

CP-668

08 May 2014

LEVEL 8.2
A Computer Program
for Solving the Radial Schrödinger Equation
for Bound and Quasibound Levels

Robert J. Le Roy

Guelph-Waterloo Centre for Graduate Work in Chemistry
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
Electronic mail: leroy@UWaterloo.ca

University of Waterloo

Chemical Physics Research Report

LEVEL 8.2
A Computer Program
for Solving the Radial Schrödinger Equation
for Bound and Quasibound Levels

Robert J. Le Roy

Guelph-Waterloo Centre for Graduate Work in Chemistry
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
Electronic mail: leroy@UWaterloo.ca

This manual describes program **LEVEL**, which can solve the radial or one-dimensional Schrödinger equation for bound and/or quasibound levels of any smooth single or double-minimum potential, and calculate inertial rotation and centrifugal distortion constants, expectation values, and/or Franck-Condon factors and other off-diagonal matrix elements, either for levels of a single potential or between levels of two different potentials.

©Robert J. Le Roy, 2014

1 Introduction

Determining the number, energies and properties of vibration-rotational levels of a given one-dimensional or effective radial potential, and calculating matrix elements and transition intensities coupling levels of a single potential or levels of two separate potentials, are ubiquitous problems in chemical physics. The present report describes a robust and flexible computer program for performing such calculations. The original version of this program was based of the famous Franck-Condon intensity program of R.N. Zare [1, 2, 3], but the present version is considerably modified, and incorporates several unique features. In particular:

- (i) it will automatically locate and calculate the widths of quasibound (or orbiting resonance, or tunneling predissociation) levels;
- (ii) it can calculate diatomic molecule inertial rotation and centrifugal distortion constants for levels of a given potential;
- (iii) it can readily locate levels with dominant wave function amplitude over either well of an asymmetric double minimum potential;
- (iv) it can automatically locate and calculate expectation values for all vibration-rotation levels of any well-behaved single-minimum, “shelf state” or double minimum potential;
- (v) as an extension of (iv), it can automatically generate (for example) Franck-Condon factors and the radiative lifetimes for all possible discrete transitions allowed by specified (in the input data file) rotational selection rules between the levels of two different potentials, or among the levels of a single potential. While the present version generates the specified matrix elements and calculates Einstein \mathcal{A} coefficients using the Hönl-London factors for the case of *singlet-singlet* electronic transitions, it may be generalized to treat other cases.

In the following, Section 2 presents the basic equation being solved, describes how the program functions, and outlines some of its options. Section 3 then states the input/output conventions, indicates the units assumed for the physical parameters of interest, and presents a shell to facilitate running the program on a UNIX or Linux system. The program’s operation is controlled by the contents of a data file which is read (on channel-5) during execution. The structure of this data file and the significance of the various read-in parameters are described in Section 4. Section 5 then describes the most significant differences between the current (8.0) and earlier [4] versions of this program. Finally, the Appendices outlines the structure of the program and the roles of its various subroutines, and present listings of illustrative sample data input files and of the resulting output.

The current version of the (extensively commented) source code for **LEVEL** and a ‘pdf’ file of this manual may be obtained from the www site <http://leroy.uwaterloo.ca/programs/>. While, there are no charges associated with distribution or use of this program, its use should be acknowledged in publications through reference to this report [5]. Users are also requested *not* to distribute the program themselves, but to refer other prospective users to the above web site or to the author. The version described herein includes corrections and enhancements incorporated up to 01 February 2014. Individuals currently utilizing older versions of this code [4] will likely find it desirable to obtain the current version since it has some corrections and additional functionality. I would also appreciate having users inform me of any apparent errors or instabilities in the code, or of additional features which might appear desirable for future versions.

2 Outline of Program Operation and Options

2.1 Solving the Radial Schrödinger Equation

The core of the program is concerned with determination of the discrete eigenvalues and eigenfunctions of the radial or (effective) one-dimensional Schrödinger equation

$$-\frac{\hbar^2}{2\mu} \frac{d^2 \Psi_{v,J}(r)}{dr^2} + V_J(r) \Psi_{v,J}(r) = E_{v,J} \Psi_{v,J}(r) \quad (1)$$

in which μ is the effective or reduced mass of the system, J the rotational quantum number, r the internuclear distance, and the effective one-dimensional potential $V_J(r)$ is a sum of the rotationless (electronic) potential $V(r)$ plus a centrifugal term. For the normal problem of a diatomic molecule rotating in three dimensions, this centrifugal potential has the form $[J(J+1) - \Omega^2] \hbar^2 / 2\mu r^2$, where $\Omega = \text{OMEGA}$ is the projection of the electronic angular momentum onto the internuclear axis. However, for the special case of a diatom rotating in two dimensions, a case invoked by setting the read-in parameter $\text{OMEGA} > 99$ (see the discussion of data input statement #5 in § 4), this term becomes $[J^2 - 1/4] \hbar^2 / 2\mu r^2$. Moreover, in order to include the case of molecular ions, the program defines the reduced mass appearing in Eq. (1) as Watson's "charge-modified reduced mass" [6], $\mu = \mu_W = (M_A M_B) / (M_A + M_B - m_e Q)$, where M_A and M_B are the atomic masses of the two atoms, m_e is the electron mass, and $Q = \text{CHARGE}$ (see input READ statement #1) is the \pm integer net charge on the molecule (ion). Of course this equation reduces to that for the reduced mass of a generic 2-body system if the molecule has no charge ($Q = 0$).

The core of the calculation is the solution of Eq. (1) to determine the eigenvalues $E_{v,J}$ and eigenfunctions $\Psi_{v,J}(r)$ of the potential $V_J(r)$. This is done in subroutine `SCHRO`, which is based on the famous Cooley-Cashion-Zare routines `SCHR` [1, 2, 3, 7, 8], but incorporates the ability to automatically locate and calculate the widths of "quasibound" or tunneling-predissociation levels [9, 10, 11, 12]. These are metastable levels which lie above the dissociation limit, but whose dissociation is inhibited by a potential energy barrier.

The accuracy of the eigenvalues and eigenfunctions obtained is largely determined by the size of the (fixed) radial mesh `RH` (`READ` #4 of the input data file) used in the numerical integration of Eq. (1). For potentials that are not too steep or too sharply curved, adequate accuracy is usually obtained using an `RH` value which yields a minimum of 30 – 50 mesh points between adjacent wavefunction nodes in the classically allowed region. An appropriate mesh size may be estimated using the 'particle-in-a-box'-type expression

$$\text{RH} = \pi / \left(\text{NPN} \times [\mu \times \max\{E - V(r)\} / 16.857\,629\,20]^{1/2} \right) \quad (2)$$

in which `NPN` is the selected minimum number of mesh points per wavefunction node (say 30), $\max\{E - V(r)\}$ is the maximum of the local kinetic energy (in cm^{-1}) for the levels under consideration (in general it is \lesssim the potential well depth), and the numerical factor is identified below in Section 3. A value of `NPN` which is too small yields results which may not have the desired accuracy, while too large a value may require excessive computational effort or cause array dimensions to be exceeded. Thus, while Eq. (2) is a useful guide, a careful user *should always* perform calculations using different `RH` values in order to ensure that the results calculated achieve the accuracy desired for their particular application.

The numerical integration of Eq. (1) is performed on the range from `RMIN` to `RMAX` (see `READ` statement #4) using the Numerov algorithm [7, 13]. To initiate this integration, it is necessary to specify initial values of the wave function at two adjacent mesh points at each end of the range. For truly bound states, the wave function at the outer end of the range, $r = \text{RMAX}$, is initialized at an arbitrary value (unity), while its value at the adjacent inner mesh point is defined using the first-order semiclassical or WKB wavefunction [14]

$$\Psi_{v,J}(r) \propto [V_J(r) - E_{v,J}]^{-1/4} \exp \left(-\sqrt{2\mu/\hbar^2} \int^r [V_J(r') - E_{v,J}]^{1/2} dr' \right) \quad (3)$$

At short range, most realistic intermolecular potentials grow very steeply, causing the wavefunctions to die off extremely rapidly with decreasing r . As a result, the wave function at the inner end of the range of integration is normally initialized by placing a node at the lower bound of this range, the read-in distance $RMIN$. This is effected by setting $\Psi_{v,J}(r = RMIN) = 0$ and giving $\Psi_{v,J}(r = RMIN + RH)$ an arbitrary (non-zero) value. This is the normal case for a diatomic molecule problem. Note that one should normally set $RMIN > 0$, as the centrifugal contribution to the potential becomes singular at $r = 0$.

A special treatment of the inner boundary condition may be implemented if one is searching for eigenfunctions of a symmetric potential whose midpoint is located at $RMIN$. For asymmetric levels that would have a node at $RMIN$, the normal treatment described above will suffice. However, another approach must be implemented for symmetric levels of such a potential whose wave functions would have zero slope at $RMIN$. This option is built into subroutine `SCHRQ`, and is invoked by setting the control parameter `INNOD1` ≤ 0 . However, since this is an unusual case, varying this parameter is not one of the normal options of the current version of the main program, and a user who wishes to deal with this case may chose to either add parameter `INNOD1` to `READ #17` (line #367 of the `MAIN` program), or to recompile the code with the value of `INNOD1` and `INNOD2` defined on lines #365 & 366 of the code reset to 0.

