to hold the two-electron integrals. Each of these files are unformatted (binary recording to compact the information) and the records are designed to be of 4096 bytes under the Microsoft standard. The *dumpfile* default name is designed as <fn>.ED3 for **HAVMOL**, <fn>.ED1 for **HAVMOLE** and as <fn>.ED5 for **HAVMOLW** programs. The *mainfile* default name is designed as <fn>.ED2 for **HAVMOL**, as <fn>.ED0 for **HAVMOLE** and as <fn>.ED4 for **HAVMOLW** programs. The record structure of these files are not compatible among them. It means that .ED1 files of **HAVMOLE** cannot be used to supply *dumpfile* information to **HAVMOLW** program.

These files contain the information for restarting procedures. They must be preserved for proper continuations of large restarting runs for further calculations on a given input set (see run types in this section). The direct access mode used to build these files allows to restart also those externally or abnormally interrupted runs from the last point in which the *dumpfile* was updated during the interrupted run. Updating of the *dumpfile* is a continuously repeated routine in the program run.

If some files exist in the current drive with the same name as the new *dump* and *mainfiles*, they are used, but the previous information could be lost, except in the case of a **RESTORE** directive (see *Section 6* below), to get the one of the current run. In the case of using the same file again, some incompatibilities could occur if basis sets or molecular dimensions are changed. If files with such filename are not present, the program will allocate them as new ones, which must be always the prefered choice.

There is also a scratchfile <fn>.ED7 to be used as temporary workspace, which is allocated automatically by the program in the more convenient drive for the user, after the appropriate request in the console. This file is deleted by the program during normal ends of run, including those which require a restart, because no permanent information is recorded in it. During abnormal ends, the scratch file could appear in the directory. It can be deleted, because no use of it is made in the restart procedure.

2.2.Input format of standard IMM files.

The program opens the run with an interactive section. Details of answers and questions are given in this manual (*Section 5*). However, the first question in the console asks if the run is an *initial* or *restoring* job. The following explanation applies mainly for initial jobs, although most directive keywords can also be used for restoring. Details for restarting jobs will be explained in *Section 6.*

Input data must be supplied in a file with any DOS name and extension. However, the TC-HABANA system extension .IMM is recommended, which will be taken as the default one. The normal input IMM file contains an ASCII written text with the full information about the desired calculation. This file can be prepared with an usual programming editor or with the EDLIN or EDIT facilities of DOS (*care must be taken to not use document modes in most commercial text editing programs).* The input deck consists of a series of **directives** which can be written either in lower and uppercase letters.

2.3. Directive format in a typical standard input file.

A typical standard IMM input file is prepared with a text edition program, and it can look like the following:

```
TITLE
Furan, (C4H4O), MIDI1 basis set
RUNTYP OPTIMIZE
TIME 120 h
SYMMETRY
Cnv 2
END
ANGSTROM
ATOMS
Oxygen  8.0             .0            .0            .0
midi1
Carbon  6.0           1.11            .0           .82
midi1
Carbon  6.0            .72            .0           2.1
midi1
Hydrogen  1.0         2.06            .0           .33
midi1
Hydrogen  1.0         1.36            .0          2.96
midi1
END
START
NOTE End of the optimizarion run
TITLE
Furan (C4H4O), vibrational modes, MIDI1|MIDI1
RUNTYP FORCE
START
TITLE
Furan (C4H4O), electron density dependent properties, MIDI1|MIDI1
RUNTYP PROPERTY
START
TITLE
Furan, (C4H3DO), vibrational modes with deuteration in C2, MIDI1|NIDI1
RUNTYP FORCE
WEIGHTS
Deuterium 6 2.0
END
START
NOTE Preserving main and dumpfile and ending
STOP
```

The block, down to the first **NOTE**, corresponds to an *ab initio* SCF-MO geometry optimization of furan with the MIDI1 basis set. It must exist as a data file in the work directory with the name MIDI1.BFN. Furan ring belongs to C_{2v} point group and it is taken advantage of it, meaning the reproduction of symmetric atoms and avoiding the corresponding calculations. It will reduce substantially the amount of time which is required to end them and the disk room for integral data. After optimization, a force constant calculation is to be performed, with the same basis set..

Then electron density dependent results will be obtained, like Mulliken gross atomic charges and dipole moments. A new force constant calculation is then performed with an isotopic substitution of hydrogen by deuterium in site 2.

In the later case, it is not necessary a new calculation of 3N+1 SCF points, because atomic masses are not entering in derivatives of the energy, which have been already calculated for the non substituted molecules. In this way, only diagonalizations for force constants are necesary with the previously calculated and stored hessian. Then a very complete output will be available.

Bold type words in the example above are directives. They are written, and they can occur in any order, and if no one is present all the defaults apply. In general, a directive consists of a text line with a **KEYWORD** (which are appearing in the whole text in capital bold letters) followed by one or more parameters,

KEYWORD *[parameter] [parameter] ...*

Parameters enclosed in brackets are optional. Words and numbers in such directives must be separated to each other by one or more blank, equal (=), or comma (,) characters.

Width of lines are of 80 characters or columns in input files. However, if longer or smaller lines are needed, the first directive must be:

WIDTH *ncol*

where *ncol* is the maximum number of columns to be read. It can be changed several times during input. If console (internal) files are used in restart runs (see *Section 6* below) or input is taken from alternative input file formats (see *Section 7* below), it can not be used and the default value of 80 columns is compulsory.

All input is in free format. Real numbers must be written without exponents. Blank lines are ignored and may be used to improve the layout of the data.

In the above example exist some directives which are governing the run or providing information about the structure of the polyatomic system to be calculated, which are treated in individual sections of this text. **ATOMS** is treated in *Section 3* , **SYMMETRY** in *Section 4*, and **START** and **STOP**, which are commanding the run in *Sections 5* and *6*. All other kinds of directives are described in this *Section*.

2.4.Options for a calculation.

The type of calculation is determined by the directive:

RUNTYP *runtyp*

where *runtyp* coud be:

INTEGRAL to calculate the 1- and 2-electron integrals. See options for the integral package control below.

SCF to perform a single SCF calculation (assuming integrals to exist in *dump* and *mainfiles*). Options for the SCF routines are in sections below.

HFSCF to perform a single HF-LCAO-MO calculation (i.e. **INTEGRALS** followed by **SCF**). This is the default calculation type.

GRADIENT to perform a single energy and gradient evaluation.

GRADONE to perform a calculation of the one-electron component of the gradient only. This is useful if the Hellmann-Feyman force is all that is required.

OPTIMIZE to perform a geometry optimization. The algorithm used is a version of the Murtagh-Sargent method.

FORCE to perform a force constant matrix calculation by taking finite differences of gradients.

PROPERTY to calculate properties of the molecule, like dipole and quadrupole moments, and rotational constants.

ROTATION to calculate only rotational constants of the current geometry.

CHECK to print initial data and end the run, in order to check the input as it is read and processed by the starting sections of the program.

Following the set of directive lines corresponding to each **RUNTYP** directive and data associated to it, there must appear one **START** keyword as it is descibed in *Section 5* below.

2.5.Directives for defining external data files.

The following directives serves for defining the *dumpfile* and *mainfile* names and the starting blocks in each of them. They can be useful when it is desired the preservation of previous run informations in the existing files. Otherwise, they are not necessary in most jobs, taking into account the DOS like file managing of **HAVMOL** package, which names, opens and closes them with the name of the main input file.

DUMPFILE *<fn>.EDn iblk*

It is used to assign *dumpfile* (which contains general information and one electron integrals) on direct-access file *<fn>.EDn* (defaults are <fn>.ED3, or <fn>.ED1, or <fn>.ED5) starting at block *iblk* (default 1).

MAINFILE *<fn>.EDn iblk*

It is used to assign *mainfile* (which contains two-electron integrals) on direct-access file *<fn>.EDn* (default <fn>.ED2, or <fn>.ED0, or <fn>.ED4) starting at block *iblk* (default 1).

The *scratchfile* is created and deleted during each run. It is not useful for any permanent storage purpose. Therefore, no assignments are provided to it in directives.

2.6.Integral package controls.

Directives for this purpose are the following:

ACCURACY *itol icut*

This directive controls the accuracy of the integral routines, where *itol* is used to skip the products of primitives whose preexponential factor exp(-g) is less than 10.0**(-*itol*). Default value 10.0**(-18), i.e., *itol* = 18. The *icut* parameter is used to not store on the integral file those which values are less than 10.0**(-*icut*) in absolute value. Default value 10.0**(-10), i.e., *itol* = 10.

The authors of **HAVMOL** parent programs quoted that these values of *itol*, and *icut* are more than adequate for ordinary molecules near their equilibrium geometries. The only case in which it may be advisable to increase them (i.e. to decrease the threshold) is when one is attempting to calculate a weak long range interaction between two systems. The basic accuracy of the program's integral routines should allow a maximum value of *icut* about 14, in which case *itol* should be increased to about 25. The default values of *itol* and *icut* give energies virtually identical to the ATMOL programs.

NORM *normf normp*

normf is set to 0 (default) to normalize the basis functions. If equal to 1, no normalization of basis functions is performed. The value of *normp* must be equal to 1 if contraction coefficients correspond to unnormalized primitive functions or 0 (default) if contraction coefficients correspond to normalized primitive functions. In this case, the contraction coefficients of the d-type primitive functions should be the ones for the d(xx) primitives. Note that in the case where *normf* is zero (that is, the default), the coefficients of the occupied orbitals are given in terms of normalized basis functions.

SKIP

This keyword is used to skip calculation of 2-electron integrals. This option can be used when only the 1-electron integrals are to be re-evaluated in a given run, due to input changes in atom qualities (like nuclear charges), but basis function nor molecular geometry information. It is useful, for example, for ghost orbital calculations during the *counterpoise* procedure to avoid the basis set superposition error in certain long distance (particularly between two different bodies) interaction calculations. In such a case, the full system calculation must be followed by independent calculations with the ultimate single point geometry of each molecular component, taking atomic core charges of atoms in ghost orbital molecules as 0.0 (see *Section 3*). In this case the previous <fn>.ED2, or ED0, or ED4 *mainfile* must be preserved by using the ending option **STOP** in place of **FINISH** (see *section 5*). An example of this kind of calculation input is showed in *Section 8*.

2.7.Initial eigenvector guesses.

Convergence in the first SCF cycle is one of the most troublesome aspects of *ab initio* calculations, due to the great variety of basis functions. The succesive SCF cycles in geometry optimizations and force constant calculations are improved by the use of previous density matrices to build the initial Hartree-Fock one. However, many cases of non convergence are originated from wrong input geometry or parameters. The user must follow, in the first instance, the revision of input data if convergence presents problems. This version of **HAVMOL** provides tools to improve SCF convergence, both during iterations and from the first moment, with an improved initial density matrix. Directives are the following:

GUESS [*guess_density_matrix*]

where

guess_density_matrix = **HCORE**

In this case, the initial density matrix is taken from a diagonalization of the corresponding 1-electron hamiltonian matrix (default).

guess_density_matrix = **RESTORE [[<fn>.]ED***n ia*]

Then, the initial density matrix is taken from an existing **[<fn>.]ED***n* file, reading the orbitals starting at block *ia*. If *ia* is omitted the orbitals are assumed to occupy the same blocks as in the current dumpfile (usually ED3 for **HAVMOL**, or ED1 for **HAVMOLE,** or ED5 for **HAVMOLW**). If **[<fn>.]ED***n* is omitted, the current dumpfile is used. This directive is issued internally in optimization and force constant runs after the first energy evaluation. To activate this option it is required to begin the new run with the option **RESTORE**, or using the console input.

guess_density_matrix = **HUCKEL** *[const]*

Allows the use of an internal effective single shell hamiltonians, from exact atomic calculations, to create a guess eigenvector matrix by the extended Hückel theory (EHT). *const* is an empirical constant to be used as a scaling factor for calculations of off diagonal matrix elements. The default value is 0.84. Values around 0.8 give good results with typical EHT effective single shell hamiltonians. This procedure helps dramatically in certain hardly convergent cases. Keyword **HUCKEL** can be used alone, without **GUESS**, to obtain the same result. *This option is only valid for elements 1 to 18 of the periodic table in this program version.* Basis function information must be prepared to distinguish between different levels of atomic orbitals. See *Section 3* below.

