Computer Physics Communications 75 (1993) 379–395
North-Holland

Computer Physics
Communications

DVR3D: programs for fully pointwise calculation of vibrational spectra

James R. Henderson, C. Ruth Le Sueur and Jonathan Tennyson

Department of Physics and Astronomy, University College London, Gower St., London WC1E 6BT, UK

Received 18 September 1992

DVR3D calculates rotationless ($J = 0$) vibrational energy levels and wavefunctions for triatomic systems using scattering (Jacobi) coordinates, or optionally unsymmetrised Radau coordinates, for a given potential energy surface. The program uses a discrete variable representation (DVR) based on Gauss–Legendre and Gauss–Laguerre quadrature for all 3 internal coordinates and thus yields a fully pointwise representation of the wavefunctions. Successive diagonalisation and truncation is implemented for 4 of the possible 6 possible coordinate orderings. DVR3D is best used for problems for which many (several hundred) vibrational states are required. Given appropriate dipole surfaces, the accompanying program DIPJ0DVR computes vibrational band intensities for wavefunctions generated by DVR3D.

PROGRAM SUMMARY

Title of program: DVR3D

Catalogue number: ACNE

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

Licensing provisions: none

Computer: Convex C3840 running BSD unix; *Installation*: University of London Computer Centre

Other machines on which program has been tested: Cray-YMP8i, IBM RS6000

Programming language used: FORTRAN 77

Memory required to execute with typical data: case dependent

Peripherals used: card reader, line printer, disk files.

No. of lines in distributed program including test data, etc.: 3848

Keywords: vibrations, body-fixed, discrete variable representation, Coriolis decoupled, finite elements, Gaussian quadrature, vectorised

Nature of physical problem
DVR3D calculates the bound vibrational or Coriolis decoupled rovibrational states of a triatomic system in body-fixed Jacobi (scattering) or Radau coordinates coordinates [1].

Method of solution
All co-ordinates are treated in a discrete variable representation (DVR). The angular coordinate uses a DVR based on (associated) Legendre polynomials and the radial coordinates utilise a DVR based on either Morse oscillator-like or spherical oscillator functions. Intermediate diagonalisation and truncation is performed on the hierarchical expression of the Hamiltonian operator to yield the final secular problem. DVR3D provides the data necessary for DIPJ0DVR [2] to calculate vibrational band intensities.

Restrictions on the complexity of the problem
(1) The size of the final Hamiltonian matrix that can practically be diagonalised. DVR3D allocates arrays dynamically at execution time and in the present version the total

---

Correspondence to: J. Tennyson, Department of Physics and Astronomy, University College London, Gower St., London, WC1E 6BT, UK.

space available is a single parameter which can be reset as required. (2) The order of integration in the radial co-ordinates that can be dealt within the machine exponent range. Some adjustment in the code may be necessary when large order Gauss-Laguerre quadrature is used.

Typical running time
Case dependent but dominated by the final (3D) matrix diagonalisation. The test data takes 229 s on a Convex C3840.

Unusual features of the program
A user supplied subroutine containing the potential energy as an analytic function (optionally a Legendre polynomial expansion) is a program requirement.

References

[1] J.R. Henderson, PhD Thesis, University of London (1990).
[2] J.R. Henderson, C.R. Le Sueur and J. Tennyson, this article, second program (DIPJ0DVR).

PROGRAM SUMMARY

Title of program: DIPJ0DVR

Catalogue number: ACNF

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

Computer: Convex C3840 running BSD unix; *Installation*: University of London Computer Centre

Other machines on which program has been tested: Cray-YMP8i.

Programming language used: FORTRAN 77

Memory required to run with typical data: case dependent

Peripherals used: card reader, line printer, at least two disk files

No. of lines in distributed program, including test data, etc.: 1999

Keywords: vibrational band intensities, vectorised, discrete variable representation

Nature of physical problem
DIPJ0DVR calculates the vibrational band intensities between eigenstates calculated using the DVR method.

Method of solution
The amplitude of the eigenstates at the DVR points are read in. The value of the dipole components are then calculated at these points and transformed into components along the Eckart axes [1]. The vibrational band intensity is the sum of the products of the dipole and the eigenstate amplitudes at all these points.

Restrictions on the complexity of the problem
The number of transitions to be calculated. DIPJ0DVR allocates arrays dynamically at execution time and in the present version the total space available is a single parameter which can be reset as required.

Typical running time
7 seconds to run test data on Convex C3840.

Unusual features of the program
Most data is read directly from DVR3D [2]. A user supplied subroutine providing the dipole at any point in configuration space is a program requirement.

References

[1] C.R. Le Sueur, J. Tennyson, S. Miller and B.T. Sutcliffe, Mol. Phys. 76 (1992) 1147.
[2] J.R. Henderson, C.R. Le Sueur and J. Tennyson, this article, first program (DVR3D).

LONG WRITE-UP

1. Introduction

The discrete variable representation, DVR, along with other pointwise methods, is becoming more and more popular and successful as a technique for determining highly excited bound rotation–vibration states of triatomic molecules [1]. The experimental community is also enjoying new found enthusiasm and success in this area of spectroscopy and molecular dynamics, due largely to advanced laser technology and techniques such as stimulated emission pumping.

The DVR was introduced by Harris et al. [2] and further developed by Light and co-workers [1,3,4] who showed the power of the DVR for obtaining many vibrational bound states of triatomic molecules. Henderson, Tennyson and co-workers subsequently exploited this [5–9] using mixed DVR and finite basis representation (FBR) studies, which also included rotationally excited states in a DVR (DVR^1–FBR^2) [5,11]. There are now a number of triatomic systems for which estimates of more than 500 vibrational band origins have been made using a DVR. This is a goal more traditional FBR based methods struggle hard to meet. More recently Whitnell and Light [10] and us [12–14] have studied triatomic species using a full 3-dimensional (3D) DVR, ie. treating all three internal co-ordinates in a pointwise manner. This method has proved remarkably successful in determining all the vibrational ($J=0$) bound states of H_3^+ [14] and even more powerful than a comparable DVR^1–FBR^2 study [5].

We present here our DVR^3 program, DVR3D, which utilises a DVR in each of the scattering (or Radau) coordinates r_1, r_2 and θ. The code calculates vibrational ($J=0$) eigenenergies and wavefunction values, and has been used successfully on H_3^+ [14], HCN/HNC [13] and H_2S. Optionally the code will also calculate Coriolis decoupled rotationally excited states of the molecule, although there is as yet no means of coupling these states to give a full rotational calculation.

Using the wavefunctions provided by DVR3D, we have developed a further program to calculate vibrational band intensities using the theory of Le Sueur et al. [15]. This program, DIPJ0DVR, is included as part of the suite.

2. Method

2.1. The vibrational problem: DVR3D

2.1.1. Formulating and solving the 3D DVR Hamiltonian

We use a multidimensional DVR in scattering (Jacobi) or Radau co-ordinates. In scattering coordinates r_1 represents the “diatom” distance between atom 1 and atom 2, and r_2 the separation of the third atom from the diatom centre of mass. The angle between r_1 and r_2 is θ. A formal definition of (r_1, r_2, θ) in Radau coordinates can be found elsewhere [16].

