



ELSEVIER

Available online at www.sciencedirect.com



Computer Physics Communications 163 (2004) 85–116

Computer Physics
Communications

www.elsevier.com/locate/cpc

DVR3D: a program suite for the calculation of rotation–vibration spectra of triatomic molecules [☆]

Jonathan Tennyson ^{a,*}, Maxim A. Kostin ^{a,b}, Paolo Barletta ^a, Gregory J. Harris ^a,
Oleg L. Polyansky ^{a,b}, Jayesh Ramanlal ^a, Nikolai F. Zobov ^{a,b}

^a Department of Physics and Astronomy, University College London, London, WC1E 6BT, UK

^b Permanent address: Institute of Applied Physics, Russian Academy of Science, Uljanov Street 46, Nizhni Novgorod, Russia 603950

Accepted 28 October 2003

Available online 21 September 2004

Abstract

The DVR3D program suite calculates energy levels, wavefunctions, and where appropriate dipole transition moments, for rotating and vibrating triatomic molecules. Potential energy and, where necessary, dipole surfaces must be provided. Expectation values of geometrically defined functions can be calculated, a feature which is particularly useful for fitting potential energy surfaces. The programs use an exact (within the Born–Oppenheimer approximation) Hamiltonian and offer a choice of Jacobi or Radau internal coordinates and several body-fixed axes. Rotationally excited states are treated using an efficient two-step algorithm. The programs use a Discrete Variable Representation (DVR) based on Gauss–Jacobi and Gauss–Laguerre quadrature for all 3 internal coordinates and thus yields a fully point-wise representation of the wavefunctions. The vibrational step uses successive diagonalisation and truncation which is implemented for a number of possible coordinate orderings. The rotational, expectation value and transition dipole programs exploit the savings offered by performing integrals on a DVR grid. The new version has been rewritten in FORTRAN 90 to exploit the dynamic array allocations and the algorithm for dipole and spectra calculations have been substantially improved. New modules allow the *z*-axis to be embedded perpendicular to the plane of the molecule and for the calculation of expectation values.

Program summary

Title of the program: DVR3D suite

Catalogue number: ADTI

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTI>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Programming language: Fortran 90

[☆] This paper and its associated computer programs are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author. Fax: +(44) 20 7679 7145; telephone: +(44) 20 7679 7809.

E-mail address: j.tennyson@ucl.ac.uk (J. Tennyson).

No. of lines in distributed program, including test data, etc.: 61 574

No. of bytes in distributed program, including test data, etc.: 972 404

Distribution format: tar.gz

New version summary

Title of program: DVR3DRJZ

Catalogue number: ADTB

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTB>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Reference in CPC to previous version: 86 (1995) 175

Catalogue identifier of previous version: ADAK

Authors of previous version: J. Tennyson, J.R. Henderson and N.G. Fulton

Does the new version supersede the original program?: DVR3DRJZ supersedes DVR3DRJ

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix; SGI Origin 2000, Sunfire V750 and V880 systems running SunOS, IBM p690 Regatta running AIX

Programming language used in the new version: Fortran 90

Memory required to execute: case dependent

No. of lines in distributed program, including test data, etc.: 4203

No. of bytes in distributed program, including test data, etc.: 30 087

Has code been vectorised or parallelised?: The code has been extensively vectorised. A parallel version of the code, PDVR3D has been developed [1], contact the first author for details

Additional keywords: perpendicular embedding

Distribution format: gz

Nature of physical problem: DVR3DRJZ calculates the bound vibrational or Coriolis decoupled rotational–vibrational states of a triatomic system in body-fixed Jacobi (scattering) or Radau coordinates [2]

Method of solution: All coordinates 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. DVR3DRJ provides the vibrational wavefunctions necessary for ROTLEV3, ROLEV3B or ROTLEV3Z to calculate rotationally excited states, DIPOLE3 to calculate rotational–vibrational transition strengths and XPECT3 to compute expectation values

Restrictions on the complexity of the problem: (1) The size of the final Hamiltonian matrix that can practically be diagonalised. (2) The order of integration in the radial coordinates that can be dealt with 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 usually dominated by the final (3D) matrix diagonalisation. The test runs take minutes on a fast PC

Unusual features of the program: A user supplied subroutine containing the potential energy as an analytic function is a program requirement

References:

[1] H.Y. Mussa, J. Tennyson, *Comput. Phys. Commun.* 128 (2000) 434.

[2] J. Tennyson, B.T. Sutcliffe, *Internat. J. Quantum Chem.* 42 (1992) 941.

New version summary

Title of program: ROTLEV3

Catalogue number: ADTC

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTC>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Reference in CPC to previous version: 86 (1995) 175

Catalogue identifier of previous version: ADAL

Authors of previous version: J. Tennyson, J.R. Henderson and N.G. Fulton

Does the new version supersede the original program?: Yes

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix; SGI Origin 2000, Sunfire V750 and V880 systems running SunOS

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 1514

No. of bytes in distributed program, including test data, etc.: 12 652

Has code been vectorised or parallelised?: The code has been extensively vectorised. A parallel version of the code, PROTLEV3 has been developed [1], contact the first author for details

Distribution format: gz

Nature of physical problem: ROTLEV3 performs the second step in a two-step variational calculation for the bound rotational–vibrational levels of a triatomic system represented in either Jacobi or unsymmetrised Radau coordinates

Method of solution: A basis is constructed from the solutions of the Coriolis decoupled problem provided by DVR3DRJZ. The angular coordinate is transformed back to a basis set representation. The sparse Hamiltonian matrix can be diagonalised iteratively or in core

Restrictions on the complexity of the problem: The size of matrix that can practically be diagonalised

Typical running time: Case dependent. The sample data takes less than a minute on a fast PC

Unusual features of the program: Most data is read directly from DVR3DRJZ. ROTLEV3 can provide data to drive DIPOLE3 and/or XPECT3

References:

[1] H.Y. Mussa, J. Tennyson, *Comput. Phys. Commun.* 128 (2000) 434.

New version summary

Title of program: ROTLEV3B

Catalogue number: ADTD

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTD>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Reference in CPC to previous version: 86 (1995) 175

Catalogue identifier of previous version: ADAM

Authors of previous version: J. Tennyson, J.R. Henderson and N.G. Fulton

Does the new version supersede the original program?: Yes

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix, Sunfire V750 and V880 systems running SunOS

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 2215

No. of bytes in distributed program, including test data, etc.: 16 595

Has code been vectorised or parallelised?: The code has been extensively vectorised. A parallel version of the code, PROTLEV3B has been developed [1], contact the first author for details

Distribution format: gz

Nature of physical problem: ROTLEV3B performs the second step in a two-step variational calculation for the bound rotational–vibrational levels of a triatomic system represented by symmetrised Radau coordinates using a bisector embedding [2]

Method of solution: A basis is constructed from the solutions of the Coriolis decoupled problem provided by DVR3DRJZ. The problem is constructed entirely within the DVR. The Hamiltonian matrix can be diagonalised iteratively or in core

Restrictions on the complexity of the problem: The size of matrix that can practically be diagonalised

Typical running time: Case dependent. The sample data takes a few minutes on a fast PC

Unusual features of the program: Most data is read directly from DVR3DRJZ. ROTLEV3B can provide data to drive DIPOLE3 and/or XPECT3

References:

- [1] H.Y. Mussa, J. Tennyson, *Comput. Phys. Commun.* 128 (2000) 434.
- [2] J. Tennyson, B.T. Sutcliffe, *Internat. J. Quantum Chem.* 42 (1992) 941.

Program summary

Title of program: ROTLEV3Z

Catalogue number: ADTE

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTE>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix, Sunfire V750 and V880 systems running SunOS

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 2919

No. of bytes in distributed program, including test data, etc.: 17 241

Keywords: rotationally excited state, Coriolis coupling, secondary variational method, sparse matrix, vectorised, perpendicular embedding, Radau coordinates

Has code been vectorised or parallelised?: The code has been extensively vectorised

Distribution format: gz

Nature of physical problem: ROTLEV3Z performs the second step in a two-step variational calculation for the bound rotational–vibrational levels of a triatomic system represented by symmetrised Radau coordinates using a perpendicular embedding [1]

Method of solution: A basis is constructed from the solutions of the Coriolis decoupled problem provided by DVR3DRJZ. The problem is constructed entirely within the DVR. The Hamiltonian matrix is diagonalised in core

Restrictions on the complexity of the problem: The size of matrix that can practically be diagonalised

Typical running time: Case dependent. The sample data takes a few minutes on a fast PC

Unusual features of the program: Most data is read directly from DVR3DRJZ

References:

- [1] M.A. Kostin, O.L. Polyansky, J. Tennyson, *J. Chem. Phys.* 116 (2002) 7564.

New version summary

Title of program: DIPOLE3

Catalogue number: ADTF

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTF>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Reference in CPC to previous version: 86 (1995) 175

Catalogue identifier of previous version: ADAN

Authors of previous version: J. Tennyson, J.R. Henderson and N.G. Fulton

Does the new version supersede the original program?: Yes

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix; SGI Origin 2000; sunfire V750 and V880 systems

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 1921

No. of bytes in distributed program, including test data, etc.: 15 685

Has code been vectorised or parallelised?: The code has been extensively vectorised. Commands to parallelise the code using OpenMP are included in the source

Distribution format: gz

Nature of physical problem: DIPOLE3 calculates dipole transition intensities between previously calculated wavefunction for both rotational and rotational–vibrational transitions

Method of solution: Integrals over dipole surfaces are constructed using a DVR in all three coordinates, this requires a transformation of the angular wavefunctions. Wavefunctions generated by DVR3DRJZ and ROTLEV3 or ROTLEV3B are then used to give transition intensities for individual pairs of states

Restrictions on the complexity of the problem: The complexity of the problem that can be solved by DVR3DRJZ, ROTLEV3 or ROTLEV3B

Typical running time: Case dependent. The test data takes a few seconds on a fast PC

Unusual features of the program: Most data is read directly from DVR3DRJZ and ROTLEV3 or ROTLEV3B. DIPOLE provides data to drive SPECTRA

New version summary

Title of program: SPECTRA

Catalogue number: ADTG

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTG>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix

Reference in CPC to previous version: 75 (1993) 339

Catalogue identifier of previous version: ACNB

Authors of previous version: J. Tennyson, S. Miller and C.R. Le Sueur

Does the new version supersede the original program?: Yes

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 1037

No. of bytes in distributed program, including test data, etc.: 9159

Has code been vectorised or parallelised?: As execution times are very short this is usually not important

Distribution format: gz

Nature of physical problem: SPECTRA generates synthetic, frequency ordered spectra as a function of temperature. Absolute intensities can be calculated if the necessary data to calculate the partition function is supplied

Method of solution: Transitions are sorted by frequency and weighted using Boltzmann statistics

Restrictions on the complexity of the problem: The complexity of problem that can be solved by other programs in the suite

Typical running time: Case dependent, but very small for sample data

Unusual features of the program: Most data is read directly from DIPOLE3. Some data from DVR3DRJZ and ROTLEV3 or ROTLEV3B may also be required

Program summary

Title of program: XPECT3

Catalogue number: ADTH

Program summary URL: <http://cpc.cs.qub.ac.uk/summaries/ADTH>

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Computer: PC running Linux

Installation: desktop

Other machines on which program tested: Compaq running True64 Unix

Programming language used: Fortran 90

High speed storage required: case dependent

No. of lines in distributed program, including test data, etc.: 1214

No. of bytes in distributed program, including test data, etc.: 9361

Distribution format: gz

Keywords: Expectation values, Hellmann–Feynman theorem, potential fitting

Has code been vectorised or parallelised?: The code has been extensively vectorised

Nature of physical problem: XPECT3 calculates expectation of geometrically defined operators using previously calculated wavefunctions

Method of solution: Integrals over the user defined surfaces are constructed using a DVR in all three coordinates. Wavefunctions generated by DVR3DRJZ and ROTLEV3 or ROTLEV3B are then used to give expectation values for each state

Restrictions on the complexity of the problem: The complexity of problem that can be solved by DVR3DRJZ, ROTLEV3 or ROTLEV3B

Typical running time: Case dependent. The test data takes a few seconds on a fast PC

Unusual features of the program: Most data is read directly from DVR3DRJZ and ROTLEV3 or ROTLEV3B.

© 2004 Elsevier B.V. All rights reserved.

PACS: 33.20; 39.30

Keywords: Molecular spectra; Infrared; Microwave; Variational principle; Expectation values; Born–Oppenheimer approximation; Triatomic molecules

1. Introduction

The calculation of rotation–vibration spectra for triatomic species has become almost routine for molecules with a single low-lying potential energy surface. This has led to increasing activity looking at highly excited states of such molecules and using the observed spectroscopic data to determine highly accurate potential energy surfaces by successive refinement of the surface [1]. Indeed high accuracy nuclear motion calculations have begun to make a significant impact on key spectroscopic problems [2].