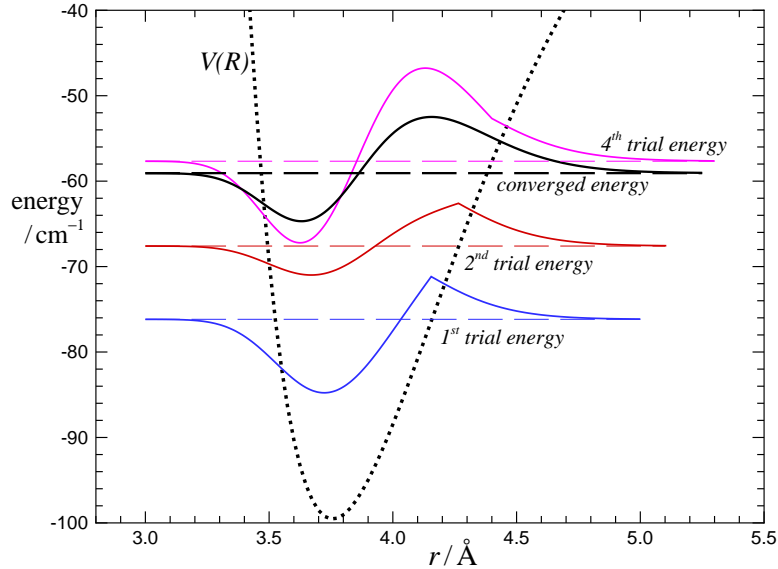


Figure 1. Illustration of the Löwdin/Cooley eigenvalue convergence procedure[15, 7] showing wavefunctions at the first four trial energies that converge on the $v=1$ eigenvalue of a model potential.

If desired, a hard wall outer boundary condition may be imposed by setting the input integer $IV(i)$ (see `READ #18`), which would otherwise represent the vibrational quantum number, equal to a large negative number whose absolute value would specify the radial mesh point where this wall would be placed. In particular, setting $IV(i) < -10$ causes a hard wall (wave function node) to be placed at mesh point number $|IV(i)|$ for level- i .

The Cooley procedure for finding an eigenvalue of Eq. (1) [7, 8] is illustrated by Fig. 1. For any given trial energy the numerical integration proceeds inward from $RMAX$ and outward from $RMIN$ until the two solution segments meet at a chosen matching point r_x . The discontinuity in the slopes of the inward and outward trial functions at r_x is then used to estimate the energy correction required to converge on the eigenvalue closest to the given trial energy [15], and this process is repeated until the energy improvement is smaller than the chosen convergence criterion (parameter `EPS` of `READ #4`). This procedure usually converges very rapidly, and for a single-minimum potential it is insensitive to the choice of the matching point r_x , as long as it lies in the classically-allowed region where the wavefunction amplitude is relatively large. However, to ensure high accuracy of calculated expectation values or matrix elements, `EPS` should

usually be set 2 orders of magnitude smaller than the the actual eigenvalue precision required.

For an asymmetric double-well potential, wavefunctions usually have amplitudes of very different magnitude over the two wells, and the eigenvalue correction algorithm [15] used by SCHRQ tends to become unstable if the matching distance r_x lies in the well where the wavefunction has very small amplitude. As a result, it is usually necessary to require r_x to lie in the well where its amplitude is the largest. In the current version of the program, this choice is set by the internal control parameter `INNER` (= `SINNER`), which in turn is set to the appropriate value by the automatic vibrational level-finder subroutine `ALF`. As a result, calculations involving vibrational levels of a double well or “shelf-state” potential are (usually) performed just as routinely (for the user) as those for a normal single-well potential.

In general, the outward and inward numerical integration must start at distances `RMIN` and `RMAX` (input via `READ #4`), respectively, that lie sufficiently far into the classically-forbidden regions (where $V_J(r) > E_{v,J}$) that the wavefunction has decayed by several orders of magnitude relative to its amplitude in the classically-allowed region. The present version of the code prints warning messages if this decay is not by a factor of at least 10^{-9} ; if such warnings are printed, a smaller `RMIN` or larger `RMAX` value will be needed to ensure the desired accuracy for such cases. On the other hand, if `RMIN` or `RMAX` lie sufficiently far into the classically-forbidden regions that $[V_J(r) - E]$ becomes extremely large, the integration algorithm can become numerically unstable for the given mesh size. For realistic diatomic molecule potential curves, this situation is only likely to occur near `RMIN`. If it does, a warning message is printed and the beginning of the integration range is automatically shifted outward until the problem disappears. However, use of a slightly larger value of `RMIN` will cause these warning messages to disappear and (marginally) reduce the computational effort. For most diatomic molecules, a reasonable value of `RMIN` is ca. 0.6 – 0.8 times the smallest inner turning point encountered in the calculation, but for hydrides or other species of low reduced mass, even smaller values may be necessary.

The program internally defines the upper bound on the range of numerical integration `RMAX` as the smaller of the value read in (`READ #4`) and the largest distance consistent with the specified mesh and the internally-defined (see § 3) potential and distance array dimension `NDIMR`. As with `RMIN`, the choice of `RMAX` is not critical, as long as (for truly bound states) the wave function has decayed to an amplitude much smaller than that in the classically allowed region, and the same amplitude decay test of 10^{-9} is used for it. However, due to the anharmonicity of typical molecular potential curves, the requisite values of `RMAX` are much larger for highly excited vibrational levels than for those lying near the potential minimum. In order to reduce computational effort, an integration range upper bound $r_{\text{end}}(v, J)$ is therefore determined for each level using the semiclassical result of Eq. (3), which shows that the wavefunction dies off exponentially in the classically forbidden region with an exponent of

$$- \sqrt{2\mu/\hbar^2} \int_{r_2(v,J)}^{r_{\text{end}}(v,J)} [V_J(r) - E_{v,J}]^{1/2} dr \quad (4)$$

where the turning point $r_2(v, J)$ marks the outer end of the classically accessible region at this energy $E_{v,J}$. For each level it considers, SCHRQ first locates $r_2(v, J)$, and then determines a value of $r_{\text{end}}(v, J)$ which is sufficiently large to ensure that this starting amplitude is smaller than that in the classically-allowed region by a factor of at least 10^{-9} . In calculations for levels spanning a wide range of energies, the program’s use of this procedure can reduce the overall computation time by a factor of two or more.

2.2 Locating Quasibound Levels and Determining Their Widths

Quasibound or orbiting resonance or tunneling-predissociation levels are metastable eigenstates of Eq. (1) that lie at energies above the potential asymptote $V_J(r = \infty)$, but below a maximum in the outer part of the potential. Although they are part of the continuum of states with $E > V_J(r \rightarrow \infty)$, they are distinct in that their wavefunction amplitude at small distances, inside the potential barrier, is very much larger

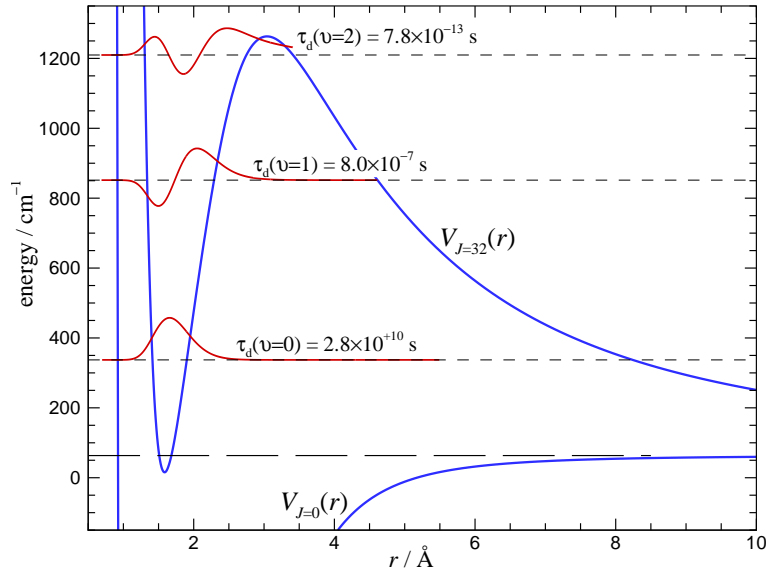


Figure 2. Quasibound level wavefunctions (red curves) and the associated tunneling lifetimes for the three vibrational levels supported by the $J = 32$ centrifugally-distorted potential (upper blue curve) for the ($^1\Pi$) state of CH^+ (adapted from Fig. 2 of Ref. [16]).

that those for neighbouring continuum states, so that they give rise to distinct sharp spectroscopic lines. The nature of the ‘inside-the-barrier’ portion of the wavefunctions for such states is illustrated by Fig. 2.