SWAP

```
  K1  K1'
  K2  K2'
  .
  .
END
```

This directive allows to exchange the molecular orbitals K1 and K1', K2 and K2', etc., from the initial guess. It can be introduced during an interrupted SCF session to favor convergence. A **SWAP** directive is canceled after the initial guess.

2.8.Controlling the SCF program.

SCFTYP *scftyp*

Controls the type of SCF calculation. If closed shell restricted Hartree-Fock calculations are desired (default) *scftyp* = **CLOSED**. If high spin Hartree-Fock calculations are to be performed, then *scftyp* = **OSCF**.

MULTIPLICITY *mult*

where *mult* = 1 for singlets (default), 2 for doublets, 3 triplets, etc. Only needed when *scftyp* is **OSCF.**

NCOORB *n*

n is the number of canonical orbitals wanted to be kept in SCF calculations, from lower to higher eigenvalues. It means the limitation of virtual orbitals in the molecular basis. It could be useful during calculations of large molecules with split basis set, where high energy virtual molecular wave functions could not be significant for the SCF procedure.

EXTRAPOLATE ON / OFF

The SCF routines have a method for extrapolating the Fock matrices which generally improves convergence. In some cases however this routine seems to make matters worse and can be switched off with this option. The default in **ON**.

CONVERGE *nconv*

nconv = SCF convergence criterion (default = 5). Convergence is reached when the density variation between two consecutive SCF cycles is less than 10.0**(-*nconv*) in absolute value. The program executes one more cycle after satisfying the convergence criterion. This value of *nconv* is adequate for calculating the energy, but the convergence should be tightened if proceeding to the optimization or force constant sections. However, it is difficult to force the convergence past about *nconv* = 8 or 9. If the run type is **OPTIMIZE** or **FORCE,** *nconv* must be set to a minimum value of 7 (or a maximum value of 10**-7). This version of **HAVMOL** sets it automatically in such cases.

MAXIT *maxit*

maxit is the maximum number of allowed SCF iterations up to convergence (default = 50). If this value is surpassed during a run, the program aborts and allows to restart with or without a new *maxit*, beginning from the last SCF iteration. It must be increased, from the first moment, if the calculated polyatomic system is very large or could be expected to present some initial problems with convergence. In such a case, an initial larger value can avoid unnecessary interruptions.

LOCK *lokcyc*

Normally, the program inspects the SCF eigenvalues to determine whether the occupied orbitals are those of lowest energy, and reorders the molecular orbitals, if necessary, to ensure that they are. Specification of the **LOCK** option prevents this happening, and eventually allows an exited state to be found, so long as it is the lowest of a particular symmetry species. The locking starts from the cycle *lokcyc* (default 999).

SHIFT *s1 n s2* **<FIXED / VARY>**

Level shifters are aids to convergence which modify density matrices each cycle. (a) If **FIXED** is specified, then it applies level shifter *s1* for the first *n* cycles of the SCF procedure, followed by *s2* for the remaining cycles. (b) Otherwise it applies level shifters as necessary, (varying from cycle to cycle) so that the separation between occupied and virtual orbitals is never less than *s1* for the first *n* cycles, and *s2* thereafter. Default: **SHIFT** *1.0 5 0.5* **VARY.** Note: **LEVEL** is a synonym directive keyword for **SHIFT**.

DAMP13 *d13*

This directive specifies a damp factor which multiplies certain off-diagonal blocks of the Fock matrix. The default value is *d13* = 1.0. **DAMP13** is for the doubly occupied-virtual block. The use of damp factors diminishes mixing of orbitals in different classes and may be of use in aiding convergence.

CANONICAL *a b c*

It is a control for canonicalization factors in open shell calculations (**SCFTYP OSCF**). In such a case, the total Fock matrix is of the form:

$$\begin{array}{|l|l|l|}
\mathbf{F+(a/2)K} & \mathbf{F+K/2} & \mathbf{F} \\
\mathbf{F+K/2} & \mathbf{F+(b/2)K} & \mathbf{F-K/2} \\
\mathbf{F} & \mathbf{F-K/2} & \mathbf{F+(c/2)K}
\end{array}$$

where **F** is the Fock operator defined over the total density **D** matrix (i.e. by taking into account the sum of both and spin molecular orbital density matrices),

$$\mathbf{F = h_0 + J(D) - K(D)/2}$$

and **K** is the exchange operator over the open shell density **O**,

$$\mathbf{K = K(O)}$$

The total energy and the converged density matrices are independent of the choice of the parameters *a, b,* and *c*. These do, however, affect the eigenvalues and the rate of convergence. The default values *a = b = c = 0* correspond to the ATMOL program default. Other choices are *a = 2*, *b = 0*, *c = -2* which is the Roothaan single Fock operator formalism, and *a = 0, b = -1,* and *c* arbitrary, which gives the same results at convergence as the Roothaan double Fock operator method.

2.9. Some advice on converging SCF calculations.

Previous experiences with MICROMOL, quoted by their authors provided the following comments, which could be applied to **HAVMOL**:

(1) The program is capable of altering the level shifters in certain situations, e.g. the calculation is diverging or oscillating, and this can often reverse such a situation. However some cases get stuck in an oscillation. If this occurs try increasing the value of *n* (the cycle parameter) in the **SHIFT** command to a considerably larger value (15-20), which has effect of inhabiting state-swapping. The use of this, which or without a **LOCK** command, increases the likelihood that the calculation will converge to and exited state. However this is generally easier to deal with cases which do not converge at all.

(2) Some calculations will converge to an exited state. A clear sign of this (apart from the symmetries of the occupied orbitals being wrong) is the presence of virtual orbitals with negative eigenvalues (though of course for a few systems with very strong electron affinities this might be genuine). To cure this, use the **SWAP** directive, either on the initial guess or (better) on the converged exited state orbitals. The program should not change states in the middle of an optimization.

2.10.Constants and Variables.

HAVMOL, as its predecessor MICROMOL, is a cartesian coordinate program. Molecular internal coordinates are only generated during output generation to help users for understanding results. However, a proper translation in the coordinate framework can use a certain axis as an interatomic distance to be maintained fixed in sucesive calculations to simulate reaction pathways. Also, certain polar or cartesian coordinates can be left fixed during molecular geometry optimization to save time. Two directives are useful for such purpose:

VARIABLES

name value [unit]
name value [unit]
.
.
END

CONSTANTS
name value [unit]
name value [unit]
.
.
END

Quantities needed to specify the geometry (see **ATOMS** and **GEOMETRY** directives, in *Section 3* below) can be given by reference to a parameter which *name* and *value* is declared in a **VARIABLES** and **CONSTANTS** directive. These differ only in their effect on any subsequent geometry optimization; the values of variables are optimized whereas those of constants remain fixed. It is also possible to give values explicitly in the **ATOMS** or **GEOMETRY** directive; such values also remain constant. The only exception to this is that if no variables are declared at all, and the optimization is called for, then all geometrical parameters are varied in the optimization.

If no *unit* is provided the *value* is assumed to represent distance in either bohrs (default) or angstroms if a **ANGSTROM** directive (below) has been used. Individual parameters may be *units*: **A** for Angstroms, **B** for bohrs, **D** for degrees or **R** for radians. Angles should always have a unit. There is one predefined constant **T**, which has the value *109.47* **D** = *arccos (-1/3)* **R**.

ANGSTROM

Nuclear coordinates in the input, and distance parameters used to define nuclear positions, are assumed by default to be in atomic units. This option causes them to be read as Ångstrom units.

2.11.Force constant calculations.

The following are directives which modify force constant calculations:

NPOINTS *nvib*

In force constant calculations (see **FORCE** option above) the second derivatives are obtained by finite differencing of the gradients using either a forward difference 1-point (*nvib* = 1, default) or a central difference 2-point (*nvib* = 2) formula. It must be taken into account that this second option implies doubling the amount of SCF and gradient calculations to be performed. A simplex method is also available which uses the same number of energy evaluations (3*nuclei+1) as the 1-point differentiation but with accuracy about that of the two point formula. To use this set *nvib* to any value except 1 or 2.

STEPSIZE *delta*

In force constant calculations a step size for differencing is required. Default *delta* = 0.001 Bohrs. WARNING: It should not be assumed that the default stepsize is reasonable, as the choice depends upon (1) the number of differentiation points, (2) whether using the simplex method or not, (3) the accuracy required, and (4) the molecule being studied, i.e. it is necessary to experiment.

2.12.Property calculation directives.

Several kinds of properties can be calculated for each polyatomic system. Of them, we can select two groups: those depending on density matrix and those non dependent. Clearly, the first group can only be calculated after the calculation and storage in corresponding *dumpfile* of a density matrix. Other properties, as rotational constants, can be calculated after the availability of a molecular geometry.

Properties depending on a density matrix are Mulliken gross atomic charges, dipole and other electric moments, and electric fields at nuclea. If a **PROPERTY** directive is governing a run type, some additional directives can be entered:

DENSITY *isec1 [ityp1 [isec2 ityp2]]*

When this directive appears, one can redirect the choice of density matrices to a particular section in *dumpfile*. In this case the density matrix is read from section *isec1* of type *ityp1* in *dumpfile*. By default *isec1* = 7 and for *ityp1* = 0. If an open shell calculation is performed, values of *isec2* and *ityp2* can be provided, where densities of spin electrons are stored. By default *isec2* = 10 and *ityp2* = 0.

MOS *[isec] [ityp]*

In this case, the density matrix used for electronic properties is transformed to be in terms of molecular orbital basis. Then, eigenvectors must be stored in section *isec* of type *ityp* in dumpfile. Default values are *isec* = 8 and *ityp* = 0.

GAUGE *x y z*

Electronic properties that are dependent on coordinates, like dipole and quadrupole moments, can be calculated with respect to a certain position in the molecule, which can be made coincident with a certain atom or other point. Such position is denoted by *x*, *y*, and *z*, which became like the center of coordinates for the particular property. Default is *0 0 0*

Calculations of thermodynamic data by statistical procedures are included when rotational properties are calculated and after force constant calculations. The default temperature is 298.15 K to which are usually referred experimental reports. However, sometimes it could be necessary to obtain data for other temperatures. In such a case the following directive is used:

TEMPERATURE *temper*

where *temper* is the value of the desired temperature in Kelvins.

HAVMOL calculates rotational constants always after a geometry optimization, or before a force constant calculation (if it has not been already calculated for the current geometry), or during a call for **PROPERTY** calculation (if it has not been already calculated for the current geometry). However, for calculations of rotational thermodynamic constants, it is necessary certain symmetry index referred to internal axis. It is entered by:

AXIS *nax*

where *nax* is the maximum order among any of the internal rotational axis in the molecule, according its symmetry. If it is coincident with the one provided in the **SYMMETRY** directive (see below), this directive is not necessary. However, sometimes the **SYMMETRY** directive is used with a lower symmetry group than the one truly corresponding to the molecule, because it is provided only to allow saving of calculation time for equivalent atoms. In this case, the order of the axis in the simulated point group could be lower than the true one, and then this **AXIS** directive is compulsory to obtain more accurate thermodynamic values for rotational components. This directive is not necessary for non symmetrical molecules (i.e. for molecules of C_1 point group).

2.13.Miscellaneous commands.

TIME *n [units]*

Causes the job to run for *n* seconds (default if units are not written) and then to terminate in a controlled manner. The default time limit is 86400 seconds (24 hours). When termination occurs in a time controlled manner, a restart (see below) is very easy. Accidental terminations can also be restarted because the way the *dump* and *mainfiles* are accessed, but a small part of the previous calculation is usually lost. The *units* can be either **H** if *n* is in hours, or **M** if it is in minutes, or **S** (default) if it is in seconds.

