New vibration-rotation code for tetraatomic molecules exhibiting wide-amplitude motion: WAVR4

Igor N. Kozin [a], Mark M. Law [a], Jonathan Tennyson [b], and Jeremy M. Hutson [c]

[a] *Department of Chemistry, University of Aberdeen, Meston Walk, Aberdeen AB24 3UE, UK*

[b] *Department of Physics and Astronomy, University College London, London WC1E 6BT, UK*

[c] *Department of Chemistry, University of Durham, South Road, Durham, DH1 3LE, UK*

Abstract

A general computational method for the accurate calculation of rotationally and vibrationally excited states of tetraatomic molecules is developed. The resulting program is particularly appropriate for molecules executing wide-amplitude motions and isomerizations. The program offers a choice of coordinate systems based on Radau, Jacobi, diatom-diatom and orthogonal satellite vectors. The method includes all six vibrational dimensions plus three rotational dimensions. Vibration-rotation calculations with reduced dimensionality in the radial degrees of freedom are easily tackled via constraints imposed on the radial coordinates via the input file.

Key words:

PACS:

PROGRAM SUMMARY

Title of program: WAVR4

Catalogue number: (supplied by Elsevier)

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue).

Licensing provisions: Persons requesting the program must sign the standard CPC nonprofit use license (see licence agreement printed in every issue).

Computer: Developed under Tru64 UNIX, ported to Microsoft Windows and Sun Unix.

Operating systems under which the program has been tested: Tru64 Unix, Microsoft Windows, Sun Unix.

Programming language used: Fortran 90.

Memory required to execute with typical data: case dependent.

No. of lines in distributed program, including test data, etc: 10385.

Keywords: ro-vibrational, bound states, wavefunctions, body-fixed, discrete variable representation, finite basis representation, tetraatomic, 4-atom

Nature of physical problem: WAVR4 calculates the bound ro-vibrational levels and wavefunctions of a tetraatomic system using body-fixed coordinates based on generalised orthogonal vectors.

Method of solution: The angular coordinates are treated using a finite basis representation (FBR) based on products of spherical harmonics. A discrete variable representation (DVR) [1] based on either Morse-oscillator-like or spherical-oscillator functions [2] is used for the radial coordinates. Matrix elements are computed using an efficient Gaussian quadrature in the angular coordinates and the DVR approximation in the radial coordinates. The solution of the secular problem is carried through a series of intermediate diagonalisations and truncations.

Restrictions on the complexity of the problem: (1) The size of the final Hamiltonian matrix that can be practically diagonalised; (2) The DVR approximation for a radial coordinate fails for values of the coordinate near zero - this is remedied only for one radial coordinate by using analytical integration.

Typical running time: problem-dependent

Unusual features of the program: A user-supplied subroutine to evaluate the potential energy is a program requirement.

References:

[1] J. C. Light, I. P. Hamilton and J. V. Lill, J. Chem. Phys. **92**, 1400 (1985).

[2] J. R. Henderson, C. R. Le Sueur, and J. Tennyson, Comp. Phys. Comm. **75**, 379 (1993).

LONG WRITE-UP

1 Introduction

Recent interest in understanding wide-amplitude ('floppy') molecular motions has been stimulated by the drive to develop theories of intermolecular forces, isomerization and coherent control of chemical reactions. Methods for calculating the rotation-vibration energy levels and wavefunctions of floppy systems have advanced greatly in the last decade but remain technically demanding and computationally expensive even for molecules and complexes as small as tetraatomics [1–12]. In this paper we present a new code, WAVR4, specially adapted for calculating the bound ro-vibrational energy levels and wavefunctions of tetraatomic systems executing wide-amplitude motions.

Coordinate systems based on generalised orthogonal vectors have become a very popular choice in dealing with wide-amplitude motions in polyatomic systems. The approach used in this work was suggested by Chapuisat and Iung [1] and developed further in Refs. [13] and [14]. Recently Mladenović gave a very concise account of the approach together with a detailed description of applications to some molecules [5]. Because of singularities in the Hamiltonian, we employ a non-direct-product finite basis representation (FBR) for angular coordinates [5]. While the discrete variable representation (DVR) method proved to be very efficient [15,16] and we use it for radial functions, the non-direct-product part of our basis cannot be transformed effectively to a DVR and so integration of the angular potential function must be performed. Therefore we use a mixed FBR-DVR basis representation and the traditional explicit sequential diagonalization and truncation approach [16]. Thus the computation is performed in several steps.

The program offers a choice of coordinate systems based on Radau, Jacobi, diatom-diatom and orthogonal satellite vectors (see Fig. 1). Vibration-rotation calculations with reduced dimensionality in the radial degrees of freedom are easily tackled via constraints imposed on the radial coordinates via the input file. So far WAVR4 has been extensively tested for the Ar_2-HF trimer (5D vibration plus 3D rotation) [17] and acetylene (6D vibration) [18]. The program is general and should be applicable to a large range of four-atom systems, particularly those undergoing large amplitude motion. There is no restriction on the form of the molecular potential energy function.

The WAVR4 code is highly complementary to the RVIB4 and RVIBNH3 codes of Carter, Colwell and Handy [19]. The latter codes employ non-orthogonal coordinate systems and kinetic energy operators based on valence coordinates adapted for formaldehyde-, hydrogen peroxide- acetylene-, and ammonia-like systems. Finite basis representations are used for all coordinates in RVIB4 and RVIBNH3 and potential energy functions up to sextic in any of the six coordinates are allowed. RVIB4 and RVIBNH3 are available via the CCP6 web site [20].

2 Computational method

2.1 Formulating and solving the 6D vibration + 3D rotation Hamiltonian

One of the most attractive features of generalised orthogonal coordinates is the simplicity of the kinetic energy operator,

$$T = \sum_{\alpha} -\frac{\hbar^2}{2\mu_\alpha}\left(\frac{\partial^2}{\partial q_\alpha^2} + \frac{2}{q_\alpha}\frac{\partial}{\partial q_\alpha}\right) + T_{\text{ang}}, \tag{1}$$

where μ_α are reduced masses, q_α are the lengths of internal vectors $\mathbf{q}_\alpha$ and T_{ang} is the angular kinetic operator given in Eq. (37) of Ref. [5]. The operator T_{ang} describes both bending and rotation of the molecule and so our reference to it as an "angular" operator is only for brevity. A very important feature of Eq. (1) is its invariance under various choices of orthogonal vectors. We have implemented several choices such as Radau, Jacobi, diatom-diatom and orthogonal satellite vectors considered in Refs. [3] and [5]. For the case of tetraatomic molecules there are three values of α. Once the coordinate scheme is chosen, the body-fixed axis system is defined so that the z-axis goes along vector $\mathbf{q}_3$, and the xz plane is defined by $\mathbf{q}_3$ and $\mathbf{q}_1$.