Using a finite basis representation (FBR), the zero rotational angular momentum ($J=0$) Hamiltonian matrix can be written as [17]

$$\begin{aligned}\langle m, n, j \,|\, \hat{H} \,|\, m', n', j' \rangle = {} & \langle m \,|\, \hat{h}^{(1)} \,|\, m' \rangle \delta_{n,n'} \delta_{j,j'} + \langle n \,|\, \hat{h}^{(2)} \,|\, n' \rangle \delta_{m,m'} \delta_{j,j'} \\ & + \big(\langle m \,|\, \hat{g}^{(1)} \,|\, m' \rangle \delta_{n,n'} + \langle n \,|\, \hat{g}^{(2)} \,|\, n' \rangle \delta_{m,m'} \big) j(j+1) \delta_{j,j'} \\ & + \langle m, n, j \,|\, V(r_1, r_2, \theta) \,|\, m', n', j' \rangle . \end{aligned} \tag{1}$$

If $J > 0$ and the projection of J along the body-fixed z-axis is designated k, then an extra, diagonal in k term is added to (1):

$$\begin{aligned}\langle m, n, j, J, k \,|\, \hat{H} \,|\, m', n', j', J, k \rangle = {} & \langle m, n, j \,|\, \hat{H} \,|\, m', n', j' \rangle \\ & + \langle t \,|\, \hat{g}^{(i)} \,|\, t' \rangle \delta_{j,j'} \delta_{s,s'} \big(J(J+1) - 2k^2 \big) . \end{aligned} \tag{2}$$

In (2), if the body-fixed z-axis is taken along r_1 then $|t\rangle = |m\rangle$, $s = n$ and $i = 1$; conversely if z is along r_2, $|t\rangle = |n\rangle$, $s = m$ and $i = 2$.

A DVR is essentially a unitary transformation of an FBR defined for some quadrature scheme associated with the FBR polynomials. When using a multidimensional DVR (i.e. a totally pointwise representation), one needs always to start (at least notionally anyway) by defining the FBR. Further, it is comforting to know that one can, at any time, transform between the two representations uniquely.

In this work the angular basis functions $|j\rangle$ are Legendre polynomials, or associated Legendre functions if $k \neq 0$. The radial basis functions are similar to those employed in our FBR^3 and DVR^1–FBR^2 programs [6,19]. These are either Morse oscillator-like functions or spherical oscillators. The Morse oscillator-like functions are defined as [18]

$$|n\rangle = H_n(r) = \beta^{1/2} N_{n\alpha} \exp(-\tfrac{1}{2}y)\; y^{(\alpha+1)/2} L_n^{\alpha}(y), \quad y = A \exp[-\beta(r - r_e)], \tag{3}$$

where

$$A = 4D_e/\beta, \qquad \beta = \omega_e(\mu/2D_e)^{1/2}, \qquad \alpha = \text{integer}(A). \tag{4}$$

The parameters μ, r_e, ω_e and D_e can be associated with the reduced mass, equilibrium separation, fundamental frequency and dissociation energy of the relevant coordinate, respectively. In practice (r_e, ω_e, D_e) are treated as variational parameters and optimised accordingly. $N_{n\alpha} L_n^{\alpha}$ is a normalised associated Legendre polynomial [20].

The spherical oscillator functions are particularly useful for systems which have significant amplitude for $r_2 = 0$. These functions are defined by [21]

$$\begin{aligned} |n\rangle &= H_n(r) = 2^{1/2}\beta^{1/4} N_{n\alpha+1/2} \exp(-\tfrac{1}{2}y)\; y^{(\alpha+1)/2} L_n^{\alpha+1/2}(y), \\ y &= \beta r^2, \end{aligned} \tag{5}$$

where

$$\beta = (\mu\omega_e)^{1/2} \tag{6}$$

and (α, ω_e) are treated as vibrational parameters.

It should be noted that our usual practice is to optimise the parameters for both Morse-like and spherical oscillator functions using an FBR isomorphic to the DVR in which the final calculation is to be performed. To this end optimisation is generally performed using the FBR^3 code TRIATOM [19] or the two-dimensional (ZTWOD = T) option in the DVR^1–FBR^2 code DVR1D [11].

In (1) V is the potential, and the radial kinetic energy integrals are given by

$$\langle t|\hat{h}^{(i)}|t'\rangle = \left\langle t \left| \frac{-\hbar^2}{2\mu_i r_i^2} \frac{\partial^2}{\partial r_i^2} \right| t' \right\rangle, \tag{7}$$

$$\langle t|\hat{g}^{(i)}|t'\rangle = \left\langle t \left| \frac{\underline{\hbar^2}}{2\mu_i r_i^2} \right| t' \right\rangle, \tag{8}$$

where $|t\rangle = |m\rangle$ for $i = 1$ and $|t\rangle = |n\rangle$ for $i = 2$. μ_i are the appropriate reduced masses given by [16]

$$\mu_1^{-1} = g_2^2 m_1^{-1} + m_2^{-1} + (1 - g_2)^2 m_3^{-1}, \qquad \mu_2^{-1} = m_1^{-1} + g_1^2 m_2^{-1} + (1 - g_1)^2 m_3^{-1}. \tag{9}$$

where for scattering coordinates

$$g_1 = \frac{m_2}{m_2 + m_3}, \qquad g_2 = 0 \tag{10}$$

and for Radau coordinates [16]:

$$g_1 = 1 - \frac{\alpha}{\alpha + \beta - \alpha\beta}, \quad g_2 = 1 - \frac{\alpha}{1 - \beta + \alpha\beta}, \quad \alpha = \left(\frac{m_3}{m_1 + m_2 + m_3}\right)^{1/2}, \quad \beta = \frac{m_2}{m_1 + m_2}. \tag{11}$$

In principle it is possible to find other combinations of (g_1, g_2) which give other orthogonal coordinate systems [16], but this possibility has not been investigated.

A 1D DVR transformation for either of r_1, r_2 or θ is defined in terms of points, η, and weights, w_η, of the N-point Gaussian quadrature associated with the orthogonal polynomials used for the FBR in that coordinate [6]:

$$T_t^\eta = (w_\eta)^{1/2} | t(\eta) \rangle, \tag{12}$$

where $|t\rangle = |m\rangle$, $|n\rangle$, $|j\rangle$ for $\eta = \alpha$, β, γ, respectively. DVR3D automatically generates Gaussian quadrature schemes using routines adapted [21] from Stroud and Secrest [22].

The required composite transformation is written as a product of 1D transformations:

$$\mathbf{T} = T_{m,n,j}^{\alpha,\beta,\gamma} = T_m^\alpha T_n^\beta T_j^\gamma. \tag{13}$$

A three-dimensional DVR is obtained by applying the transformation $\mathbf{T}^{\mathrm{T}}\mathbf{H}\mathbf{T}$. Thus the 3D $J = 0$ Hamiltonian is

$$^{(3D)}H_{\alpha,\alpha',\beta,\beta',\gamma,\gamma'} = \sum_{m,n,j} \sum_{m',n',j'} T_{m,n,j}^{\alpha,\beta,\gamma} \langle m, n, j | H | m', n', j' \rangle T_{m',n',j'}^{\alpha',\beta',\gamma'}. \tag{14}$$

The transformed Hamiltonian can be written at the DVR grid points as [12]

$$\begin{aligned} ^{(3D)}H_{\alpha,\alpha',\beta,\beta',\gamma,\gamma'} = {} & K_{\alpha,\alpha'}^{(1)} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'} + K_{\beta,\beta'}^{(2)} \delta_{\alpha,\alpha'} \delta_{\gamma,\gamma'} + L_{\alpha,\alpha',\gamma,\gamma'}^{(1)} \delta_{\beta,\beta'} + L_{\beta,\beta',\gamma,\gamma'}^{(2)} \delta_{\alpha,\alpha'} \\ & + (J(J+1) - k^2) M_{\alpha,\alpha',\beta,\beta'}^{(i)} \delta_{\gamma,\gamma'} + V(r_{1\alpha}, r_{2\beta}, \theta_\gamma) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'}, \end{aligned} \tag{15}$$

where $i = 1$ for z embedded along r_1 and $i = 2$ for z along r_2.