WEIGHTS
name i weight
name i weight
. . .
END

When isotopic substitution effects are wanted to be calculated for both rotational and vibrational effects, isotope atomic weights of one or more atoms can be entered by **WEIGHTS** keyword. It also serves when atoms different to the first 36 elements in the periodic table are included in the polyatomic system. The entry is made in the above format where *name* is any pnemotecnic single word of up to eight characters to identify the isotope which will replace the target atom name in relevant output, *i* is the number of the atom in the final molecular geometry. Care must be taken to consider multiple atoms which are usually generated when a symmetry feature is input. *This **i** parameter is the final number of the atom in the order of the definitive polyatomic system, after* *inclusions of multiple atoms are made because symmetry*. Finally, *weight* is the input atomic weight.

HINT: Sucesive isotope effect calculations can be made by placing the necessary isotopic substitutions after sucesive **START** (see below) directives for both **FORCE** and **ROTATION** run types. It saves the time of a complete calculation of the force matrix, which is a very much time consuming procedure because it requires at least 3*natoms+1 SCF cycles, including two electron integral matrices and derivatives of them. *Dumpfile* must be preserved if a **RESTORE** calculation is desired.

NOTE *text*

The *text* is a comment which is ignored during the input process. **NOTE**s may occur anywhere an option statement may occur. See in *Section 7* below an special use of this directive for cartesian and internal coordinate input files.

GRADTOL *n*

Optimization runs stop when the magnitude of the largest component of the energy gradient is smaller than a specified tolerance. This directive sets the gradient tolerance to 10**-*n*. The default tolerance is 10**-3. *n* can be a real number. As helping numbers, see the table for different fractionary values:

n	*minimum gradient*
2.30103	0.005
2.39794	0.004
2.52288	0.003
2.69897	0.002
3.12494	0.00075
3.30103	0.0005

This magnitude becomes very important when a geometry is optimized for further force constant calculations. Note that the accuracy of a force constant run at the "minimum" geometry will depend upon the degree of convergence, as the cartesian force-constant matrix is only rotationally invariant at the stationary point.

TITLE
title (maximum of 80 characters)

The line following the **TITLE** directive can be written with any comment about the current run. Recommended items are the name of the polyatomic system and that of the basis set in use to be displayed in the final output. The default title is a blank line.

CHARGE *charge*

charge is the molecular charge with the appropriate sign. e.g.: *charge* = +1 for NH_4^+, -1 for F^-, etc. Default is zero (a neutral system).

PRINT *nprint*

nprint = 0 for a standard printing.
1 for extra printing of basis set plus symmetry.
2 for extra printing of MO guess routines.
3 for printing out 1e-integrals.
4 for printing out 2e-integrals.
5 for printing out SCF data for each cycle.
-1 to omit density matrix and virtual orbitals in standard printing (default).
-3 for printing 1e-contribution to gradient.
-4 for printing 2e-contribution to gradient.
-5 to Inhibit printing of records and time controls for integral evaluations.

The default printer option gives a reasonable amount of output in a standard SCF run. The alternatives all increase the amount of output.

2.14.Reduced set of directives for other standard input file formats.

HAVMOL presents a variety of alternative input file formats (see *Section 7*). These formats include both cartesian and internal coordinates for a given polyatomic system. If any alternative input file format is used, a reduced set of directives is implemented in order to simplify the input building job for users. This reduced set includes all of the most important directives, and they are the following:

AXIS *nax*

CHARGE *charge*

CONVERGE *nconv*

GRADTOL *n*

MAXIT *maxit*

MULTIPLICITY *mult*

NPOINTS *nvib*

PRINT *nprint*

SCFTYP *scftyp*

SKIP

STEPSIZE *delta*

STOP

TEMPERATURE *temper*

Descriptions and default values are the same as the ones described in previous items of this section.

It must be taken into account that, in this case, directives *must* appear with appropriate parameters (**SKIP** and **STOP** have no parameters). Default values are adopted only when both directives and their parameters are absent.

The **RUNTYP** directives must appear in the order in which calculations are desired. It means that, for example, a *runtyp* which needs a density matrix must be following a previous *runtyp* which creates such data in *dumpfile*. Several **RUNTYP** directives can appear in the same input file. The format is:

RUNTYP *runtyp*

RUNTYP keyword can either appear or not explicitly, being present the parameter or sucesive parameters *runtyp*, which are **HAVMOL** kewords, themselves.
The **TIME** directive must be expressed in hours and decimal fractions and the unit term **H** is not allowed. Note that this differs respect to the format in the standard input file, where default units are seconds. Thus, the format is:

TIME *hours*

Default is *hours* = 24.

The **NOTE** directive is used in a similar way as in normal files:

NOTE *comment*

This directive can be used to insert a true comment card as second title of a .CAR file. It discards any further comment or command line and conducts to read numeric data of cartesian coordinates in this kind of files.

Additionally, a special directive is included for this kind of input files:

BASLIB *basfile*

where *basfile* is the name (**without extension**) of an existing .BFN file which contains basis set information for all kinds of atoms in the calculation. It must be taken into account that the same basis set is used for the whole molecule if an standard .BFN file is taken. The default is

MINI1.BFN, which also can be written as **BASLIB MINI1** in directive line. It corresponds to a Huzinaga's contracted gaussian minimal basis set.

An unlimited amount of continuation lines can be included. In order to tell the computer to read a following parameter line, the only requirement is to write as the last word in the preceeding one of the signs **&** or **#** *alone*.

Note that neither **SYMMETRY** and **ATOMS** (or **GEOMETRY**) directives are allowed with alternative input file formats. They are discussed below in *Sections 3* and *4*, respectively.

3.SPECIFICATION OF GEOMETRY AND BASIS SET

3.1.Geometry and basis set general input format.

This section describes the way to input data about the initial geometry of the polyatomic system and the basis set to be used for each constituent atom.

The normal form in which the geometry and basis set are specified is as follows:

ATOMS
unique-centre-specification
unique-centre-specification
.
.
.
END

The program generates symmetry-equivalent centres automatically (see *Section 4* below). A *unique centre specification* takes one of the forms (a) or (b) below:

(a)
Atom-name Znuc position-specification
Basis-specification
END

(b)
Dummy-name -1 position-specification

Where:

atom-name is a character string (max. 8 characters, no spaces)
Znuc is the nuclear charge (which may be zero in the case of ghost orbitals). A negative nuclear charge specifies the dummy, which may be used to simplify the construction of the geometry and which is subsequently ignored.
position-specification is a set of coordinates corresponding to the current atom position and will be described with its alternatives in sect. 3.2.
basis-specification will be described in sect. 3.3.

An alternative form for the **ATOMS** directive is available when the basis set has been specified for a previous calculation and stored in the *dumpfile* <fn>.ED3, or <fn>.ED1, or <fn>.ED5, and it is only necessary to specify a new geometry:

GEOMETRY
Atom-name Znuc position-specification
Atom-name Znuc position-specification
.

.

END

If the **GEOMETRY** directive is used, the nuclear charges must match those given in the previous **ATOMS** directive which set up the basis; it means that the **GEOMETRY** directive can not be used for ghost orbital calculations. Dummies may be used, as with the **ATOMS** directive.

To use stored basis set the run must begin under a **RESTORE** directive. If a **CONTINUE** directive is used, it will restart calculation as a new one, being or not the *dumpfile* present in the default directory. See *Section 5* below.

See also *Section 2*, regarding variable and constant geometric parameters.

3.2.Position specification

The *position-specification* takes one of the forms:

[**XYZ**]	*x*	*y*	*z*
POL	*r*	*theta*	*phi*

The first form specifies cartesian and the second polar coordinates with the usual definition. As it is noted, the **XYZ** cartesian coordinate specification is optional, and only useful if a mixed coordinate is provided in an input set.

3.3.Basis specification

The specification of the basis normally consists of the set of shell-specifications, one for each shell of basis functions, numbered from 1 for each atom. The shell may consist of one of an *s,p*, or *d* set of functions. The list of shell specifications for a single center must be followed by an **END** line.

If the user have data with certain, standard or not, basis sets which will be used currently, he can create DOS ASCII files named as <bfn>.BFN (.BFN is an standard TC-HABANA package extension and it is compulsory in this case) which must contain the values of scale factors, exponents, and coefficients in the same format as demanded by this program. In such a case the user must copy the <bfn>.BFN file to the same DOS drive and directory where the input file is, and follow the *Atom-name Znuc position-specification* card by a single line with the basis file name <bfn> (*without* the .BFN extension) in place of the full *Basis-specification*. Some <bfn>.BFN kind files are supplied with this package.

In such a case, the program will look in the <bfn>.BFN file for a basis function of an element which atomic number must be the same as the one in the input file, and subsequent reading of the unique atom basis function will be done in such auxiliary file, up to its own **END** directive (see one of the enclosed <bfn>.BFN files) for such atom. Then, the control returns to the main input file to continue with the next atom or finish the **ATOMS** directive with the corresponding **END** (see examples below).

General_basis

The specification for each shell takes the form (in both <fn>.IMM or the <bfn>.BFN files):

```
ishell  itype  n  factor
1  a1  c1
2  a2  c2
   .
   .
   .
n   an   cn
```

where:

ishell is the shell number on this atom. For each atom, ishell takes sequential values 1,2,3,...,NSHELL. NSHELL is the total number of s, p, or d shells to be considered in the atom.

itype is the shell type:

S, **1S**, **2S** or **3S** for s shells;
P, **2P** or **3P** for p shells;
D or **3D** for d shells.

1S, **2S**, **3S**, **2P**, **3P**, or **3D** nomenclature must be used to take advantage of the Huckel matrix guess in starting the SCF routine (see **GUESS**, in section 2).

n is the number of contracted primitives in the shell.

factor is a scaling factor to multiply the gaussian exponents of primitives in the shell by *factor***2. The default is 1.

ai is the Gaussian exponential parameter of the ith primitive;

ci is the contraction coefficient for *s*, *p* and *d* shells.

The following is an example of a valence split basis function of Nitrogen:

```
1    1S    3
1      218.36449       0.06787
2      32.59889        0.39020
3      6.91739         0.67008
2    2S    2
1      8.32638         -0.08089
2      0.65919         0.56720
```

```
  3    2S    1
  1      0.21009          0.51109
  4    2P    2
  1      6.12035          0.11592
  2      1.25938          0.46996
  5    2P    1
  1      0.29145          0.61845
END
```

See more examples in section 7.

3.4.Change of coordinate origin

The coordinate origin can be changed with the directive **ORIGIN**. It must be shifted to any of three options: the center of mass, the center of charge, or any desired point in the current framework.

The correspondent directive line is:

ORIGIN [*shift_type*] [**CAR**]

where

shift_type is the kind of traslation desired, which is denoted by any of the following keywords:

COM for a traslation to the **c**enter **o**f **m**ass. It can be ordered during a **ROTATION** run type. If a geometry optimization (**RUNTYP OPTIMIZE**), or a force constant (**RUNTYP FORCE**), or a property (**RUNTYP PROPERTY**) calculation is performed, rotational constants are implicitly calculated, and a shift to the center of mass can also be done. In this case *x*, *y* and *z* coordinates are not necessary and *they must be absent*.

COC for a translation to the center of charge. It can only be ordered during a property run type, to have available molecular charges. In this case *x*, *y* and *z* coordinates are not necessary and *they must be absent*.

It can also be used as:

ORIGIN [*x*   *y*   *z*] [**CAR**]

in the case where a simple traslation to another point during the calculation is desired. Then *x*, *y* and *z* coordinates must be entered. The implicit units are Bohrs. However, if **ANGSTROM** directive is active, the new center of coordinates must be specified in this unit. A move must be asked in the initial input *after reading atomic coordinates* by any of the **ATOMS** or **GEOMETRY** directives, and ending the corresponding block with the **END** directive.

In both cases the **CAR** directive asks for printing the cartesian coordinates in a .CAR file format.

If a <fn>.CAR file exists, it will be appended with the shifted coordinates.