There are two types of singularities associated with T_{ang}. One singularity corresponds to the angle between $\mathbf{q}_3$ and $\mathbf{q}_1$ being zero or π (so that the xz plane is not defined). Since we are interested in wide-amplitude motion we must consider the full angular dynamical range. The nature of this singularity is known and can be handled by using a non-direct-product angular basis [21] as discussed below. For a recent, general discussion see for example Ref. [22]. Another type of singularity arises when a radial coordinate is equal to zero. As discussed below, we have considered only the most frequently occurring case when q_3 is equal to zero (so that the z-axis is not defined).

Our treatment of the angular problem is essentially close to that of Mladenović [5] who gave the matrix elements of T_{ang} in a parity-adapted angular basis. For convenience and also because we use slightly different phase factors, the matrix elements that arise in our treatment are presented in the Appendix. Our primitive angular basis functions are

$$\begin{aligned} |\gamma\, K\, k\, j\, l, Jp\rangle = N_{Kk}\gamma^k P_j^{|k-\gamma K|} \Big[& Y_l^{\gamma k}|J,K,M\rangle + \\ & (-1)^{J+p+K+k} Y_l^{-\gamma k}|J,-K,M\rangle \Big] \end{aligned} \tag{2}$$

where γ is an auxiliary number taking the values -1 and $+1$, J and K are the usual rotational quantum numbers associated with the total angular momentum and its projection on the body-fixed z-axis, j and l are angular momenta associated with rotation of $\mathbf{q}_1$ and $\mathbf{q}_2$ respectively, k is the projection of l onto $\mathbf{q}_3$, p takes the values 0 and 1 for even and odd total parity respectively, N_{Kk} is a normalization factor, P_j^k are associated Legendre functions of the angle θ_1 between $\mathbf{q}_3$ and $\mathbf{q}_1$, Y_j^k are spherical harmonics of the body-fixed angles (θ_2, φ) defining the direction of $\mathbf{q}_2$ and $|J,K,M\rangle$ are symmetric top eigenfunctions. When $\gamma = +1$ our angular basis functions are the same as in Ref. [5], but they differ by a factor $(-1)^{J+p+K}$ when $\gamma = -1$.

The radial basis functions are similar to those employed in the triatomic code DVR3D [24]. The two possible choices are Morse-oscillator-like functions and spherical-oscillator functions. Morse-oscillator-like functions are defined as [25]

$$\beta^{1/2} N_{n\alpha} \exp(-y/2) y^{(\alpha+1)/2} L_n^{\alpha}(y) \tag{3}$$

where $y = \alpha \exp[-\beta(r - r_e)]$, $\alpha = 4D_e/\beta$, $\beta = \omega_e(\mu/2D_e)^{1/2}$, $L_n^\alpha(y)$ is a Laguerre polynomial, μ is the reduced mass associated with radial distance r and $N_{n\alpha}$ is a normalization factor. The set $\{r_e,\ \omega_e,\ D_e\}$ is treated as a set of parameters to be optimized, although they can be associated with the equilibrium distance, fundamental frequency and dissociation energy. In the case where the distance r can be zero, spherical-oscillator functions [23] are a better choice,

$$\sqrt{2}\beta^{1/4} N_{n\eta+1/2} \exp(-y/2) y^{(\eta+1)/2} L_n^{\eta+1/2}(y) \tag{4}$$

where $y = \beta r^2$, $\beta = (\mu\omega_e)^{1/2}$ and the parameter set $\{\eta,\ \omega_e\}$ is optimized. It is straightforward to convert the basis functions (3) and (4) to a DVR basis [24].

Our full primitive basis is a product of the FBR angular basis (2) and a radial DVR basis obtained from Eq. (3) or Eq. (4). A major advantage of Eq. (1) is that it helps separate radial and angular coordinates because no mixed-derivative angular-radial operators are present. Thus, having radial coordinates represented in DVR, the whole problem can be constructed from a set of angular sub-problems. Furthermore the use of the DVR approximation for the potential energy and $1/q_i^2$ in T_{ang} requires only the angular integrals to be computed explicitly. Therefore the whole structure of the Hamiltonian matrix is very sparse: it consists of angular (bending-rotation) sub-blocks which are coupled by the matrix elements of radial kinetic energy operators that are diagonal in all angular indices.

The primitive secular matrix is very big and we implemented explicit sequential diagonalization and truncation to solve the corresponding eigenvalue problem (see for example Ref. [26,16]). The computation is performed in several steps. First the angular kinetic energy operator is separated into sub-blocks diagonal in K, $T^{K}_{\rm ang}$, and sub-blocks off-diagonal in K with $\Delta K = \pm 1$, $T^{K',K}_{\rm ang}$. The angular problem associated with $T^{K}_{\rm ang} + V$ is solved separately for every K sub-block for all radial grid triple points (q_1, q_2, q_3). Here V is the full (6D) potential with fixed q_1, q_2 and q_3. Only eigenfunctions below a certain energy cutoff, $E^{(1)}_{\rm cut}$ are selected for later use. Then the kinetic energy operators in the radial coordinates q_1 and q_2 are included and the respective matrices are constructed in the angular plus $(q_1,\, q_2)$ basis and solved for eigenvectors for every q_3 point. Again only the lowest states, this time below $E^{(2)}_{\rm cut}$, are selected. Then the kinetic energy operator in q_3 is included and the full 6D vibrational matrix is computed. During this step only the eigenvalues below $E^{(3)}_{\rm cut}$ are found. Up to and including this stage, in our implementation, terms in the Hamiltonian are evaluated setting J equal to K [27]. Hence for $K = 0$ the eigenvalues give the desired final vibrational levels. For $K > 0$ K-optimized eigenvectors are obtained for further use in the final ro-vibrational step (where the value of J is restored and the corresponding missing terms are properly included in the Hamiltonian).