Following Refs. [9] and [10], program **LEVEL** locates such levels by combining the standard $\Psi_{v,J}(r = \text{RMIN}) = 0$ inner boundary condition with an outer boundary condition defined by the slope of the inward-increasing Airy function at the third (outermost) turning point. This procedure is the most accurate and efficient method of locating quasibound or tunneling predissociation levels proposed to date [9, 10, 11, 17, 18]. It is virtually exact for narrow (long-lived) states, while for the very broadest levels lying marginally below barrier maxima (such as $v = 2$ in Fig. 2), its differences with alternative methods is at most a small fraction (say, $1/5$) of the level width (FWHM). More accurate predictions for such short-lived states would require a detailed simulation of the actual process by which they are observed, since different methods of observing a given quasibound level may yield apparent energies differing by a fraction of the level width. For example, if such levels are being observed spectroscopically, the peaks in the actual bound \rightarrow continuum spectrum would be calculated using a photodissociation simulation code [19, 20].

Calculation of the level width or tunneling lifetime of a quasibound state by program **LEVEL** is based on Eq. (4.5) of Connor and Smith [11]; a more transparent description of this procedure may be found in §II.B of Ref. [12]. This is a uniform semiclassical procedure in which the predissociation rate may be thought of as being the product of the probability of tunneling past the barrier at the specified energy, times the vibrational frequency (inverse of the vibrational period) for the system trapped in the well behind the barrier. The actual calculation requires the evaluation of an integral of the type seen in Eq. (4) across the interval between the two outermost classical turning points (i.e., with the upper bound $r_{\text{end}}(v, J)$ replaced by the outermost turning point $r_3(v, J)$), and of an analogous integral across the classically allowed interval between the two inner turning points:

$$\sqrt{2\mu/\hbar^2} \int_{r_1(v,J)}^{r_2(v,J)} [E_{v,J} - V_J(r)]^{-1/2} dr \quad (5)$$

together with (the energy derivative of) a phase correction factor which takes account of the proximity to the barrier maximum [10, 11]. This procedure yields widths which are expected to be very reliable, particularly for narrow (long-lived) levels, but may have uncertainties of up to *ca.* 10% or more for the

very broadest levels [10, 11]. To obtain more accurate results for such levels would again require one to perform a direct simulations of the process by which they are observed. The results in Fig. 2, where the tunneling lifetimes range from *ca.* 1 fs to 900 years, illustrates the extreme range of the tunneling lifetimes that may arise.

On a practical note, if the end of the numerical integration range **RMAX** is smaller than the outermost turning point $r_3(v, J)$ of the metastable level of interest, the program attempts to generate a reasonable estimate of the width by completing the quadrature of Eq. (4) over the barrier analytically while approximating the potential on the remainder of the interval by a centrifugal-type term C_2/r^2 attached to the potential function at **RMAX**. If this approximation is invoked, warning messages are written to the main channel-6 output file (e.g., see the output for Case 3 in Appendix G).

2.3 Calculating Diatomic Molecule Centrifugal Distortion Constants

The rotational sublevels of a given vibrational level of a molecule are conventionally represented by the power series

$$\begin{aligned} E_{v,J} &= G(v) + B_v[J(J+1)] - D_v[J(J+1)]^2 + H_v[J(J+1)]^3 + \dots \\ &= \sum_{m=0} K_m(v) [J(J+1)]^m \end{aligned} \quad (6)$$

If desired, the program will calculate values of the inertial rotation constant $B_v = (\hbar^2/2\mu)\langle v, J|1/r^2|v, J\rangle$ and of the first six centrifugal distortion constants associated with this expansion ($-D_v$, H_v , L_v , M_v , N_v and O_v). These constants have their normal significance [21] when calculated for levels of a rotationless ($J = 0$) potential, and are simply related to derivatives of the energy with respect to $[J(J+1)]$ when calculated for vibration-rotation levels with $J > 0$. Calculation of these constants is invoked by setting input parameter **LCDC** > 0 (see **READ #17**), and are performed using a subroutine based on Tellinghuisen's reformulation [22] of the exact quantum mechanical method of Hutson [23], which has been extended to higher order to allow the calculation of N_v and O_v . To ensure stable, fully converged calculations, it is often necessary to make the eigenvalue convergence parameter **EPS** of **READ #4** quite small (e.g., $\lesssim 10^{-6} \text{ cm}^{-1}$).

2.4 Calculating Expectation Values, Matrix Elements, and Einstein A Coefficients

If desired, the program will calculate expectation values or matrix elements of a function $M(r)$ which may be defined either by interpolating over an array of input values, by a user-defined analytic function, or as a power series in a user-specified radial variable **RFN**(r):

$$M(r) = \sum_{i=0}^{\text{MORDR}} \text{DM}(i) \times \text{RFN}(r)^i \quad (7)$$

where parameters **MORDR**, **IRFN** and **DREF** defining the extent of the power series and the nature of the radial variable **RFN**(r) are input via **READ #19**, and the power series coefficients **DM**(i) are input via **READ #20**. In this last option, the radial variable **RFN**(r) is defined by the choice of input variable **IRFN** in the range $-4 \leq \text{IRFN} \leq 9$ (see comments for **READ #19** in § 4), while setting **IRFN** ≥ 10 causes $M(r)$ to be defined by interpolating over and extrapolating beyond a set of read-in numerical values, and setting **IRFN** ≤ -10 causes $M(r)$ to be a user-defined analytic radial function. For this last case, code for calculating the desired function should be inserted in the program in the manner illustrates by the example in lines #508–529 of the main program. If $M(r)$ is to be defined by interpolating over an array of read-in points, the necessary information is input via **READS #21 – 24**.

The conventional Franck-Condon factor $\text{FCF} = |\langle \Psi_{v',J'} | \Psi_{v'',J''} \rangle|^2$ is the square of the matrix element of the zeroth power of **RFN**(r), and will be generated whenever any off-diagonal matrix elements are calculated

(i.e., whenever input parameter $\text{LXPCT} \geq 3$). In this case, the program also assumes that $M(r)$ is the transition dipole function (in debye), and uses its matrix element to calculate the Einstein \mathcal{A} coefficient coupling the two levels in question. For cases in which a pointwise ($\text{IRFN} \geq 10$) or analytic user-defined ($\text{IRFN} \leq -10$) matrix element argument function $M(r)$ is chosen, MORDR and DREF are dummy variables, and no $\text{DM}(i)$ coefficients are read in. Note: to calculate *only* Franck-Condon factors, one should set $\text{MORDR} < 0$, in which case IRFN and DREF are dummy variables and no $\text{DM}(i)$ values are read in.

The Einstein \mathcal{A} coefficient for the rate of spontaneous emission from initial-state level (v', J') into final-state level (v'', J'') is defined by the expression [21, 24]

$$\mathcal{A} = 3.1361891 \times 10^{-7} \frac{S(J', J'')}{2J' + 1} \nu^3 |\langle \Psi_{v', J'} | M(r) | \Psi_{v'', J''} \rangle|^2 \quad (8)$$

Here: \mathcal{A} has units s^{-1} , $M(r)$ is the dipole moment (or transition dipole) function in units debye, ν the emission frequency in cm^{-1} , $S(J', J'')$ the Hönl-London rotational intensity factor and $\Psi_{v', J'}$ and $\Psi_{v'', J''}$ are the unit normalized initial and final state radial wave functions. The present version of the code incorporates $S(J', J'')$ expressions for *singlet* \leftrightarrow *singlet* transitions obeying the parity selection rule, with $\Delta\Lambda = 0$ or ± 1 . However, while versions of this code prior to 7.7 used the $S(J', J'')$ expressions of Herzberg [21], subsequent versions use the revised Hönl-London factors recommended by Hansson and Watson [25], which for $\Pi - \Sigma$ or $\Sigma - \Pi$ transitions are a factor of four larger than those reported by Herzberg [21] (see also Bernath [24]). To generalize these selection rules for other cases, a user will need to modify lines #60 – 106 of subroutine MATXEL.

2.5 Defining the Rotationless Potential $V(r)$

The potential function package which reads required input and returns the potential array and associated parameters is controlled by subroutine PREPOT. It uses subroutine package GENINT for interpolation/extrapolation over a set of read-in turning points, and subroutine POTGEN for generating analytic potential functions. Values of the necessary input parameters enter via READ statements #5–16; for the 2-state case invoked by inputting $\text{NUMPOT} = 2$, this block of input statements is read twice.