4.SYMMETRY SPECIFICATION

This section describes directives for molecular symmetry specifications which could be very convenient for saving computing time when equivalent atoms can be generated by symmetry operations in a given polyatomic system. However, *it can not be used with alternative input file formats,* as they are described in *Section 7*.

In default, any system is assumed to have C1 symmetry, i.e. no symmetry. In such a case, the program makes no attempt to find any symmetry. If symmetry is present, it must be specified explicitly for the program to take advantage of it. If a **SYMMETRY** directive is included, it must come before the geometry specification, which will be described in *Section 4*. The symmetry is defined as follows:

SYMMETRY
symbol n
<x1 y1 z1   <x2 y2 z2> >
<<x3 y3 z3> <parameter>>
END

where:

symbol is the Shoenfiles symbol of the symmetry point group, according to the following code:

C1, Cs, Ci, Cn, Cnh, Cnv,
Dn, Dnh, Dnd,
S2n,
T, Th, Td,
O

and *n* is the order of principal axis, where relevant.

Molecular systems corresponding to relevant point groups can be built according fixed relationships with respect to a cartesian coordinate set. Therefore,

C_{nv} point groups:	the generator ${}_{v}$ plane is the (x,z) plane of the local frame
C_{nh} and dihedral groups:	the generator ${}_{h}$ plane is the (x,y) of the local frame
Dihedral groups:	the generator C_2 axis is the x-axis of the local frame.

The principal axis of the molecule is defined using (*x1*,*y1*,*z1*) and (*x2*,*y2*,*z2*). The x-axis of the molecular symmetry is defined with the aid of (*x3*,*y3*,*z3*). *Therefore, x1,y1,z1, x2,y2,z2, x3,y3,z3* are parameters which are used to specify the orientation of the local (i.e. molecular) frame relative to the master (space-fixed) frame. The default values of *x1,y1,z1* etc., are,

(*x1*,*y1*,*z1*) = (0,0,0)
(*x2*,*y2*,*z2*) = (0,0,1)
(*x3*,*y3*,*z3*) = (1,0,0)

If the molecular frame coincides with the master frame, as specified above, no values of *x1,y1*, etc. need to be specified.

To change the orientation from the default, supply information as follows:

C1 group: no additional information required. Supply **END** card only. Note, however, that the entire symmetry specification can be omitted in this case.

Cs group: (*x1,y1,z1*) etc. are any three points in the symmetry plane. (Default: put the molecule in the xz plane).

Ci group: (*x1,y1,z1*) is the center of inversion. (Default: centre of inversion is the origin)

Cn, Cnh, Cnv, Dn, Dnh, Dnd, S2n axial groups: (*x1,y1,z1*) and (*x2,y2,z2*) are any two points on the principal axis of rotation. (Default: the principal axis coincides with the z-axis).

T, Th, Td tetrahedral groups: (*x1,y1,z1*) and (*x2,y2,z2*) are any two points on a two-fold axis.

O octahedral group: (*x1,y1,z1*) and (*x2,y2,z2*) are any two points on a four-fold axis.

The perpendicular to the principal axis of rotation passing through the third point (*x3,y3,z3*) defines a direction called *d1*. If *parameter* = **PARALLEL** (default), the x-axis of the local frame coincides with the direction *d1*. If *parameter* = **NORMAL**, then the x-axis of the local frame is the common perpendicular to *d1* and the principal axis, passing through the intersection point of these two lines. Thus *d1* coincides in this case with the negative *y* axis.

Notes:

Previous experiences in parent versions of this program recommend the following aspects:

(i) Choice of symmetry group

As far as the integral and SCF routines are concerned the only symmetry elements which are useful are those which exchange atoms. Thus in a linear molecule like HCl there are no usable symmetry elements, and in a centrosymmetric molecule such as N2 there is no point in using D_{2h} as C_i will do just as well.

(ii) An important restriction

In constructing the Fock matrix, the program assumes that the wave function belongs to a non-degenerate representation of the molecular point group, i.e. it assumes that the density matrix transforms as the totally symmetric representation. If this is not true, the SCF procedure could give ridiculous results (usually a very large negative value for energy). *In cases where the ground state wave function is genuinely spatially degenerate, this problem can be avoided by using a non-degenerate subgroup of the true molecular point-group.* In some cases, because the

SCF procedure can alter the order of the orbitals, it may happen that the SCF procedure will pass through a spatially degenerate state on the way to convergence, even though this is not the state intended as the final solution. This occurs very rarely but, if it does, SCF procedure may collapse (sometimes it recovers after a few cycles).

Another way to avoid this undesirable effect is causing the SCF procedure to take a different level shifters, etc. Note that the fewer times the programs swaps orbital order, the less likely this is to happen. See comments under **LOCK**, **SWAP** and **SHIFT** options in *Section 2*, above.

(iii) A warning

Since only unique 2-electron integrals are stored on the *mainfile* <fn>.ED2 (or ED0, or ED4), the structure of this file is dependent upon the choice of group. This means that it is difficult to change the point group once the *mainfile* has been calculated, unless the symmetry operations added or taken away do not interchange any nuclei. The symmetry is also used in constructing the Fock matrix from the 2-electron integrals. Obviously, therefore if one is intending a series of calculations using the same integral file, the lowest symmetry must be used from the outset.

(iv) A hint

To test correctness of input symmetry and the corresponding atom definition it is strongly recommended to perform, at first, a **CHECK** run type. The output of such a run demonstrates how symmetric atoms are generated and helps very much to avoid mistakes.

(v) False overflows

When symmetry information is used to generate equivalent atoms, a mistake in such input can give an unexpected overflow in the allowed number of atoms, basis functions, or shells. It is due to an abnormal increase in the total number of atoms, which are generated internally by the program, due to symmetry operations. In such cases, output error flag should be absent of any advice respect to problems with symmetry directives, being them the truly origin of error.

5.BEGINNING AND RESTARTING THE CALCULATION

This section describes directives for starting a given calculation from a standard input file.

5.1. Running the program.

The program uses data from the file <fn>.IMM and produces output in a file called <fn>.OMM by default (see description in *Section 2*). It also produces certain messages on the screen, so that the user can check on the progress and major time consumption of his job. The *dumpfile* and the *mainfile* must be kept after the run if the job is to be restarted; otherwise all that is necessary to run a job is to type:

>havmol

or

>havmole

or double marking the file **havmolw.exe** in WINDOWS file manager, depending which option is available and desired.

5.2. Starting directives.

The starting directive is:

START

which initiates a calculation of the type as specified in the **RUNTYP** directive (see *Section 2*, above), or, if no calculation type has been explicitly mentioned, does an **HFSCF** calculation.

When the specified calculation has been completed, the program reads further data from the input file (i.e., the <fn>.IMM file remains opened during the whole calculation). To perform another **RUNTYP** the corresponding directive lines must be entered, and eventual new parameters. All other data will be read from the previous **RUNTYP** in the dumpfile.

5.3. Stream calculations.

To beguin a calculation of a new system in the stream of the same input file, one of the following *equivalent* directives must appear:

CONTINUE
NEW

Then, a new set of input data for another polyatomic system can be read. In this case, former *dumpfile* and *mainfile* are deleted and a new calculation is initialized.

If no further calculations are required, one of the following directives can be used:
FINISH
END

which are equivalent and causes the program to stop. If one of these directives are presented after the geometry has been input, i.e. without the **START** directive, input data is processed, printed out, and the run stops. This is useful for checking the input data. It is equivalent to a **CHECK** run type. *If these directives are used, when the calculation is finished after obtaining the desired output, files <fn>.ED2 (or .ED0, or .ED4) and <fn>.ED3 (or .ED1, or .ED5) are deleted.*

If deletion of *dumpfile* and *mainfile* is not desired, the run must be finished with:

STOP

which causes the same effect as **FINISH** or **END**, but avoids loosing such files. If this option is selected such files must be planned to be used for further calculations.

This is particularly useful *i)* for ghost orbital calculations with the **SKIP** directive, *ii)* to restart a hardly converging new calculation with **GUESS RESTORE** from an auxiliary *dumpfile, iii)* to obtain isotopic effects in vibrational modes with the **WEIGHTS** directive.

5.4.Internal restarting features.

The program monitors the elapsed time while running. If this approaches the overall time limit, it closes down the job in a controlled manner, placing the information required to restart the job into the *dumpfile*. When an interruption of the run occurs due to the reach of time limit, both *dumpfile* and *mainfile* are preserved, being or not active the **STOP** directive.

The sections monitored most carefully are those which consume the bulk of the time i.e. the two-electron integrals, the SCF routines, and the two-electron contributions to the gradient. Provided the program closes down correctly and that the necessary files have been saved, these sections will restart automatically from the point where they stopped. For other less time consuming sections, such as the 1-electron integrals, the program simply checks that is not very close to the overall time limit before starting. Consequently, it is possible to run out of time in these sections. If this occurs, the program should be restarted from the beginning of the relevant section.

The files which have to be kept to ensure a successful restart are as follows,

Integrals -	*dumpfile* and *mainfile*
SCF -	*dumpfile* and *mainfile*
Gradient -	*dumpfile* only

No permanent information, at all, is kept on the *scratchfile*, so this is never to be saved.

5.5. Restarting directives. Storage of restarting data.

The restarting data file must be written as the former <fn>.IMM file. However, it can be named as <fn>.IM1, which is always the default extension for restarting input files. This is done in order to save the initial input file for further use or modification.

When restarting input comes from an existing <fn>.IM1 file, to restart a job the first line will be,

RESTORE [[*<fn>.EDn*] *iblk*]

This causes the program to restore all information about the job from the *dumpfile*. If the *dumpfile* is not on <fn>.ED3 (or <fn>.ED1, or <fn>.ED5) then, appropriate values of *<fn>.EDn* and *iblk* must be included, where *iblk* is the block number in such direct access file from where dump data must be restored.

WARNING: If an old dumpfile is used with an existing <fn>.IM1 file without the **RESTORE** command, the program has no way of knowing that it is and old dumpfile, and consequently will procceed as it is where a new dumpfile, and will reinitialize certain sections. Accordingly, if there is anything on a dumpfile which the restarted job needs, then the data must begin with **RESTORE**.

Further directives from *Section 2* (above) may then be given if it is wished to change any of the parameters for the job. The **RUNTYP** type of calculation may be changed using one of the directives in this section (though, of course, if the old calculation had not finished completely nonsense may result from doing this). The job restarts on encountering an **START** directive of the form:

START [*irest*]

If the parameter *irest* is omitted the program restarts from the point where it left off. Giving a value to *irest* forces the program to begin again at a specified point, as follows,

irest = 0, restart the calculation from the beginning. (For optimizations this means from the beginning of the current energy evaluation).

irest = 1, 2-electron integral restart (1-electron integrals saved on *dumpfile*); the restart point is saved on the *dumpfile*. Obviously, if you wish to restart during the 2-electron integral evaluation, the *mainfile* must have been kept as well as the *dumpfile*.

irest = 2, restart the SCF (1-electron integrals saved on *dumpfile*; 2-electron saved on *mainfile*). This option generates a new initial guess and begins the SCF procedure again, which is useful if the previous SCF failed to converge to the correct state.

irest = 3, restart during SCF. Used to continue a non-converged SCF from the point it stopped.

irest = 4, restart at the beginning of 1-electron section of gradient evaluation.
irest = 5, restart during 2-electron section of gradient evaluation.

For the simplest restart, it is only necessary to provide the data file:

```
RESTORE
START
FINISH
```

The following are examples of jobs in which the parameters are changed and the restart point forced:

- to increase the convergence of the SCF procedure from the default of 5:

```
RESTORE
CONVERGE 7
START 3
END
```

- to calculate a gradient when the dumpfile contains the results of an SCF run:

```
RESTORE
TITLE
Evaluate gradient
GRADIENT
START 4
FINISH
```

6. CONSOLE AND COMMAND LINE INPUT AND OUTPUT.

6.1. Restoring or beginning a job.

All versions of **HAVMOL** receive some input data from the computer console. The only difference among them is the presentation of certain questions when the MS WINDOWS variant is used, which implements system facilities for such purpose in the sake of clarity and the use of mouse. On the other hand, console input can be substituted by command line input in the case of **HAVMOL** and **HAVMOLE**.