An important feature of our basis (2) is that the contracted eigenvectors for $p = 0$ and $p = 1$ are identical for the $K > 0$ diagonal sub-blocks because our basis is such that the matrix elements of $T^{K}_{\rm ang} + V$ are identical. Hence the contracted eigenvectors for $p = 0$ can be used for both $p = 0$ and $p = 1$ in the final step. In this final step, $\Delta K = \pm 1$ sub-blocks are included and ro-vibrational levels are computed.

Depending on the size of the angular basis, the computation of the three-dimensional angular integrals of the potential function may be the most timeconsuming part of the calculation. Therefore it is important to make it as efficient as possible. To facilitate this the code offers two options. The first option is to expand the potential function in the $J = 0$, totally symmetric angular functions defined by Eq. (2) at every radial triple and compute the angular integrals analytically. This allows the expansion to be re-used for all angular integrals. This strategy is particularly useful when high accuracy integrals are desired. The second option is to perform direct integration over the angles using Gaussian quadrature. It turned out that in the real applications considered to date, the second option employing the minimal number of quadrature points gave us the best ratio of performance to accuracy. This was achieved after implementing an algorithm [28] which takes into account the symmetry properties of the product of two primitive basis functions. The summation is performed over only half of the quadrature points: it uses the symmetric part of the potential if the product is symmetric and the anti-symmetric part if the product is anti-symmetric. In choosing the number of quadrature points, we used the minimum number of points required to maintain the orthogonality of the basis functions. These numbers are $j^{\max}+1$, $k^{\max}+1$ and $l^{\max}+1$ for the coordinates θ_1, φ and θ_2 respectively. Without taking into account the symmetry properties of the basis functions one would have to use $2j^{\max}+1$, $2k^{\max}+1$ and $2l^{\max}+1$ points respectively. So the approximately twofold saving for every angular coordinate gave an almost eightfold saving overall.

2.2 Symmetry

The symmetry which is always present is space-fixed inversion. It separates the states of even ($p = 0$) and odd ($p = 1$) parity. The inversion symmetry is implemented in the angular basis (see Appendix). Furthermore, it allows a further twofold reduction of the Gaussian quadrature grid for the angle φ because the inversion transforms $\varphi \to 2\pi - \varphi$. So the range for the grid can be chosen from zero to π only although φ is physically defined from zero to 2π.

In addition, the permutation of two identical atoms may be also feasible. In the present code only those symmetries are supported that reverse a vector connecting two identical atoms. This has a straightforward effect on the angular matrix, making it block diagonal in even and odd quantum numbers: j if the permutation reverses q_1, l if q_2, and $j + l$ if q_3. To separate these we use the numbers j-parity, l-parity, and jl-parity which take values 0 (even) and 1 (odd).

2.3 Breakdown of the DVR approximation for q_3

As mentioned above it is possible for some systems for a radial coordinate to take values close to zero. This results in failure of our quadrature approximation for the $1/q_i^2$ term in the angular kinetic energy operator T_{ang}. The problem of the $1/r^2$-type singularity is well known in triatomic systems such as H_3^+ and Ar_3, for example if treated using Jacobi coordinates [29,30]. The reader is referred to an excellent paper [31] which explicitly considered the problem of singularities and implications made by the choice of direct-product or non-direct-product bases. The conclusion of Ref. [31] is that strictly speaking one needs a non-direct-product angular-radial basis to account fully for the $1/r^2$ singularity. However frequently a simpler direct-product approach works well [29,30]. This simpler approach involves two ingredients: (1) the use of basis functions with nonzero probability density at $r = 0$ if they are allowed by symmetry and basis functions with zero probability density at $r = 0$ otherwise; (2) the matrix elements of $1/r^2$ are first computed analytically in FBR and then converted to DVR.

The spherical-oscillator functions (4) have nonzero probability density at $r = 0$ if $\eta = 0$ and zero if $\eta > 0$. In fact the Gaussian quadrature points in r never take the value zero exactly, but sample the area near it. Therefore we may use $\eta = 0$ functions if the symmetry allows non-zero probability density at $r = 0$ and $\eta = 1$ functions otherwise. To deal with the breakdown of the DVR approximation for q_3, an option is implemented to compute $1/q_3^2$ (required in T_{ang}) analytically [29]. We have not at present implemented similar options for q_1 and q_2.

3 Program structure

The program is written in Fortran 90 taking full advantage of dynamic memory allocation. All variables are explicitly declared. A diagram of the program flow is present in Fig. 2.

The program `main` starts with initialization during which the input data are read. The input file defines the molecule, its coordinate system and symmetry, and the type and size of the basis. The size of the radial basis is essentially the size of the radial grid. The size of the angular grid required for numerical evaluation of expansion terms or matrix elements can be set manually or automatically. After the radial and angular grids are computed, the transformations of radial kinetic energy operators from FBR to DVR representation are performed. Next the expansion of the potential is computed and stored for further use if necessary (`expansion_flag` = 1). To avoid calculating eigenvectors at radial points which are too high in energy the subroutine `test_potential` is used: if the minimal potential energy on the angular grid is higher than a predefined value (`margin1`) then that radial point is excluded from further processing.

Then the program enters main loops over K, p, and permutation symmetries (j-parity, l-parity and jl-parity). In step 1, the subroutine `angular_states_qnumbers` returns the full set of all possible angular quantum numbers for the given K, symmetry and the basis limits provided by the input file. Next, the angular kinetic energy matrix is computed in `angleT3D`. Then the matrix elements of the potential function are computed and the angular problem is solved for eigenvalues and eigenvectors. All diagonalisations are performed by calling standard LAPACK subroutines [32]. At the end of every step the program saves the selected eigenvectors and eigenvalues to disk. These vectors are read later by the subroutine `mainj` and optionally they can be used again for another run of the program provided that the basis has not been changed.

The program proceeds through steps 2 and 3 where further diagonalisations and truncations are performed. In step 3, `angleT3D` is called again only if analytical calculation of $1/q_3^2$ has been requested. It is the responsibility of the user to specify in the input file how the truncation proceeds. The truncation can be based either on the number of eigenstates computed for a given grid point (`icut`) or on the energy cutoff (`encut`). The former should be used only when the radial grid is so small that selecting eigenstates using the energy cutoff gives poor results because some of the grid points are completely or nearly removed. The details of relevant settings in the input file are given below. The final step 4 is performed by the subroutine `mainj` which is called only for $K > 0$.