A number of methods are available which will compute vibration–rotation spectra of triatomics using Hamiltonians which are exact within the Born–Oppenheimer approximation and basis function expansions to represent the nuclear wavefunctions [3–7]. These variational procedures have proved very successful, particularly for problems where highly excited states are not required.

An alternative procedure, based on finite element representations of the nuclear wavefunctions, has been developed by Light and co-workers [8,9]. This approach, generally known as the discrete variable representation or DVR, was based on earlier work by Harris et al. [10]. The DVR is not strictly variational, but has a number of advantages over the more traditional basis set methods in both the development of compact representations for multi-dimensional Hamiltonians and the simplicity of most of the matrix elements. This latter property has been increasingly exploited in a number of methods using iterative diagonalisation, see [11] and references therein. Such methods have proved very powerful for studying very high-lying states, for example, near and just above dissociation, but have been little used to compute actual spectra.

Advances in computer technology have meant that variational or DVR methods can be routinely used to compute spectra of triatomics on desktop computers. At the same time these methods are being increasingly used to compute large datasets which are much too big to be obtained experimentally. Large variational calculations on triatomics have been used to obtain data to give temperature dependent thermodynamic functions [12–15] and radiative transport models [16–21].

In 1995 Tennyson and co-workers published a triatomic rotational–vibrational spectral package based on the use of a discrete variable representation for all three internal coordinates, called DVR3D [22]. The present work updates and extends this program. The whole package has been re-written into FORTRAN 90 to, in particu-

lar, take advantage of the dynamic array allocation procedures offered. Key portions of the programs utilise BLAS routines for optimal efficiency. A new module has been added, ROTLEV3Z, which allows the body-fixed z -axis of the system to be placed perpendicular to the plane defined by the molecule [23]. The main driving module, renamed from DVR3DRJ to DVR3DRJZ, has been extended to cope with this case. Additional options in DRV3DRJZ and the various ROTLEV programs allow for the inclusion of non-adiabatic effects which thus go beyond the Born–Oppenheimer separation of electronic and nuclear motion [24,25]. The transition moments routine DIPOLE3 has been reprogrammed with a new algorithm, described below, which is both substantially faster for large runs and also significantly reduces input/output. The reduction in i/o means that the interface between ROTLEV3/ROTELEV3B and DIPOLE3 has also been re-written. Module SPECTRA, previously published as part of the finite basis set package TRIATOM [26], has been re-written to be more efficient for large problems and to increase its functionality. Finally a new module, XPECT3, has been introduced which calculates expectation values for a given geometric operator. This procedure has a number of uses but we have found it particularly important for fitting potential energy surfaces using spectroscopic data [22,27].

2. Method

2.1. The vibrational problem: DVR3DRJZ

2.1.1. The 3D DVR Hamiltonian matrix and its solution

We use a multidimensional DVR in scattering (Jacobi) or Radau coordinates. In scattering coordinates r_1 represents the ‘diatom’ distance between atom 2 and atom 3, and r_2 the separation of the atom 1 from the diatom centre of mass. The angle between \underline{r}_1 and \underline{r}_2 is θ . A formal definition of (r_1, r_2, θ) in Radau coordinates is given below.

Using a finite basis representation (FBR), the zero rotational angular momentum ($J = 0$) Hamiltonian matrix can be written [3]

$$\begin{aligned} \langle m, n, j | \hat{H}^{J=0} | m', n', j' \rangle = & \langle m | \hat{h}^{(1)} | m' \rangle \delta_{n,n'} \delta_{j,j'} + \langle n | \hat{h}^{(2)} | n' \rangle \delta_{m,m'} \delta_{j,j'} \\ & + (\langle m | \hat{g}^{(1)} | m' \rangle \delta_{n,n'} + \langle n | \hat{g}^{(2)} | n' \rangle \delta_{m,m'}) j(j+1) \delta_{j,j'} \\ & + \langle m, n, j | V(r_1, r_2, \theta) | m', n', j' \rangle. \end{aligned} \quad (1)$$

A DVR is a unitary transformation of an FBR defined for some quadrature scheme associated with the FBR polynomials. The angular basis functions $|j\rangle$, when $J = 0$, are Legendre polynomials. The radial basis functions are Laguerre polynomials. These are either Morse oscillator-like functions or spherical oscillators. The Morse oscillator-like functions are defined as [28]:

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

where

$$A = \frac{4D_e}{\omega_e}, \quad \beta = \omega_e \left(\frac{\mu}{2D_e} \right)^{1/2}, \quad \alpha = \text{integer}(A). \quad (3)$$

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 Laguerre polynomial [29].

When optimizing the parameters for the Morse oscillator-like functions it usual to take initial guesses based on the shape of the potential for the initial coordinate. Experience shows

- (a) that the results are largely insensitive to the choice of D_e ,
- (b) the optimal value of r_e is usually larger than the value given by the minimum of the potential,
- (c) values for r_e and w_e are often quite strongly coupled and need to be varied together.

In particular if more states are required, increasing r_e and reducing w_e extends the range of the basis. When optimising parameter one should concentrate on a few states at the top end of the required energy range as these are the most sensitive to the parameters.

The spherical oscillator functions are particularly useful for systems which have significant amplitude for $r_2 = 0$. These functions are defined by [30]:

$$|n\rangle = H_n(r) = 2^{1/2}\beta^{1/4}N_{n\alpha+1/2}\exp\left(-\frac{y}{2}\right)y^{(\alpha+1)/2}L_n^{\alpha+1/2}(y), \quad y = \beta r^2, \quad (4)$$

where

$$\beta = (\mu\omega_e)^{1/2} \quad (5)$$

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

It can be advantageous 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. This is because optimisation is usually performed on cut-down problems for which a DVR can become unreliable because of the linkage between accuracy of integration and size of basis [1]. To this end optimisation is generally performed using the FBR³ code TRIATOM [26] or the two-dimensional (ZTWOD = T) option in the DVR¹–FBR² code DVR1D [31].

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

$$\langle t|\hat{h}^{(i)}|t'\rangle = \langle t|\frac{-\hbar^2}{2\mu_i}\frac{\partial^2}{\partial r_i^2}|t'\rangle, \quad (6)$$

$$\langle t|\hat{g}^{(i)}|t'\rangle = \langle t|\frac{\hbar^2}{2\mu_i r_i^2}|t'\rangle, \quad (7)$$

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 [32]:

$$\begin{aligned} \mu_1^{-1} &= g_2^2 m_1^{-1} + m_2^{-1} + (1 - g_2)^2 m_3^{-1}, \\ \mu_2^{-1} &= m_1^{-1} + g_1^2 m_2^{-1} + (1 - g_1)^2 m_3^{-1}, \end{aligned} \quad (8)$$

where for scattering coordinates

$$g_1 = \frac{m_2}{m_2 + m_3}, \quad g_2 = 0 \quad (9)$$

and for Radau coordinates [32]:

$$\begin{aligned} g_1 &= 1 - \frac{A}{A + B - AB}, & g_2 &= 1 - \frac{A}{1 - B + AB}, \\ A &= \left(\frac{m_3}{m_1 + m_2 + m_3}\right)^{1/2}, & B &= \frac{m_2}{m_1 + m_2}. \end{aligned} \quad (10)$$

The relationship between the g 's and geometrically defined coordinates is given by Fig. 1.

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 [8]:

$$T_i^\eta = (w_\eta)^{1/2}|t(\eta)\rangle, \quad (11)$$

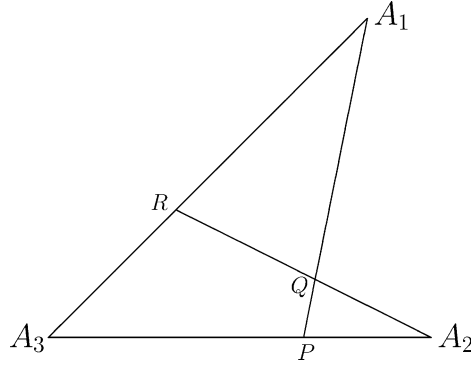


Fig. 1. Internal coordinate system of Sutcliffe and Tennyson [32]: A_i represents atom i . The coordinates are given by $r_1 = A_2 - R$, $r_2 = A_1 - P$ and the angle $\theta = \angle A_1 Q A_2$. The geometric parameters are defined by $g_1 = \frac{A_3 - P}{A_3 - A_2}$; $g_2 = \frac{A_3 - R}{A_3 - A_1}$.

where $|t\rangle = |m\rangle, |n\rangle, |j\rangle$ for $\eta = \alpha, \beta, \gamma$ respectively. DVR3DRJZ automatically generates the appropriate Gaussian quadrature schemes [33].

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

$$\underline{T} = T_{m,n,j}^{\alpha,\beta,\gamma} = T_m^\alpha T_n^\beta T_j^\gamma. \tag{12}$$

A three-dimensional DVR is obtained by applying the transformation $\underline{T}^T \underline{H} \underline{T}$. For $J = 0$, the transformed Hamiltonian can be written at the DVR grid points as

$$\begin{aligned} (3D) H_{\alpha,\alpha',\beta,\beta',\gamma,\gamma'} &= K_{\alpha,\alpha'}^{(1)} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'} + K_{\beta,\beta'}^{(2)} \delta_{\alpha,\alpha'} \delta_{\gamma,\gamma'} + L_{\alpha,\alpha',\gamma,\gamma'}^{(1)} \delta_{\beta,\beta'} \\ &+ L_{\beta,\beta',\gamma,\gamma'}^{(2)} \delta_{\alpha,\alpha'} + V(r_{1\alpha}, r_{2\beta}, \theta_\gamma) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'}. \end{aligned} \tag{13}$$

In (13), the potential energy operator is diagonal because of the quadrature approximation [8,34]

$$\sum_{m,n,j} \sum_{m',n',j'} T_{m,n,j}^{\alpha,\beta,\gamma} \langle m, n, j | V(r_1, r_2, \theta) | m', n', j' \rangle T_{m',n',j'}^{\alpha',\beta',\gamma'} \simeq V(r_{1\alpha}, r_{2\beta}, \theta_\gamma) \delta_{\alpha,\alpha'} \delta_{\beta,\beta'} \delta_{\gamma,\gamma'} \tag{14}$$

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

The kinetic energy terms in Hamiltonian (13) are represented by

$$K_{\eta,\eta'}^{(i)} = \sum_{t,t'} T_t^\eta \langle t | \hat{h}^{(i)} | t' \rangle T_{t'}^{\eta'} \tag{15}$$

$$L_{\eta,\eta',\gamma,\gamma'}^{(i)} = J_{\gamma\gamma'} \sum_{t,t'} T_t^\eta \langle t | \hat{g}^{(i)} | t' \rangle T_{t'}^{\eta'} \tag{16}$$

$$\simeq \frac{J_{\gamma\gamma'} \hbar^2}{2\mu_i r_{i\eta}^2} \delta_{\eta\eta'}, \tag{17}$$

again applying the quadrature approximation, and where

$$J_{\gamma\gamma'} = \sum_j T_j^\gamma j(j+1) T_j^{\gamma'}. \tag{18}$$

The calculation is generally set up as a series of diagonalisations and truncations [35,36]. 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

$$(1D) H_{\gamma,\gamma'}^{\alpha,\beta} = L_{\alpha,\alpha,\gamma,\gamma'}^{(1)} + L_{\beta,\beta,\gamma,\gamma'}^{(2)} + V(r_{1\alpha}, r_{2\beta}, \theta_\gamma) \delta_{\gamma\gamma'}. \quad (19)$$

Amplitudes for the h th level, with eigenenergy $\varepsilon_h^{\alpha,\beta}$, are given at each grid point, α, β , by $C_{\gamma,h}^{\alpha,\beta}$.

The solutions with $\varepsilon_h^{\alpha,\beta} \leq E_{\max}^{1D}$ are then selected and used to solve 2D problems for each value of β . This gives

$$(2D) H_{\alpha,\alpha',h,h'}^\beta = \varepsilon_h^{\alpha,\beta} \delta_{\alpha\alpha'} \delta_{hh'} + \sum_{\gamma} C_{\gamma,h}^{\alpha,\beta} C_{\gamma,h'}^{\alpha',\beta} K_{\alpha\alpha'}^{(1)}. \quad (20)$$

Solutions for the l th level, with eigenenergy ε_l^β are given by $C_{\alpha,l,h}^\beta$.