In (15), the potential energy operator is diagonal because of the quadrature approximation [25]

$$\sum_{m,n,j} \sum_{m',n',j'} T_{m,n,j}^{\alpha,\beta,\gamma} \langle m, n, j | V(r_1, r_2, \theta) | m', n', j' \rangle T_{m',n',j'}^{\alpha',\beta',\gamma'} \simeq V(r_{1\alpha}, r_{2\beta}, \theta_\gamma) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'}, \tag{16}$$

where $(r_{1\alpha}, r_{2\beta}, \theta_\gamma)$ is the value of (r_1, r_2, θ) at (α, β, γ). Note a major attraction here, in that no integration at all is required over the potential; it is diagonal in every co-ordinate.

The kinetic energy terms in (15) are represented by

$$K_{\eta,\eta'}^{(i)} = \sum_{t,t'} T_t^n \langle t | \hat{h}^{(i)} | t' \rangle T_{t'}^{n'}, \tag{17}$$

$$L_{\eta,\eta',\gamma,\gamma'}^{(i)} = J_{\gamma\gamma'} \sum_{t,t'} T_t^n \langle t | \hat{g}^{(i)} | t' \rangle T_t^{\eta'} \tag{18}$$

$$\simeq \frac{J'_{\gamma\gamma} \hbar^2}{2\mu_i r_{i\eta}^2} \delta_{\eta\eta'}, \tag{19}$$

again applying the quadrature approximation, and where

$$J_{\gamma\gamma'} = \sum_j T_j^\gamma j(j+1) T_j^{\gamma'}. \tag{20}$$

For $J > 0$, the extra term is given by

$$M^{(1)}_{\alpha,\alpha'} = \sum_{m,m'} T^{\alpha}_{m} \langle m \,|\, \hat{g}^{(1)} \,|\, m' \rangle T^{\alpha'}_{m'} \simeq \frac{\hbar^2}{2\mu_1 r^2_{1\alpha}}, \tag{21}$$

if $i = 1$, when it is diagonal in β, and

$$M^{(2)}_{\beta,\beta'} = \sum_{n,n'} T^{\beta}_{n} \langle n \,|\, \hat{g}^{(2)} \,|\, n' \rangle T^{\beta'}_{n'} \simeq \delta_{\beta,\beta'} \frac{\hbar^2}{2\mu_2 r^2_{2\beta}}, \tag{22}$$

if $i = 2$, when it is diagonal in α.

The calculation is generally set up as a series of diagonalisations and truncations [12,23]. Assume for the moment that the coordinate ordering θ then r_1 then r_2 is used, i.e. diagonalise on γ first and β last ($\theta \rightarrow r_1 \rightarrow r_2$). With this ordering, the 1D problems that have to be solved for each α and β are given by

$${}^{(1\mathrm{D})}H^{\alpha,\beta}_{\gamma,\gamma'} = L^{(1)}_{\alpha,\alpha,\gamma,\gamma'} + L^{(2)}_{\beta,\beta,\gamma,\gamma'} + V(r_{1\alpha}, r_{2\beta}, \theta_{\gamma})\delta_{\gamma,\gamma'}. \tag{23}$$

Amplitudes for the kth level, with eigenenergy $\epsilon^{\alpha,\beta}_{k}$, are given at each grid point by $C^{\alpha,\beta}_{\gamma,k}$.

The solutions with $\epsilon^{\alpha,\beta}_{k} \leq E^{1\mathrm{D}}_{\max}$ are then used to solve 2D problems for each value of β. This gives

$${}^{(2\mathrm{D})}H^{\beta}_{\alpha,\alpha',k,k'} = \epsilon^{\alpha,\beta}_{k}\delta_{\alpha,\alpha'}\delta_{k,k'} + \sum_{\gamma} C^{\alpha,\beta}_{\gamma,k} C^{\alpha',\beta}_{\gamma,k'} K^{(1)}_{\alpha,\alpha'}. \tag{24}$$

Solutions for the lth level, with eigenenergy ϵ^{β}_{l} are given by $C^{\beta}_{\alpha,\mathrm{l},\mathrm{k}}$.

The solutions with $\epsilon^{\beta}_{l} \leq E^{2\mathrm{D}}_{\max}$ are then used to solve the full 3D problem of dimension N:

$${}^{(3\mathrm{D})}H_{\beta,\beta',l,l'} = \epsilon^{\beta}_{l}\delta_{\beta,\beta'}\delta_{l,l'} + \sum_{\alpha,k,k'} C^{\beta}_{\alpha,l,k} C^{\beta'}_{\alpha,l',k'} \sum_{\gamma} C^{\alpha,\beta}_{\gamma,k} C^{\alpha,\beta'}_{\gamma,k'} K^{(2)}_{\beta,\beta'}. \tag{25}$$

Solutions of this diagonalisation are the required eigenenergies, ϵ_i, and wavefunction coefficients $C_{\beta,i,l}$.

2.1.2. Wavefunctions

Once the eigenvectors of the 3D Hamiltonian have been obtained one would usually like to use them to express a more physically meaningful quantity. This can be achieved by transforming them to yield values for the amplitude of the wavefunction at the DVR grid points. These wavefunctions can then, in principle, be put to many quantum mechanical and spectroscopic uses. Such uses include the calculation of vibrational band intensities, as explained in the next section, and contour plots of the wavefunctions (or linear combinations of the wavefunctions [24]) which we have found very informative.

The wavefunction amplitude for the ith eigenstate at the DVR grid points is simply

$$\Psi^{i}_{\alpha,\beta,\gamma} = \sum_{l,k} C_{\beta,i,l} C^{\beta}_{\alpha,l,k} C^{\alpha,\beta}_{\gamma,k}.$$

Computationally this process is more efficiently written as

$$\Psi^{i}_{\alpha,\beta,\gamma} = \sum_{l} C_{\beta,i,l} \sum_{k} C^{\beta}_{\alpha,l,k} C^{\alpha,\beta}_{\gamma,k}. \tag{26}$$

Transformations to FBR are also fairly straightforward to write down, but may be computationally expensive.

2.1.3. A problem with the quadrature approximation

The quadrature approximation can be used when evaluating the DVR transformation of FBR matrix elements of an operator that leaves the ket unchanged. Its validity is due to the unitarity of the DVR transformation matrix. We have experienced one particular failure of the quadrature approximation [14], which occurred when evaluating the r_2^{-2} integrals.

In scattering co-ordinates, it is possible for the r_2 co-ordinate to be equal, or very close, to zero if this linear geometry is energetically accessible or favourable. As stated earlier, it is desirable to use the spherical oscillator functions in this case. Under these circumstances it was found [14] that the quadrature approximation had to be abandoned for the r_2^{-2} integral because of its non-polynomial behaviour as $r_2 \to 0$.