4 Program use

4.1 Installing and running the code

BLAS and LAPACK 3.0 libraries [32] are supported by many vendors and also can be freely downloaded from Netlib.org. They are assumed to be available. The user is supplied with `makefile` and `make.inc` files which should be modified appropriately for correct references to compiler, linker and libraries. The code is structured into several directories: the directory `12-3` contains the core files and `COMMON` contains various auxiliary subroutines. The program comes with a subset of LAPACK95 (see Netlib.org) which is a collection of Fortran 90 interfaces to LAPACK. These are kept in directory `LAPACK95`. While standard LAPACK, the interfaces and the code describe integers as 32-bit, some LAPACK implementations require 64-bit integers (e.g. Sun computers). An alternative set of LAPACK interfaces is provided in `LAPACK95.sun64` for this purpose. Finally there is a directory `test` with sample input file, potential function and test output file. Both `input.txt` and `potential.f90` need to be placed in a root directory where `makefile` resides before running `make`. After running `make`, the user needs only to execute `main.exe` which reads from `input.txt` and writes to `output.txt`.

At run time the program creates a number of unformatted data files. The files `x_h6.dat` and `x_expansion.dat` are temporary files for storing the matrix in step 3 and the potential expansion respectively. The files `xnnnn_Kn-n.dat` store eigenvectors and eigenvalues. The naming convention for these files uses six integer numbers to represent symmetry (p, j-parity, l-parity and jl-parity, all either 0 or 1), value of K (currently from 0 to 9) and step number (from 1 to 4).

4.2 The potential subroutine

The user must supply potential function `v(q1,q2,q3,theta1,theta2,phi)` for the system under consideration. This must contain the following module and be compatible with its interface

```
MODULE potential

INTERFACE v
  FUNCTION v(q1,q2,q3,theta1,theta2,phi)
    REAL(8), INTENT(IN) :: q1,q2,q3,theta1,theta2,phi
    REAL(8) :: v
  END FUNCTION
END INTERFACE

END MODULE
```

A sample potential file is provided for the HCCH molecule [33].

WAVR4 requires an input file for all runs. Not all parameters are used but all must be present anyway. Apart from the first line, the data is given in free format but line positions are fixed. Any lines starting with "!" are comments that can hold a brief explanation of parameters. The units are: energies expressed as the equivalent wavenumbers in cm^{-1}, lengths in Å, angles in radians, masses in unified atomic mass units.

Line 1: `title`

– A80 format, 80-character title.

Line 3: `mass1, mass2, mass3, mass4`

– reals, atom masses in unified atomic mass units.

[Here and below, 'real' is used to denote 'double precision real'.]

Line 5: `opt`

– integer, defines coordinate system: = 1, Jacobi; = 2, Radau; = 3, diatom-diatom; = 4, orthogonal-satellite vectors. See Fig. 1

Lines 7-9: `igq(i), re(i), we(i), De(i), nn(i)`

– parameters for the radial basis in q_1, q_2, and q_3.

`igq(1:3)`, integer: = 1, Morse-oscillator-like functions; = 2, spherical-oscillator functions.

`re(1:3)`, real: r_e for Morse-oscillator-like functions and η for spherical-oscillator functions.

`we(1:3)`, real: w_e in cm^{-1}.

`De(1:3)`, real: D_e in cm^{-1} (not used for spherical-oscillator functions).

`nn(1:3)`, integer: number of functions (i.e. grid size).

Note: If `nn(i)` = 1 then the `i`-th coordinate is frozen to the value given below (see Line 26).

Line 11: `angular_problem_only_flag, optimize_flag, test_flag, expansion_flag`

– integer flags.

`angular_problem_only_flag`: $= 0$ (default), normal operation; > 0, only 3D angular problem (in the coordinates θ_1, φ and θ_2) is computed for the radial triple point specified in Line 26.

`optimize_flag`: currently not used.

`test_flag`: $= 0$, default; $= 1$ or 2, controls output level giving progressively more test output.

`expansion_flag`: $= 0$, compute 3D FBR angular matrix elements using quadrature integration; $= 1$, compute angular matrix elements through the expansion of the angular potential.

Line 13: `jmax, lmax, kmax, jrmax, krmax, j_parity_max, l_parity_max, jl_parity_max`

– all integers, define the bending-rotation basis. The first five numbers indicate maximum quantum numbers included for j, l, k, J, and K respectively. The last three numbers indicate the presence of permutational symmetry:

`j_parity_max`: $= 1$, q_1 is reversible; $= 0$, no symmetry.

`l_parity_max`: $= 1$, q_2 is reversible; $= 0$, no symmetry.

`jl_parity_max`: $= 1$, q_3 is reversible; $= 0$, no symmetry.

Line 15: `ne1, ne2, ne3`

– all integers, specify maximum quantum numbers for angular expansion in j, l, and k respectively.

Line 17: `nt1, nt, nphi, iauto`

– all integers. The first three numbers specify the size of the angular quadrature in θ_1, θ_2 and φ respectively. `iauto`: $= 0$, WAVR4 uses user-supplied numbers; $= 1$, 2, or 3, WAVR4 automatically defines minimum, double or quadruple grid sizes as described in Table 1.

Line 19: `enzero`

– real, zero-point energy, used only if not computed in the present run.

Line 21: `icut0, icut1, icut2, icut3, icut4`

– all integers. `icutN` defines the number of eigenvectors computed during every diagonalisation in step N. If `icutN` $= 0$ then all eigenvectors are found. If `icutN` > 0 then only `icutN` lowest eigenvectors are found. If `icutN` < 0 then `encutN` controls the number of computed eigenvalues and defines the maximum energy for computed eigenvalues.

Note: step 0 is reserved for future use.

Line 22: `encut0, encut1, encut2, encut3, encut4`

– all reals. `encutN` defines the energy cutoff used in step N. The selection based on energy is always applied.

Note: if a cutoff based on a number of levels is desired, `encut1` and/or `encut2` need to be set to a sufficiently large number. This may be useful for removing some very high eigenstates (particularly those on the edges of the radial grid).