The first question to the user is:

Are you trying to restore a previous calculation? (Y/[N])

Console input data begins with a definition of a *new* or *restoring* job. It is answered with **Y**es or **N**ot, accordingly. Default is **N**ot.

6.1.1. Restoring jobs.

In the case of a restoring job, other two options are asked to the user which are related with the kind of main input file to use for restart. An special feature for entering data is implemented in all **HAVMOL** programs which allows the user to enter directives directly to a volatile memory buffer. In this case, a restoring input file is not necessary because it is constructed *on line* in the computer core. The other option is to enter the restoring data form a file. A question is asked from console to enter restoring data:

Do you want to enter restart commands from the keyboard? (Y/[N])

It is answered with **Y**es or **N**ot, accordingly. Default is **N**ot.

6.1.2. The internal restoring file.

In the case when the user prefers to enter data for restoring to a file in volatile memory, immedialetly after answering the later questions appears:

Enter up to 47 lines of HAVMOL commands (including the START <n> and FINISH
or STOP lines, RESTORE must NOT be written). SYMMETRY directives can not be
changed.
=>_

A **=>** prompt in console appears asking for directives to be written. In this case, directives are allowed up to 47 lines of 80 characters, each one. Their rules are the same as described in *Sections 2, 3,* and *5*. **SYMMETRY** directive is *not* alowed in this input file. Entering data finishes with any of the **STOP** or **FINISH** keywords. **RESTORE** *directive can not be written, because it is included by the program.*

6.2. New jobs: options for input files.

Different kinds of formats for input files are allowed in **HAVMOL**. Standard .IMM file format, as described in all previous sections, is the default. However, Cartesian and internal coordinate input file formats are allowed for new jobs. The kind of input file is characterized by certain standard file name extensions. It is explained by a question in console:

```
Enter the code for a kind of main input file:
  HAVMOL (.IMM)                    [1]
  Cartesian coordimate (CAR)        2
  Internal coordinate (IMC)         3
  AMPAC (AMI)                       4
  Z matrix (ZMT)                    5
=>
```

Details of each kind of input file format are described in *Section 7* below.

6.3. Input file name.

The corresponding input file is then asked to be named, either if this is the one for starting a new job, or it is a restoring true file, or it is a restoring volatile memory dummy one:

```
Enter the main|dummy input file name <fn>[.IM?] =>
```

In the case of restoring jobs its name *must* coincide with the name of existing *main* and *dump-files*, to allow data recovering from them in the new run. In any case, the input file is called with the default standard file name extension .IMM, for starting a job, or .IM1 for restoring it.

If the input comes from the console by the *dummy* <fn>.IM1 file, the **RESTORE** directive is not necessary and it *must not be entered*. In such a case, the default <fn>.ED3, or <fn>.ED1, or <fn>.ED5 files are used to begin from the *iblk* = 1.

6.4. Output file name.

In any case, *the output file name is not asked to the user*. The default name is the same as the input, with a changed extension. For starting jobs the default extension is .OMM and for restarting runs, it is *.nnn*, where this is the number of the run in the restarting job, considering the initial run as 0.

6.5. Location of scratch files.

The next entry is devoted to locate the scratch file:

```
If you have a virtual disk with sufficient capacity for the scratch file,
enter now the DOS virtual drive and path to write such scratch file. The
default is the current drive =>
```

The default location is the active drive and directory. However, it is convenient to locate the scratch file in the fastest input/output drive, to save much time during SCF iterative procedures. Many matrices are taken from permanent *dumpfile* and located in the scratch file, during the Fock matrix building process, and this handling is considerably faster in fast devices. The more adequate of them are virtual drives. A typical personal computer used for this kind of calculations can do things much better if they are provided with an active virtual drive device driver. They are present in the most modern versions of DOS, as RAMDRIVE.SYS or VDISK.SYS.

Thus, this demand of input from the console must be supplied with the path of the virtual disk, if it is present. If not, it can contain a directory path in the fastest disk of the computer.

6.6. Console output.

Due to the fact that **HAVMOL** system is intended not only for research, but teaching quantum chemistry, it is provided with an abundant console output. It means that the user is permanently informed about the active step of each calculation, in any moment, and some results can be obtained directly from the console, during the run. It is particularly useful in WINDOWS version. However, it takes a significant amount of time in calculations due to all the internal processes that must be followed for such output to console in any computer. Therefore, a way to inhibit the major part of the console output is provided, frequently resulting in a dramatic increase of the calculation speed. Moreover, all screen output is always saved in the main output file, according to the selected option of **PRINT** directive.

This is the reason because the last console input is to answer if the console output is supressed or not:

Do you want to supress screen output? (Y/[N])

It is recommended to take the default full console output if the program is used for teaching, or it is wanted a carefully look of the run. It means **N**o supression of screen output (default). In other cases, where results are wanted as fast as possible, the answer to this question must be **Y**es, to supress the screen output.

6.7. IMM file output.

When alternative input file format is used, the program allows to output a standard .IMM file which is absolutely equivalent to selected options and geometry. It can be used for further runs, to be modified and chained with other input data. Its creation is asked from the console to the user by:

Do you want to create a <fn>[.IMM] file? (Y/[N])

It is answered with **Y**es or **N**ot, accordingly. Default is **N**ot.

6.8. Command line input.

In order to make more easy batch runs where no interactive data can be used, command line input is accepted for **HAVMOL** and **HAVMOLE**. It can supply all information to load the program and start any run in most cases, except when a volatile memory input file is used to restore a job. In this case, command line can provide only partial details, because console interaction with the user is compulsory to create the dummy file.

The command line input syntax is the following:

HAVMOL *[job] [file] [screen] [scratch file path] [create input file]*

or

HAVMOLE *[job] [file] [screen] [scratch file path] [create input file]*

The meaning of items in command line is the following,

The *job* can be either **RESTORE** or **NEW** keywords. **RESTORE** indicates to begin a restoring run of an already started job, which means the same as answering **Y** to the first console question (see this section above). **NEW** means that the run is a new job.

The *file* could be either an input file name, or the **INTERNAL** directive. In the first case the default extension is .IMM. If one of the extensions which are designed by default for alternative input file formats (see this section above) is used, such format will be expected. When **INTERNAL** directive is used it is undestood that a restoring run is tried, and that the user will enter data to the volatile memory input file. In such a case no more command line directives are read, and the user must continue input from the console.

The *screen* must be **NOSCREENOUT** directive if the supression of most screen output is desired. **SCREENOUT** directive (or any other character string) is compulsory if more data is to be entered from command line.

The *scratch file path* is a string with the DOS path where scratch file is desired to be located during the run.

The *create input file* must be **IMMOUT** when an alternative input file format is used for input (which has been indicated previously by the input file name extension) and an IMM file is desired to be created with the equivalent standard input data. If it is not desired, it can be left without any further command line word.

7. ALTERNATIVE INPUT FILE FORMATS

7.1. Cartesian coordinate file input.

7.1.1. Cartesian coordinate files in the program package.

HAVMOL is part of a full package of programs for modelling molecules and chemical processes in personal computers named as TC-HABANA. This system of programs is characterized by a normalized set of input/output file formats to which belongs the above mentioned IMM, OMM, and BFN ones. One of the more common files in this system is the one containing cartesian coordinates of a polyatomic system, which is used by almost all the programs. It is called as its extension .CAR, and it have a fixed format. It is used as output for optimized geometries in the former Havana's version of MICROMOL, as well as in the present program.

The shortest and easiest input file for **HAVMOL** is also a typical .CAR file, which can be obtained as output from other programs, like the one of molecular mechanics, or the semiempirical **HAPAC**, or from a previous run of this program, .

7.1.2.Cartesian coordinate file format (.CAR).

These files are very simple, containing a single number with the amount of atoms on the system in the first line, and two comment cards, which can use up to 80 columns, each one. Then, follows a set of lines, one for each atom up to the amount registered in the first line. Each atom line contains *x*, *y* and *z* coordinates and the atomic number of the corresponding element. Units in the cartesian coordinate frames are Angstroms. A detailed description follows:

i) a card or line with the number of atoms between columns 1 to 4,

ii) a card with titles,

iii) a second title car, which is used for comments in other programs, and it is taken as *command line* for **HAVMOL** input, with the posibility to be increased with continuation lines (allowed directives which can be written in this line, one after the other, are described above in section 2),

iv) and a set of cartesian coordinate and atomic number lines, one for each atom. The format of single atom cartesian coordinate cards is *x* (cols.1 to 11), *y* (cols. 12 to 21), *z* (cols.22 to 31) and atomic number (cols. 32 to 35). Units of cartesian coordinates are Angstroms.

7.1.3.Limitations.

The presentation of **HAVMOL** in the computer console allows the user to deviate the input to a CAR file from the beginnings of its execution. Default is the above described IMM file.

Cartesian coordinate files used as input present the limitation of a reduced set of directives and this version allows the exclusive use of it for beginning a calculation, not for a series, nor continuation run. *A continuation of a run which have been started from a .CAR cartesian coordinate* *file must be entered in the usual .IM1 file format, as described above.* I f a batch run is desired of various sucessive atomic conformations, an operating system batch run file (.BAT) can be prepared with command line inputs for each run of **HAVMOL**.

7.2. Internal coordinate file input.

Three different standard formats of internal coordinates are allowed to enter data for **HAVMOL**: TC-HABANA formatted internal coordinate (.IMC standard DOS extension), AMPAC-MOPAC free format (.AMI standard DOS extension), and Gaussian's Z matrix free format (ZMT).

There are *3N-6* internal coordinates, where *N* is the total number of atoms in the system. To define the coordinates of a certain D atom, it is necessary to refer three already localized A, B, and C atoms. Atom D is always related directly to C. Dummy atoms can be used to create a ghost point to which refer a given atom. The atomic number of dummy atoms must be set to 99.

The first internal coordinate is always the interatomic distance in Angstroms between the target D and a previously localized C atoms. This is usually, although not necessarily, the bond distance between both atoms.

The second internal coordinate is the planar angle in degrees between the segments C-D and C-B, where B is another already localized atom related to C. The apex of such angle must be in C. This is usually the bond angle BCD.

The third internal coordinate is the dihedral or twist angle between semiplanes defined by atoms ABC on one hand, and BCD on the other hand, with an axis defined by the segment between B and C. Angles of rotation across this axis are in degrees and clockwise positive in the B-C direction. Null angle value implies that A and D as vicinals and the four atoms A, B, C, and D in the same plane. 180 degrees is consequently the opposite, being A and D in an anti conformation.

All input angles are in degrees and decimal fractions of them (not minutes, nor seconds).

It is recommended to draw an outline of the polyatomic system with an appropriate ordinal numeration for each atom. An user owned conventional order of different kinds of atoms is recommended to his convenience. A table is included in the appendix with approximate interatomic distances for the most common organic bond lengths.

7.2.1. Formatted IMC internal coordinate files.

The first card, or cards, of a formatted IMC input file are *command lines*, in the sense of the second title card of cartesian coordinate .CAR files (see above). The only difference is that they *must* begin with the > sign. If a continuation line exists, the previous line must be ended with **&** or # signs. Formats of command lines are free, although geometry data are strictly formatted in IMC files.

After the *command line* (or lines) a title line is compulsory. It must contain any information in ASCII characters to identify the run.

In the following description of .IMC files a rigid FORTRAN type format is required and instructions include the input variable name, the field where it must be written (in terms of columns), the type - real (R) which is recommended to include a decimal point, or integer (I) which point *must* be absent -, and the meaning of the value of such input variable. Blank fields are read as null values for both real and integer variables.