The solutions with $\varepsilon_l^\beta \leq E_{\max}^{2D}$ are then used to solve the full 3D problem of dimension N :

$$(3D) H_{\beta,\beta',l,l'} = \varepsilon_l^\beta \delta_{\beta\beta'} \delta_{ll'} + \sum_{\alpha,h,h'} C_{\alpha,l,h}^\beta C_{\alpha,l',h'}^{\beta'} \sum_{\gamma} C_{\gamma,h}^{\alpha,\beta} C_{\gamma,h'}^{\alpha,\beta'} K_{\beta\beta'}^{(2)}. \quad (21)$$

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

2.1.2. Order of solution

As stated the 3D DVR Hamiltonian matrix of Eq. (13) can be solved by successive diagonalisation and truncation. In Section 2.1.1 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. The most efficient ordering is to treat the coordinate accommodating the highest density of states last [36,37]. In principle the coordinate holding the lowest density of states should come first but this has been found to matter little in practice.

Four of the possible coordinate orders have been implemented in DVR3DRJZ. 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.

In symmetrised Radau coordinates, see below, the coordinates r_1 and r_2 are mixed and it no longer makes sense to diagonalise and truncate in these coordinates separately. Furthermore, in all the problems we have tackled with Radau coordinates, diagonalisation of θ last has been the method of choice. For this reason we have only implemented this option in DVR3DRJZ for symmetrised Radau coordinates.

2.1.3. Symmetry

Scattering coordinates can exploit permutation symmetry of an AB₂ system. This symmetry is carried by the \underline{J} matrix of Eq. (18). Then the symmetrised \underline{J} matrix becomes

$$J_{\gamma\gamma'q} = 2 \sum_{j=0}^{N/2-1} T_{2j+q}^{\gamma'} (2j+q)(2j+q+1) T_{2j+q}^{\gamma}, \quad q = 0, 1. \quad (22)$$

It should be noted that for problems including rotational excitation, this symmetry is preserved with the body-fixed z -axis placed either along either r_2 (ZEMBED = .TRUE. in the code) or r_1 (ZEMBED = .FALSE.).

In Radau coordinates, the permutation symmetry of an A₂B molecule is carried by the radial coordinates. This is more complicated to treat in a DVR since it involves coupling two coordinates. Symmetrising the DVR by taking suitable combinations of points in r_1 and r_2 has been shown to have distinct advantages over other methods of including symmetry [38]. With this procedure the new functions become:

$$|\alpha, \beta, q\rangle = [2(1 + \delta_{\alpha,\beta})]^{-1/2} (|\alpha, \beta\rangle + (-1)^{q+k} |\beta, \alpha\rangle), \quad q = 0, 1, \quad (23)$$

where $\alpha > \beta$ for $q + k$ odd or $\alpha \geq \beta$ for $q + k$ even. Note that this definition [39] of q , which is not the one used in most of our previous works, identifies the ortho and para symmetry blocks for AB₂ molecules each by a single q value.

Symmetry with $J > 0$ and the z -perpendicular embedding is quite subtle. We refer to Kostin et al. [23] for a full discussion of this.

2.1.4. Rotational excitation within DVR3DRJZ

Besides solving the pure vibrational ($J = 0$) problem, DVR3DRJZ also solves the first step in the two-step [40] solution to the full rotational–vibrational problem. For $J > 0$, DVR3DRJZ assumes that the projection of J along the body-fixed z -axis, k , is a good quantum number. The choice of body-fixed axes is crucial to a rapid solution of problems with rotational excitation. There are four possible options included in the program:

- (1) r_1 embedding: z is parallel to \underline{r}_1 , with x in the plane of the molecule.
- (2) r_2 embedding: z is parallel to \underline{r}_2 , with x in the plane of the molecule.
- (3) Bisector embedding: x bisects the angle θ , with z in the plane of the molecule.
- (4) Perpendicular embedding: x bisects the angle θ , with z perpendicular to the plane of the molecule.

In each case the y -axis is defined to give a right-handed set. Options 1 and 2 are available for scattering coordinates and non-symmetrised Radau coordinates; options 3 and 4 are available only for symmetrised Radau coordinates.

For the standard \underline{r}_1 or \underline{r}_2 embeddings and assuming k to be a good quantum number, it is only necessary to add one extra term to Hamiltonian (1):

$$\langle m, n, j, J, k | \hat{H}^{J,k} | m', n', j', J, k \rangle = \langle m, n, j | \hat{H}^{J=0} | m', n', j' \rangle + \langle t | \hat{g}^{(i)} | t' \rangle \delta_{j,j'} \delta_{s,s'} (J(J+1) - 2k^2). \quad (24)$$

In (24), if the z -axis is taken along \underline{r}_1 then $|t\rangle = |m\rangle$, $s = n$ and $i = 1$; conversely if z is along \underline{r}_2 , $|t\rangle = |n\rangle$, $s = m$ and $i = 2$.

In the bisector embedding [41]:

$$\begin{aligned} \langle m, n, j, J, k | \hat{H}^{J,k} | m', n', j', J, k \rangle &= \langle m, n, j | \hat{H}^{J=0} | m', n', j' \rangle + (\langle m | \hat{g}^{(1)} | m' \rangle \delta_{n,n'} + \langle n | \hat{g}^{(2)} | n' \rangle \delta_{m,m'}) \\ &\times \left(\frac{1}{4} \langle jk | \frac{1}{(1 - \cos\theta)} | j'k \rangle (J(J+1) - 3k^2) + \frac{1}{8} \delta_{j,j'} (J(J+1) - k^2) \right). \end{aligned} \quad (25)$$

In both the standard and bisector embedding the angular basis functions for $J > 0$ calculations are associated Legendre polynomials of order k . For the perpendicular embedding conversely they are Jacobi polynomials $J_j^{a,b}$ defined by [42]

$$a = b = \left(\frac{1}{2} [J(J+1) - k^2] \right)^{1/2}. \quad (26)$$

With these functions, the perpendicular embedding gives [23]:

$$\begin{aligned} \langle m, n, j, J, k | \hat{H}^{J,k} | m', n', j', J, k \rangle &= \langle m, n, j | \hat{H}^{J=0} | m', n', j' \rangle \\ &- \left(\delta_{n,n'} \langle m | \hat{g}^{(1)} | m' \rangle + \delta_{m,m'} \langle n | \hat{g}^{(2)} | n' \rangle \left[(j+a)(j+a+1) - \frac{k^2}{4} \right] \right) \end{aligned} \quad (27)$$

where a is as defined in Eq. (26).

In both the bisector and perpendicular embeddings, $k = 1$ is a special case and has an extra term on the diagonal. For the bisector embedding:

$$+ \frac{(-1)^{1-p}}{16} (\delta_{n,n'} \langle m | \hat{g}^{(1)} | m' \rangle + \delta_{m,m'} \langle n | \hat{g}^{(2)} | n' \rangle) \langle jk | \frac{(1 + \cos\theta)}{(1 - \cos\theta)} | j'k \rangle J(J+1), \quad (28)$$

where p , the rotational parity, is defined below. Similarly for the perpendicular embedding the extra term is:

$$+ \frac{(-1)^{1-q}}{4} (\delta_{n,n'} \langle m | \hat{g}^{(1)} | m' \rangle + \delta_{m,m'} \langle n | \hat{g}^{(2)} | n' \rangle) \langle jk | \frac{\cos \theta}{(1 - \cos^2 \theta)} | j'k \rangle J(J+1), \quad (29)$$

where q is the symmetry label defined previously, (22).

The extra matrix elements present for the $J > 0$ case need to be transformed into a DVR. For the standard r_1 and r_2 embeddings this gives

$$(J(J+1) - k^2) M_{\alpha,\alpha',\beta,\beta'}^{(i)} \delta_{\gamma,\gamma'} \quad (30)$$

where the \underline{M} -matrix is given by:

$$M_{\alpha,\alpha'}^{(1)} = \sum_{m,m'} T_m^\alpha \langle m | \hat{g}^{(1)} | m' \rangle T_{m'}^{\alpha'} \simeq \delta_{\alpha\alpha'} \frac{\hbar^2}{2\mu_1 r_{1\alpha}^2} \quad (31)$$

if $i = 1$ and z embedded along r_1 , when it is diagonal in β , and

$$M_{\beta,\beta'}^{(2)} = \sum_{n,n'} T_n^\beta \langle n | \hat{g}^{(2)} | n' \rangle T_{n'}^{\beta'} \simeq \delta_{\beta\beta'} \frac{\hbar^2}{2\mu_2 r_{2\beta}^2} \quad (32)$$

if $i = 2$ and z embedded along r_2 , when it is diagonal in α .

In the bisector embedding the J -dependent term becomes in a DVR:

$$\frac{1}{8} (M_{\alpha,\alpha',\beta,\beta'}^{(1)} + M_{\alpha,\alpha',\beta,\beta'}^{(2)}) \delta_{\gamma,\gamma'} \left(\frac{1}{8} (J(J+1) - k^2) + \delta_{k,1} \frac{(-1)^{1-p}}{16} J(J+1) \frac{(1+\gamma)}{(1-\gamma)} \right) \quad (33)$$

plus an extra term for the $k = 1$ case. The extra term for the perpendicular embedding case is somewhat more complicated and can be found in Kostin et al. [23].

For the bisector and perpendicular embeddings, the extra angular integrals are evaluated using the quadrature approximation. It should be noted, however, that the operators which depend on $(1 - \cos \theta)^{-1}$ are singular when $\theta = 0$. These ABB linear geometries correspond to very high energy for many AB_2 molecules, for instance water. The present code assumes that the wavefunction has zero amplitude in this region; this can actually be enforced for rotationally excited states by using the switch ZLIN in DVR3DRJZ. Other angular integrals are evaluated in the FBR and then transformed to the DVR as discussed for the J -matrix, Eq. (18) above.

2.1.5. Wavefunctions

The eigenvectors of the 3D Hamiltonian are obtained as coefficients of the intermediate basis. To use these vectors it is usually necessary to express them in a more physically meaningful fashion. This can be achieved by transforming the vectors 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.

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

$$\Psi_{\alpha\beta\gamma}^i = \sum_{l,h} C_{\beta il} C_{\alpha lh}^\beta C_{\gamma h}^{\alpha\beta} = \sum_l C_{\beta il} \sum_h C_{\alpha lh}^\beta C_{\gamma h}^{\alpha\beta} \quad (34)$$

where, for definiteness, the $\theta \rightarrow r_1 \rightarrow r_2$ ordering has been assumed, as above.

Reverse transformations to FBR are also fairly straightforward to write down. In particular ROTLEV3 works with the angular functions represented as a basis set of associated Legendre polynomials. In this case the wavefunction is written

$$\Psi_{\alpha\beta}^{Ji} = \sum_{jk} d_{j\alpha\beta}^{Jki} | j \rangle. \quad (35)$$

The new coefficients, d^{Jki} , are obtained from the back transformation

$$d_{j\alpha\beta}^{Jki} = \sum_{\gamma} \Psi_{\alpha\beta\gamma}^i \omega_{\gamma}^{1/2} |j(\gamma)\rangle. \quad (36)$$

DIPOLE3 uses a similar transformation to place all wavefunctions on a single grid, see [Appendix A](#).

A utility, wfnread.f90, has been included in the distribution to allow users to read wavefunctions for further analysis such as plotting. The utility requires no user input and will read the wavefunction files produced by DVR3DRJZ, ROTLEV3 and ROTLEV3B. As written wfnread.f90 does no more than echo the contents of these wavefunction files and it will therefore need to be adapted before use. For further information users should read the source of the program.

2.1.6. 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 [43], which occurred when evaluating the r_2^{-2} integrals.

In scattering coordinates, it is possible for the r_2 coordinate to be equal, or very close, to zero if this linear geometry is energetically accessible or favorable. It is desirable to use the spherical oscillator functions in this case. Under these circumstances it was found [43] that the quadrature approximation had to be abandoned for the r_2^{-2} integral because of its non-polynomial behaviour as $r_2 \rightarrow 0$.

After extensive tests on the H_3^+ molecular ion [43], an alternative procedure was implemented which continues to use the quadrature approximation to construct $^{(1D)}\underline{H}$, and then constructs $^{(3D)}\underline{H}$ using the full-matrix transformation of the r_2^{-2} integrals, correcting for the fact that the quadrature approximation was used in $^{(1D)}\underline{H}$. So the only change in the formulation is that now

$$^{(3D)}\tilde{H}_{\beta,\beta',l,l'} \approx ^{(3D)}H_{\beta,\beta',l,l'} + \sum_{\alpha,k,k'} C_{\alpha,l,k}^{\beta} C_{\alpha,l',k'}^{\beta'} (\tilde{M}_{\beta,\beta'}^{(2)} - M_{\beta,\beta'}^{(2)}) \sum_{\gamma,\gamma'} C_{\gamma,k}^{\alpha,\beta} C_{\gamma',k'}^{\alpha,\beta'} J_{\gamma\gamma'} \quad (37)$$

where $^{(3D)}H_{\beta,\beta',l,l'}$ is defined by (21) and

$$\tilde{M}_{\beta,\beta'}^{(2)} = \sum_{n,n'} T_n^{\beta} \langle n | \hat{g}^{(2)} | n' \rangle T_{n'}^{\beta'}, \quad (38)$$

and $M_{\beta,\beta'}^{(2)}$ is given by the approximation (32).