WARNING: special considerations apply to steps 3 and 4. If `icut3/4` = 0 then all eigenvalues will be computed but no eigenvectors. Most importantly, because Hamiltonian matrices in steps 3/4 are created in a packed form, storage for eigenvectors must be allocated separately. In practice this means that, for initial calculations, it is safer to use truncation based on numbers (i.e. setting positive `icut3/4`) rather than based on energy `encut3/4`. If `icut3/4` are negative, `abs(icut3/4)` are used to reserve the storage for eigenvectors and it could happen that there are more than `abs(icut3/4)` states below `encut3/4`. However it is still possible to use the energy cutoff: `icut3/4` must be sufficiently big but not too big to make selective computation of eigenvectors inefficient (i.e. less than 10% of the matrix size).

Line 23: `margin0, margin1, margin2, margin3, margin4`

– all reals. The only one used currently is `margin1`, which is used to skip 3D angular calculation at a radial triple point if minimum potential energy computed on the angular grid is higher than `margin1`.

Line 24: `imargin0, imargin1, imargin2, imargin3, imargin4`

– all integers, currently not used.

Line 26: `qe(1), qe(2), qe(3), qe(4), qe(5), qe(6)`

– all reals. The values of `qe(1)`, `qe(2)` and `qe(3)` define fixed values of q_1, q_2 and q_3 respectively if required by the options specified in Lines 7-9 and 11. The values of `qe(4)`, `qe(5)` and `qe(6)` define fixed values of θ_1, θ_2 and φ respectively used to compute pure 3D stretching states when `jmax` = `lmax` = 0.

Line 28: `stage_flag`

– integer, from 0 (default) to 4. If set to a value greater than 0 the program will attempt to read from disk eigenvectors saved on a previous run of the program; the value indicates up to which step the eigenvectors are already available. It is the responsibility of the user to make sure that the eigenvectors correspond to the current primitive basis and potential. The eigenvectors are saved to disk files with names of the form `xnnnn-Kn-n.dat` where the digits 'n' indicate the numbers p, j-parity, l-parity, jl-parity, K and the step number respectively.

Line 30: `oner_flag`

– integer: = 0, $1/q_3^2$ is computed in DVR approximation; = 1, $1/q_3^2$ is computed analytically in the FBR and transformed to the DVR (this is only valid for `igq(3)` = 2, see Lines 7-9).

iauto =	0	1	2	3
	user grid	minimal grid	double grid	quadruple grid
θ_1	input	$j_{\max}+1$	$2j_{\max}+1$	$4j_{\max}+1$
θ_2	input	$l_{\max}+1$	$2l_{\max}+1$	$4l_{\max}+1$
φ	input	$(k_{\max}+1)/2$, $k_{\max}$ is odd	$k_{\max}+1$	$2k_{\max}+1$
		$k_{\max}/2+1$, $k_{\max}$ is even		

Table 1

The size of the angular grid depending on input options.

4.4 Test output

A test input file has been prepared. It is based on the HCCH potential energy surface of Carter *et al* [33]. The basis set used for the test is rather small to reduce the run time. Full $J=1$ calculation takes about 10 minutes on a current computer (e.g. Compaq Alpha or Sun workstation). The output file for the test run is presented at the end of this paper. This output is slightly adapted and truncated after the first few purely vibrational levels. The beginning of the file reproduces the input file as read by the program.

Acknowledgements

We are grateful to Mirjana Mladenović and Christoph Iung for very useful discussions and to Tucker Carrington Jr for comments on the manuscript. This work was funded by the UK Engineering and Physical Sciences Research Council as the "flagship" project of CCP6 (the Collaborative Computational Project on Molecular Quantum Dynamics). Part of the work has been carried out on the Ra Supercomputer at the HiPerSPACE Computing Centre, UCL.

Appendix

Here we present matrix elements of $T_{\rm ang}$ in the angular basis given by Eq. (2). It is useful to introduce radial functions

$$b_1 = \frac{\hbar^2}{2\mu_1 q_1^2} + \frac{\hbar^2}{2\mu_3 q_3^2} \tag{5}$$
$$b_2 = \frac{\hbar^2}{2\mu_2 q_2^2} + \frac{\hbar^2}{2\mu_3 q_3^2} \tag{6}$$
$$b_3 = \frac{\hbar^2}{2\mu_3 q_3^2}. \tag{7}$$

J and p are strictly conserved and are therefore omitted in the formulae below.

The matrix elements diagonal in K are given by:

$$\langle \gamma, K, k, j, l | T_{\rm ang} | \gamma, K, k, j, l \rangle = b_1 j(j+1) + b_2 l(l+1)$$
$$+ b_3 \left[J(J+1) - 2(K^2 + k^2 - \gamma K k) \right], \tag{8}$$
$$\langle \gamma, K, k', j, l | T_{\rm ang} | \gamma, K, k, j, l \rangle = b_3 \,\mathrm{sign}(k - \gamma K) \sqrt{1 + \delta_{K0}\delta_{k0}} C^+_{j,k-\gamma K} C^+_{l,k} \delta_{k',k+1}$$
$$+ b_3 \,\mathrm{sign}(k' - \gamma K) \sqrt{1 + \delta_{K0}\delta_{k1}} C^-_{j,k-\gamma K} C^-_{l,k} \delta_{k',k-1}, \tag{9}$$
$$\langle \gamma' = 1, K, k' = 1, j, l | T_{\rm ang} | \gamma = -1, K, k = 0, j, l \rangle = -b_3 \, C^-_{j,K} C^-_{l,0}, \tag{10}$$

where $C^{\pm}_{lk} = \sqrt{l(l+1) - k(k \pm 1)}$. The quantity sign$(I)$ takes the value -1 if $I < 0$ and $+1$ if $I \geq 0$. Inspection shows that these matrix elements do not depend on the parity quantum number p. This allows the eigenvectors for the $K > 0$ diagonal blocks to be re-used in the construction of the matrix elements for the off-diagonal blocks.