The first line or card must contain data for up to the first three atoms in the polyatomic system:

	cols.	type	description
<nat(i)> *i=1,3*	1- 3 4- 6 7- 9	I	Atomic numbers of the first three atoms, in their appropriate order, which are being taken as references for the whole molecule. Atom 1 will be in the center of the Cartesian coordinate system. The segment 1,2 is colinear with the X axis. For diatomics, nat(3) must be null or blank.
<kwik>	10-12	I	is an 123 planar angle switch. If: = 0 angle 123 (apex in 2) is tetrahedral (109.47 deg.) = 1 angle 123 is trigonal (120 deg.) = 2 angle 123 is explicit in this input line or card as *<theta(3)>*.
<r(2)>	13-19	R	is the interatomic distance between atoms 1 and 2. It is an internal coordinate of atom 2.
<r(3)>	20-26	R	is the interatomic distance between atoms 2 and 3. It is an internal coordinate of atom 3.
<theta(3)>	27-33	R	is the angle 123 (with the apex in 2). It must be entered only if *<kwik>* is 2. It is an internal coordinate of atom 3.

All the further atom definition line or cards have the same format, one for each new atom, to be finished with an **END** directive.

<*na*>,<*nb*>,<*nc*> 1- 3 I are the assigned numbers in the
4- 6 polyatomic system to the three
7- 9 reference atoms for defining the
current atom which corresponds to
this line or card. All of them must
be defined previously.

<*nd*> 10-12 I is the current number of the atom to
be defined in this line or card. Its
corresponding internal coordinates
will be defined respect to the reference atoms <*na*>, <*nb*>, and <*nc*>.

<*nat(nd)*> 13-15 I is the atomic number of the element
corresponding to atom <*nd*>.

<*kwikd*> 16-18 I is the angle switch of atom <*nd*>.In
order to make more easy the input,
it can take the following values:

= 0, 1, 2, 3, 4, 5 if the planar
angle formed by atoms
<*nb*><*nc*><*nd*> with the apex in
<*nc*> is tetrahedral (109.47 deg.) and
dihedral angles corresponding to a
clockwise rotation between the planes
formed by *[<na>,<nb>,<nc>]* on one
hand, and *[<nb>,<nc>,<nd>]* on the
other, by the <*nb*>,<*nc*> axis are 0
(cis), 60, 120, 180 (trans), 240, and 300
deg. respectively.

= 6, 7 if the angle <*nb*><*nc*><*nd*>
is trigonal (120 deg.) and dihedral
angle is 0 or 180 deg., (cis or trans)
respectively.

= 8 when atoms <*nb*>, <*nc*>, and
<*nd*> are colinear. In this case, it results impossible the further use of
<*nb*>, <*nc*>, <*nd*> as reference
atoms in a single line or card, because
the dihedral angle is not defined.

= 9 when the value of any or both

			the previously referred angles must be defined by the user explicitly.
<r(nd)>	19-25	R	is the interatomic distance between atoms <*nc*> and <*nd*>. It is an internal coordinate of atom *nd*.
<theta(nd)>	26-32	R	is the angle <*nb*><*nc*><*nd*> (with the apex in <*nc*>). Only necessary if <*kwicd*> is 9. It is an internal coordinate of atom *nd*.
<phi(nd)>	33-39	R	is the dihedral or twist angle <*na*><*nb*><*nc*><*nd*> (with the rotation axis in <*nb*><*nc*>). Only necessary if <*kwicd*> is 9. It is an internal coordinate of atom *nd*.

7.2.2. Free format IAM (AMPAC or MOPAC) internal coordinate files.

The first card, or cards, of a free format IAM input file are *command lines*, in the sense of the second title card of cartesian coordinate .CAR files (see above). The only difference is that they *must* begin with the **>** sign. If a continuation line exists, the previous line must be ended with **&** or **#** signs. Formats of command lines are free.

After the *command line* (or lines) a title line is compulsory between it (or them) and numeric data of internal atomic coordinates. It must contain any information in ASCII characters to identify the run.

The way that internal coordinates are entered is essentially the same as the one used in AMPAC or MOPAC programs. Input data can be written in any place of each line (free format), and atomic numbers can be replaced, if it is wanted, by atomic symbols, indistinctly. Any input parameter must be separated from the other by blank spaces or commas.

The end of geometry input is marked by a single **END** directive in a line or card.

For each atom, the following data must appear in a single line or card:

<nat(nd)> is the atomic number (or symbol) of the element corresponding to atom <*nd*>.

<r(nd)> is the interatomic distance between the reference atom <*nc*> and the current <*nd*>. It is an internal coordinate of atom *nd*.

<word> is any string of alphanumeric characters.

<theta(nd)> is the angle *<nb><nc><nd>* (with the apex in *<nc>*), where *<nb>* is another reference atom. It is an internal coordinate of atom *nd*.

<word> is any string of alphanumeric characters.

<phi(nd)> is the dihedral angle *<na><nb><nc><nd>* (with the rotation axis in *<nb><nc>*). It is an internal coordinate of atom *nd*.

<nc>, <nb>, *<na>* are the assigned numbers in the polyatomic system to the three reference atoms which define the atom corresponding to this line or card. All of them must be defined in a previous line.

The first line of atomic internal coordinate data only includes the atomic number (or symbol) of atom 1. The second one, the atomic number (or symbol) of atom 2, and the distance between atom 2 and atom 1. The third line contains the atomic number (or symbol) of atom 3, the distance between atoms 2 and 3, the corresponding <*word*> and the bond angle 123.

Those <*words*> can be any string of characters, without blanks, and are present to allow compatibility with optimization parameters in AMPAC input files.

7.2.3. Free format ZMT internal coordinate files.

The first card, or cards, of a free format ZMT input file are *command lines*, in the sense of the second title card of cartesian coordinate .CAR files (see above). The only difference is that they *must* begin with the **>** sign. If a continuation line exists, the previous line must be ended with **&** or **#** signs. Formats of command lines are free.

After the *command line* (or lines) a title line is compulsory between it (or them) and numeric data of internal atomic coordinates. It must contain any information in ASCII characters to identify the run.

The way that internal coordinates are entered is essentially the same as used in GAUSSIAN programs, with some limitations respect to the latest versions. The allowed facilities are *only* the ones described in this section. Input data can be written in any place of each line (free format), and atomic numbers can be replaced, if it is wanted, by atomic symbols, indistinctly. Any input parameter must be separated from the other by blank spaces or commas.

The end of geometry input is marked by a single **END** directive in a line or card.

For each atom, the following data must appear in a single line or card:

<nat(nd)> is the atomic number (or symbol) of the element corresponding to atom *<nd>*.

<nc> is the assigned number in the polyatomic system to the atom taken as first reference with respect to *nd*. It must be defined in a previous line.

<r(nd)> is the interatomic distance between the reference atom <*nc*> and the current <*nd*>. It is an internal coordinate of atom *nd*.

<nb> is the assigned number in the polyatomic system to the atom taken as second reference with respect to *nd*. It must be defined in a previous line.

<theta(nd)> is the angle <*nb*><*nc*><*nd*> (with the apex in <*nc*>). It is an internal coordinate of atom *nd*.

<na> is the assigned number in the polyatomic system to the atom taken as third reference with respect to *nd*. It must be defined in a previous line.

<phi(nd)> is the dihedral angle <*na*><*nb*><*nc*><*nd*> (with the rotation axis in <*nb*><*nc*>). It is an internal coordinate of atom *nd*.

The first line of atomic internal coordinate data only includes the atomic number (or symbol) of atom 1. The second the atomic number (or symbol) of atom 2, the *nc* number (which *must* be 1) and the distance between atom 2 and atom 1. The third line contains the atomic number (or symbol) of atom 3, the *nc* number (which *must* be 2), the distance between atoms 2 and 3, the *nb* number (which *must* be 1), and the bond angle 123.

7.3. Comments on alternative input file formats.

1. In the case of alternative input file formats, if a directive is included and it needs an associated parameter, it *must* appear. Default values are only adopted if both directive and parameter are absent.

2. A simple .CAR file with an empty second title card will be taken as an input for a single HFSCF with MINI1 basis set. Other default parameters are the ones in *Section 2*.

3. A simple .IMC, .IAM, or .ZMT file with an empty first command card *must* begin it with the **>** sign. It will be taken as an input for a single HFSCF with MINI1 basis set. Other default parameters are the ones of each in *Section 2*.

5. In the case of all kinds of alternative format input files, the program can output an usual .IMM file which is completely equivalent as input to the original one. A possible action for cases where further calculations are needed on a certain polyatomic system is to use **RUNTYP CHECK** with the alternative format for obtaining a test output alone and the corresponding .IMM file. In such a case, this newly generated input file can be modified with new options, if desired, according its greater flexibility and allowances, and can become the main input file.

8.EXAMPLES OF INPUT FILES

8.1. Examples of standard input files.

1) Calculates the one-and two electron integrals for Hydrogen Chloride, using a 6-31G basis set.*

```
TITLE
HYDROGEN CHLORIDE  6-31G*
ATOMS
CHLORINE  17.0  0.0  0.0  0.0
 1  S  6
1  25180.1  0.001833
2  3780.35  0.014034
3  860.474  0.069097
4  242.145  0.237452
5  77.3349  0.483034
6  26.2470  0.339856
 2  S  6
1  491.765  -0.002297
2  116.984  -0.030714
3  37.4153  -0.112528
4  13.7834   0.045163
5  5.45215   0.589353
6  2.22588   0.465206
 3  S  3
1  3.18649  -0.251830
2  1.14427   0.061589
3  0.420377  1.060180
  4  S  1
1  0.142657  1.000000
  5  P  6
1  491.765   0.003989
2  116.984   0.030318
3  37.4153   0.129880
4  13.7834   0.327951
5  5.45215   0.453527
6  2.22588   0.252154
  6 P  3
1  3.18649  -0.014299
2  1.14427   0.323572
3  0.420377  0.743507
 7  P  1
1  0.142657  1.000000
 8  D  1
1  1.00   1.00
END
```

```
HYDROGEN  1.0  2.407512  0.0  0.0
 1  S  3
1  13.00773  0.033495
2  1.962079  0.234727
3  0.444529  0.813757
 2  S  1
1  0.121949  1.000000
END
END
RUNTYP INTEGRAL
START
FINISH
```

If basis function library files exist containing non polarized and polarized 6-31G basis sets, and their DOS file names are 631G.BFN and 631GPD.BFN, respectively, the same input data could be simplified to:

```
TITLE
HYDROGEN CHLORIDE  6-31G*
ATOMS
CHLORINE  17.0  0.0  0.0  0.0
631GPD
HYDROGEN  1.0  2.407512  0.0  0.0
631G
END
RUNTYP INTEGRAL
START
FINISH
```

2) Calculation of the energy of HCO^{+1} and gradients using a double zeta plus polarization basis set.

```
TITLE
    DOUBLE ZETA + POLARIZATION BASIS HCO+
PRINT -1
RUNTYP GRADIENT
CHARGE +1
CONVERGENCE 7
MAXIT 150
ATOMS
C1       6.0       0.0       0.0       0.0
  1    S     4
  1  4232.6100  0.006228
  2   634.8820  0.047676
  3   146.0970  0.231439
  4    42.4974  0.789108
  2    S    2
  1    14.1892  0.791751
  2     1.9666  0.321870
  3    S    1
  1    5.1477  1.0
  4    S    1
  1    0.4962  1.0
  5    S    1
  1    0.1533  1.0
  6    P    4
  1   18.1577  0.018534
  2    3.9864  0.115442
  3    1.1429  0.386206
  4    0.3594  0.640089
  7    P    1
  1    0.1146  1.0
  8    D    1
  1    0.6260   1.0
END
02        8.0       0.0       0.0       2.09801203
    1  S    4
    1  7816.5400   0.006436
    1  1175.8200   0.048924
    2   273.1880   0.233819
    4    81.1696   0.784798
    2    S    2
    1    27.1836   0.803381
    2     3.4136   0.316720
    3   S   1
    1     9.5322   1.0
    4   S   1
    1     0.9398   1.0
    5   S   1
    1     0.2846   1.0
    6   P   4
    1    35.1832   0.01958
    2     7.9040   0.124189
    3     2.3051   0.394727
    4     0.7171   0.627375
    7   P   1
    1     0.2137   1.0
    8   D   1
    1     1.2920   1.0
END
H3        1.0       0.0       0.0       -2.06230211
    1  S  2
```

```
   1   19.2406  0.130844
   2    2.8992  0.921539
   2  S     1
   1    0.6534  1.0
   3  S     1
   1    0.1776  1.0
   4  P     1
   1    0.7500  1.0
END
END
START
FINISH
```