After extensive tests on the H_3^+ molecular ion [14], we found a suitable alternative was to continue to use the quadrature approximation in constructing ${}^{(1D)}\mathbf{H}$, and then construct ${}^{(3D)}\mathbf{H}$ using the full-matrix transformation of the r_2^{-2} integrals, correcting for the fact that the quadrature approximation was used in ${}^{(1D)}\mathbf{H}$. So the only change in the formulation is that now

$$ {}^{(3D)}\tilde{\tilde{H}}_{\beta,\beta',l,l'} = {}^{(3D)}H_{\beta,\beta',l,l'} + \sum_{\alpha,k,k'} C^{\beta}_{\alpha,l,k} C^{\beta'}_{\alpha,l',k'} \left(\tilde{M}^{(2)}_{\beta,\beta'} - M^{(2)}_{\beta,\beta'} \right) \sum_{\gamma,\gamma'} C^{\alpha,\beta}_{\gamma,k} C^{\alpha,\beta'}_{\gamma',k'} J_{\gamma,\gamma'}, \tag{27} $$

where

$$ \tilde{M}^{(2)}_{\beta,\beta'} = \sum_{n,n'} T^{\beta}_{n} \langle n \,|\, \hat{g}^{(2)} \,|\, n' \rangle T^{\beta'}_{n'}, \tag{28} $$

and $M^{(2)}_{\beta,\beta'}$ is given by eq. (22).

Note that the FBR matrix elements above can be evaluated analytically, and are given by [21]

$$ \langle n \,|\, \hat{g}^{(2)} \,|\, n' \rangle = \frac{\underline{\hbar^2 \beta}}{(2\alpha + 1)\mu_2} \left(\frac{n!}{n'!} \frac{\Gamma\left(n' + \alpha + \frac{3}{2}\right)}{\Gamma\left(n + \alpha + \frac{3}{2}\right)} \right)^{1/2}, \quad n \geq n'. \tag{29} $$

We have implemented a user input variable (ZQUAD2) in the program DVR3D so that either of eqs. (25) or (27) can be used.

2.1.4. Order of solution

We have shown above how the 3D DVR Hamiltonian matrix of eq. (14) can be solved by successive diagonalisation and trucation. Above the angle θ was dealt with first, and r_2 last. It also is possible for the problem to be solved in any of the 5 other orders. We have recently suggested how one may determine in advance the most efficient order of solution for a particular molecule [13]. It is thought that the co-ordinate accomodating the highest density of states should be treated last and the co-ordinate holding the lowest density of states coming first, although this was found to be less important in practice.

Four of the possible co-ordinate orders have been implemented in DVR3D. The two orders where θ is considered second have been omitted; it is unlikely that such orders will offer significant savings over the case in which the order of θ and the first coordinate are swapped.

2.1.5. Symmetry

Scattering co-ordinates are capable of exploiting the C_{2v} symmetry of a molecule. Thus they can split the full permutation symmetry of an A_2B system and can therefore be of some help in A_3 molecules. We have shown how to do this in a DVR-in-θ approach [5,11]; it involves only a property of the so-called **L** matrix. In our multidimensional DVR we have exactly the same matrix in the **J** matrix of eq. (20). Then the symmetrised **J** matrix becomes

$$ J_{\gamma,\gamma'} = 2 \sum_{j=0}^{N/2-1} T^{\gamma'}_{2j+q} (2j + q)(2j + q + 1) T^{\gamma}_{2j+q}, \quad q = 0, 1. \tag{30} $$

In a symmetrised DVR3D calculation even ($q = 0$) and odd ($q = 1$) calculations are done separately unlike DVR1D [11]. If θ is the first co-ordinate to be treated then no saving can be made from doing both odd and even runs at once. There could be some saving for other orders, although this has not yet been implemented in program DVR3D.

It should be noted that the current version of DVR3D does not employ the full symmetry of an AB_2 system in Radau coordinates. To do this involves symmetrising points taken from different steps in the diagonalisation–truncation procedure and will therefore require major changes to the algorithm employed here. A code to solve this problem is currently under development [26].

2.2. *Vibrational band intensities*

The equation to be evaluated for vibrational band intensities may be written as

$$S_{fi}^{\mathrm{Vib}} = \left| \langle \psi_f | \mu^{\mathrm{E}} | \psi_i \rangle \right|^2, \tag{31}$$

where μ is expressed in terms of components along the Eckart axes [15]. This expression is easy to evaluate in the DVR, where the integral is reduced via the quadrature approximation to a sum:

$$S_{fi}^{\mathrm{Vib}} = \left| \sum_{\alpha\beta\gamma} \psi_i(\alpha\beta\gamma)\ \mu^{\mathrm{E}}(\alpha\beta\gamma)\ \psi_f(\alpha\beta\gamma) \right|^2 . \tag{32}$$

DIPJ0DVR uses wavefunctions produced by DVR3D to evaluate this expression and hence to obtain vibrational band intensities.

Complications can arise if the bra and ket wavefunctions were not calculated at the same points. In this case the transformation matrix must be used to obtain the function expansion of the wavefunctions. DIPJ0DVR can cope with this additional problem for a limited number of cases [27].

3. Program structure

Card input is needed for both DVR3D and DIPJ0DVR. Both programs follow the convention that names beginning with letters A–H and O–Y are for 8-byte real variables, I–N are for integers, and variables whose name begins with Z are logicals.

The calling sequences of DVR3D and DIPJ0DVR are given in figs. 1 and 2. The role of the individual subroutines is described in comment cards included in the source programs.

3.1. *DVR3D*

DVR3D uses dynamical assignment of array space in which one big vector is subdivided in routine CORE. In the current versions, this array is a single fixed length array ARRAY of dimension NAVAIL (set to 500 000 in the version supplied) in subroutine GTMAIN. For efficient storage management a call to a local GETMAIN, MEMORY or HPALLOC command should be implemented.

The CPU time requirement of DVR3D is dominated by the final diagonalisation of the 3D Hamiltonian matrix HAM3. The required diagonaliser has to give all eigenvalues and eigenvectors of a real symmetric matrix. The present implementation uses subroutine EGVQR [28] to mimic NAG routine F02ABF [30]. All of the intermediate diagonalisations of the various HAM1 and HAM2 matrices use this routine. We would strongly recommend that EGVQR is replaced either by the local NAG implementation or by some diagonaliser appropriate to the architecture of the machine.