The matrix elements off-diagonal in K are given by:

$$\begin{aligned}
&\langle \gamma, K' = K-1, k', j, l | T_{\text{ang}} | \gamma, K, k, j, l \rangle = -\gamma b_3 C^{-}_{JK} \Big[C^{-\text{sign}(\gamma)}_{l,k} \delta_{k',k-\gamma} \\
&+\text{sign}(k - \gamma K) \sqrt{1 + \delta_{K1}\delta_{k0}}\, C^{\text{sign}(\gamma)}_{j,k-\gamma K} \delta_{k',k} \Big] , \qquad (11) \\
&\langle \gamma, K' = K+1, k', j, l | T_{\text{ang}} | \gamma, K, k, j, l \rangle = -\gamma b_3 C^{+}_{JK} \Big[C^{\text{sign}(\gamma)}_{l,k} \delta_{k',k+\gamma} \\
&+\text{sign}(k - \gamma K') \sqrt{1 + \delta_{K0}\delta_{k0}}\, C^{-\text{sign}(\gamma)}_{j,k-\gamma K} \delta_{k',k} \Big] , \qquad (12) \\
&\langle \gamma' = 1, K = 1, k, j, l | T_{\text{ang}} | \gamma = -1, K = 0, k, j, l \rangle \\
&= -b_3 C^{+}_{J,0} C^{-}_{j,k} (-1)^{J+p}, \qquad (13) \\
&\langle \gamma' = 1, K = 1, k+1, j, l | T_{\text{ang}} | \gamma = -1, K = 0, k, j, l \rangle \\
&= -b_3 \sqrt{1 + \delta_{k0}}\, C^{+}_{J,0} C^{+}_{l,k} (-1)^{J+p}, \qquad (14) \\
&\langle \gamma' = 1, K' = K+1, k' = 1, j, l | T_{\text{ang}} | \gamma = -1, K, k = 0, j, l \rangle \\
&= -b_3 \sqrt{1 + \delta_{K0}}\, C^{+}_{J,K} C^{+}_{l,0}. \qquad (15)
\end{aligned}$$

If $K = k = 0$ and $J + p$ is even then the last two equations are the same and need be used only once.

References

[1] X. Chapuisat and C. Iung, Phys. Rev. A **45**, 6217 (1992).

[2] A. R. Cooper and J. M. Hutson, J. Chem. Phys. **98**, 5337 (1993).

[3] D. W. Schwenke, J. Phys. Chem. **100**, 2868 (1996); **100**, 18884 (1996).

[4] M. M. Law, I. A. Atkinson and J. M. Hutson (editors), "*Rovibrational Bound*

States in Polyatomic Molecules", (CCP6, Daresbury 1999).

[5] M. Mladenović, J. Chem. Phys. **112**, 1070 (2000).

[6] Y. Volobuev, W. C. Necoechea and D. G. Truhlar, Chem. Phys. Lett. **330**, 471 (2000).

[7] M. Mladenovic and M. Lewerenz, Chem. Phys. Lett. **321**, 135 (2000).

[8] Z. Bačić, Comp. Phys. Comm. **128**, 46 (2000).

[9] J. Koput, S. Carter and N. C. Handy, J. Chem. Phys. **115**, 8345 (2001).

[10] H.-S. Lee and A. B. McCoy, J. Chem. Phys. **116**, 9677 (2002).

[11] H.-G. Yu and J. T. Muckerman, J. Mol. Spectrosc. **214**, 11 (2002).

[12] I. N. Kozin, M. M. Law and J. N. L. Connor (editors), "*Wide-amplitude Rovibrational Bound States in Polyatomic Molecules*", (CCP6, Daresbury 2002).

[13] F. Gatti, C. Iung, M. Menou, Y. Justum, A. Nauts, and X. Chapuisat, J. Chem. Phys. **108**, 8804 (1998).

[14] C. Iung, F. Gatti, A. Viel, and X. Chapuisat, Phys. Chem. Chem. Phys. **1**, 3377 (1999).

[15] J. C. Light, I. P. Hamilton and J. V. Lill, J. Chem. Phys. **92**, 1400 (1985).

[16] J. Tennyson, J. R. Henderson and N. G. Fulton, Comp. Phys. Comm. **86**, 175 (1995).

[17] I. N. Kozin, M. M. Law, J. M. Hutson and J. Tennyson, J. Chem. Phys. **118**, 4896 (2003).

[18] I. N. Kozin, M. M. Law, J. Tennyson, and J. M. Hutson, manuscript in preparation.

[19] RVIB4 is a tetraatomic rovibrational variational code written by S. Carter and N.C. Handy. See J. Molec. Spectrosc., 192, 263 (1998); J.Phys. Chem., A102, 6325 (1998); Molec. Phys., 90, 729 (1997); J. Molec. Spectrosc., 157, 301 (1993) and references therein; RVIBNH3 is a tetraatomic rovibrational variational code written by S. Carter, S.M. Colwell and N.C. Handy. See Molec. Phys., 96, 477 (1999).

[20] CCP6 (the Collaborative Computational Project on Molecular Quantum Dynamics), URL: http://www.ccp.ac.uk/ .

[21] J. Tennyson and A. van der Avoird, J. Chem. Phys. **77**, 5664 (1982).

[22] J. C. Light and T. Carrington, Jr., Adv. Chem. Phys. **114**, 263 (2000).

[23] J. Tennyson and B. T. Sutcliffe, J. Mol. Spectrosc. **101**, 71 (1983).

[24] J. R. Henderson, C. R. Le Sueur, and J. Tennyson, Comp. Phys. Comm. **75**, 379 (1993).

[25] J. Tennyson and B. T. Sutcliffe, J. Chem. Phys. **77**, 4061 (1982).

[26] Z. Bačić and J. C. Light, Ann. Rev. Phys. Chem. **40**, 469 (1989).

[27] When we construct the matrix elements defined by equation (8) we temporarily set $J = K$: that is we omit the term $b_3[J(J+1) - K(K+1)]$ from the diagonal elements; this term is properly included only in the final ro-vibrational diagonalisation. This allows the K-optimised eigenvectors/values to be used for all J ($\geq K$).

[28] I. N. Kozin, J. Tennyson, and M. M. Law, Comp. Phys. Comm. (accepted December 2003).

[29] J. R. Henderson J. Tennyson and B. T. Sutcliffe, J. Chem. Phys. **98**, 7191 (1993).

[30] N. J. Wright and J. M. Hutson, J. Chem. Phys. **110**, 902 (1999).

[31] M. J. Bramley, J. W. Tromp, T. Carrington, Jr., and G. C. Corey, J. Chem. Phys. **100**, 6175 (1994).

[32] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, "*LAPACK Users' Guide*", Third Edition, (Society for Industrial and Applied Mathematics, Philadelphia 1999).