Note that the FBR matrix elements above can be evaluated analytically, and are given by [28]

$$\langle n | \hat{g}^{(2)} | n' \rangle = \frac{\hbar^2 \beta}{(2\alpha + 1)\mu_2} \left(\frac{n! \Gamma(n' + \alpha + 3/2)}{n'! \Gamma(n + \alpha + 3/2)} \right)^{1/2}, \quad n \geq n'. \quad (39)$$

We have implemented a user input variable (ZQUAD2) in the program DVR3DRJZ so that either of Eqs. (21) or (37) can be used when working in Jacobi or unsymmetrised Radau coordinates.

2.2. Rotational motion with standard embeddings: ROTLEV3

The rotational problem can be fully expanded in terms of a finite set of functions, generally called rotation matrices [44]. These functions are symmetrised to reflect the rotational parity of the system, given by $(-1)^{J+p}$ with $p = 0, 1$; these functions will be denoted $|J, k, p\rangle$ below. To deal satisfactorily with the transition from linear to non-linear geometries it is usual to couple these rotation matrices with the k -dependent angular functions denoted

$|j\rangle$ above [32]. The FBR Hamiltonian matrix for the fully coupled vibration–rotation problem can be expressed as

$$\begin{aligned} \langle m, n, j, J, k, p | \widehat{H} | m', n', j', J, k', p \rangle &= \delta_{k,k'} \langle m, n, j | \widehat{H}^{J,k} | m', n', j' \rangle \\ &- (1 + \delta_{k,0} + \delta_{k',0})^{-1/2} \delta_{k',k\pm 1} \langle t | \hat{g}^{(i)} | t' \rangle \delta_{j,j'} \delta_{s,s'} C_{J,k'}^{\pm} C_{j,k'}^{\pm}, \quad k = p, p+1, \dots, J, \quad p = 0, 1, \end{aligned} \quad (40)$$

where if the body-fixed z -axis is taken parallel to \underline{r}_1 then $|t\rangle = |m\rangle$, $s = n$ and $i = 1$; and if z is along \underline{r}_2 , $|t\rangle = |n\rangle$, $s = m$ and $i = 2$. The angular factors are given by

$$C_{\ell,k}^{\pm} = (\ell(\ell+1) - k(k\pm 1))^{1/2}. \quad (41)$$

Using the solutions of the first step as a basis to expand the full problem means that the first term in (40) is simply $\varepsilon_h^{J,k}$ and the problem of constructing the Hamiltonian matrix reduces to the one of constructing the second, off-diagonal in k , term.

The quadrature approximation means that the matrix element $\langle t | \hat{g}^{(i)} | t' \rangle$ is diagonal in the DVR. Conversely the angular contribution is diagonal in the FBR. ROTLEV3 therefore transforms the DVR wavefunctions generated by DVR3DRJZ to an FBR in θ by

$$\psi_{\alpha,\beta,j}^{J,k,h} = \sum_{\gamma} T_j^{\gamma} \psi_{\alpha,\beta,\gamma}^{J,k,h}. \quad (42)$$

This means that only transformations of diagonal matrix element are required, see (43) below. It should be noted that the above transformation, unlike that of the matrix elements, is one-dimensional and therefore rapid.

In this DVR²–FBR¹ representation, the Hamiltonian matrix in terms of solutions of the first step is

$$\begin{aligned} \langle h, k, p | \widehat{H} | h', k', p \rangle &= \delta_{h,h'} \delta_{k,k'} \varepsilon_h^{J,k} - (1 + \delta_{k,0} + \delta_{k',0})^{-1/2} \delta_{k',k\pm 1} \sum_{\alpha,\beta,j} \psi_{\alpha,\beta,j}^{J,k,h} \psi_{\alpha,\beta,j}^{J,k',h'} C_{J,k'}^{\pm} C_{j,k'}^{\pm} M_{\alpha,\alpha,\beta,\beta}^{(i)}, \\ k &= p, p+1, \dots, J, \quad p = 0, 1. \end{aligned} \quad (43)$$

For a given J , DVR3DRJZ solves the $J+1$ unique Coriolis decoupled ‘vibrational’ problems. ROTLEV3 selects the IBASS lowest energy solutions of these calculations [45] as a basis for the full problem. The angular coordinate of these solutions is transformed back to an FBR and the Hamiltonian constructed. Because of its sparse nature, only the diagonal elements and $(k, k+1)$ off-diagonal blocks are computed and stored. The matrix can be diagonalised iteratively, a procedure which has proved very successful when a small fraction (less than 20%) of the eigenvalues are of interest; where many levels are required full matrix diagonalisation is more efficient. The $p=1$ matrix is simply a submatrix of the $p=0$ problem with the $k=0$ rows and columns deleted. This matrix thus need not be recalculated (option KMIN = 2).

2.3. Rotational motion with a bisector embedding: ROTLEV3B

ROTELEV3B uses a DVR for all three coordinates. In this representation, the Hamiltonian matrix in terms of solutions of the first step is

$$\begin{aligned} \langle h, k, p, q | \widehat{H} | h', k', p, q \rangle &= \delta_{h,h'} \delta_{k,k'} \varepsilon_h^{J,k} \\ &- (1 + \delta_{k,0} + \delta_{k',0})^{-1/2} \delta_{k',k\pm 1} C_{J,k}^{\pm} \sum_{\alpha,\beta,\gamma,\gamma'} \psi_{\alpha,\beta,\gamma}^{J,k,h} \psi_{\alpha,\beta,\gamma'}^{J,k',h'} (M_{\alpha,\alpha,\beta,\beta}^{(1)} - M_{\alpha,\alpha,\beta,\beta}^{(2)}) J_{k\pm 1,k,\gamma,\gamma'}^{(1)} \\ &- (1 + \delta_{k,0} + \delta_{k',0})^{-1/2} \delta_{k',k\pm 2} C_{J,k\pm 1}^{\pm} C_{J,k}^{\pm} \\ &\times \sum_{\alpha,\beta,\gamma} \psi_{\alpha,\beta,\gamma}^{J,k,h} \psi_{\alpha,\beta,\gamma}^{J,k',h'} \left(\frac{1+\gamma}{1-\gamma} \right) (M_{\alpha,\alpha,\beta,\beta}^{(1)} + M_{\alpha,\alpha,\beta,\beta}^{(2)}) J_{k\pm 2,k,\gamma,\gamma'}^{(2)}, \\ k &= p, p+1, \dots, J, \quad p = 0, 1. \end{aligned} \quad (44)$$

In (44), the angular integrals are obtained by quadrature in terms of basis functions and then transformed:

$$J_{k',k,\gamma,\gamma'}^{(1)} = \sum_{j,j'} T_j^{\gamma(k)} \left(\delta_{j,j'} + \left(\frac{k+k'}{2} \right) \langle j, k | \frac{(1+\cos\theta)}{\sin\theta} | j', k' \rangle \right) T_{j'}^{\gamma'(k')}, \quad (45)$$

$$J_{k',k,\gamma,\gamma'}^{(2)} = \sum_{j,j'} T_j^{\gamma(k)} \left(\delta_{j,j'} + \langle j, k | \frac{(1+\cos\theta)}{(1-\cos\theta)} | j', k' \rangle \right) T_{j'}^{\gamma'(k')} \quad (46)$$

where the notation $\gamma(k)$ has been used to emphasis that the DVR points for the bra and ket are based on the Gaussian quadrature points of associated Legendre functions of different degrees.

2.4. Rotational motion with a perpendicular embedding: ROTLEV3Z

The structure of the Hamiltonian matrix for the z -perpendicular embedding is somewhat different from the other cases since the couplings are between cases with $\Delta k = \pm 2$ only, but there are additional couplings between blocks with positive and negative k . In this case the rotational parity, p , is given by the parity of k . To help define the matrix elements it is useful to define an extra quantum number, denoted s below, which depends on the sign of k and takes the value 0 or 1. A detailed discussion of these quantum numbers and the structure of the resulting matrices is given by Kostin et al. [23]; note however that the definition of s and q are swapped in that work compared to here.

ROTLLEV3Z uses a DVR for all three coordinates. The rotational kinetic energy operator (KEO) also has a diagonal contribution as well as coupling k blocks differing by two

$$\begin{aligned} \langle Jksj | T^{\text{rot}} | Jk's'j' \rangle = & -(1 + \delta_{0k})^{1/2} \frac{1}{2} M^+ \langle j | \frac{\cos\theta}{1 - \cos^2\theta} | j' \rangle A_{Jk'}^{\pm} \delta_{k,k' \pm 2} \delta_{s,s'} \\ & + (1 + \delta_{0k})^{1/2} \frac{1}{4} M^- \langle j | \frac{1}{(1 - \cos^2\theta)^{1/2}} | j' \rangle A_{Jk'}^{\pm} \delta_{k,k' \pm 2} \delta_{s,1-s'} \end{aligned} \quad (47)$$

where

$$A_{Jk}^{\pm} = A_{J-k}^{\mp} = ([J(J+1) - k(k \pm 1)][J(J+1) - k(k \pm 3) - 2])^{1/2}, \quad (48)$$

and where

$$M^{\pm} = \frac{1}{2} \left[\frac{1}{\mu_1 R_1^2} \pm \frac{1}{\mu_2 R_2^2} \right]. \quad (49)$$

Finally the Coriolis KEO couples terms diagonal in k but differing in s

$$\begin{aligned} \langle Jksj | T^{\text{cor}} | Jk's'j' \rangle = & -\frac{k}{2} M^- \left(\langle j | -\frac{\cos\theta(1+2a+2j')}{(1-\cos^2\theta)^{1/2}} | j' \rangle \right. \\ & \left. + \langle j | \frac{2(j'+a)}{(1-\cos^2\theta)^{1/2}} \frac{(h_{j'-1}^{ab})^{1/2}}{(h_j^{ab})^{1/2}} | j'-1 \rangle \right) \delta_{kk'} \delta_{s,1-s'}. \end{aligned} \quad (50)$$

Besides these terms, there is a special case which occurs when $k = 1$. For this case there are extra terms both on the diagonal, augmenting $\langle Jkqs | T^{\text{vib}} | Jk's'j \rangle$:

$$\langle Jksj | T_{k=1}^{\text{vib}} | Jk's'j \rangle = \frac{(-1)^{1+s}}{4} J(J+1) M^+ \langle j | \frac{\cos\theta}{1 - \cos^2\theta} | j' \rangle \delta_{k,1} \delta_{k',1} \delta_{s,s'} \quad (51)$$

and on the off-diagonal augmenting $\langle Jksj | T^{\text{cor}} | Jk's'j \rangle$

$$\langle Jksj | T_{k=1}^{\text{rot}} | Jk's'j \rangle = \frac{(-1)^{s'+q}}{4} J(J+1) M^- \langle j | (1 - \cos^2\theta)^{-1/2} | j' \rangle \delta_{k,1} \delta_{k',1} \delta_{s,1-s'} \quad (52)$$

where q is the parity of the radial basis, see Eq. (23). The above matrix elements include a cancellation between vibrational and rotational terms which is potentially singular at linear (i.e., $\theta = 0$ or π) geometries.

2.5. Non-adiabatic corrections

Polyansky and Tennyson [24,25] showed that non-adiabatic corrections to the Born–Oppenheimer approximation could, at least partially, be modelled by using separated reduced masses for the vibrational and rotational motions, μ_i^V and μ_i^R , respectively. If $\mu_i^V \neq \mu_i^R$, a new term has to be added the Hamiltonian

$$\widehat{K}_{\text{NBO}} = \delta_{k,k'} k^2 \langle j', k' | \sin^{-2} \theta | j, k \rangle \left(\frac{\hbar^2}{2r_1^2} \left(\frac{1}{\mu_1^R} - \frac{1}{\mu_1^V} \right) + \frac{\hbar^2}{2r_2^2} \left(\frac{1}{\mu_2^R} - \frac{1}{\mu_2^V} \right) \right) \quad (53)$$

for the in-plane z embeddings [24] and

$$\langle Jksj | T^{\text{NBO}} | Jk's'j \rangle = \left(1 - \frac{\mu_r}{\mu_v} \right) \frac{1}{4} [J(J+1) - k^2] M^+ \langle j | \frac{1}{1 - \cos^2 \theta} | j' \rangle \delta_{k,k'} \delta_{s,s'} \quad (54)$$

for the z -perpendicular embedding [23].

As μ_i^V is usually greater than μ_i^R , this extra term, which only occurs for $J > 0$, is generally positive. The term is computed if separate vibrational and rotational masses are specified in the input.