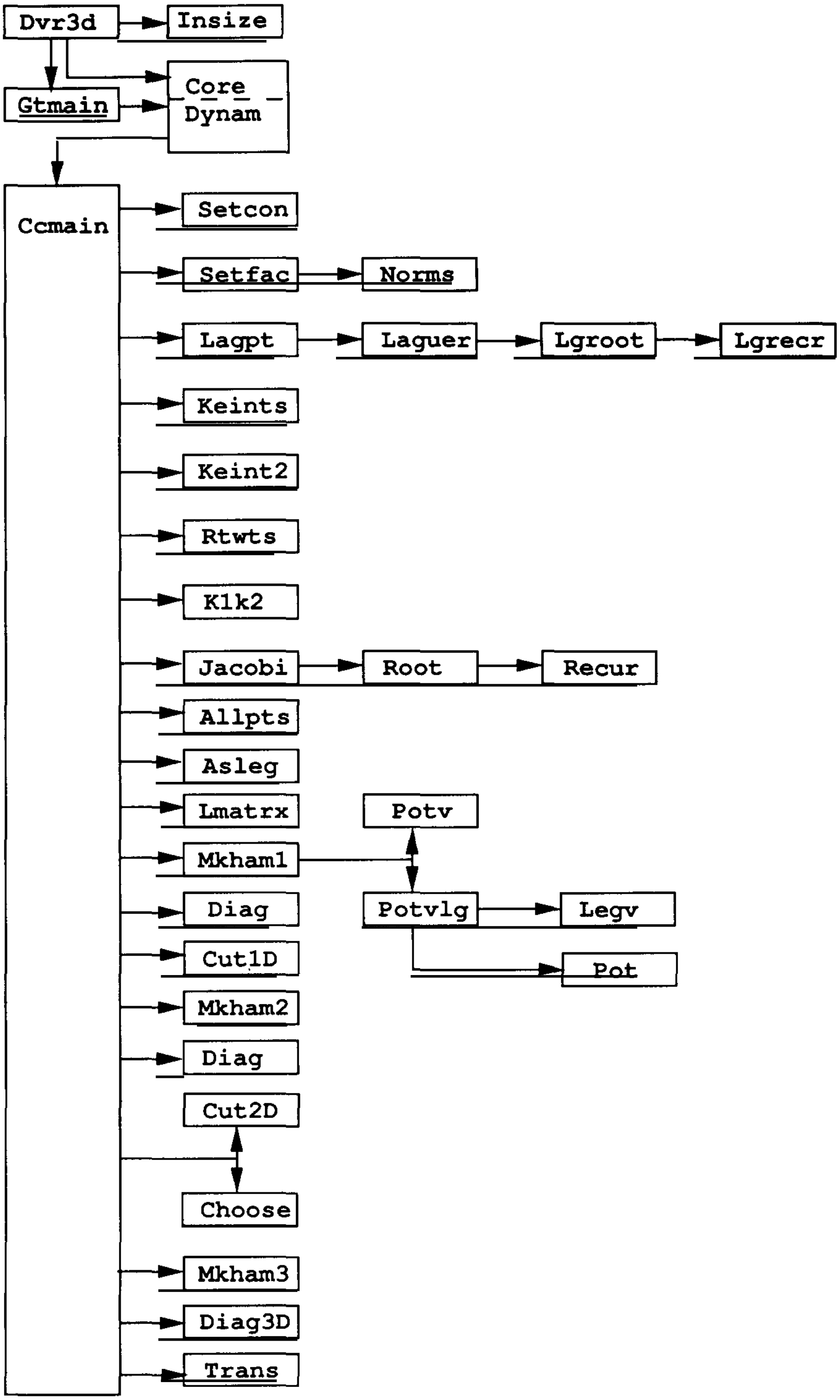

Fig. 1. Structure of program DVR3D. Service routines TIMER, GETROW and OUTROW have been omitted.

3.2. DIPJODVR

DIPJ0DVR also uses dynamical assignment of array space in which one big vector is subdivided. But in this program the fixed length array ARRAY, of length NAVAIL (set to 500000 in this version) is subdivided in routine GTMAIN. For technical reasons, part of this array is also used in the main

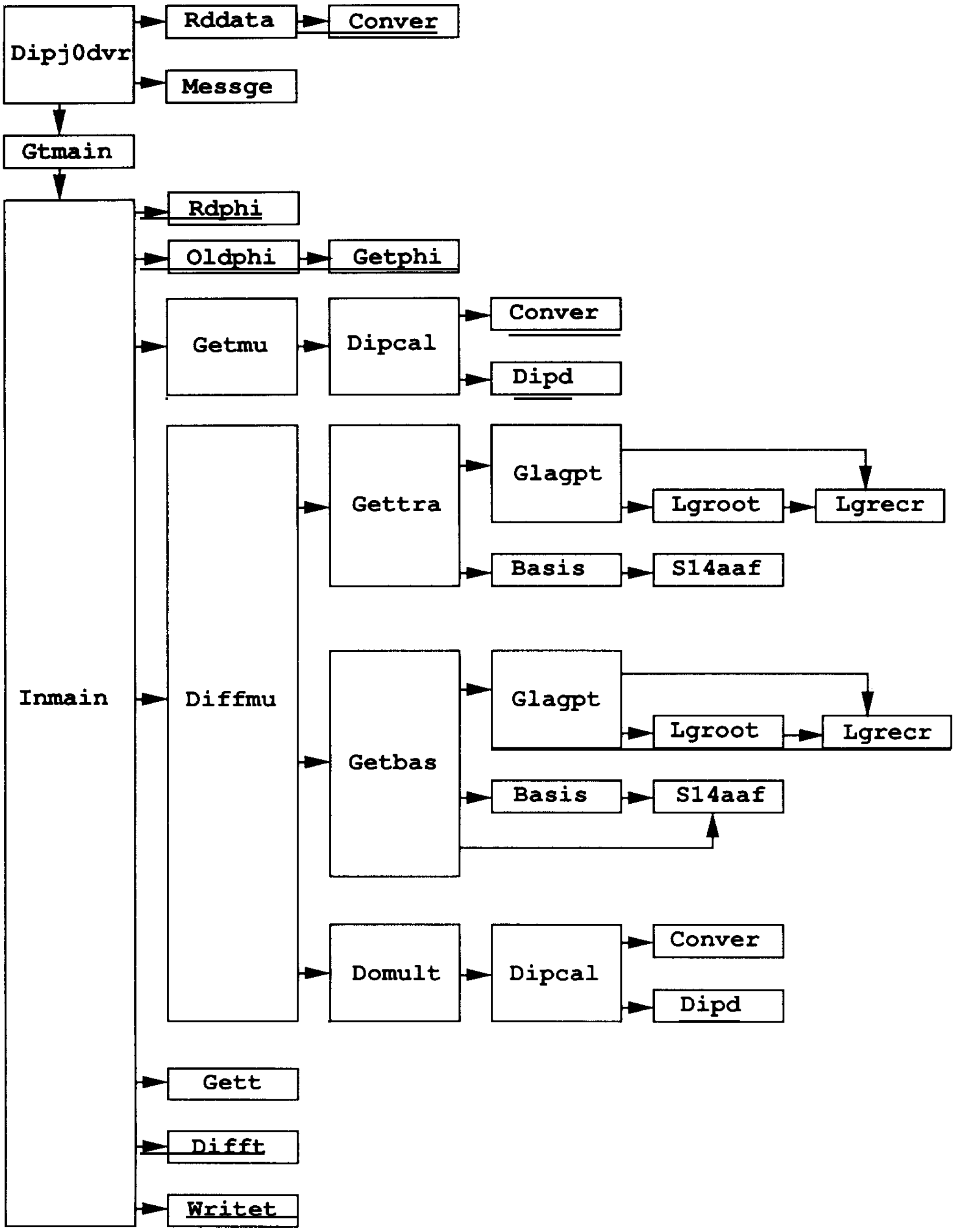

Fig. 2. Structure of program DIPJ0DVR. Service routines TIMER, GETROW and OUTROW have been omitted.

program to read in the DVR points. Each array of DVR points is assumed to be shorter than MAXQ (set to 100 in this version).

For a typical problem, DIPJ0DVR uses a negligible quantity of CPU time. For example in the test run, 7.5 s is used, of which 7.3 s is time used to calculate the dipole at the various DVR points. If, however, the bra and ket use different DVR points, then a substantial proportion of time is spent calculating the transition intensities (see below).