One may choose to define a potential either by a set of NTP turning points $\{\text{XI}(i), \text{YI}(i)\}$ input via READ #8, or (by setting $\text{NTP} \leq 0$) by an analytic function. In the former case, interpolation over the read-in turning points to produce the array with mesh size RH required for the numerical integration of Eq. (1) is performed in a manner specified by the input parameter NUSE. For $\text{NUSE} > 0$ this involves the use of piecewise NUSE-point polynomials (typically $\text{NUSE} = 8$ or 10), while for $\text{NUSE} \leq 0$ the interpolation uses a cubic spline function. If the range of numerical integration $[\text{RMAX}, \text{RMIN}]$ extends beyond that of the input turning points, appropriate extrapolation procedures are invoked. In particular, at distances smaller than the second of the read-in turning points $\text{XI}(2)$, the potential is extrapolated inward with an exponential function fitted to the first three turning points. Similarly, if $\text{RMAX} > \text{XI}(\text{NTP} - 1)$ the potential for $r > \text{XI}(\text{NTP} - 1)$ is extrapolated outward either as an exponential-type function or as a (sum of) inverse-power terms, as specified by parameters ILR, NCN and CNN of READ #6 (see §4).

To define the potential by an analytic function, rather than by an array of points, the integer input parameter NTP of READ #5 should be set ≤ 0 . The program then *skips* READS #6–8 and proceeds instead to #9–16 (see §4), where it reads values of the parameters defining the chosen analytic potential. The present version of the code allows for the following seven families of analytic potential energy functions.

(i) The Lennard-Jones(m, n) potential:

$$V(r) = \mathfrak{D}_e \left[\left(\frac{n}{m-n} \right) \left(\frac{r_e}{r} \right)^m - \left(\frac{m}{m-n} \right) \left(\frac{r_e}{r} \right)^n \right] \quad (9)$$

in which \mathfrak{D}_e is the well depth and r_e the equilibrium internuclear distance.

(ii) **Polynomial potentials** based on Seto’s modification [26] of Šurkus’ GPEF potential form [27], which includes the familiar Dunham [28], Simons-Parr-Finlan [29], and Ogilvie–Tipping [30] expansions as special cases invoked by particular definitions of the GPEF expansion variable (see discussion of READ #9 in § 4):

$$z = \frac{r^p - r_e^p}{a_S r^p + b_S r_e^p} \quad (10)$$

$$V(r) = \beta_0 z^2 \left(1 + \sum_{i=1}^{N_\beta} \beta_i z^i \right) \quad (11)$$

(iii) **The “Extended Morse Oscillator” or EMO function** [26, 31, 32, 33]:

$$V(r) = \mathfrak{D}_e \left[1 - e^{\beta(r) \cdot (r - r_e)} \right]^2, \quad (12)$$

in which $\beta(r)$ is a simple polynomial $\beta(r) = \sum_{i=0}^{N_\beta} \beta_i [y_q(r)]^i$ in the generalized Šurkus variable [27]

$$y_q(r) = y_q(r; r_{\text{ref}}) = y_q^{\text{ref}}(r) = \frac{r^q - r_{\text{ref}}^q}{r^q + r_{\text{ref}}^q}, \quad (13)$$

in which q is a selected small positive integer. With this potential form, the reference distance r_{ref} has most commonly been set equal to the equilibrium distance r_e , but there are substantial advantages associated with allowing it to have larger values [34, 35]. Truncating the exponent expansion at the constant term yields the familiar simple Morse potential.

Another Morse-type function allowed by POTGEN is Hua Wei’s 4-parameter potential [36]

$$V(r) = \mathfrak{D}_e \left([1 - e^{-b(r - r_e)}] / [1 - C e^{-b(r - r_e)}] \right)^2. \quad (14)$$

(iv) **The “Morse/Long-Range” or MLR function**, [37, 34, 35] whose initial version was called the “Morse/Lennard-Jones” (MLJ) [38, 33] potential, in which a flexible Morse-like function for the potential well incorporates an expression for the long-range interaction energy $u_{\text{LR}}(r)$ that consists of one (for MLJ) or more (for MLR) of the leading theoretically predicted inverse-power terms:

$$V(r) = \mathfrak{D}_e \left[1 - \frac{u_{\text{LR}}(r)}{u_{\text{LR}}(r_e)} e^{-\beta(r) \cdot y_p^{\text{eq}}(r)} \right]^2. \quad (15)$$

Here, the explicit radial variable in the exponent is a special case of the generalized Šurkus-type variable of Eq. (13) with r_{ref} fixed at the equilibrium distance r_e :

$$y_p^{\text{eq}}(r) = y_p(r; r_e) \equiv \frac{r^p - r_e^p}{r^p + r_e^p} \quad (16)$$

and the exponent coefficient function $\beta(r)$ is (normally) written as a constrained polynomial expansion defined in terms of *two* Šurkus-type variables [27] that both have the form of Eq. (13), but are defined by different integers p and q [35]:

$$\beta(r) = y_p^{\text{ref}}(r) \beta_\infty + \left[1 - y_p^{\text{ref}}(r) \right] \sum_{i=0}^{N_\beta} \beta_i [y_q^{\text{ref}}(r)]^i. \quad (17)$$

As an alternative to Eq. (17), $\beta(r)$ may be represented as a natural cubic spline in the independent variable $y_q^{\text{ref}}(r)$, passing through a specified set of input points, as described by Tao *et al.* [39].

The asymptotic behaviour of the three radial variables $y_p^{\text{eq}}(r)$, $y_p^{\text{ref}}(r)$, and $y_q^{\text{eq}}(r)$ means that

$$\beta_\infty \equiv \lim_{r \rightarrow \infty} \beta(r) = \ln \left\{ \frac{2\mathfrak{D}_e}{u_{\text{LR}}(r_e)} \right\}, \quad (18)$$

and that as $r \rightarrow \infty$, the overall potential takes on the form $V(r) \simeq \mathfrak{D}_e - u_{\text{LR}}(r)$. The long-range interaction energy function in Eq. (15) may be written either as an ‘undamped’ [40, 37]

$$u_{\text{LR}}(r) = \frac{C_{m_1}}{r^{m_1}} + \frac{C_{m_2}}{r^{m_2}} + \dots \quad (19)$$

or as a ‘damped’ [35, 41, 42]

$$u_{\text{LR}}(r) = \sum_i D_{m_i}(r) \frac{C_{m_i}}{r^{m_i}} \quad (20)$$

sum of inverse-power terms. There are two choices for the damping functions $D_m(r)$ in Eq. (20): one is based on a form proposed by Douketis *et al.* [43]:

$$D_m^{\text{DS}(s)}(r) = \left[1 - \exp \left\{ -\frac{b^{\text{ds}}(s) \cdot (\rho r)}{m} - \frac{c^{\text{ds}}(s) \cdot (\rho r)^2}{\sqrt{m}} \right\} \right]^{m+s}, \quad (21)$$

and one is based on a form proposed by Tang and Toennies [44]

$$D_m^{\text{TT}(s)}(r) = 1 - e^{-b^{\text{tt}}(s) \cdot \rho r} \sum_{k=0}^{m-1+s} \frac{(b^{\text{tt}}(s) \cdot \rho r)^k}{k!} \quad (22)$$

in both of which ρ is a system-dependent scaling parameter [43, 35] while $b^{\text{ds}}(s)$, $c^{\text{ds}}(s)$ and $b^{\text{tt}}(s)$ are known [35] system-independent constants.

Both of these generalized [35] damping function forms have the property that at very small r

$$D_m^s(r)/r^m \simeq r^s \quad (23)$$

for all values of m and s . The original Tang-Toennies model corresponded to the case $s = +1$, and that presents no problem if $u_{\text{LR}}(r)$ is an additive contribution to the potential (see cases below). However, it cannot be used in the MLR potential form which requires $s \leq 0$, since otherwise the repulsive potential wall would turn over and approach zero as $r \rightarrow 0$. The original Douketis *et al.* [43] model corresponds to the case $s = 0$, and while that presents no formal problems for either MLR potentials or the functions described below, the results of Ref. [35] suggest that $s = -1$ or $s = -1/2$ would be a better choice, since they lead to more physically realistic behaviour at very small r .

The long-range interaction energy is not restricted to being one of the sums of Eqs. (19) or (20), but could take on any form predicted by theory. In particular, in the present version of **LEVEL** it may be represented as one of the roots of a 2×2 or 3×3 diagonalization arising from the type of two-state [34, 45] or three-state [46, 47] coupling encountered for alkali dimers at the first $nS + nP$ asymptote.

Details of these options are presented in § 4.

(v) The “Double-Exponential Long-Range” or DELR potential [12].

$$V(r) = A e^{-2\beta(r) \cdot (r-r_e)} - B e^{-\beta(r) \cdot (r-r_e)} + u_{\text{LR}}(r) \quad (24)$$

in which $\beta(r) = \sum_{i=0}^{N_\beta} \beta_i [y_q(r)]^i$ is a simple power series in the Šurkus variable $y_q^{\text{ref}}(r)$ of Eq. (13) and $u_{\text{LR}}(r)$ is a function chosen to represent the long-range region that may have the form of one of Eqs. (19) or (20). The factors A and B are defined in terms of the well depth and the properties of $V_{\text{LR}}(r)$ at r_e :

$$A = \mathfrak{D}_e + V_{\text{LR}}(r_e) + V'_{\text{LR}}(r_e)/\beta_0 \quad (25)$$

$$B = 2\mathfrak{D}_e + 2V_{\text{LR}}(r_e) + V'_{\text{LR}}(r_e)/\beta_0 \quad (26)$$

in which $V'_{\text{LR}}(r_e) \equiv dV_{\text{LR}}(r)/dr|_{r=r_e}$.