2.6. Dipole transition moments: DIPOLE3

Lynas-Gray et al. [46] adapted Miller et al.'s [47] formalism for calculating dipole transition strengths between rotational–vibrational levels which use only rigorous selection rules. Their algorithm used DVRs in the radial coordinates and transformed to an FBR for the angular integrals. For very large runs this method is still not optimal and a new algorithm has been developed which retains the DVR in the radial coordinates and evaluates the angular integrals as a simple sum over Gauss–Legendre quadrature point. This gives the basic structure that a transition dipole is evaluated as the sum over grid points of the bra wavefunction times the dipole operator times the ket wavefunction [6]. Since this algorithm has not been published before, a derivation is given in [Appendix A](#).

As the angular DVR's defined by DVR3DRJZ depend on k , the method used therefore involves transforming all wavefunctions to an FBR in the angular coordinate. These FBR wavefunctions, and the dipole surface, are then evaluated at the requested Gauss–Legendre quadrature points, γ_i . It is usually necessary to use slightly more Gauss–Legendre quadrature points than the number of DVR points used to define the underlying wavefunctions. However the expression, Eq. (55) below, involves one less summation than the equivalent one of Lynas-Gray et al. [46]. In addition the structure of the wavefunction files created by ROTLEV3 and ROTLEV3B has been reorganised to minimise the i/o requirements in DIPOLE3 (and XPECT3). Note that ROTLEV3Z cannot be used in conjunction with these programs.

With the new algorithm, the expression for the transition strength $S(f - i)$ is given by

$$S(f - i) = \frac{1}{4} [(2J' + 1)(2J'' + 1)] [(-1)^{J''+J'+1} + (-1)^{p'+p''}]^2 \times \left[\sum_{v=-1}^{+1} \sum_{k=p''}^{J''} a(k+v, k) (-1)^k \begin{pmatrix} J' & 1 & J'' \\ -v-k & v & k \end{pmatrix} \sum_{\alpha\beta\gamma} c_{k'\alpha\beta\gamma}^{J'p'l'} c_{k''\alpha\beta\gamma}^{J''p''l''} \mu_v^m(\alpha\beta\gamma) \right]^2, \quad (55)$$

where $\underline{c}^{JMp'l}$ is the value of the wavefunction of the l th state with rotational quantum numbers (J, M, p) , at grid point $(\alpha\beta\gamma)$. The coefficient $a(v, k)$ is given by

$$\begin{aligned} a(0, k) &= 2^{-1/2} b, \\ a(\pm 1, 0) &= \mp 2^{-1/2} b, \\ a(\pm 1, k) &= \mp b \end{aligned} \quad (56)$$

where the factor b depends on the embedding used:

$$\begin{aligned} b &= 1 && \text{for the standard } r_1 \text{ embedding;} \\ b &= (-1)^v && \text{for the standard } r_2 \text{ embedding;} \\ b &= (-1)^d && \text{for the bisector embedding,} \\ d &= \text{integer}\left(\frac{k+p'}{2}\right) + \text{integer}\left(\frac{k+v+p''}{2}\right) \end{aligned} \quad (57)$$

where the *integer* specifies integer arithmetic rounded towards zero.

2.7. Synthetic spectra: SPECTRA

There are a number of ways of representing transition data. The simplest is the Einstein A-coefficient which can be defined in terms of our linestrength (55):

$$A_{if} = \frac{64\pi^4}{3c^3h} \omega^3 \frac{S(f-i)g_i}{2J'+1}. \quad (58)$$

In programs the constants in the first fraction is set to 3.136186×10^{-7} which also converts from a linestrength in D^2 to an A-coefficient in s^{-1} .

Integrated absorption coefficients, in cm/molecule, can be generated as a function of frequency, ω_{if} , and temperature, T , using the formula

$$I(\omega_{if}) = \frac{4.162034 \times 10^{-19} \omega_{if} g_i [\exp(E''/kT) - \exp(E'/kT)]}{Q(T)} S(f-i) \quad (59)$$

where $Q(T)$ is the partition function of the system, and ω' and ω'' are respectively the energies of the upper and lower state. The degeneracy factor g_i for a particular level is determined by nuclear spin statistics.

The emissivity, in Ergs/molecule/Sr, is defined by

$$J(\omega_{if}) = \frac{(2J'+1)g_i hc \omega_{if} \exp(-E'/kT)}{4\pi Q(T)} A_{if}. \quad (60)$$

If enough energy levels of the system are known it is possible to obtain an expression for the partition function of the system

$$Q(T) = \sum_J \sum_i g_i (2J+1) \exp\left(-\frac{E_i^J}{kT}\right). \quad (61)$$

2.8. Expectation values: XPECT3

Expectation values of geometrically defined operators can be computed in a straightforward manner using a DVR and the quadrature approximation simply by summing the square of the wavefunction and the operator over all grid points. This is useful for obtaining such things as average coordinates or rotational constants. Unlike DIPOLE3, all expectation values involve diagonal summations and these can be evaluated straightforwardly on the original DVR grid. Such expectation values have been used in a number of schemes for automating vibrational assignments [48,49].

Expectation values are particularly useful for least squares fitting of potential energy surfaces to spectroscopic data [50]. If, as is often the case, the potential is represented as linear expansion of some function set

$$V(r_1, r_2, \theta) = \sum_i c_i f_i(r_1, r_2, \theta) \quad (62)$$

then for state l the expectation value of the derivative with respect to the expansion parameters, $\frac{dV}{dc_i}$, can be obtained using the Hellmann–Feynman theorem as

$$\frac{d\langle l|V|l\rangle}{dc_i} = \langle l|\frac{dV}{dc_i}|l\rangle = \langle l|f_i(r_1, r_2, \theta)|l\rangle, \quad (63)$$

where use has been made of the fact that the kinetic energy operator does not depend on c_i .

Use of Eq. (63) means that it is cheap and straightforward to test the sensitivity of all expansion parameters of the potential during a fit. Formally the Hellmann–Feynman theorem is only satisfied for exact wavefunctions, which has caused difficulties in some numerical implementations. However, experience and many tests have shown that converged wavefunctions obtained with present programs are accurate enough for this procedure to be completely reliable with well converged basis sets.

There is one final point that needs to be considered when fitting potentials to spectroscopic data. Fits are not made to absolute energies but to energy differences such as transition frequencies or term values which are energies relative to the ground state. This means that one actually needs to compute the difference in the derivative of Eq. (63) rather than the absolute difference. So for the l th term value, E_l , one needs:

$$\frac{E_l}{dc_i} = \langle l|\frac{dV}{dc_i}|l\rangle - \langle 0|\frac{dV}{dc_i}|0\rangle \quad (64)$$

where $|0\rangle$ is the wavefunction of the ground state. To facilitate this process for term values, the code can optionally read in derivatives for the ground state and take the differences. If transition frequencies are being fitted it is necessary for the user to take these differences which are, of course, dependent on the quantum numbers associated with each transition.

3. Program structure

User input is needed for all modules. The modules 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 data flow through the modules is given in Fig. 2. The role of the individual subroutines is described in comments included in the source programs. All modules now make full use of the facility to allocate arrays dynamically

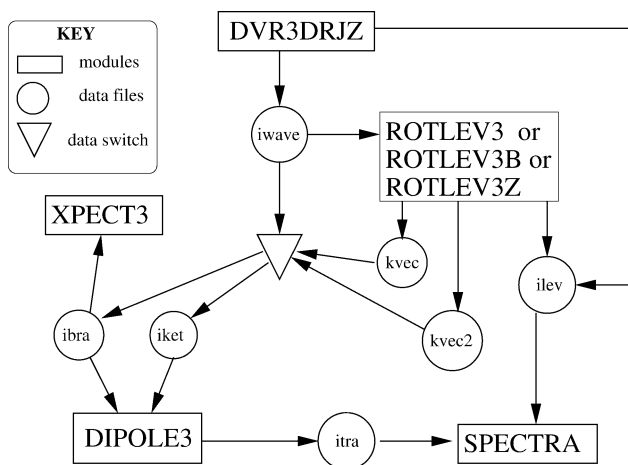


Fig. 2. Structure and data flow for the DVR3D program suite. Note that module ROTLEV3Z cannot drive DIPOLE3 or XPECT3. Scratch disk files used by individual modules are not shown.

offered under FORTRAN 90. In addition calls to a number of basic linear algebra subroutines (BLAS) have been implemented. The local implementation of these routines should be used where possible otherwise FORTRAN sources can be downloaded from the web.

The only remaining machine depend aspect of the codes concerns reading to the end of files. Certain disk files (ILEV in DVR3DRJZ, ROTLEV3 ROTLEV3B and ROTLEV3Z, ITRA in DIPOLE3 and XPECT3) may be read to the end and then written to. Unfortunately whether it is necessary to BACKSPACE the file between the reads and the writes depends on the machine being used. The present implementation *does* BACKSPACE and it may be necessary to remove these statements. The occurrences, at most once per module, are all clearly labelled in the sources of the programs.

Internally the programs work and pass data between them entirely in atomic units. Atomic mass units, used for input, are converted using $1 u = 1822.8883 m_e$; note that this value differs in earlier versions of the program. For printing purposes the conversion factors $1 E_h = 219474.624 \text{ cm}^{-1}$ is used for energies and $1 \text{ au} = 2.5417662 \text{ D}$ for dipoles.

3.1. DVR3DRJZ

For large runs, the CPU time requirement of DVR3DRJZ is usually dominated by the diagonalisation of the final 3D Hamiltonian matrix HAM3. The required diagonaliser has to compute all eigenvalues and eigenvectors of a real symmetric matrix. The present implementation uses LAPACK [51] subroutine dsyev.

3.2. ROTLEV3, ROTLEV3B and ROTLEV3Z

These routines have rather different performance characteristics with ROTLEV3 taking little time to construct the Hamiltonian matrix unlike the other two. Execution times for ROTLEV3B and ROTLEV3Z are dominated by two processes, transformation and diagonalisation. The step which transforms the matrix elements has been programmed using the BLAS rank-one-update routine dger.

In all three cases in core diagonalisation is performed by dsyev [51]. As the matrices constructed by ROTLEV3 and ROTLEV3B can be sparse, particularly for high values of J , iterative diagonalisation is offered in the programs as an alternative which should be used if less than 20% of the eigenvalues are required. Iterative diagonalisation is performed using NAG Library routine F0JFJF [52] and associated subroutines This diagonaliser uses the algorithm of Nikolai [53] and again timing is dominated by the vector matrix multiplies performed by BLAS routine dgemv.

When using ROTLEV3B for rotational states with $J > 1$, it is *essential* that the vibrational wavefunctions output file IWAVE is input on *both* streams IVEC and IVEC2, which *must* correspond to separate files. This duplication is required to prevent excessive repositioning of the input file.

Similarly for ROTLEV3Z for rotational states with $J > 1$ all calculations require *four* wavefunction input files. It is essential that the vibrational wavefunctions output file IWAVE generated by DVR3DRJZ run with IPAR = 0 is copied to units IVEC and IVEC1, and the IWAVE file generated by DVR3DRJZ run with IPAR = 1 is copied to IVEC2 and IVEC3.

3.3. DIPOLE3

DIPOLE3 uses the BLAS matrix multiply routine dger to perform transformations. This section of the code dominates the CPU usage and the coding to parallelise it for large runs is in place using OpenMP [54]. The parallel version has been used successfully on 8 processors of an SGI Origin 2000 machine [21] and a Sun V750 machine, it has also been tested on a dual processor Linux PC.

4. Program use

4.1. Potential, dipole and property subroutines

DVR3DRJZ requires the user to provide the appropriate potential energy surface as a subroutine using the structure:

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 $XCOS = \cos \theta$.

DVR3DRJZ includes COMMON /MASS/ XMASS(3),G1,G2,XMASSR(3) where XMASS contains the (vibrational) masses in atomic mass units, $G1 = g_1$, $G2 = g_2$ and XMASSR contains the rotational masses also in atomic mass units. This enables users to write flexible potential subroutines which allow for changes in coordinates or isotopic substitution.

DIPOLE requires a subroutine defining the dipole surfaces.

SUBROUTINE DIPD(DIPC,R1,R2,XCOS,NU) must be supplied. DIPD returns the NUth component of the dipole in atomic units (1 au = 2.5417662 Debye) at point $R1 = r_1$, $R2 = r_2$ (both in Bohr) and $XCOS = \cos \theta$ where $NU = 0$ corresponds to μ_z and $NU = 1$ μ_x .

DIPOLE3 includes COMMON /MASS/ XMASS(3),G1,G2,ZEMBED,ZBISC where XMASS contains the (vibrational) masses, $G1 = g_1$, $G2 = g_2$, ZEMBED is the axis embedding parameter defined below and ZBISC = .TRUE. for the bisector embedding case ($IDIA = -2$, $JROT > 0$ and $ZPERP = .FALSE.$) and .FALSE. otherwise. This enables users to write dipole subroutines which allow for changes in coordinates, embeddings or isotopic substitution.

XPECT3 requires a subroutine defining the properties for which expectation values are requested.