The dipole is provided via a user-supplied subroutine DIPD in the internal coordinate system of the DVR calculation. It is subsequently rotated by DIPJ0DVR into the Eckart system [15]. In order to do this, the equilibrium values of r_1, r_2 and $\cos\theta$ are required. The values of the transition moments can be quite sensitive to these values, and so it is important that they are correct. Accordingly, the program prints out the cartesian coordinates of the atoms corresponding to the given equilibrium values. It is recommended that these are checked before placing any credence on the rest of the output!

The greatest part of the program is taken up with a type of calculation that has not been generalised yet, which was used to calculate transition moments for H_3^+ [27]. Because the particular boundary conditions for this ion require that the basis functions for r_2 are different for even and odd symmetry runs, there are parts of the program (governed by ZSAME = F) which can cope with the bra and ket functions having different points for r_2. This option has only been implemented for spherical oscillator functions in the r_2 coordinate as in our experience the same Morse-like functions are always used for even and odd symmetries. In this case additional input is required and the program is much slower. When calculating the transition moments, the program is written to use quadrature points specified by both the even and odd r_2 functions, and to take the average. However, if as in the original work done on H_3^+ [14], the even and odd functions differ by integer powers of r_2, then it is important to use only quadrature points resulting from the ones specifying lower powers of r_2 in order to avoid inaccuracies of quadrature [27].

Calculations run with ZSAME = F require Γ functions. Subroutine S14AAF is a NAG routine [29] which calculates Γ functions. The present implementation uses routine GAMMLN from "Numerical Recipes" [30] to mimic the operation of S14AAF.

4. Program use

4.1. The potential subroutine

DVR3D requires a user supplied potential energy subroutine. There are two ways of supplying the potential. If it is specified as a Legendre expansion,

$$V(r_1, r_2, \theta) = \sum_{\lambda} V_\lambda(r_1, r_2)\, P_\lambda(\theta), \tag{33}$$

which corresponds to option ZLPOT = .TRUE., then the expansion must be supplied by

SUBROUTINE POT(V0, VL, R1, R2)

which returns V0 = $V_0(r_1, r_2)$ and VL(λ) = $V_\lambda(r_1, r_2)$ in Hartree for R1 = r_1 and R2 = r_2 in Bohr. If NCOORD = 2, R1 contains the rigid diatom bondlength r_e.

If a general potential function, ZLPOT = .FALSE., is to be used then

SUBROUTINE POTV(V, R1, R2, XCOS)

must be supplied. POTV returns the potential V in Hartree for an arbitrary point given by R1 = r_1, R2 = r_2 (both in Bohr) and XCOS = $\cos\theta$.

DVR3D includes COMMON/MASS/XMASS(3), G1, G2 where XMASS contains the atomic masses in atomic units (*not* amu), $G1 = g_1$ and $G2 = g_2$. This enables users to write flexible potential subroutines which allow for changes in coordinates or isotopic substitution. See, for example, the version of POTV supplied.

4.2. Input for DVR3D

DVR3D requires 9 lines of card input for all runs. Cards giving data not required or for which the defaults [given below in parenthesis] are sufficient should be left blank.

Card 1: NAMELIST/PRT/

ZPHAM[F]	= T, request printing of the Hamiltonian matrix.
ZPRAD[F]	= T, requests printing of the radial matrix elements.
ZP1D[F]	= T, requests printing of the results of 1D calculations.
ZP2D[F]	= T, requests printing of the results of 2D calculations.
ZPMIN[F]	= T, requests only minimal printing.
ZPVEC[F]	= T, requests printing of the eigenvectors.
ZLMAT[F]	= T, requests printing of the *L*-matrix.
ZCUT[F]	= T, final dimension selected using an energy cut-off given by EMAX2. = F, final dimension determined by NHAM3.
ZEMBED[T]	Used only if $J > 0$. = T, *z*-axis embedded along r_2; = F, *z*-axis embedded along r_1.
ZMORS1[T]	= T, use More oscillator-like functions for r_1 coordinate; = F, use spherical oscillator functions.
ZMORS2[T]	= T, use Morse oscillator-like functions for r_2 coordinate; = F, use spherical oscillator functions.
ZLPOT[F]	= T, potential supplied in POT; = F, potential supplied in POTV.
ZVEC[F]	= T, store the eigenvector from all the parts of the calculation (1D, 2D and 3D) on stream IOUT2. Further information relating to this (arrays IV1 and IV2) is stored on stream IOUT1.
ZALL[F]	= T, requests *no* truncation of the intermediate solution.
ZTHETA[T]	= T, let θ be first in the order of solution; = F, let θ be last in the order of solution.
ZR2R1[T]	= T, let r_2 come before r_1 in the order of solution; = F, let r_1 come before r_2 in the order of solution.
ZTRAN[F]	= T, perform the transformation of the solution coefficients to the expression for the wavefunction amplitudes at the grid point, eq. (26). Store the data on stream IWAE. ZTRAN = T automatically sets ZVEC = T.
ZQUAD2[T]	= T, use the DVR quadrature approximation for the integrals of the r_2^{-2} matrix, and hence make its DVR transformation diagonal. = F, evaluate the r_2^{-2} integrals fully and perform the DVR transformation on them. Note that ZQUAD2 = F is only implemented for ZMORS2 = F and for ZTHETA = T.
ZDIAG[T]	= F, do not do final diagonalisation, instead the final Hamiltonian matrix is written on units IDIAG [20] and IDIAG [21]. For further details see the source code.
IEIGS[7]	stream for eigenvalues of the 1D solutions.
IVECS1[3]	stream for eigenvectors of the 1D solutions.

IEIGS2[2]	stream for eigenvalues of the 2D solutions.
IVECS2[4]	stream for eigenvectors of the 2D solutions.
IVINT[14]	scratch file used for storing intermediate vectors in building the final Hamiltonian.
IBAND[15]	scratch file used for storing bands of the final Hamiltonian.
INTVEC[16]	scratch file for intermediate storage of the 2D vectors.
IOUT1[24]	stream for arrays IV1 and IV2, which record the sizes of the truncated vectors. Used when ZVEC = T.
IOUT2[25]	stream for the 1D, 2D and 3D vectors for use when ZVEC = T.
IWAVE[26]	stores the wavefunction amplitudes at the grid points when ZTRAN = T.

Card 2: NCOORD (I5)

NCOORD[3]	the number of vibrational coordinates of the problem: = 2 for an atom rigid diatom system, = 3 for a full triatomic.

Card 3: NPNT2, JROT, NEVAL, NALF, MAX2D, MAX3D, IDIA, KMIN, NPNT1, IPAR (10I5)

NPNT2	number of DVR points in r_2 Gauss–(associated) Laguerre quadrature.
JROT[0]	total angular momentum quantum number of the system, J.
NEVAL[10]	number of eigenvalues and eigenvectors required.
NALF	number of DVR points in θ from Gauss–(associated) Legendre quadrature.
MAX2D	maximum dimension of the largest intermediate 2D Hamiltonian.
MAX3D	maximum dimension of the final Hamiltonian. If ZCUT = F, it is the actual dimension, if ZCUT = T, MAX3D is $\geq$ than the number of function selected.
IDIA	= 0 for Radau coordinate, = 1 for scattering coordinates with a heteronuclear diatomic, = 2 for scattering coordinates with a homonuclear diatomic.
KMIN[0]	= k for JROT > 0.
NPNT1	number of DVR points in r_1 from Gauss–(associated) Laguerre quadrature.
IPAR[0]	parity of basis for the homonuclear diatomic (IDIA = 2) case: IPAR = 0 for even parity and = 1 for odd.