3) Calculation of the energy of chloro benzene using a Huzinaga-Dunning contracted double zeta basis set, in the minimal basis STO-3G equilibrium geometry. The basis function data file is HD9S4S.BFN.

```
TITLE
C6H5Cl, (HUZINAGA/DUNNING |STO-3G)
RUNTYP     OPTIMIZE
CONVERGE    7
TIME        51000
PRINT       0
SYMMETRY
CNV 2
END
ANGSTROM
ATOMS
Carbon  6.0     .00000    .00000    .04233
HD9S4S
Carbon  6.0    1.21457    .00000    .71332
HD9S4S
Carbon  6.0    1.20332    .00000   2.10016
HD9S4S
Carbon  6.0     .00000    .00000   2.79160
HD9S4S
Chlorine 17.0   .00000    .00000  -1.75860
HD9S4S
Hydrogen  1.0  2.14480    .00000    .15923
HD9S4S
Hydrogen  1.0  2.14349    .00000   2.63777
HD9S4S
Hydrogen  1.0   .00000    .00000   3.87371
HD9S4S
END
START
```

```
FINISH
```

The library file with the basis function is the following:

```
Hydrogen 1.0 HUZINAGA-DUNNING (9S/2S) SCALED BY 1.2
    1    S    3
    1     19.2406     0.032828
    2      2.8992     0.231208
    3      0.6534     0.817238
    2    S    1
    1      0.1776     1.0
END
Boron 5.0 HUZINAGA-DUNNING (9S/4S)
    1    S    6
    1   2788.4100     0.002122
    2    419.0390     0.016171
    3     96.4683     0.078356
    4     28.0694     0.263250
    5      9.3760     0.596729
    6      1.3057     0.230397
    2    S    1
    1      3.4062     1.0
    3    S    1
    1      0.3245     1.0
    4    S    1
    1      0.1022     1.0
    5    P    4
    1     11.3413     0.017987
    2      2.4360     0.110339
    3      0.6836     0.383111
    4      0.2134     0.647860
    6    P    1
    1      0.0701     1.0
END
Carbon 6.0 HUZINAGA-DUNNING (9S/4S)
    1    S    6
    1   4232.6100     0.002029
    2    634.8820     0.015535
    3    146.0970     0.075411
    4     42.4974     0.257121
    5     14.1892     0.596555
    6      1.9666     0.242517
    2    S    1
    1      5.1477     1.0
    3    S    1
    1      0.4962     1.0
```

```
   4   S   1
   1     0.1533    1.0
   5   P   4
   1    18.1557    0.018534
   2     3.9864    0.115442
   3     1.1429    0.386206
   4     0.3594    0.640089
   6   P   1
   1     0.1146    1.0
END
Chlorine 17.0 HUZINAGA-DUNNING (11S/6S)
   1   S   5
   1  40850.        0.002532
   2   6179.        0.019207
   3   1425.        0.095257
   4    409.2       0.345589
   5    135.5       0.636401
   2   S   3
   1    135.5       0.120956
   2     50.13      0.648511
   3     20.21      0.275487
   3   S   1
   1     6.283      1.0
   4   S   1
   1     2.460      1.0
   5   S   1
   1     0.5271     1.0
   6   S   1
   1     0.1884     1.0
   7   P   4
   1    240.8       0.014595
   2     56.56      0.099047
   3     17.85      0.330462
   4     6.350      0.682874
   8   P   2
   1     6.350     -0.561785
   2     2.403      1.351901
   9   P   1
   1     0.6410     1.0
  10   P   1
   1     0.1838     1.0
END
LIMIT
```

If a Huckel type of starting density matrix is desired to improve convergence in SCF routine, labels of shells must be accompained by the principal quantum number. In the case of this type of basis functions where labeling is meaningless, it is recommended to try the lowest S shells with the 1S label, and the others to be distributed among 2S and 3S, discretionally. The same must be done with 2P and 3P labels.

4) Performs a default HF-SCF calculation of methyl ion with a library file for 4-31G standard basis set which filename is just 4-31G.BFN. Use of EHT guess matrix and previously calculated two electron integrals in mainfile.

```
TITLE
CH3+, 4-31G
charge 1
skip
TIME 2 h
HUCKEL
PRINT 2
ATOMS
Carbon      6.0    1.09130      0           0
4-31g
Hydrogen    1.0    0            0           0
4-31g
Hydrogen    1.0    1.63689      .94510      0
4-31g
Hydrogen    1.0    1.63674      -.94519     -.00179
4-31g
END
START
stop
```

The basis function file can contain:

```
Hydrogen
1   1S   3
   1     18.731137     0.033495
   2      2.825394     0.234727
   3      0.640122     0.813757
2   1S   1
   1      0.161278     1.000000
END
Carbon
   1   1S   4
  1    486.966928     0.017726
  2     73.371094     0.123478
  3     16.413458     0.433875
  4      4.344984     0.561504
    2   2S   3
  1      8.673525    -0.121384
```

```
 2       2.096619     -0.227338
 3       0.604651      1.185174
   3   2S    1
 1       0.183558      1.000000
   4   2P    3
 1       8.673525      0.063545
 2       2.096619      0.298268
 3       0.604651      0.762103
   5   2P    1
 1       0.183558      1.000000
END
LIMIT
```

Observe that shell labels are with the principal quantum number, in order to create an appropriate EHT matrix for initial guess of the density matrix.

5) Optimizes geometry of benzene at the minimal basis MINI1 level. It uses the great symmetry of benzene ring to input only two atoms. All others are generated by the program.

```
TITLE
C6H6, mini1
RUNTYP    OPTIMIZE
time 16 h
SYMMETRY
DNH 6
END
ANGSTROM
ATOMS
Carbon  6.0      1.4    0    0
mini1
Hydrogen  1.0   2.48    0    0
mini1
END
START
FINISH
```

6) Minimal basis force constant calculation for the water molecule at the equilibrium geometry for this basis, using the simplex method with a stepsize of 0.0002 Bohr. Sequential calculation of the isotopic effect of ^{18}O and mono and disubstituted with deuterium.

```
TITLE
STO-3G  MINIMUM BASIS WATER FORCE CONSTANT RUN
MAXIT 20
RUNTYP FORCE
NPOINTS 0
```

```
STEPSIZE 0.002
TIME 6000
CONVERGENCE 9
SYMMETRY
CNV 2
END
ATOMS
OXYGEN 8.0  0.0  0.0  0.0
  1   S   3
  1   130.709321   0.154329
  2    23.808866   0.535328
  3     6.443608   0.444635
  2   S   3
  1   13.069241   -0.059945
  2    0.793609    0.596039
  3    0.304678    0.458179
  3   P   3
  1   4.653642     0.162395
  2   1.194342     0.566171
  3.  0.405496     0.422307
END
HYDROGEN 1.0 -1.4299469 0.0 1.2072984
  1   S   3
  1    3.425251    0.154329
  2    0.623914    0.535328
  3    0.168855    0.444635
END
END
START
TITLE
STO-3G  MINIMUM BASIS WATER FORCE CONSTANT RUN, O18 isotopic substitution
WEIGHTS
O  1  18.
END
RUNTYP FORCE
START
TITLE
STO-3G H2O, Deuterium isotopic single substitution
WEIGHTS
O  1  16.
D  2  2.
END
RUNTYP FORCE
START
TITLE
STO-3G H2O, Deuterium isotopic double substitution
```

```
WEIGHTS
D  3  2.
END
RUNTYP FORCE
START
STOP
```

7) Calculation of the optimized geometry of tetrahedral silicium tetrafluoride at the split MIDI1 basis set. It is in a MIDI1.BFN file, present in the default directory.

```
TITLE
SiF4, midi1
time 12 h
RUNTYP     OPTIMIZE
CONVERGE 7
PRINT 0
SYMMETRY
TD
END
ANGSTROM
ATOMS
Silicon  14.0   0    0    0
midi1
Fluorine  9.0   .82063    .82063    .82063
midi1
END
START
STOP
```

8.2. Examples of alternative format input files.

1) Cartesian coordinate file with the input geometry of ethyl silane, previously optimized geometry at the MINI1 level, to be optimized now at the MIDI1 level, further calculation of force constants, use of extended Huckel guess density matrix, convergence forced to 10^{-8} in electron density terms, calculation of electrical properties of the resulting geometry. The first run in the job is allowed up to al elapsed time of 2.5 hours:

```
  11
 C2H5-SiH3, (midi1|mini1)
 baslib midi1 optimize force print 1 huckel #
 converge 8 property time 2.5 h
-.05768    .00000   -.02374   6
    1.51316    .00000   -.09074   6
    2.30042    .00000   1.70633  14
    -.47979    .00000  -1.05541   1
    -.42696   -.90052    .51643   1
    -.42696    .90052    .51643   1
    1.87011   -.89825   -.63749   1
    1.87011     .89825   -.63749   1
    3.82719     .00000   1.57222   1
    1.83521  -1.25529   2.45674   1
    1.83521   1.25529   2.45674   1
```

2) Internal coordinate file in the IMC format. Single point HF SCF of 2 furan carboxaldehyde with the standard 6-31G basis set.

```
> baslib 6-31G
 Method PM3 ; furfural, OO trans,  dHf: -156.9 kJ/m
  8  6  6  2 1.3927 1.3788109.773  1  1  1
  1  2  3  4  6  9 1.4362106.496   .109  1  1  1
  2  3  4  5  6  9 1.3759106.333359.959  1  1  1
  4  3  2  6  6  9 1.4699133.028179.347  1  1  1
  1  2  3  7  1  9 1.0878127.151180.185  1  1  1
  2  3  4  8  1  9 1.0864126.010179.954  1  1  1
  3  4  5  9  1  9 1.0864134.026179.929  1  1  1
  3  2  6 10  8  9 1.2107122.020   .000  1  1 -1
  3  2  6 11  1  9 1.1019117.173180.000  1  1  0
end
```

3) Internal coordinate file in the AMI (AMPAC-MOPAC) free format. Geometry optimization at the HF SCF of 2 furan carboxaldehyde with the standard 3-21G basis set. Initial guess of density matrix by the EHT method. Observe thar asteriscs are used in place of 0 or 1 for optimization parameter indicators in AMPAC or MOPAC. All parameters are to be optimized.

```
> runtyp optimize baslib 3-21G huckel
 Method PM3 ; furfural, OO trans                      dHf: -156.9 kJ/m
O
C  1.3927 *
C  1.3788 * 109.773 *
C  1.4362 * 106.496 *    .109 *  3  2  1
C  1.3759 * 106.333 * 359.959 *  4  3  2
C  1.4699 * 133.028 * 179.347 *  2  3  4
H  1.0878 * 127.151 * 180.185 *  3  2  1
H  1.0864 * 126.010 * 179.954 *  4  3  2
H  1.0864 * 134.026 * 179.929 *  5  4  3
O  1.2107 * 122.020 *    .000 *  6  2  3
H  1.1019 * 117.173 * 180.000 *  6  2  3
end
```

4) Internal coordinate file in the Z matrix (ZMT) of GAUSSIAN free format. Geometry optimization at the HF SCF of 2 furan carboxaldehyde with the standard MINI1 basis set (default). Initial guess of density matrix by the EHT method. All parameters are to be optimized.

```
> runtyp optimize huckel
 Method PM3 ; furfural, OO trans                      dHf: -156.9 kJ/m
O
C  1 1.3927
```

```
C  2 1.3788 1 109.773
C  3 1.4362 2 106.496 1
C  4 1.3759 3 106.333 2 359.959
C  2 1.4699 3 133.028 4 179.347
H  3 1.0878 2 127.151 1 180.185
H  4 1.0864 3 126.010 2 179.954
H  5 1.0864 4 134.026 3 179.929
O  6 1.2107 2 122.020 3    .000
H  6 1.1019 2 117.173 3 180.000
end
```

8.3. An example of basis set superposition error correction.

1) Calculation of boron trifluoride dimer optimized geometry at the MIDI1 level with polarization functions.