(vi) *The generalized HFD function* [48, 49] defined in terms of the reduced distance variable $x = r/r_e$:

$$V(r) = A x^\gamma e^{-\beta_1 x - \beta_2 x^2} - D(x) \sum_m C_m / r^m \quad (27)$$

in which $D(x) = \exp \{ -\alpha_1 [\alpha_2/x - 1]^{\alpha_3} \}$ is a global damping function that cuts off the inverse-power terms at short distances.

(vii) *The “X-Expansion” or “Hannover Polynomial Potential” (HPP) form* that was introduced by Tiemann and co-workers [50] represents the main part of the potential by a GPEF-like power series:

$$V(r) = \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + \beta_3 \xi^3 + \beta_4 \xi^4 + \dots \quad (28)$$

in which $\xi \equiv \xi(r, b, r_m) = (r - r_m)/(r + b r_m)$. The possible presence of a non-zero linear term ($\beta_1 \neq 0$) in Eq. (28) means that when $\beta_1 \neq 0$ the parameter r_m (which is input as $r_e = \text{REQ}$) only approximately corresponds to the equilibrium distance. At a specified small distance, this power series is smoothly joined to an exponential, and beyond a specified large distance, it is attached to an inverse-power sum with the form of Eq. (19).

Note that except for the simple polynomial potentials of *type (ii)*, all of these analytic potentials are defined relative to the absolute energy at the asymptote, which is specified by input parameter VLIM (see READ #5). For the HPP potential this is effected by setting $a_0 = \text{VLIM} - \mathfrak{D}_e$. However, GPEF-type potentials of type (ii) are defined relative to their energy at the potential minimum.

A user may readily introduce their own analytic potential function form by simply replacing subroutine POTGEN with their own potential routine and making input parameter NTP of READ #5 a negative integer. To retain consistency with the rest of the present code, such a user-prepared POTGEN subroutine should have the argument list:

POTGEN(LNPT, NPP, IAN1, IAN2, IMN1, IMN2, VLIM, R, RM2, VV, NCN, CNN)

The first argument, parameter LNPT, is an integer which in program **LEVEL** is fixed as LNPT = 1.* The other input quantities are the integer NPP specifying the size of the array of radial distances $R(i)$ (in Å) at which potential values are to be generated, the squared inverse distance array $\text{RM2}(i) = 1/R(i)^2$, the absolute energy VLIM (in cm^{-1}) at the potential asymptote, and integers giving the atomic numbers (IAN1 & IAN2) and mass numbers (IMN1 & IMN2) of that particular isotopologue (required for calculating BOB terms, see below). The subroutine is expected to return the desired NPP-point array of potential function values $\text{VV}(i)$ (in units cm^{-1}), as well as the integer NCN and real positive coefficient CNN. Under the option in which the program automatically searches for many or all vibrational levels of a given potential (when input parameter NLEV1 is large and negative, see READ #17), the limiting long-range potential is assumed to have the form $V(r) \simeq \mathfrak{D} - \text{CNN}/r^{\text{NCN}}$, and the parameters NCN and CNN returned from POTGEN may be used in a near-dissociation theory [51, 52, 53] algorithm to estimate the number and energies of missing levels. If the user’s analytic potential has a barrier maximum or dies off exponentially rather than as an inverse power, NCN should be set at some large integer value (e.g., NCN = 99).

2.6 Born-Oppenheimer Breakdown Radial Strength functions

In recent years, it has become increasingly common for combined-isotopologue spectroscopic data analyses to require the inclusion of atomic-mass-dependent Born-Oppenheimer breakdown (BOB) corrections to the rotationless and centrifugal potential energy functions. **LEVEL** will include such terms if the value

* While not used in **LEVEL**, LNPT is retained in the calling sequences to facilitate the use of this subroutine package in other programs.

of input parameter $\text{IBOB} > 0$. These (optional) functions are defined as in Ref. [54]. In particular, for each atom $a = A$ or B , the additive correction to the potential energy function is defined as a constrained polynomial in the Šurkus variable of Eq. (13)

$$\left(\frac{M_a - M_a^{\text{ref}}}{M_a} \right) \left(y_{p_{\text{ad}}}^{\text{eq}}(r) u_{\infty}^a + [1 - y_{p_{\text{ad}}}^{\text{eq}}(r)] \sum_{j=0}^{N_{\text{ad}}^a} u_j^a [y_{q_{\text{ad}}}^{\text{eq}}(r)]^j \right) \quad (29)$$

in which M_a is the mass of the particular isotope of atom- a , M_a^{ref} the mass of the chosen reference isotope of that species [54], $y_{p_{\text{ad}}}(r)$ and $y_{q_{\text{ad}}}(r)$ have the form of Eq. (13), and the integers p_{ad} and q_{ad} are chosen as discussed in Refs. [54] and [34].

The centrifugal BOB correction is a multiplicative factor $[1 + g_A(r) + g_B(r)]$ applied to the centrifugal contribution to the overall potential function $V_J(r)$ of Eq. (1), in which the terms associated with the two atoms $a = A$ or B have the same type of radial form as the “adiabatic” potential function corrections of Eq. (29):

$$g_a(r) = \frac{M_a^{\text{ref}}}{M_a} \left(y_{q_{\text{na}}}^{\text{eq}}(r) t_{\infty}^a + [1 - y_{q_{\text{na}}}^{\text{eq}}(r)] \sum_{j=0}^{N_{\text{na}}^a} t_j^a [y_{q_{\text{na}}}^{\text{eq}}(r)]^j \right) \quad (30)$$

Note, however, that this expression has no separate integer p_{na} , since we know of no theoretical prediction regarding the limiting long-range behaviour of these functions. However, the algebraic structure of Eqs. (17) and (29) is retained to allow for the case of molecular ions, for which t_{∞}^a would be non-zero [54].

3 Units, Physical Constants, Array Dimensions, Input/Output Conventions, and Program Execution

Unless otherwise specified, the units of length and energy used throughout this program, and assumed for all input data, are Å and cm^{-1} , respectively. The main exception is that the transition dipole function $M(r)$ of Eqs. (7) used for calculating the Einstein coefficients of Eq. (8), defined in terms of the expansion coefficients $\text{DM}(i)$ of `READ #20` (see § 4), is assumed to be in debye (where $1 \text{ debye} = 3.335\,640\,952 \times 10^{-30} [\text{C} \cdot \text{m}] = 0.393\,430\,295 [\text{au}]$). Note, however, that in the `IRFN ≥ 10` option for generating the radial function defining the matrix element argument by numerical interpolation over a set of read-in points (see `READ #19`), the channel-6 output describing the read-in transition moment function values being interpolated over may (incorrectly) refer to their units as cm^{-1} rather than debye, since the interpolation is done by the same subroutine package set up to deal with an input pointwise potential. Note too that if set of read-in points is used to define the potential or the matrix element argument $M(r)$, the values may be in any convenient units, as appropriate conversion factors are always also read in (see `READS #7 & 23`) to convert them to the appropriate units.

The values of the physical constants appear in the program in two places. The first is the dimensionless factor $\hbar^2/(2 u_0 E_0 r_0^2) = 16.857\,629\,206$ in the radial Schrödinger equation of Eq. (1), where the choice of reference mass $u_0 = 1 \text{ amu}$, distance $r_0 = 1 \text{ Å}$, and energy $E_0 = 1 \text{ cm}^{-1}$ effectively define the units of the input/output variables. The second is in the collections of terms defining the numerical factor in Eq. (8) used in calculating the Einstein coefficient for the rate of spontaneous emission. Our current values of these constants are based on the 2012 CODATA recommended values of Ref. [55], while the masses for all stable isotopes of all atoms stored in subroutine `MASSSES` were taken from the The AME2012 Atomic Mass Evaluation of Ref. [56].

The array dimension limits which a user may wish to change are set in `PARAMETER` statements in the main driver routine and in subroutines `GENINT` and `SPLINT`. In the former, `NDIMR` (currently 250001) is the maximum dimension of the radial mesh array on which the potential, wave functions and radial

expectation value/matrix element arguments are defined. For systems of small reduced mass, it could be safely set considerably smaller than this. The second parameter set in the main program is VIBMX (currently 400), which defines the maximum number of vibration/rotation levels for which vibrational eigenvalues may be read and stored, and the upper bound on the number of rotational sublevels which may be stored when applying the $\text{NJM} > 0$ option to automatically generate many J sublevels for a given v (see READ #19). The two other array-size parameters set inside the code are NTPMX (currently 1600) which is the maximum number of potential function points (or radial matrix element $M(r)$ values, for $\text{IRFN} > 10$) that may be read in, which is set in GENINT, and MAXSP (set in SPLINT), the number of spline coefficients required when interpolating over the read-in function values, and should be set equal to $4 \times \text{NTPMX}$.