Card 4: TITLE (9A8)

A 72 character title.

Card 5: (XMASS(I), I = 1, 3) (3F20.0)

XMASS(I)	contains the mass of atom I in atomic mass units.

Card 6: EMAX1, EMAX2 (2F20.0)

EMAX1	is the first cut-off energy in cm^{-1} with the same energy zero as the potential. This determines the truncation of the 1D solutions.
EMAX2	is the second cut-off energy in cm^{-1} with the same energy zero as the potential. This controls the truncation of the 2D solutions (i.e. the size of the final basis). If ZCUT = F it is ignored and the size of the final Hamiltonian is MAX3D.

Card 7: RE1, DISS1, WE1 (3F20.0)

If NCOORD = 2, RE1 is the fixed diatomic bondlength, DISS1 and WE1 ignored

If NCOORD = 3, RE1 = r_e, DISS1 = D_e and WE1 = ω_e are Morse parameters for the r_1 coordinate when ZMORS1 = T, and are spherical oscillator parameters when ZMORS1 = F.

Card 8: RE2, DISS2, WE2 (3F20.0)

If ZMORS2 = T, RE2 = r_e, DISS2 = D_e and WE2 = ω_e are Morse parameters for the r_2 coordinate.

If ZMORS2 = F, RE2 is ignored; DISS2 = α and WE2 = ω_e are spherical oscillator parameters for the r_2 coordinate.

Card 9: GSTATE (F20.0)

The ground state of the system in cm^{-1} relative to the energy zero.

This *must* be supplied when IDIA = 2, IPAR = 1 and JROT = 0, and can be blank otherwise.

4.3. The dipole subroutine

DIPJ0DVR processes batches of transitions between states defined by the input file. This file is produced by DVR3D stream IWAVE. Usually there will only be one input file, but if the Hamiltonian has two symmetry blocks which were calculated separately, then two may be given. In that case, the inputs for the two DVR runs must agree absolutely except for IPAR which will obviously be different, and, if ZSAME = F, the value of α used to define spherical oscillator functions in the r_2 coordinate.

The stream IWAVE from the DVR run contains most of the data necessary to characterise the run. The rest must be supplied in the input for DIPJ0DVR. The program uses streams IBRA0, IKET0, IBRA1, IKET1, IWAVE0 and IWAVE1, and will stop if any two of them are set of the same channel. All arrays are internally allocated within an area of memory, the size of which is specified by a parameter NAVAIL. The program will stop if NAVAIL is not large enough.

DIPJ0DVR requires a subroutine DIPD(DIPC,R1,R2,XCOS,NU) which must supply the dipoles in the x and z direction of the DVR coordinate embedding. The specifications of DIPD are as follows:

DIPC (DOUBLE PRECISION):	value of the x dipole in atomic units if NU = 1; value of the z dipole in atomic units if NU = 0.
R1 (DOUBLE PRECISION):	value of r_1 in Bohr.
R2 (DOUBLE PRECISION):	value of r_2 in Bohr.
XCOS (DOUBLE PRECISION):	value of cos θ.
NU (INTEGER):	0 or 1.

DIPJ0DVR requires 5 lines of card input for most runs. One extra card is required if the original calculation is done in two symmetry blocks (even and odd) and these use different points for r_2. Cards giving data not required or for which the defaults [given below in square brackets] are sufficient should be left blank.

4.4. Card input for DIPJ0DVR

Card 1: NAMELIST/PRT/

IPTOT[0]	only relevant if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation. = 0 results from just one block are to be calculated. = 2 results from both blocks are to be calculated.
IBRA[50]	the stream on which the wavefunctions for the bra are stored.
IKET[51]	the stream on which the wavefunctions for the ket are stored.

IBRA[52]	the stream on which the wavefunctions from the second (odd) DVR calculation for the bra are stored. This is only relevant if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation.
IKET1[53]	the stream on which the wavefunctions from the second (odd) DVR calculation for the ket are stored. This is only relevant if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation.
IWAVE0[26]	the stream on which the results written to IWAVE (default 26) in the DVR calculation are stored.
IWAVE1[55]	the stream on which the results written to IWAVE (default 26) in the second (odd) DVR calculation are stored. This is only relevant if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation.
ZSAME[T]	= F if the Hamiltonian has two symmetry blocks which were calculated separately and which used different points for r_2.

Card 2: Title (A80)
An eighty character title.

Card 3: LBRA0, LKET0, LBRA1, LKET1 (4I5)

LBRA0	number of lowest state to be used in the bra.
LKET0	number of lowest state to be used in the ket.
LBRA1	number of lowest state to be used from the second symmetry block in the bra (only needed if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation).
LKET1	number of lowest state to be used from the second symmetry block in the ket (only needed if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation).

Card 4: NBRA0, NKET0, NBRA1, NKET1 (4I5)

NBRA0	number of highest state to be used in the bra. (−1 means use the value of NEVAL from IWAVE).
NKET0	number of highest state to be used in the ket.
NBRA1	number of highest state to be used from the second symmetry block in the bra (only needed if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation).
NKET1	number of highest state to be used from the second symmetry block in the ket (only needed if the Hamiltonian has two symmetry blocks which were calculated separately in the DVR calculation.

Card 5: R1E, R2E, XCOSE (3F20.0)

R1E	is the equilibrium value of r_1 in atomic units.
R2E	is the equilibrium value of r_2 in atomic units.
XCOSE	is the equilibrium value of $\cos\theta$.

Card 6: NQE, NQO (2I5)

NQE	is the number of quadrature points to be generated using the parameters from the even symmetry calculation.
NQO	is the number of quadrature points to be generated using the parameters from the odd symmetry calculation.

4.5. Test output

A test deck which runs both DVR3D and DIPJ0DVR has been prepared. This uses the H_2S potential energy and dipole surfaces of Senekowitch et al. [31]. The run mimics the benchmark calculations on H_2S performed by Carter et al. [32] and the vibrational band intensity calculations of Le Sueur et al. [15], although the size of the calculations have been reduced in the test data.

The LiCN (CN frozen) scattering coordinate surface of Essers et al. is supplied in subroutine POT.

Acknowledgements

We would like to thank Dr. Brian Sutcliffe for helpful discussions during the course of this work. JRH thanks SERC for a Fellowship. This work was supported by SERC grants GR/G34339 and GR/H41744.