SUBROUTINE PROPS(PROP,R1,R2,XCOS,N) must be supplied. PROPS returns the N properties required in array PROP for an arbitrary point given by $R1 = r_1$, $R2 = r_2$ (both in Bohr) and $XCOS = \cos \theta$. Like DIPOLE3, XPECT3 includes COMMON /MASS/ XMASS(3),G1,G2,ZEMBED,ZBISC where the variables are as defined above.

4.2. Input for DVR3DRJZ

DVR3DRJZ requires 9 lines of user input for all runs. Lines giving data not required or for which the defaults [given below in parenthesis] are sufficient should be left blank.

Line 1: NAMELIST /PRT/

ZPHAM[F] = T requests 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 MAX3D (MAX3D2).
 ZROT[T] = T DVR3DRJZ to perform first step in a two-step variational calculation.
 = F Rotational excitation calculation with Coriolis coupling neglected.
 ZPERP[F] = F z -axis embedded in plane.
 = T z -axis embedded perpendicular to molecular plane.
 ZEMBED[T] Used only if $J > 0$, ZBISC = F and ZPERP = F.
 = T z -axis embedded along r_2 ;
 = F z -axis embedded along r_1 .

- ZLIN [F] = T forces suppression of functions at last DVR point (ZBISC = T or ZPERP = T only).
- ZMORS1[T] = T use Morse oscillator-like functions for r_1 coordinate;
= F use spherical oscillator functions (IDIA > 0 only).
- ZMORS2[T] = T use Morse oscillator-like functions for r_2 coordinate;
= F use spherical oscillator functions (IDIA > 0 only).
- ZLADD[T] = T NALF kept constant as k increases,
= F NALF decreases with k
(used if ZROT = T only).
- ZTWOD[F] = T perform 2D calculation only at specified grid point.
- ZVEC[F] = T store the eigenvectors 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;
(used if IDIA > -2 only).
- 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;
(used if IDIA > -2 only).
- ZTRAN[F] = T perform the transformation of the solution coefficients to the expression for the wavefunction amplitudes at the grid points, Eq. (29). Store the data on stream IWAVE.
ZTRAN = T automatically sets ZVEC = T if IDIA > -2.
- 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, 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 IDIAG1[20] and IDIAG2[21]. For further details see the source code.
- ZPFUN[F] = T eigenvalues concatenated on stream ILEV.
Warning, the first eigenvalues on this file must be for JROT = 0, IPAR = 0.
- ILEV[14] output stream for eigenvalue data (formatted).
- IEIGS1[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[17] 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.

Line 2: NCOORD (15)

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

Line 3: NPNT2,JROT,NEVAL,NALF,MAX2D,MAX3D,IDIA,KMIN,NPNT1,IPAR,MAX3D2(11I5)

NPNT2 number of DVR points in r_2 from 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, (ignored if IDIA = -2).

MAX3D maximum dimension of the final Hamiltonian.

If ZCUT = F, it is the actual number of functions selected.

If ZCUT = T, MAX3D must be \geq than the number of functions selected using EMAX2.

IDIA = 1 for scattering coordinates with a heteronuclear diatomic,

= 2 for scattering coordinates with a homonuclear diatomic,

= -1 for Radau coordinates with a heteronuclear diatomic,

= -2 for Radau coordinates with a homonuclear diatomic.

KMIN[0] = k for JROT > 0 and ZROT = F,

= $(1 - p)$ for JROT > 0 and ZROT = T.

Note:

For IDIA > 0, KMIN must be 1 in DVR3DRJ if KMIN = 2 in ROTLEV3.

For ZBISC = T, setting KMIN = 2 performs $p = 0$ and 1 calculations.

For ZPERP = T and ZROT = T, use KMIN = 1.

NPNT1 number of DVR points in r_1 from Gauss-(associated) Laguerre quadrature, (ignored if IDIA = -2).

IPAR[0] parity of basis for the AB₂ molecule (i.e., |IDIA| = 2) case:

IPAR = 0 for even parity and = 1 for odd.

MAX3D2[MAX3D] maximum dimension of odd parity final Hamiltonians.

(IDIA = -2, ZROT = T only).

Line 4: TITLE (9A8)

A 72 character title.

Line 5: FIXCOS (F20.0)

If ZTWOD = T, FIXCOS is the fixed value of $\cos\theta$ for the run.

If ZTWOD = F, this line is read but ignored.

Line 6: (XMASS(I),I = 1,3) (3F20.0)

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

Line 7: (XMASSR(I),I = 1,3) (3F20.0)

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

If XMASSR(1) is not set, XMASSR is set equal to XMASS.

Line 8: 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 (IDIA > -2 only).

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.

Line 9: 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.

Line 10: 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.

If IDIA = -2 line read but ignored.

Line 11: EZERO [0.0] (F20.0)

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

Optional and only read when IDIA = ± 2 , IPAR = 1 and JROT = 0.

4.3. Input for ROTLEV3

Most of the data for ROTLEV3, which must have been prepared previously by DVR3DRJZ run with IDIA > -2, is read from stream IWAVE. 3 or 4 lines of data are read.

Line 1: NAMELIST/PRT/

TOLER[0.0D0] tolerance for convergence of the eigenvalues by F02FJF [52], zero gives machine accuracy, 1×10^{-4} is usually sufficient for most applications.

(Ignored if ZDCORE = T.)

ZPVEC[F] = T requests printing of the eigenvectors.

THRESH[0.1d0] threshold for printing eigenvector coefficients, zero requests the full vector (only used if ZPVEC = T).

ZPHAM[F] = T requests printing of the Hamiltonian matrix.

ZPTRAF[F] = T requests printing of the transformed vectors.

IWAVE[26] stream for input data from DVR3DRJZ (unformatted).

IVEC[4] scratch file for the transformed input vectors (unformatted).

JVEC[3] stream for first set of eigenvalue/vector output (unformatted).

JVEC2[2] stream for second set of eigenvalue/vector output (unformatted), KMIN = 2 only.

ZTRAN[F] = T eigenvectors transformed back to original basis.

ZVEC[F] = T eigenvalue and eigenvector data to be written to disk file.

(= T forced if ZTRAN = T.)

KVEC[8] stream for first set of transformed eigenvector output (unformatted).

KVEC2[9] stream for second set of transformed eigenvector output (unformatted), KMIN = 2 only.

ISCR[1] stream for scratch file storing Hamiltonian matrix (unformatted).

JSCR[7] stream for scratch file storing DVR3DRJZ vectors transformed to an FBR in θ (unformatted).

Used only if ZTRAN = T.

IRES[0] restart flag:

= 0 normal run.

= 1 full restart.

= 2 restart second diagonalisation only (for KMIN = 2 only).

= -1 perform vector transformation only (stream JVEC must be supplied).

ZPFUN[F] = T eigenvalues concatenated on stream ILEV. The first eigenvalues on this file must (with $J = 0$, j even) be already present.

ILEV[14] stream for eigenvalue data (formatted).

ZDIAG[T] = F do not diagonalise the Hamiltonian matrix.

ZDCORE[F] = T diagonalisation performed in core using LAPACK routine dsyev,
 = F diagonalisation performed iteratively using NAG Routine F02FJF.

Line 2: NVIB,NEVAL,KMIN,IBASS,NEVAL2 (5I5)

NVIB number of vibrational levels from DVR3DRJZ for each k to be read, and perhaps selected from, in the second variational step.

NEVAL[10] the number of eigenvalues required for the first set.

KMIN[0] = 0, f or $p = 1$ parity calculation.

= 1, e or $p = 0$ parity calculation.

= 2, do both e and f parity calculation.

IBASS[0] = 0 or $> NVIB*(JROT + KMIN)$, use all the vibrational levels.

Otherwise, select IBASS levels with the lowest energy.

NEVAL2[NEVAL] the number of eigenvalues required for the second set.

Line 3: TITLE (9A8)

A 72 character title.

Line 4: EZERO [0.0] (F20.0)

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

4.4. Input for ROTLEV3B

Most of the data for ROTLEV3B, which must have been prepared previously by DVR3DRJZ run with IDIA = -2, is read from streams IVEC and IVEC2. 3 or 4 lines of data are read.

Line 1: NAMELIST/PRT/

TOLER[0.0D0] tolerance for convergence of the eigenvalues by F02FJF [52], zero gives machine accuracy, 1×10^{-4} is usually sufficient for most applications.

(Ignored if ZDCORE = T.)

ZPVEC[F] = T requests printing of the eigenvectors.

THRESH[0.1d0] threshold for printing eigenvector coefficients, zero requests the full vector (only used if ZPVEC = T).

ZPHAM[F] = T requests printing of the Hamiltonian matrix.

ZPTRAF[F] = T requests printing of the transformed vectors.

IVEC[26] stream for input data from DVR3DRJZ (unformatted).

IVEC2[4] second stream for input data from DVR3DRJZ (unformatted); this file is simply a copy of that on stream IVEC.

ZVEC[F] = T eigenvalue and eigenvector data to be written to disk file (= T forced if ZTRAN = T).

JVEC[3] stream for first set of eigenvalue/vector output (unformatted).

JVEC2[2] stream for second set of eigenvalue/vector output (unformatted), KMIN = 2 only.

ZTRAN[F] = T eigenvector transformed back to original basis.

ZVEC[F] = T eigenvalue and eigenvector data to be written to disk file (= T forced if ZTRAN = T).

KVEC[8] stream for first set of transformed eigenvector output (unformatted).

KVEC2[9] stream for second set of transformed eigenvector output (unformatted), KMIN = 2 only.

ISCR[1] stream for scratch file storing array OFFDG (unformatted).

IRES[0] restart flag:

= 0 normal run.

= 1 full restart.

= 2 restart second diagonalisation only (for $KMIN = 2$ only).
 = -1 perform vector transformation only (stream JVEC must be supplied).
 ZPFUN[F] = T eigenvalues concatenated on stream ILEV. The first eigenvalues on this file must (with $J = 0$, j even) be already present.
 ILEV[14] stream for eigenvalue data (formatted).
 ZDIAG[T] = F do not diagonalise the Hamiltonian matrix.
 ZDCORE[F] = T diagonalisation performed in core using LAPACK routine dsyev,
 = F diagonalisation performed iteratively using NAG Routine F02FJF.

Line 2: NVIB,NEVAL,KMIN,IBASS,NEVAL2,NPNT (5I5)

NVIB number of vibrational levels from DVR3DRJZ for each k to be read, and perhaps selected from, in the second variational step.
 NEVAL[10] the number of eigenvalues required for the first set.
 KMIN[0] = 0, f or $p = 1$ parity calculation.
 = 1, e or $p = 0$ parity calculation.
 = 2, do both e and f parity calculation.
 IBASS[0] = 0 or $> NVIB*(JROT + KMIN)$, use all the vibrational levels.
 Otherwise, select IBASS levels with the lowest energy.
 NEVAL2[NEVAL] the number of eigenvalues required for the second set.
 NPNT[NALF] number of quadrature points for angular integrals.

Line 3: TITLE (9A8)

A 72 character title.

Line 4: EZERO [0.0] (F20.0)

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

4.5. Input for ROTLEV3Z

Most of the data for ROTLEV3Z, which must have been prepared previously by DVR3DRJZ run with IDIA = -2, is read from streams IVEC and IVEC2. To avoid repeating i/o a second copy of each of these files is needed linked to streams IVEC1 and IVEC3 respectively. The program also uses six scratch files which are associated with streams 1, 11, 60, 61, 70, 80 and 81. 3 or 4 lines of data are read.

Line 1: NAMELIST/PRT/

ZPVEC[F] = T requests printing of the eigenvectors.
 IVEC[26] stream for input data from DVR3DRJZ run with IPAR = 0 (unformatted).
 IVEC1[27] stream holding a copy of stream IVEC.
 IVEC2[4] stream for input data from DVR3DRJZ run with IPAR = 1 (unformatted).
 IVEC3[28] stream holding a copy of stream IVEC2.
 ZVEC[F] = T eigenvalue and eigenvector data to be written to disk file (= T forced if ZTRAN = T).
 JVEC[3] stream for eigenvalues and untransformed vectors (unformatted).
 ZTRAN[F] = T eigenvector transformed back to original basis.
 ZPTR[A] = T requests printing of the transformed vectors.
 ZVEC[F] = T eigenvalue and eigenvector data to be written to disk file (= T forced if ZTRAN = T).
 KVEC[8] stream for eigenvalues and transformed eigenvector output (unformatted).
 ZPFUN[F] = T eigenvalues concatenated on stream ILEV. The first eigenvalues on this file must (with $J = 0$, j even) be already present.

ILEV[14] stream for eigenvalue data (formatted).
 ZDIAG[T] = F do not diagonalise the Hamiltonian matrix.