The program reads input data on channel-5, writes standard output to channel-6, and optionally (controlled by parameters LPPOT, LCDC and LXPCT of READS #5 & 19) writes a condensed output file to one or more of channels 7-10. Those executing the program using a UNIX or Linux operating system environment may wish to create and store in the system or user's 'bin' directory a shell named (say) 'rlev', such as that shown here:

```
# UNIX shell 'rlev' to execute the compiled version of program LEVEL named
#   lev.x, which is stored in the user's directory /upath/ with input data
#   file $1.5, and write output to $1.6, $1.7, etc. all in the current
#   directory.
#
time ~/upath/lev.x < $1.5 > $1.6
mv fort.7 $1.7 >& /dev/null
mv fort.8 $1.8 >& /dev/null
mv fort.9 $1.9 >& /dev/null
mv fort.10 $1.10 >& /dev/null
```

This shell allows the program to be executed with the simple command: `rlev <filename>` where `<filename>.5` is the input data file the user has created (`<filename>` may be any name, usually chosen to identify a particular case). In this case the standard output from channel-6 will be written to file `<filename>.6`, and the channel-7, channel-8, channel-9, and channel-10 output will be written to files `<filename>.7`, `...`, `<filename>.10`, respectively.

4 Data file Structure and input parameter definitions

All of the READ statements for inputting data to the program, and the associated logical structure, are listed here. The following subsection then provides a detailed description of the nature and/or options associated with each of the input variables.

```
#1      READ(5,*,END=999) IAN1, IMN1, IAN2, IMN2, CHARGE, NUMPOT
#2a     IF(IAN1.LE.0) READ(5,*) NAME1, MASS1
#2b     IF(IAN2.LE.0) READ(5,*) NAME2, MASS2
#3      READ(5,*,END=999) TITL
#4      READ(5,*) RH, RMIN, RMAX, EPS
      DO IPOT= 1, NUMPOT
#5      READ(5,*) NTP, LPPOT, OMEGA, VLIM
      IF(NTP.GT.0) THEN
#6      READ(5,*) NUSE, IR2, ILR, NCN, CNN
#7      READ(5,*) RFACT, EFACT, VSHIFT
#8      READ(5,*) (XI(I), YI(I), I= 1,NTP)
      ELSE
#9      READ(5,*) IPOTL, PPAR, QPAR, APSE, Nbeta, IBOB
#10     READ(5,*) DSCM, REQ, Rref
#11     IF(IPOTL.GE.4) READ(5,*) NCMM, rhoAB, IVSR, IDSTT
#12     IF(IPOTL.GE.4) READ(5,*) (MMLR(I), CMM(I), I= 1, NCMM)
      IF(NVARB.GT.0) THEN
#13a    READ(5,*) (XPARM(I), PARM(I), I=1,NVARB)
      ELSE
#13b    READ(5,*) (PARM(I), I=1,NVARB)
      ENDIF
      ENDIF
      IF(IBOB.GT.0) THEN
#14     READ(5,*) MN1R, MN2R, pAD, qAD, NU1, NU2, pqNA, NT1, NT2
#15     IF(NU1.GE.0) READ(5,*) U1INF, (U1(I), I=0,NU1)
#16     IF(NU2.GE.0) READ(5,*) U2INF, (U2(I), I=0,NU2)
#17     IF(NT1.GE.0) READ(5,*) T1INF, (T1(I), I=0,NT1)
#18     IF(NT2.GE.0) READ(5,*) T2INF, (T2(I), I=0,NT2)
      ENDIF
      ENDIF
      ENDDO
#19     READ(5,*) NLEV1, AUTO1, LCDC, LXPCT, NJM, JDJR, IWR, LPRWF
      NLEV= MAX(1,NLEV1)
#20a    IF(AUTO1.GT.0) READ(5,*) (IV(I), IJ(I), I= 1,NLEV)
#20b    IF(AUTO1.LE.0) READ(5,*) (IV(I), IJ(I), GV(I), I= 1,NLEV)
      IF((LXPCT.NE.0).AND.(LXPCT.NE.-1)) THEN
#21     READ(5,*) MORDR, IRFN, DREF
#22     IF((IABS(IRFN).LE.9).AND.(MORDR.GE.0)) READ(5,*) (DM(J), J= 0,MORDR)
      IF(IRFN.GE.10) THEN
#23     READ(5,*) NRFN, RFLIM
#24     READ(5,*) NUSEF, ILRF, NCNF, CNNF
#25     READ(5,*) RFACTF, MFACTF
#26     READ(5,*) (XIF(I), YIF(I), I= 1,IRFN)
      ENDIF
      ENDIF
      IF(IABS(LXPCT).GE.3) THEN
#27     READ(5,*) NLEV2, AUTO2, J2DL, J2DU, J2DD
#28a    IF(AUTO2.GT.0) READ(5,*) (IV2(I), I= 1,NLEV2)
#28b    IF(AUTO2.GT.0) READ(5,*) (IV2(I), GV(I), I= 1,NLEV2)
      ENDIF
```

Definitions and Descriptions of Input File Data

Read integers identifying the molecule or system.

#1. READ(5,*) IAN1, IMN1, IAN2, IMN2, CHARGE, NUMPOT

IAN1 & IAN2: integer atomic numbers of the atoms/particles #1 & 2 forming the molecule. If both are positive and ≤ 109 , atomic masses from the tabulation in subroutine MASSES will generate the reduced mass of the system. If either is ≤ 0 or > 109 the mass of that particle will be input via READ #2.

IMN1 & IMN2: integer mass numbers of the atoms/particles #1 & 2 forming the molecule. For a normal stable atomic isotope, its mass was taken from the tabulation in subroutine MASSES; if outside the range for the normal stable isotopes of that atom, the abundance-averaged atomic mass will be used.

CHARGE: \pm integer for the total charge on the molecule. Used to generate Watson's charge-modified reduced mass for molecular ions [6]: $\mu = M_A M_B / (M_A + M_B - m_e \times \text{CHARGE})$, where m_e is the electron mass.

NUMPOT: the number of potentials considered: =1 for calculations involving only a single potential function; =2 to input and generate two different potentials and calculate matrix elements coupling their levels.

In the special case when IAN1 and/or IAN2 is either ≤ 0 or > 109 , read in a two-character alphanumeric name for that particle and its mass (in amu). This facilitates the treatment of model systems or exotic species such as muonium or positronium "molecules".

#2.a IF(IAN1.LE.0) READ(5,*) NAME1, MASS1

#2.b IF(IAN2.LE.0) READ(5,*) NAME2, MASS2

NAME1 & NAME2: a two-character alphanumeric name for the (1 or 2) particle(s) whose mass is being read, enclosed in single quotes, as in 'mu'.

MASS1 & MASS2: the masses of particles 1 and 2, in amu.

Read a text title or description for the calculation.

#3. READ(5,510) (TITL(I), I= 1,20)

TITL: a title or output header for the calculation, consisting of up to 78 characters on a single line, enclosed between single quotes: e.g., 'title of problem'.

Read real numbers defining the mesh and range of the numerical integration (all in Å), and the eigenvalue convergence criterion to be used (in cm^{-1}).

#4 READ(5,*) RH, RMIN, RMAX, EPS

RH: the numerical integration mesh size; see discussion associated with Eq. (2) in § 2.1.

RMIN & RMAX: the inner and outer limits, respectively, of the range of numerical integration (see § 2). Plausible zeroth order estimates would be $\text{RMIN} \approx 0.6 \times (\text{potential inner wall position})$ and **RMAX** fairly large (say 40 Å). Internally **RMAX** is set to the smaller of this read-in value and the largest distance allowed by **RMIN**, **RH** and the array dimension **NDIMR** (see § 3).

EPS: the eigenvalue convergence parameter used by SCHRQ (in cm^{-1}). To ensure that appropriately accurate expectation values or matrix elements are generated, it should usually be set ca. 2 orders of magnitude smaller than the eigenvalue precision actually required.

Some combination of the next 13 READ statements defines the potential energy function. A particular case always starts with READ #5, but then uses *either* READs #6–8 for numerical interpolation over a set of input turning points, *or* (some of) READs #9–18 for the case of an analytic potential function.

#5. READ(5,*) NTP, LPPOT, OMEGA, VLIM

NTP: An integer which is set ≤ 0 to generate an analytic potential using POTGEN, in which case the program skips READs #6–8 and goes directly to READ #9. If $NTP > 0$, it is the number of turning point pairs to be input via READ #8.