References

[1] Z. Bačić and J.C. Light, Ann. Rev. Phys. Chem. 40 (1989) 469.
[2] D.O. Harris, G.O. Engerholm and W. Gwinn, J. Chem. Phys. 43 (1965) 1515.
[3] Z. Bačić and J.C. Light, J. Chem. Phys. 85 (1986) 4594.
[4] J.C. Light, I.P. Hamilton and J.V. Lill, J. Chem. Phys. 92 (1985) 1400.
[5] J. Tennyson and J.R. Henderson, J. Chem. Phys. 91 (1989) 3815.
[6] J.R. Henderson and J. Tennyson, Mol. Phys. 69 (1990) 639.
[7] J.R. Henderson, S. Miller and J. Tennyson, J. Chem. Soc. Faraday Trans. 86 (1990) 1963.
[8] J.R. Henderson, H.A. Lam and J. Tennyson, J. Chem. Soc. Faraday Trans. 88 (1992) 3187.
[9] J.A. Fernley, S. Miller and J. Tennyson, J. Mol. Spectrosc. 150 (1991) 597.
[10] R.M. Whitnell and J.C. Light, J. Chem. Phys. 90 (1989) 1774.
[11] J.R. Henderson and J. Tennyson, Comput. Phys. Commun. 75 (1993) 365, this issue.
[12] J.R. Henderson, PhD Thesis, University of London (1990).
[13] J.R. Henderson, C.R. Le Sueur, S.G. Pavett and J. Tennyson, Comput. Phys. Commun. 74 (1993) 193.
[14] J.R. Henderson, J. Tennyson and B.T. Sutcliffe, J. Chem. Phys., in press.
[15] C.R. Le Sueur, S. Miller, J. Tennyson and B.T. Sutcliffe, Mol. Phys. 76 (1992) 1147.
[16] B.T. Sutcliffe and J. Tennyson, Int. J. Quantum Chem. 29 (1991) 183.
[17] J. Tennyson, Comput. Phys. Rep. 4 (1986) 1.
[18] J. Tennyson and B.T. Sutcliffe, J. Chem. Phys. 77 (1982) 4061.
[19] J. Tennyson, S. Miller and C.R. Le Sueur, Comput. Phys. Commun. 75 (1993) 339, this issue.
[20] I.S. Gradshteyn and I.H. Ryzhik, Tables of Integrals, Series and Products (Academic, New York, 1980).
[21] J. Tennyson and B.T. Sutcliffe, J. Mol. Spectrosc. 101 (1983) 71.
[22] A.H. Stroud and D. Secrest, Gaussian Quadrature Formulas (Prentice-Hall, London, 1966).
[23] J.C. Light, R.M. Whitnell, T.J. Pack and S.E. Choi, in: Supercomputer Algorithms for Reactivity, Dynamics and Kinetics of Small Molecules, ed. A. Laganà, NATO ASI series C, Vol. 277 (Kluwer, Dordrecht, 1989) p. 187.
[24] D. Sadavoski, N.G. Fulton, J.R. Henderson, J. Tennyson and B.I. Zhilinski, J. Chem. Phys., submitted.
[25] A.S. Dickinson and P.R. Certain, J. Chem. Phys. 49 (1968) 4204.
[26] N.G. Fulton, J.R. Henderson and J. Tennyson, work in progress.
[27] C.R. Le Sueur, J.R. Henderson and J. Tennyson, Chem. Phys. Lett., submitted.
[28] Subroutine EGVQR, originally due to T.J. Dekker, Amsterdam (1968).
[29] NAG Fortran Library Manual, Mark 14 (1990).
[30] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes (Cambridge Univ. Press, Cambridge, 1986).
[31] J. Senekowitsch, S. Carter, A. Zilch, H.-J. Werner, N.C. Handy and P. Rosmus, J. Chem. Phys. 90 (1989) 783.
[32] S. Carter, P. Rosmus, N.C. Handy, S. Miller, J. Tennyson and B.T. Sutcliffe, Comput. Phys. Commun. 55 (1989) 71.
[33] R. Essers, J. Tennyson and P.E.S. Wormer, Chem. Phys. Lett. 89 (1982) 223.

TEST RUN OUTPUT

```
         PROGRAM DVR3D (VERSION OF 15 Sept 1992)

         FULL TRIATOMIC VIBRATIONAL PROBLEM WITH

            18   RADIAL R1 DVR POINTS USED,
            18   RADIAL R2 DVR POINTS USED,
            40   ANGULAR DVR POINTS USED, WITH
            10   LOWEST EIGENVECTORS CHOSEN FROM
    UP TO  600   DIMENSION SECULAR PROBLEM

         TITLE:  H2S J=0 scattering coordinates 3D DVR

         PROBLEM SOLVED IN THE ORDER: THETA -> R1 -> R2

          600 EIGENVALUES SELECTED FROM     0.1106398948D-01 TO    0.1246613142D+00

    LOWEST  10 EIGENVALUES IN CM-1:

     0.329745814930D+04     0.448789782008D+04     0.566947448478D+04     0.591781492896D+04     0.684093215232D+04

     0.709201630442D+04     0.800114410352D+04     0.825755962359D+04     0.845166121552D+04     0.854865836085D+04

          600 EIGENVALUES SELECTED FROM     0.1106398948D-01 TO    0.1246613142D+00

          0 MINS  0.00 SECS CPU TIME USED


          0 MINS  0.00 SECS CPU TIME USED


    LOWEST  10 EIGENVALUES IN HARTREES:

     0.150243253147D-01     0.204483677351D-01     0.258320273272D-01     0.269635496857D-01     0.311695813741D-01

     0.323136049862D-01     0.364558961656D-01     0.376242112782D-01     0.385086032339D-01     0.389505547614D-01


    LOWEST  10 EIGENVALUES IN CM-1:

     0.329745814930D+04     0.448789782008D+04     0.566947448478D+04     0.591781492896D+04     0.684093215232D+04

     0.709201630442D+04     0.800114410352D+04     0.825755962359D+04     0.845166121552D+04     0.854865836085D+04

    Program DIPJ0DVR - version 1.1 (September 1992)

 test data for H2S even eigenstates only

The equilibrium values of r1, r2, and cos(theta)
correspond to the following atom coordinates:

                                          x                   z
Atom 1 (mass    31.972071 amu):          0.000000           0.105563
Atom 2 (mass     1.007825 amu):          1.870000          -1.674437
Atom 3 (mass     1.007825 amu):         -1.870000          -1.674437

     ************************************************

    PRINT OUT OF DIPOLE TRANSITION MOMENTS AND S(F-I)

        FREQUENCIES IN WAVENUMBERS
        TRANSITION MOMENTS IN DEBYE (2.54174A.U.)
        S(F-I) IN DEBYE**2
        EINSTEIN A-COEFFICIENT IN SEC-1

    ************************************************

IE1   IE2     KET ENERGY   BRA ENERGY    FREQUENCY  Z TRANSITION    X TRANSITION       DIPOLE        S(F-I)       A-COEFFICIENT

  1     1          0.000        0.000        0.000  -0.982740E+00   0.000000E+00   0.982740E+00   0.965779E+00   0.000000E+00
  1     2          0.000     1190.440     1190.440  -0.147225E-01   0.000000E+00   0.147225E-01   0.216751E-03   0.114679E+00
  1     3          0.000     2372.016     2372.016  -0.367882E-02   0.000000E+00   0.367882E-02   0.135337E-04   0.566464E-01
  1     4          0.000     2620.357     2620.357  -0.721566E-02   0.000000E+00   0.721566E-02   0.520658E-04   0.293789E+00
  1     5          0.000     3543.474     3543.474  -0.106987E-02   0.000000E+00   0.106987E-02   0.114463E-05   0.159718E-01
  1     6          0.000     3794.558     3794.558  -0.676743E-02   0.000000E+00   0.676743E-02   0.457980E-04   0.784752E+00
  1     7          0.000     4703.686     4703.686  -0.225805E-03   0.000000E+00   0.225805E-03   0.509878E-07   0.166411E-02
  1     8          0.000     4960.101     4960.101   0.354321E-03   0.000000E+00   0.354321E-03   0.125543E-06   0.480470E-02
  1     9          0.000     5154.203     5154.203  -0.257982E-02   0.000000E+00   0.257982E-02   0.665546E-05   0.285802E+00
  1    10          0.000     5251.200     5251.200  -0.862820E-03   0.000000E+00   0.862820E-03   0.744459E-06   0.338079E-01
```