Line 2: NVIB,NEVAL,KPAR,IBASS,IQPAR,NPNT (6I5)

NVIB number of vibrational levels from DVR3DRJZ for each k to be read, and perhaps selected from, in the second variational step.
 NEVAL[10] the number of eigenvalues required for the first set.
 KPAR = 0, for k even calculation,
 = 1, for k odd calculation.
 IBASS[0] = 0 or $>$ NVIB*(JROT + 1), use all the vibrational levels.
 Otherwise, select IBASS levels with the lowest energy.
 IQPAR parity of Hamiltonian, i.e., 0 or 1.
 NPNT[NALF] number of quadrature points for angular integrals.

Line 3: TITLE (9A8)

A 72 character title.

Line 4: EZERO [0.0] (F20.0)

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

5. Input for DIPOLE3

DIPOLE3 takes most of its input from the output streams IWAVE (from DVR3DRJZ) or KVEC and KVEC2 (from ROTLEV3 or ROTLEV3B). It has the option to produce output files for SPECTRA [26] to calculate simulated spectra at a given temperature. Input and output on streams IKET, IBRA and ITRA are in atomic units. The data printed at the end of DIPOLE is given in wavenumbers, Debye for the transition dipoles and s^{-1} for the Einstein A coefficients. The user must supply the following three lines of input

Line 1: NAMELIST/PRT/

ZPRINT[F] = T supplies extra print out for debugging purposes.
 ZTRA[T] = T writes data for SPECTRA to stream ITRA.
 ZSTART[F] = T initiates the output file for the data for SPECTRA.
 = F writes data to the end of existing file on stream ITRA.
 IKET[11] input stream from DVR3DRJZ/ROTELEV3/ROTELEV3B for the ket (unformatted).
 IBRA[12] input stream for the bra (unformatted).
 ITRA[13] output stream to SPECTRA (if ZTRA = T) (unformatted).

Line 2: TITLE (9A8)

A 72 character title.

Line 3: NPOT, NV1, NV2, IBASE1, IBASE2 (5I5)

NPOT number of Gauss–Legendre quadrature points.
 NV1[all] number of ket eigenfunctions considered.
 NV2[all] number of bra eigenfunctions considered.
 IBASE1[0] number of lowest ket eigenfunctions skipped.
 IBASE2[0] number of lowest bra eigenfunctions skipped.

Line 4: EZERO [0.0] (F20.0)

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

6. Input for SPECTRA

SPECTRA takes most of its input from stream ITRA generated by DIPOLE. Data for generating partition functions may optionally be provided by DVR3DRJZ and one of the ROTLEV programs via stream ILEV. The user must supply 5 (4 if ZSPE = .FALSE.) lines of input.

Line 1: NAMELIST/PRT/

ZOUT[F] = T, print sorted transition frequencies and line strengths.
 ZOUT is set automatically to T if ZSPE = F.
 ZSORT[T] = T sort transition data and write it to stream ISPE.
 = F sorted transition data is to be read from stream ISPE.
 ZSPE[T] = F if the program is to stop after sorting only.
 ZPFUN[T] = T calculate the partition function from data on stream ILEV;
 = F the partition function is not calculated but set to value Q which is read in.
 ITRA[13] input stream transitions file from DIPOLE3 (unformatted).
 ILEV[14] formatted input stream with energy levels for the partition function.
 The first eigenenergy on ILEV must be the ground state, i.e., $J = 0$, (IPAR = 0).
 This is the zero energy of the problem if ZPFUN = T.
 ISPE[15] stream for the sorted transitions (unformatted).
 ITEM[16] scratch file (unformatted).

The following are only used if ZSORT = T.

WSMIN[0.0d0] minimum transition frequency in cm^{-1} , for which data sorted.
 WSMAX[1.0d6] maximum transition frequency in cm^{-1} , for which data sorted.
 EMIN[−1.0d27] minimum value of energy of lower state, E'' , in cm^{-1} , for which data sorted.
 EMAX[+1.0d27] maximum value of energy of lower state, E'' , in cm^{-1} , for which data sorted.
 JMAX[all] maximum value of J'' , the angular momentum of the lower state, for which data sorted.
 SMIN[0.0] minimum linestrength (in D^2) for which data sorted.
 GZ[0.0] ground state energy in cm^{-1} .

Line 2: TITLE (9A8)

A 72 character title.

Line 3: GE, GO (2D10.0)

GE[1.0d0] nuclear-spin times symmetry-degeneracy factors for AB_2 molecule (i.e., $|\text{IDIA}| = 2$) for the even (IPAR = 0).
 GO[1.0d0] nuclear-spin times symmetry-degeneracy factors for homonuclear AB_2 molecule (i.e., $|\text{IDIA}| = 2$) for the odd (IPAR = 1).

Line 4: TEMP, XMIN, WMIN, WMAX, DWL, Q (6F10.0)

TEMP temperature in K.
 XMIN lowest relative intensity printed.
 WMIN[0.0] minimum transition frequency required in cm^{-1} .
 WMAX[all] maximum transition frequency required in cm^{-1} .

DWL[0.0] profile half width, in cm^{-1} or μm depending on ZFREQ, used if ZPROF = T and ZDOP = F.
 Q[1.0] value of the partition function, only used if ZPFUN = F.

Line 5: NAMELIST/SPE/

EMIN1[−1.0d27] minimum value of energy of lower state, E'' , in cm^{-1} , for which data printed.
 EMAX1[+1.0d27] maximum value of energy of lower state, E'' , in cm^{-1} , for which data printed.
 EMIN2[−1.0d27] minimum value of energy of upper state, E' , in cm^{-1} , for which data printed.
 EMAX2[+1.0d27] maximum value of energy of upper state, E' , in cm^{-1} , for which data printed.
 JMAX[all] maximum value of J' , the lower state angular momentum, for which data printed.
 ZEMIT[F] = F calculates integrated absorption coefficient.
 = T calculates emissivities.
 ZPLOT[F] = T writes computed spectrum to stream IPLOT.
 IPLOT[20] output stream for formatted file containing spectral data.
 ZPROF[F] = T gives spectrum at NPOINTS points with Gaussian line profiles;
 only used with ZPLOT = T, when results are written on stream IPLOT.
 = F generates stick spectrum.
 NPOINTS[3000] number of points at which spectrum is stored if ZPROF = T.
 ZDOP[F] = T use thermal Doppler half width for spectral profile.
 XMOLM[18.0] molecular mass in amu. Only used if ZDOP = T.
 ZENE[F] = T write assignments with lines to IPLOT.
 = F only line position and intensity to IPLOT.
 Only used if ZPROF = F.
 IDAT[19] Scratch file used to construct profiles (formatted), used if ZPROF = T and ZPLOT = T.
 ZLIST[F] = T write transition data to linelist file on stream ILIST.
 ILIST[36] output stream for formatted linelist if ZLIST = T.
 PRTPR[0.1D0] relative intensity threshold for printing results.
 TINTE[1.0D−15] absolute intensity threshold for printing results.
 ZFREQ[T] = T, stream IPLOT contains wavenumber (cm^{-1}) as first column.
 = F, stream IPLOT contains wavelength (μm) as first column.
 ZEINST[F] = T, stream IPLOT contains spin weighted Einstein A coefficient as second column. In this case
 ZEMIT = T is forced.
 = F, stream IPLOT contains intensity as second column.

7. Input for XPECT3

XPECT3 requires wavefunctions as generated by DVR3DRJZ on unit IWAVE (ZPERP = F only), or ROTLEV3/ROTLV3B on units KVEC or KVEC2. An additional four lines of standard input are required:

Line 1: NAMELIST/PRT/

ZPRINT[F] = T supplies extra print out for debugging purposes.
 ZTRA[T] = T writes property data to stream ITRA.
 ZFORM[F] = T use formatted writes to stream ITRA.
 ZFIT[T] = T potential energy fit being performed.
 = F standard expectation value run.
 ZSTART[F] = T initiates the output file ITRA;
 = F writes data to the end of existing file on stream ITRA.
 (Only used if ZFIT = F.)

IKET[11] input stream from DVR3DRJZ/ROTLEV3/ROTLEV3B (unformatted).
 ITRA[12] output stream for properties (if ZTRA = T).
 ITRA0[28] derivatives computed by XPECT3 for ground state.
 (Only used if ZFIT = T.)

Line 2: TITLE (9A8)

A 72 character title.

Line 3: LPOT, NPROPIN, NPRT, NV1 (515)

LPOT number of Gauss–Legendre quadrature points.
 (Ignored if $J = 0$.)

NPROPIN[1] number of properties to be considered.

NPRT [NPROPIN] total possible number of properties.

NV1[all] number of eigenfunctions considered.

Line 4: (IPROP(I), I = 1, NPROPIN) (2015)

Pointers to the properties in array IPROP to be considered. Default is 1, 2, . . . , NPROPIN.

Values must be unique and in the range 1 to NPRT.

8. Test runs

Short scripts, input data and associated output files have been included with the programs for four different cases. These cases, ensure that there is a test run for each module, are as follows:

HCN tests use the potential and dipole surfaces of Van Mourik et al. [55]. The calculations use scattering coordinates and run modules DVR3DRJZ, ROTLEV3, DIPOLE3 and SPECTRA. These runs are a cut down version of the calculations reported by Harris et al. [56].

Water tests use the potential energy surface of Shirin et al. [57] and a preliminary version of the dipole surfaces of Lynas-Gray et al. [58]. A properties file used by Shirin et al. for their fits of the potential using spectroscopic data is also given. The calculations use Radau coordinates with a bisector embedding and run modules DVR3DRJZ, ROTLEV3, DIPOLE3 and XPECT3.

H_3^+ in Jacobi coordinates uses the global H_3^+ potential of Polyansky et al. (Fit 2) [59] and the dipole surface of Röhse et al. [60]. This calculation runs modules DVR3DRJZ, ROTLEV3, DIPOLE3 and SPECTRA.

H_3^+ in Radau coordinates with a perpendicular embedding. This calculation uses the same potential as the other H_3^+ test run. It runs modes DVR3DRJZ and ROTLEV3Z only.

All test runs require Basic Linear Algebra Subroutines (BLAS) and LAPACK routine dsyev [51]. The programs have been tested using several compilers. The test runs use the Intel Fortran compiler which we found significantly better than others we tested for use under Linux. This compiler can be downloaded free. To keep run times short, not all the results in the test runs are fully converged.

Acknowledgements

We thank the various users of these programs who both check them and provided helpful feedback. We thank NAG for permission to publish their routine F02FJF. This work has been supported by EPSRC, The Royal Society and the Russian Fund for Fundamental Science.

Appendix A

This appendix gives a derivation of the line strength calculation algorithm used in DIPOLE3. The derivation is based upon that of Miller et al. [47] for an FBR but exploits the properties of a DVR to yield an algorithm which is significantly more efficient. This derivation is applicable to ‘standard’, r_1 and r_2 embeddings, as well as the bisector embedding but does not apply to the perpendicular embedding.

J is the total angular momentum, with k the projection of J onto the z -axis. M spans the magnetic sub levels of the wavefunction. The angular part of the wavefunction is represented in an angular (FBR) basis given by

$$\begin{aligned} |J_M, k, j, p\rangle &= \left(\frac{2J+1}{8\pi^2}\right)^{1/2} \Theta_{j0}(\theta) D_{M0}^J(\alpha\beta\gamma)^*, \quad k=0, p=0, \\ &= (-1)^k \frac{(2J+1)^{1/2}}{4\pi} \Theta_{jk}(\theta) (D_{M-k}^J(\alpha\beta\gamma)^* + (-1)^p D_{Mk}^J(\alpha\beta\gamma)^*), \quad k > 0, p=0, 1. \end{aligned} \quad (\text{A.1})$$

Below the l th eigenfunction for the symmetry block defined by angular momentum J and rotational parity p is denoted by $|J_M, p, l\rangle$. Its value is given at radial grid points α, β by $d_{j\alpha\beta}^{Jki}$, see Eq. (36), where j runs over angular functions of the form of Eq. (A.1).