[33] S. Carter, I. M. Mills, J. N. Murrell, Mol. Phys. **41**, 191 (1980); L. Halonen, M. S. Child, S. Carter, Mol. Phys. **47**, 1097 (1982).

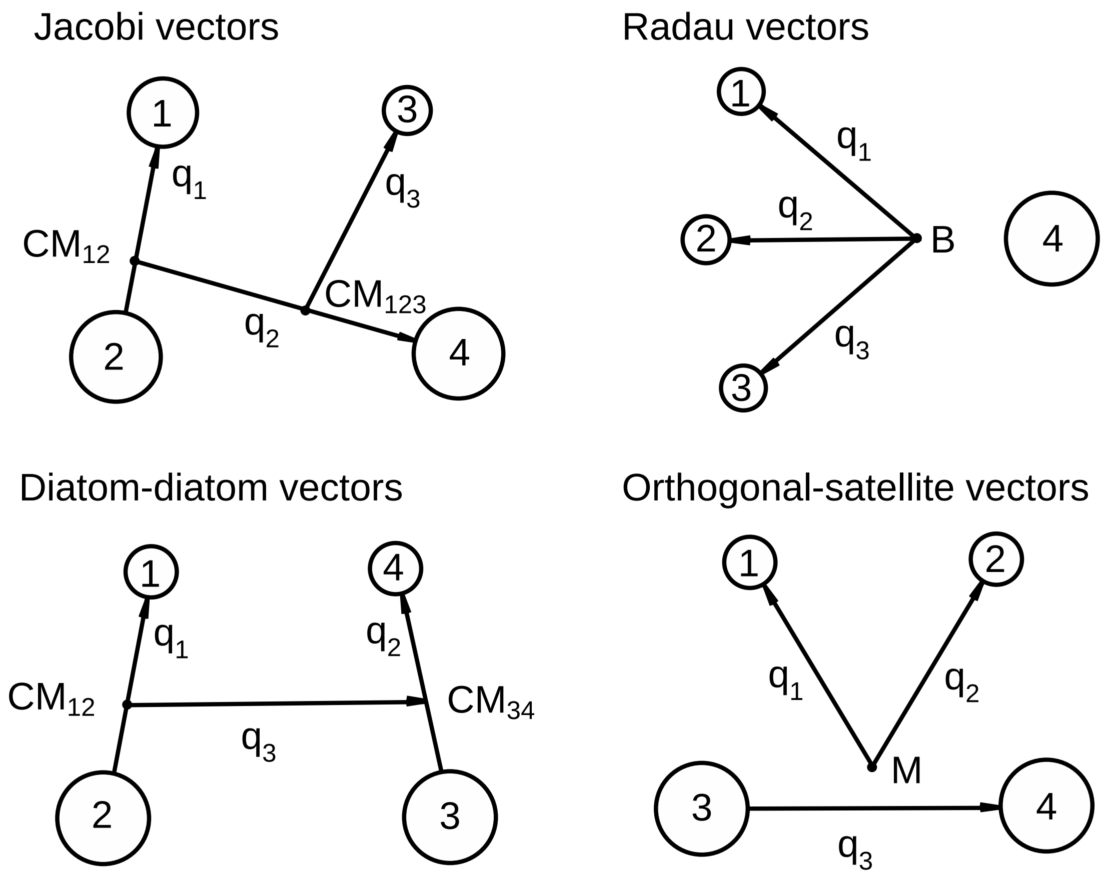

Fig. 1. Coordinate systems supported by WAVR4. The positions of the points CM_{12}, CM_{123}, CM_{34}, B and M are defined in Ref. [5].