The initial job for geometry optimization is the following. Observe the intensive use of molecular symmetry to avoid the explicit calculation of less than one half of the atoms.

```
TITLE
B2H6, (midi1pd)
TIME 54000
PRINT 0
RUNTYP  OPTIMIZE
SYMMETRY
CNH 2
END
ANGSTROM
ATOMS
Boron  5.0   .90232    .02207    0
midi1pd
Hydrogen  1.0   1.45863    -.04067    1.01095
midi1pd
Hydrogen  1.0   .02598    .97251    0
midi1pd
end
START
runtyp force
start
FINISH
```

2) Calculation of single point SCF of boron trifluoride monomer, with the optimized geometry of it in the dimer, with basis sets corresponding to the other monomer. Subsequent calculation of the same monomer geometry without ghost basis sets.

```
TITLE
```

```
B2H6, (optimized to midi1pd level), ghost orbital calculation of monomer
TIME 54000
PRINT 0
ANGSTROM
ATOMS
Boron  0.0   -.89460    .01830    .00000
midi1pd
Boron  5.0    .89460   -.01830    .00000
midi1pd
Hydrogen  0.0   -1.47210   -.03100   1.04506
midi1pd
Hydrogen  1.0    1.47210   -.03100  -1.04506
midi1pd
Hydrogen  0.0   -1.47210    .03100  -1.04506
midi1pd
Hydrogen  1.0    1.47210   -.03100   1.04506
midi1pd
Hydrogen  0.0    -.01989   -.97789    .00000
midi1pd
Hydrogen  1.0     .01989    .97789    .00000
midi1pd
end
START
continue
TITLE
BH3, (optimized to midi1pd level in B2H6), calculation of single monomer
PRINT 0
ANGSTROM
ATOMS
Boron  5.0    .89460   -.01830    .00000
midi1pd
Hydrogen  1.0    1.47210   -.03100  -1.04506
midi1pd
Hydrogen  1.0    1.47210   -.03100   1.04506
midi1pd
Hydrogen  1.0     .01989    .97789    .00000
midi1pd
end
START
stop
```

3) Calculation of BH_3 monomer as a free molecule with full geometry optimization.

```
TITLE
BH3, (midi1pd)
PRINT 0
```

```
RUNTYP     optimize
SYMMETRY
Cnv 3
END
ANGSTROM
maxit 99
ATOMS
Boron  5.0   0    0    .28469
midi1pd
Hydrogen  1.0   1.16024    0    .2851
midi1pd
END
START
runtyp force
start
finish
```

9.ERROR MESSAGES

If an error occurs in a calculation or in interpretation of the input data, the program prints message with a number and the name of the subroutine where the error was detected, and stops. A printing out of the current input line in a letter case appears both in the main output file and the console. The message identifies the error by means of a number. This number is comming, in some cases, from ancient HONDO error codes. In the present version the error numbers are as follows:

(1) Unrecognized option directive.
(2) Error in **SYMMETRY** directive.
(3) Error in **GEOMETRY** directive: the atomic numbers or the number of atoms are not in agreement with the data given in an earlier **ATOMS** directive.
(4) Error in **ATOMS** directive: unrecognized basis function type.
(5) Error in **ATOMS** directive: incorrect format for basis function.
(6) Too many primitives in the system. Check also symmetry directive, if exist.
(7) Too many atoms in the system. Check also symmetry directive, if exist.
(8) Too many basis functions. Check also symmetry directive, if exist.
(9) Too many primitives in a single contracted basis function.
(10) Too many shells. Check also symmetry directive, if exist.
(11) Error in **SWAP** directive: orbital label out of range.
(12) Error in **GUESS** directive: unrecognized option.
(13) The amount of allowed coordinates is exceeded.
(14) No basis set available in external library file.
(15) Unrecognized **START** directive.
(16) More than expected lines read in an internal file for a restarting input.
(17) Unknown SCF type.
(18) Inconsistent multiplicity.
(19) Error in **SYMMETRY** assignment.
(20) A file name must be specified with this keyword.
(21) An illegal file name is specified with this keyword.
(22) The refered file name must be used as scratch file.
(23) The restoring file must be opened during restart.
(24) The number of words is exceeding the available core.
(25) Error in SCF locker.
(26) Trying to change the origin of coordinates before reading geometry.

(61) Index block of dumpfile has incorrect format.
(62) Input / output error usually caused by failing to assign the file.
(63) Section number less then 1.
(64) Section on dumpfile not found.
(65) Type of section incorrect.
(66) File has not been opened.
(67) Illegal search of data set; i.e. the block does not exist.
(68) Illegal character while attempting to read a REAL (F format) number.
(69) Illegal character while attempting to read an INTEGER (I format) number.

(70) Attempt to reduce the maximum length of the dumpfile below its current length.

(71) Attempt to write beyond the end of the dumpfile.

(72) Section on dumpfile already exists and its current length is less than that requested.

(73) Output error in external file.

(74) Output <fn>.CAR file is not properly formatted.

(75) Word not allowed in input .CAR file.

(76) Word not allowed in input internal coordinate file.

(77) Non defined atom as reference in internal coordinate input file.

(78) Colinear atoms taken as reference in internal coordinate input file.

(79) Atomic symbol used in input file corresponds to an element greater than Krypton.

(80) An error occured in command line parameters.

10. FILE MANAGEMENT

10.1. Assignment of sections in the dumpfile.

The sections on the dumpfile are allocated using internal routines SECPUT/SECGET, etc. Different section sizes are used for each of the .ED1, .ED3, and .ED5 versions, due to the different amounts of data allocated for them. It means that HAVMOL, HAVMOLE and HAVMOLW use different *dumpfile* section dimensions.

However, the number of each section are the same to localize a given type of information. Each **HAVMOL** output shows a resume of *dumpfile* section occupations.

The default section map for a typical *dumpfile* is as follows,

Section	***Content***
1	Symmetry data 1
2	System data 1 (geometry)
3	Basis functions
4	Options and restart information 1
5	Overlap matrix
6	1-electron hamiltonian (T + V)
7	Density matrix (closed and open shell)
8	Wave functions (eigenvectors)
9	Eigenvalues
10	Density matrix (open shell)
11	Common HUCKEL
12	Common ATMWT (atomic weights)
13	SCF data
14	Gradients
15	Optimized coordinates and other related data
16	Inverse Hessian
17	Force constant data
18	Force constant matrix
19	Lagrangian for open shells
20	K matrix for open shells
21	Symmetry data 2
22	System data 2
23	Density matrix (optional)
24	Options and restart information 2

These sections are not allocated into *dumpfile* until they are needed. It means that more of the dumpfile will be used as the calculation progresses.

Each section has a number and a type. The type is set equal to the default section number in order to help for remembering which section is. Block numbers are the place where they have been

allocated in the file. They are equivalent to the record numbers where the given sections begin in direct access *dumpfile*.
Finally, each block in **HAVMOL** *dumpfiles* is of 4096 bytes length. Many of them need no more for allocating the corresponding data. However, certain matrices require much more, and the section length corresponds to the amount of contiguous blocks (4096 bytes each one) which are used in it.

10.2. Assignment of sections in the mainfile.

The *mainfile* have a unique structure. It has been nominated with different file name extensions due to formal advantages, but they are fully compatible across different **HAVMOL** versions. They consist in blocks of 4096 bytes, containing 510 integrals, indexes and register values, each one.

Only one section exists in this file, and it is devoted to bielectronic integrals.

10.3. Summary of HAVMOL files

10.3.1. Input files.

1.- Standard input file for beginning a job: **<fn>.IMM**
2.- Standard input file for continuing a job: **<fn>.IM1**
3.- Alternative input file format with cartesian coordinates: **<fn>.CAR**
4.- Alternative input file format with internal coordinates (TC-HABANA): **<fn>.IMC**
5.- Alternative input file format with internal coordinates (AMPAC-MOPAC): **<fn>.AMI**
6.- Alternative input file format with internal coordinates (GAUSSIAN): **<fn>.ZMT**
7.- Dummy input file for continuing a job (prepared in console): **<fn>.IM1**

10.3.2. Output files.

1.- Main output file (initial run of the <fn> job): **<fn>.OMM**
2.- Main output file (restoring the *nnn* run of the <fn> job): **<fn>.*nnn***
3.- Cartesian coordinates (in Angstrom units) of the optimized geometry: **<fn>.CAR**
4.- Standard input file from alternative input file formats: **<fn>.IMM**

10.3.3. Direct access storage files in HAVMOL.

1.- Mainfile with two electron integrals: **<fn>.ED2**
2.- Dumpfile with two electron integrals: **<fn>.ED3**
3.- Scratch file **<fn>.ED7**

10.3.4. Direct access storage files in HAVMOLE.

1.- Mainfile with two electron integrals: **<fn>.ED0**
2.- Dumpfile with two electron integrals: **<fn>.ED1**
3.- Scratch file **<fn>.ED7**

10.3.5. Direct access storage files in HAVMOLW.

1.- Mainfile with two electron integrals:	**<fn>.ED4**
2.- Dumpfile with two electron integrals:	**<fn>.ED5**
3.- Scratch file	**<fn>.ED7**

10.4. On the lengths of files.

10.4.1. The mainfile

This holds the two-electron integrals. If there are *M* basis functions then the maximum possible number of two-electron integrals will be of the order *(M**4)/8*. As the integrals are stored with 510 to a block, this gives a maximum length for the *mainfile*. However, as only unique non-zero integrals are stored, the *mainfile* will nearly always be much less than its maximum possible size.

10.4.2. The dumpfile.

If *MAXSHELLS* is the maximum allowed number of shells for a given **HAVMOL** version, *MAXATOMS* is the corresponding maximum allowed number of atoms, *MAXPRMTVES* is the corresponding maximum allowed number of primitives, and *MAXBASFUN* is for the maximum number of basis functions, then the fixed sections 1, 2, 3, 4, 11, 12, 13, 21, 22, and 24 in dumpfile will amount:

Section 1:	*239+12*(MAXSHELLS+1)*
Section 2:	*8+14.5*MAXATOMS*
Section 3:	*1+4*MAXPRMTVES+3.5*MAXSHELLS*
Section 4:	*51*
Section 11:	*MAXSHELLS*
Section 12:	*MAXATOMS*
Section 13:	*23*
Section 21:	*1*
Section 22:	*5*MAXATOMS+MAXBASFUN*
Section 24:	*14*

in terms of 8 bytes words. The corresponding amount of bytes is obtained if multiplied by 8. By taking into account that each block have a maximum of 4096 bytes, some **HAVMOL** versions will require more than one block for these fixed sections.

Many of the other sections e.g. 1-electron integrals, Fock matrices, density matrices etc, have a length of *NX=M*(M+1)/2* words or *(NX/511)+1* blocks, where *M* is the number of basis functions. Any sets of eigenvectors will require *((M*M)/511)+6* blocks. Force constant matrices, hessians etc., are *((9*N*N/511)+1* blocks long, where *N* is the number of nuclei. All other sections are 1 block long.

10.4.3. The scratchfile.

This is used in the SCF routines to store sets of eigenvectors, density matrices, etc. from the last few energy evaluations for use by extrapolation routines. If a density matrix is *(NX/511)+1* blocks long, then the maximum length will be about 15 times this.

Referencias

1. Roothaan, C. C. J., New Developments in Molecular Orbital Theory. *Rev. Mod. Phys.* **1951,** 23, (2), 69-89.
2. Hall, G., The molecular orbital theory of chemical valency. VIII. A method of calculating ionization potentials. *Proc. Roy. Soc. (London), A* **1951,** 205, 541-552.