The line strength $S(f-i)$ for a particular transition from an initial state i to a final state f is given by

$$S(f-i) = \sum_{M'M''\tau} (T_{if}^{M'M''\tau})^2, \quad (\text{A.2})$$

where

$$T_{if}^{M'M''\tau} = \langle J'_{M'}, p', l' | \mu_\tau^s | J''_{M''}, p'', l'' \rangle \quad (\text{A.3})$$

and μ_τ^s is the τ component of the space-fixed dipole moment. In our choice of body-fixed coordinates, only the z and x components are non-zero. The body-fixed dipole moment, $\underline{\mu}^m(r_1, r_2, \theta)$ transforms to a tensor of rank one into space fixed coordinates:

$$\mu_\tau^s = \sum_{v=-1}^{+1} \mu_v^m(r_1, r_2, \theta) D_{\tau,v}^1(\alpha\beta\gamma)^*, \quad (\text{A.4})$$

giving

$$\begin{aligned} T_{if}^{M'M''\tau} &= \langle J'_{M'}, p', l' | \mu_v^m(r_1, r_2, \theta) D_{\tau,v}^{1*} | J''_{M''}, p'', l'' \rangle \\ &= \sum_{v=-1}^{+1} \sum_{k'=p'}^{J'} \sum_{m'n'j'}^{J''} \sum_{k''=p''}^{J''} \langle J'_{M'}, k', m', n', j', p' | \mu_v^m(r_1, r_2, \theta) D_{\tau,v}^{1*} | J''_{M''}, k'', m'', n'', j'', p'' \rangle. \end{aligned} \quad (\text{A.5})$$

This equation can be separated into angular and radial parts. The treatment of the radial part is straightforward and has been discussed previously [46]. Considering the angular part

$$\sum_{v=-1}^{+1} \sum_{k'=p'}^{J'} \sum_{k''=p''}^{J''} \sum_{m'n'j'}^{J''} d_{j'\alpha'\beta'}^{J'p'l'} d_{j''\alpha''\beta''}^{J''p''l''} \langle J'_{M'}, k', j', p' | \mu_v^m(r_1, r_2, \theta) D_{\tau,v}^{1*} | J''_{M''}, k'', j'', p'' \rangle. \quad (\text{A.6})$$

Substituting in the angular functions (A.1), multiplied out and exploiting the properties of the angular algebra gives:

$$T_{if}^{M'M''\tau} = \frac{(-1)^{M'}}{2} (2J'+1)^{1/2} (2J''+1)^{1/2} [(-1)^{J''+J'+1} + (-1)^{p'+p''}]$$

$$\begin{aligned}
& \times \sum_{v=-1}^{+1} \sum_{k=p''}^{J''} (-1)^k \begin{pmatrix} J' & 1 & J'' \\ -M' & \tau & M'' \end{pmatrix} \begin{pmatrix} J' & 1 & J'' \\ -v-k & v & k \end{pmatrix} \\
& \times \sum_{j'j''} d_{j'\alpha'\beta'}^{J'p'l'} d_{j''\alpha''\beta''}^{J''p''l''} \int \mu_v^m \Theta_{j'k'}^*(\theta) \Theta_{j''k''}(\theta) d \cos \theta.
\end{aligned} \tag{A.7}$$

Evaluating the angular integral with M -point Gauss–Legendre quadrature is equivalent to transforming the wavefunctions back to a DVR on a grid of Gauss–Legendre (i.e., $k = 0$) points using transformation (11) to give new coefficients $a_{k\alpha\beta\gamma}^{Jpi}$.

$$\begin{aligned}
T_{if}^{M'M''\tau} &= \frac{(-1)^{M'}}{2} (2J' + 1)^{1/2} (2J'' + 1)^{1/2} [(-1)^{J''+J'+1} + (-1)^{p'+p''}] \\
& \times \sum_{v=-1}^{+1} \sum_{k=p''}^{J''} (-1)^k \begin{pmatrix} J' & 1 & J'' \\ -M' & \tau & M'' \end{pmatrix} \begin{pmatrix} J' & 1 & J'' \\ -v-k & v & k \end{pmatrix} \\
& \times \sum_{\alpha\beta\gamma} a_{k'\alpha'\beta'}^{J'p'l'} a_{k''\alpha''\beta''}^{J''p''l''} \mu_v^m(\alpha\beta\gamma).
\end{aligned} \tag{A.8}$$

Finally summing over all degenerate magnetic substates and summing the dipole components gives the final expression for the linestrength:

$$\begin{aligned}
S(f-i) &= \frac{1}{4} [(2J' + 1)(2J'' + 1)] [(-1)^{J''+J'+1} + (-1)^{p'+p''}]^2 \\
& \times \left[\sum_{v=-1}^{+1} \sum_{k=p''}^{J''} a(k+v, k) (-1)^k \begin{pmatrix} J' & 1 & J'' \\ -v-k & v & k \end{pmatrix} \sum_{\alpha\beta\gamma} c_{k'\alpha'\beta'}^{J'p'l'} c_{k''\alpha''\beta''}^{J''p''l''} \mu_v^m(\alpha\beta\gamma) \right]^2,
\end{aligned} \tag{A.9}$$

where $a(k+v, k)$ is defined by Eqs. (56) and (57). This form of the dipole involves dipole transition integrals which can be expressed in the computationally most efficient, general form $\sum_I \psi_i(I) \mu(I) \psi_f(I)$, where I runs over integration points, as advocated by Schwenke [6].

It should be noted that the phase factors in Eq. (A.9) give the rigorous selection rules:

$$\begin{aligned}
\Delta J &= 0, & p' &\neq p'', \\
\Delta J &= 1, & p' &= p''
\end{aligned} \tag{A.10}$$

which all transitions must obey. Apart from further symmetry-induced selection rules for systems with identical atoms, these are the only selection which rigorously apply.

References

- [1] J. Tennyson, J. Chem. Soc., Faraday Trans. 88 (1992) 3271–3279.
- [2] O.L. Polyansky, N.F. Zobov, S. Viti, J. Tennyson, P.F. Bernath, L. Wallace, Science 277 (1997) 346–349.
- [3] J. Tennyson, Comput. Phys. Rep. 4 (1986) 1–36.
- [4] S. Carter, N.C. Handy, Comput. Phys. Rep. 5 (1986) 115.
- [5] P. Jensen, J. Mol. Spectrosc. 128 (1989) 478.
- [6] D.W. Schwenke, J. Chem. Phys. 100 (1996) 2867–2884;
D.W. Schwenke, J. Chem. Phys. 100 (1996) 18884–18884 (erratum).
- [7] J.M. Hutson, Program Bound, version 5. See www.dur.ac.uk/j.m.hutson/.
- [8] Z. Bačić, J.C. Light, Annu. Rev. Phys. Chem. 40 (1989) 469.
- [9] J.C. Light, T. Carrington Jr., Adv. Chem. Phys. 114 (2000) 263.

- [10] D.O. Harris, G.G. Engerholm, W. Gwinn, *J. Chem. Phys.* 43 (1965) 1515.
- [11] S.W. Huang, T. Carrington Jr., *J. Chem. Phys.* 112 (2000) 8765–8771.
- [12] K.S. Sidhu, S. Miller, J. Tennyson, *Astron. Astrophys.* 255 (1992) 453–456.
- [13] L. Neale, J. Tennyson, *Astrophys. J.* 454 (1995) L169–L173.
- [14] M. Vidler, J. Tennyson, *J. Chem. Phys.* 113 (2000) 9766–9771.
- [15] R.J. Barber, G.J. Harris, J. Tennyson, *J. Chem. Phys.* 117 (2002) 11239–11243.
- [16] F. Allard, P.H. Hauschildt, S. Miller, J. Tennyson, *Astrophys. J.* 426 (1994) L39–L41.
- [17] L. Neale, S. Miller, J. Tennyson, *Astrophys. J.* 464 (1996) 516–520.
- [18] H. Partridge, D.W. Schwenke, *J. Chem. Phys.* 106 (1997) 4618.
- [19] R.C.M. Learner, W. Zhong, J.D. Haigh, D. Belmiloud, J. Clarke, *Geophys. Res. Lett.* 26 (1999) 3609–3612.
- [20] R. Schermaul, R.C.M. Learner, D.A. Newnham, J. Ballard, N.F. Zobov, D. Belmiloud, J. Tennyson, *J. Molec. Spectrosc.* 208 (2001) 43–50.
- [21] G.J. Harris, O.L. Polyansky, J. Tennyson, *Astrophys. J.* 578 (2002) 657–663.
- [22] J. Tennyson, J.R. Henderson, N.G. Fulton, *Comput. Phys. Commun.* 86 (1995) 175–198.
- [23] M.A. Kostin, O.L. Polyansky, J. Tennyson, *J. Chem. Phys.* 116 (2002) 7564–7573.
- [24] O.L. Polyansky, J. Tennyson, *J. Chem. Phys.* 110 (1999) 5056–5064.
- [25] J. Tennyson, P. Barletta, M.A. Kostin, O.L. Polyansky, N.F. Zobov, *Spectrochimica Acta A* 58 (2002) 663–672.
- [26] J. Tennyson, S. Miller, C.R. Le Sueur, *Comput. Phys. Commun.* 75 (1993) 339–364.
- [27] O.L. Polyansky, A.G. Császár, S.V. Shirin, N.F. Zobov, P. Barletta, J. Tennyson, D.W. Schwenke, P.J. Knowles, *Science* 299 (2003) 539–542.
- [28] J. Tennyson, B.T. Sutcliffe, *J. Chem. Phys.* 77 (1982) 4061–4072.
- [29] I.S. Gradshteyn, I.H. Ryzhik, *Tables of Integrals, Series and Products*, Academic Press, New York, 1980.
- [30] J. Tennyson, B.T. Sutcliffe, *J. Mol. Spectrosc.* 101 (1983) 71–82.
- [31] J.R. Henderson, J. Tennyson, *Comput. Phys. Commun.* 75 (1993) 365–378.
- [32] B.T. Sutcliffe, J. Tennyson, *Internat. J. Quantum Chem.* 39 (1991) 183–196.
- [33] A.H. Stroud, D. Secrest, *Gaussian Quadrature Formulas*, Prentice-Hall, London, 1966.
- [34] A.S. Dickinson, P.R. Certain, *J. Chem. Phys.* 49 (1968) 4204.
- [35] J.C. Light, R.M. Whitnell, T.J. Pack, S.E. Choi, in: A. Laganà (Ed.), *Supercomputer Algorithms for Reactivity, Dynamics and Kinetics of Small Molecules*, in: NATO ASI Ser. C, vol. 277, Kluwer, Dordrecht, 1989, p. 187.
- [36] J.R. Henderson, C.R. Le Sueur, S.G. Pavett, J. Tennyson, *Comput. Phys. Commun.* 74 (1993) 193–198.
- [37] M.J. Bramley, T. Carrington Jr., *J. Chem. Phys.* 101 (1994) 8494–8507.
- [38] O.L. Polyansky, J. Tennyson, N.F. Zobov, *Spectrochimica Acta* 55 A (1999) 659–693.
- [39] H.Y. Mussa, Ph.D. Thesis, University College London (1998).
- [40] J. Tennyson, B.T. Sutcliffe, *Mol. Phys.* 58 (1986) 1067–1085.
- [41] J. Tennyson, B.T. Sutcliffe, *Internat. J. Quantum Chem.* 42 (1992) 941–952.
- [42] P. Sarkar, N. Poulin, T. Carrington Jr., *J. Chem. Phys.* 110 (1999) 10269–10274.
- [43] J.R. Henderson, J. Tennyson, B.T. Sutcliffe, *J. Chem. Phys.* 98 (1993) 7191–7203.
- [44] D.M. Brink, G.R. Satchler, *Angular Momentum*, Clarendon Press, Oxford, 1979.
- [45] B.T. Sutcliffe, S. Miller, J. Tennyson, *Comput. Phys. Commun.* 51 (1988) 73–82.
- [46] A.E. Lynas-Gray, S. Miller, J. Tennyson, *J. Mol. Spectrosc.* 169 (1995) 458–467.
- [47] S. Miller, J. Tennyson, B.T. Sutcliffe, *Mol. Phys.* 66 (1989) 429–456.
- [48] M. Menou, C. Leforestier, *Chem. Phys. Lett.* 210 (1993) 294–302.
- [49] J.M. Bowman, B. Gazdy, J.A. Bentley, T.J. Lee, C.E. Dateo, *J. Chem. Phys.* 99 (1993) 308–323.
- [50] C.D. Paulse, J. Tennyson, *J. Mol. Spectrosc.* 168 (1994) 313–322.
- [51] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
- [52] NAG Fortran Library Manual, Mark 14 (1990).
- [53] P.J. Nikolai, *ACM Trans. Math. Software* 5.
- [54] openMP consortium, <http://www.openmp.org> (2002).
- [55] T. van Mourik, G.J. Harris, O.L. Polyansky, J. Tennyson, A.G. Császár, P.J. Knowles, *J. Chem. Phys.* 115 (2001) 3706–3718.
- [56] G.J. Harris, O.L. Polyansky, J. Tennyson, *Spectrochimica Acta A* 58 (2002) 673–690.
- [57] S.V. Shirin, O.L. Polyansky, N.F. Zobov, P. Barletta, J. Tennyson, *J. Chem. Phys.* 118 (2003) 2124–2129.
- [58] A.E. Lynas-Gray et al., *J. Chem. Phys.*, in preparation.
- [59] O.L. Polyansky, R. Prosmi, W. Klopper, J. Tennyson, *Mol. Phys.* 98 (2000) 261–273.
- [60] R. Röhse, W. Kutzelnigg, R. Jaquet, W. Klopper, *J. Chem. Phys.* 101 (3) (1994) 2231–2243.