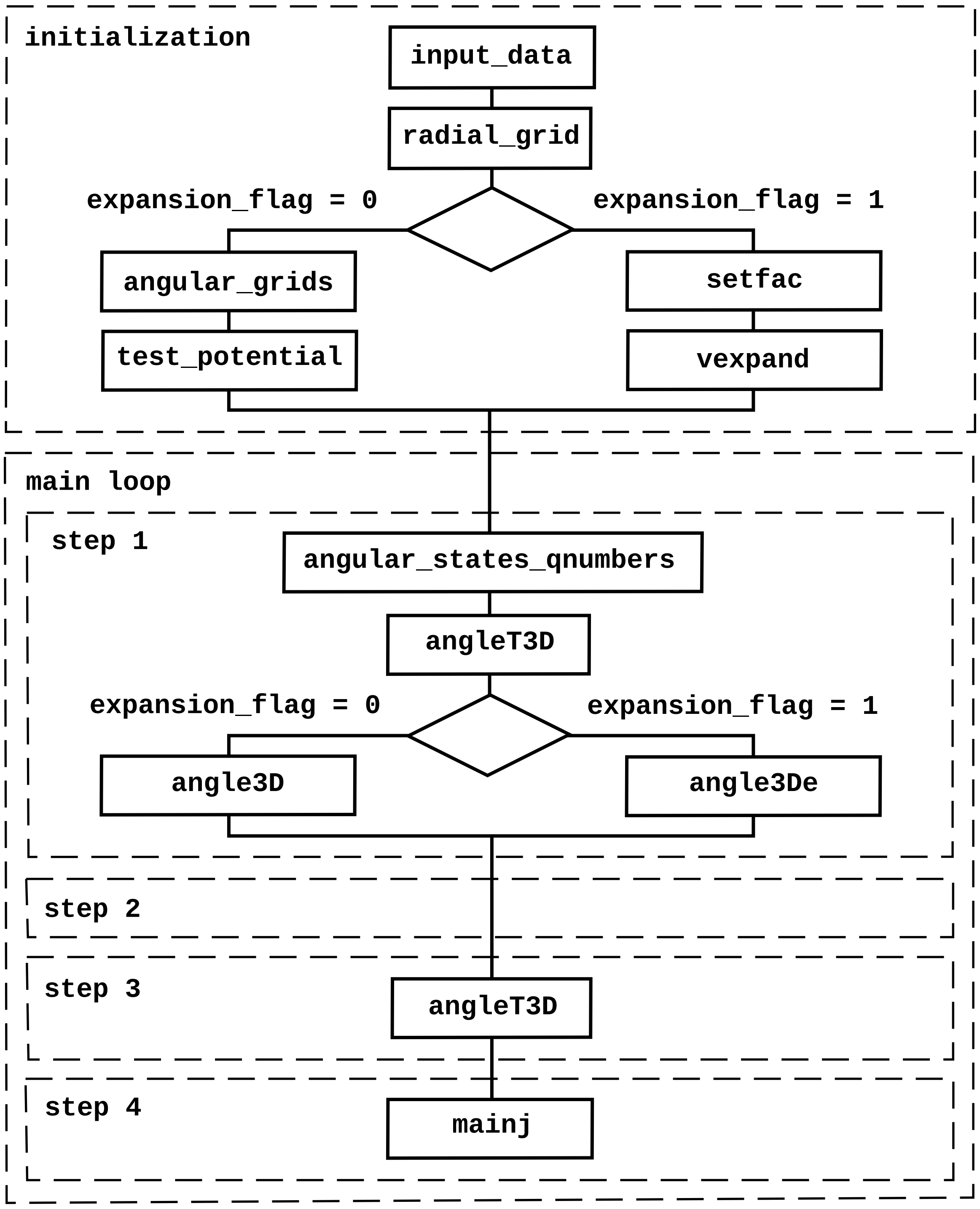

Fig. 2. Diagram of WAVR4 program flow.

```
 WAVR4 version 1.0: (1,2),3 h6swap

 Four Atomic RoVibrational Program
 INPUT: reading input...
==============================< file:  input.txt >==============================
 TITLE:  HCCH; coords: diat-diat (valence-like); potential: Carter et al (1982)
!  m1              m2              m3              m4                  <-- masses
    1.007825032   12.000000000   12.000000000    1.007825032
! Coordinate system:               <-- 1 Jacobi; 2 Radau; 3 DDiatom; 4 Orthog-Sat
    3
! type re/eta     we         De        gridsize      <-- radial basis/quadrature
   1  1.1100    3200.000   43000.000     4
   1  1.1100    3200.000   43000.000     4
   1  1.3720    2100.000   45000.000     5
! angular_problem_only_flag  optimize_flag  test_flag  expansion_flag
          0                        3             1            0
!  jmax lmax kmax Jmax Kmax j_par l_par j+l_par           <-- angular basis size
    12   12    4    1    1    0    0    0
!    j    l    k                 <-- angular potential expansn size (if needed)
     8    8    8
!  nt1  nt  nphi iauto (1 - on)   <-- angular quadrature size (autoslct: on/off)
     9    9    5    1
! enzero                         <-- zero energy is used only when not computed
        0.0000000000
!   encut0  ! encut1    encut2    encut3     encut4      <-- Ecut#/icut#/margin#
          0        50        80        40        10
          0.    30000.    30000.    30000.    30000.
          0.    30000.       0.        0.        0.
          0         0         0         0         0
!  r1(1)    r2(2)    R(3)    theta1(4) theta(5) phi(6)      <-- reference config
   1.06100  1.06100  1.36700  3.14159  0.00000  0.00000
!  stage_flag         <-- used to read in saved eigenvectors from previous stage
          0
!  oner_flag                     <-- 1/R^2 treatment: 0 - DVR approx, 1 - exact
          0
================================================================================
 INPUT: processing input...

 opt 3: Diatom-diatom vectors

 adapted masses:
    0.92974040    0.92974040    6.50391252

 Basis Optimization is ENABLED

  j-, l-, (j+l)-parity:
  NO:  q1 -> -q1 symmetry (j-parity)
  NO:  q2 -> -q2 symmetry (l-parity)
  NO:  q3 -> -q3 symmetry (jl-parity)

 angular basis size:               jmax= 12 lmax= 12 kmax=  4

 angular potential expansion size: ne1 =  8 ne2 =  8 ne3 =  8

 (auto) signle ang grid, iauto= 1:
 nt1 = 13 nt  = 13 nphi=  3

 full angular grid: nagrid = nt1*nt*nphi =   507

 input zero energy   (enzero)=       0.0
 stage #;      icut;  E cutoff  E margin
 stage 0:         0       0.0       0.0         0
 stage 1:        50   30000.0   30000.0         0
 stage 2:        80   30000.0       0.0         0
 stage 3:        40   30000.0       0.0         0
 stage 4:        10   30000.0       0.0         0

   Equilibrium/reference Valence configuration:
   Re1=          1.06100000 AA
   Re2=          1.06100000 AA
   Re3=          1.36700000 AA
   Th1=          3.14159300 rad
   Th2=          0.00000000 rad
   Phi=          0.00000000 rad

   Equilibrium/reference energy:       -0.014511

 INPUT: done.

  angular_states_max: namax=        1663
  angular_states_max: mmax =           5
  angular_states_max: nexp =         285

 stretch: processing r1...
 stretch: done.
 stretch: processing r2...
 stretch: done.
 stretch: processing r3...
 stretch: done.

 computing theta grid...
 done.
 computing theta grid...
 done.
 computing phi grid...
 done.

 preparatory stage
 Last step:      0.05s; total:      0.08s.

   %%%%     jp= 0    %%%%

  angular_states_qnumbers: na=        1663
  number of basis functions for various K
    K    # start  end
    0  615    1  615
    1 1048  616 1663

   %%%%  K= 0  p= 0  %%%%

 Last step:      0.00s; total:      0.08s.
   angular K sub-block: na =         615
  stage1 vectors will be computed
  stage2 vectors will be computed
  stage3 vectors will be computed

 starting stage 1...

 stage 1: made    3969 records
 Last step:     92.96s; total:     93.04s.
  direct product basis=     49200
  optimized      basis=      3969
  effective energy window for optimised basis was
    from        1936.438 to       30000.000
  max number of selected levels=        50
  istage1=        3969
 Last step:      0.22s; total:     93.27s.
 end of stage 1

 starting stage 2...

  max number of records per i3, nf =         800
  effective energy window for optimised basis was
    from        4880.770 to       30000.000
  max number of selected levels=        80
 stage 2: made     400 records
 Last step:      8.28s; total:    101.55s.
 end of stage 2

 starting stage 3...

   matrix size=  400
  allocating and reading in h6...
  read       80200  out of       80200
 Last step:     15.59s; total:    117.14s.
  diagonalising h6...

 zero energy=            5827.59151952

 h6 energies (relative to zero energy):
     1              0.000000
     2           1347.190247
     3           1457.040837
     4           1559.452193
     5           1943.875342

<truncated>
```