LPPOT: controls printing of the potential array (usually set = 0 to have no printing). If $LPPOT > 0$ write to standard output (channel-6) the potential and its first 2 derivatives-by-differences at every $LPPOT^{th}$ mesh point; it is sometimes useful to do this when troubleshooting. Setting $LPPOT < 0$ writes the resulting potential in condensed format to channel-8 at every $|LPPOT|^{th}$ mesh point; this is useful if one wants to input this potential into a plotting routine.

OMEGA: is the (integer) projection of the electronic orbital angular momentum on the molecular axis for this state, so that the reduced centrifugal potential becomes $[J(J+1) - OMEGA^2]\hbar^2/2\mu r^2$. For rotation constrained to a plane setting $OMEGA \geq 99$ will cause the centrifugal potential to have the appropriate form: $[J^2 - 1/4]\hbar^2/2\mu r^2$.

VLIM: The absolute energy (in cm^{-1}) of the potential asymptote. This sets the absolute energy scale seen in the output. For power-series (GPEF- or Dunham-type) potentials ($IPOTL = 2$), it is the energy at the potential function minimum where $r = r_e$.

For a pointwise potential (if $NTP > 0$), we must specify how the interpolation is to be done, and since RMAX usually lies outside the range of the input turning points, we also must specify how the potential is to be extrapolated at large r . Skip down to READ #9 if $NTP \leq 0$.

#6. READ(5,*) NUSE, IR2, ILR, NCN, CNN

NUSE: Specifies how the interpolation is to be done. If $NUSE > 0$ use NUSE-point piecewise polynomials; if $NUSE \leq 0$ perform cubic spline interpolation. For highly precise and smooth input points, such as those generated from an RKR calculation, $NUSE = 8, 10$ or 12 is usually most appropriate; for less precise or less dense points, such as those from *ab initio* calculations, low-order piecewise polynomials ($NUSE = 4$) or a spline ($NUSE \leq 0$) is usually best.

IR2: For very steep repulsive potential walls, better interpolation is often attained by actually interpolating over $r^2 \times V(r)$; setting the integer $IR2 > 0$ causes this to be done (normally recommended). A comparison between results obtained with this option turned on *vs.* off (setting $IR2 \leq 0$, which causes interpolation to be performed over $V(r)$ itself) provides an indication of the magnitude of “interpolation noise” uncertainties in the final results.

ILR: Specifies how to extrapolate from the outermost read-in turning points to RMAX. For a long extrapolation, one of $ILR = -1, 0$ or 1 is often most appropriate; however, if the outer turning points extend moderately close to the dissociation limit (at VLIM), one should set $ILR \geq 2$ and specify the theoretically appropriate value of NCN (≥ 1), and if it is available, also input an estimate of CNN (see below).

For $ILR < 0$, fit the last 3 points to: $V(r) = VLIM - A \times \exp[-b(r - r_o)^2]$

For $ILR = 0$, fit the last 3 points to: $V(r) = VLIM - A \times r^p \times \exp[-br]$.

For $ILR = 1$, fit the last 2 points to: $V(r) = VLIM - A/r^B$.

For $ILR = 2$ or 3 , respectively, fit the outermost 2 or 3 points to a sum of 2 or 3 inverse-power terms, with powers differing by 2: $V(r) = VLIM - \sum_{m=0}^{ILR-1} C_{NCN+2m}/r^{NCN+2m}$.

For $\text{ILR} \geq 4$, fit outermost ILR turning points to a sum of ILR inverse-power terms, with powers differing by 1: $V(r) = \text{VLIM} - \sum_{m=0}^{\text{ILR}-1} C_{\text{NCN}+m}/r^{\text{NCN}+m}$.

NCN: For inverse-power potential extrapolation with $\text{ILR} \geq 2$, NCN (> 0) specifies the limiting inverse-power behaviour: $V(r) \propto \text{VLIM} - \text{CNN}/r^{\text{NCN}}$. Otherwise (for $\text{ILR} \leq 1$) it is a dummy input variable.

CNN: For inverse-power potential extrapolation with $\text{ILR} \geq 2$, setting $\text{CNN} \neq 0$ causes the leading inverse-power coefficient to be fixed at the read-in value $\text{CNN} = C_{\text{NCN}} [\text{cm}^{-1} \text{\AA}^{\text{NCN}}]$ rather than be determined from a fit to the outermost turning points.

The input turning points may come from *ab initio* or RKR calculations, and may need to be converted to the units (\AA and cm^{-1}) used inside the program.

#7. READ(5,*) RFACT, EFACT, VSHIFT

RFACT & EFACT: are multiplicative factors required to convert units of the NTP input turning point distances $\text{XI}(i)$ and energies $\text{YI}(i)$ to \AA and cm^{-1} , respectively. If no conversion is required, read in factors of 1.0D+00.

VSHIFT: An energy shift (in cm^{-1}) to be added to the input potential point energies to make them consistent with VLIM. It addresses the fact that *ab initio* or RKR turning points may be expressed relative to an energy zero inconsistent with the user-specified asymptote energy VLIM.

Read in the actual turning points.

#8. READ(5,*) (XI(I), YI(I), I= 1,NTP)

XI(i) & YI(i): are the (distance, energy) input turning points defining the potential function.

If the input potential is defined by an analytic function (the $\text{NTP} \leq 0$ case) using subroutine POTGEN, the parameters characterizing it are input via some or all of READ statements #9–18. For a user-supplied POTGEN function, the input parameters (if any) are up to them, but the calling sequence must match that expected by PREPOT (see discussion at the end of § 2.5). The variable NVARB which specifies the number of parameters input through READ #13 is determined internally, as specified by the comments below.

#9. READ(5,*) IPOTL, PPAR, QPAR, APSE, Nbeta, IBOB

IPOTL: an integer specifying the type of analytic function used for the potential.

IPOTL=1 generates a Lennard-Jones($m = \text{PPAR}, n = \text{QPAR}$) function using Eq. (9). In this case APSE and Nbeta are dummy variables and NVARB = 0.

IPOTL=2 generates the GPEF power series potential of Eq. (11) of order $N_\beta = \text{Nbeta}$ using Seto's [26] rearranged form of Šurkus' [27] expansion variable $z = z(r) = (r^{\text{PPAR}} - r_e^{\text{PPAR}})/(a_S r^{\text{PPAR}} + b_S r_e^{\text{PPAR}})$, with β_0 defined by the input value of DSCM, $\text{PARM}(i) = \beta_i$ for $i = 1$ to Nbeta, $a_S = \text{PARM}(\text{Nbeta} + 1)$ and $b_S = (\text{Nbeta} + 2)$. Note that Šurkus' case of $\text{PPAR} < 0$ is accommodated by Seto's identity [26]: $z(-\text{PPAR}, a_S, b_S) = z(\text{PPAR}, -b_S, -a_S)$. Here NVARB = Nbeta + 2, while QPAR, APSE and NCMM are dummy variables.

- Dunham expansions are generated by setting $\text{PPAR} = 1$, $\alpha_S = 0.0$ & $b_S = 1.0$.
- SPF expansions are generated by setting $\text{PPAR} = 1$, $\alpha_S = 1.0$ & $b_S = 0.0$.
- Ogilvie–Tipping expansions are generated by setting $\text{PPAR} = 1$, $\alpha_S = b_S = 0.5$.
- A simple harmonic oscillator potential is obtained by setting $\text{PPAR} = 1$, $\text{Nbeta} = 0$, $a_S = 0.0$ and $b_S = 1.0$, while the harmonic force constant $k = 2\text{DSCM}/\text{REQ}^2$.
- All of these polynomial-type potentials have an undefined (or at best, indirectly-defined) asymptote, so parameter VLIM defines the potential energy minimum.

- If $\text{PPAR} = 0$, the potential is generated as an Nbeta -order polynomial in r whose constant coefficient is set as $c_0 = \text{VLIM}$ and $\text{NVARB} = \text{Nbeta}$.

$\text{IPOTL} = 3$ generates the Morse or EMO potential of Eq. (12), in which $\mathfrak{D}_e = \text{DSCM}$, $r_e = \text{REQ}$, and the expansion-variable of Eq. (13) is defined by the positive integer $q = \text{QPAR}$, while the expansion coefficients are $\text{PARM}(i) = \beta_{i-1}$ for $i = 1$ to $\text{NVARB} = (\text{Nbeta} + 1)$. In this case PPAR and APSE are dummy variables. Setting $\text{Nbeta} = 0$ yields the ordinary Morse potential.

If $\text{PPAR} \leq 0$, generate Hua Wei's [36] 4-parameter Morse-like potential

$$V(r) = \mathfrak{D}_e \left([1 - e^{-b(r-r_e)}] / [1 - C e^{-b(r-r_e)}] \right)^2 \quad \text{in which } b = \text{PARM}(1) \text{ and } C = \text{PARM}(2).$$

In this case $\text{NVARB} = 2$ while all of PPAR , APSE and Nbeta are dummy variables.

$\text{IPOTL} = 4$ generates an MLR or MLJ potential from Eqs. (15–20) [38, 57, 33, 37, 34, 35], in which $\mathfrak{D}_e = \text{DSCM}$, $r_e = \text{REQ}$, and the potential tail $u_{\text{LR}}(r)$ is defined by a sum of of undamped C_m/r^m , Eq. (19), or damped $D_m(r) C_m/r^m$, Eq. (20), inverse-power terms.

For $\text{APSE} \leq 0$, the exponent coefficient in Eq. (15) is represented by the constrained polynomial expansion of Eq. (17), in which positive integers $\text{PPAR} = p$ and $\text{QPAR} = q$ are the powers defining its two radial variables, and the upper bound on the summation is $N_\beta = \text{Nbeta}$, while the exponent expansion coefficients are $\text{PARM}(i) = \beta_{i-1}$ for $i = 1$ to $\text{NVARB} = (\text{Nbeta} + 1)$.

For $\text{APSE} > 0$, the exponent coefficient in Eq. (15) is represented as a natural cubic spline through Nbeta points at distances defined by the set of input y_q^{ref} values, and whose ordinate values $\beta_i = \text{PARM}(i)$ for $i = 1 - \text{Nbeta}$ define the potential function shape.

$\text{IPOTL} = 5$ generates the DELR potential of Eqs. (24–26) [12], in which $\mathfrak{D}_e = \text{DSCM}$, $r_e = \text{REQ}$, the power defining the expansion variable $y_q^{\text{ref}}(r)$ is $\text{QPAR} = p$. The exponent coefficient $\beta(r)$ is a simple power-series expansion in the variable $y_q^{\text{ref}}(r)$ with expansion coefficients $\text{PARM}(i) = \beta_{i-1}$ for $i = 1$ to $(\text{NVARB} = \text{Nbeta} + 1)$. The additive long-range tail function $u_{\text{LR}}(r)$ is defined by one of Eqs. (19) or (20) via READS #11 & 12, while PPAR is a dummy parameter.

$\text{IPOTL} = 6$ generates the generalized HFD-type potentials of Eq. (27), for which A and β_1 are defined (internally) by the input values of $\mathfrak{D}_e = \text{DSCM}$ and $r_e = \text{REQ}$, the parameters defining the damping function $D(x)$ are read in as $\text{PARM}(i) = \alpha_i$ for $i = 1 - 3$, and the other exponent coefficients are $\beta_2 = \text{PARM}(4)$ and $\gamma = \text{PARM}(5)$. For this case, the inverse-power coefficients C_m input through READ #12 should be dimensionless (scaled by $\mathfrak{D}_e r_e^m$), and $\text{NVARB} = 5$, while PPAR , QPAR , APSE and Nbeta are all dummy parameters.

$\text{IPOTL} = 7$ generates the ‘Hannover Polynomial Potential’ (HPP) [50] of Eq. (28), which consists of a power series of order Nbeta with coefficients $\beta_i = \text{PARM}(i + 1)$ for $i = 0 - \text{Nbeta}$, in the variable $\xi = (r - r_m)/(r + b r_m)$, where $b = \text{PARM}(\text{Nbeta} + 2)$. The well depth is specified as $\mathfrak{D}_e = \text{DSCM}$, and the reference distance r_m (which for $a_1 \neq 0$ is *not* identical to r_e) is read in as REQ . For $r < r_{\text{inn}} \equiv \text{PARM}(\text{Nbeta} + 3)$ the potential extrapolates inward by smoothly attaching the function $A e^{-\alpha(r-r_{\text{inn}})}$ at distance r_{inn} . For $r > r_{\text{out}} \equiv (\text{Nbeta} + 4)$ it extrapolates outward continuously as the sum of NCMM inverse-power terms defined by READS #11 & 12, with the (internal) addition of an additional term C_{m_L}/r^{m_L} with $m_L = [\text{MMLR}(\text{NCMM}) + 2]$, whose coefficient C_{m_L} is defined (internally) to attach this long-range tail continuously to the polynomial at $r = r_{\text{out}}$. For this model, PPAR , QPAR and APSE are dummy parameters, and $\text{NVARB} = (\text{Nbeta} + 4)$.

PPAR, QPAR, APSE & Nbeta: Integers used to characterize particular potential forms (see above).

IBOB: An integer to specify whether (for $\text{IBOB} > 0$) or not (for $\text{IBOB} \leq 0$) atomic-mass-dependent Born-Oppenheimer breakdown correction terms are to be included in the rotationless (electronic) and/or the centrifugal $\{ [J(J+1) - \Omega^2] \hbar^2 / 2\mu r^2 \}$ potential functions (see § 2.6).

#10. `READ(5,*) DSCM, REQ, Rref`

#11. `IF(IPOTL.GE.4) READ(5,*) NCMM, rhoAB, IVSR, IDSTT`

#12. `IF(IPOTL.GE.4) READ(5,*) (MMLR(I), CMM(I), I= 1,NCMM)`

DSCM: Normally (except for the $\text{IPOTL} = 2$ case), the potential well depth \mathfrak{D}_e in cm^{-1} .

REQ: Normally (except for the $\text{IPOTL} = 7$ case) the equilibrium distance r_e in Å.

Rref: The reference distance in the definition of the exponent expansion radial variable of Eq. (13).
If the input value is ≤ 0.0 , the code sets $r_{\text{ref}} = r_e$.

NCMM: Integer specifying the number of inverse-power terms to be used to define the long-range part of the potential for cases $\text{IPOTL} \geq 4$.

MMLR(I) & CMM(I): are the powers and coefficients of the NCMM inverse-power terms defining the long-range potentials of Eqs. (19) or (20).

If $\text{NCMM} \geq 3$ and $\text{MMLR}(2) = 0$, generate a long-range tail function defined as the lower root of the 2×2 diagonalization described in Ref. [34], as applied to the $A^1\Sigma_u^+$ state of Li_2 . For this case $\text{NCMM} = 4$, $\text{MMLR}(i) = 3, 0, 6$ & 8 , and $\text{CMM}(2) = A_{\text{so}}$, the $^7\text{Li}(2p)$ spin-orbit splitting energy.

If $\text{NCMM} \geq 3$ and $\text{MMLR}(2) = -1$, generate a long-range tail function defined as the lowest root of the 3×3 diagonalization described in Ref. [46], as applied to the $c^1^3\Sigma_g^+$ state of Li_2 . For this case, again, $\text{NCMM} = 4$, $\text{MMLR}(i) = 3, 0, 6$ & 8 , and $\text{CMM}(2) = A_{\text{so}}$, the $^7\text{Li}(2p)$ spin-orbit splitting energy.

If $\text{NCMM} \geq 3$ and $\text{MMLR}(2) = -1$, generate a long-range tail function defined as the *upper* root of the 2×2 diagonalization described in Ref. [34] as applied to the $b^3\Sigma_u^+$ state of Li_2 . For this case $\text{NCMM} = 4$, $\text{MMLR}(i) = 3, 0, 6$ & 8 , and $\text{CMM}(2) = A_{\text{so}}$, the $^7\text{Li}(2p)$ spin-orbit splitting energy.

rhoAB: selects whether ($\text{rhoAB} > 0$) or not ($\text{rhoAB} \leq 0$) damping functions are to be included in the long-range terms. If they are:

- ★ $\text{rhoAB} = \rho$ is the system-dependent scaling parameter,
- ★ integer $\text{IDSTT} > 0$ specifies use of the Douketis-type damping functions of Eq. (21), and
- ★ $\text{IDSTT} \leq 0$ specifies use of the Tang-Toennies-type damping functions of Eq. (22), while
- ★ integer IDF specifies the limiting short-range behaviour to be: $D_m(r) \cdot C_m / r^m \propto r^{\text{IDF}/2}$

#13 a. `IF((NVARB.GT.0).AND.(APSE.GT.0)) READ(5,*) (XPARM(I), PARM(I), I=1,NVARB)`

XPARM(i) are the NVARB values of the distance coordinate $y_q^{\text{ref}}(r_i)$ and $\text{PPARM}(i) = \beta_i$ the associated function values defining the cubic spline used to represent the MLR exponent coefficient function $\beta(r)$ when $\text{APSE} > 0$.

#13 b. `IF((NVARB.GT.0).AND.(APSE.LE.0)) READ(5,*) (PARM(I), I=1,NVARB)`

PPARM(I): are the NVARB parameters characterizing the potential functions described above for all cases *other* than an MLR with $\text{APSE} > 0$: e.g., the β_i parameters of Eqs. (11), (17) and (28) and the coefficients β_i of the exponent polynomials defining the EMO or DELR potentials.

If atomic-mass-dependent Born-Oppenheimer breakdown (BOB) terms are to be incorporated into the potential energy and/or centrifugal potential functions ($\text{IBOB} > 0$), use READS #14 and one or more (as needed) of #15–18. These BOB functions normally have the forms defined by Eqs. (29) and (30). If a given type of correction function is to be omitted, the associated expansion order (e.g., NU1, NU2, NT1 or NT2) should be set < 0 , in which case the associated parameter READ statement (of #15–18) is omitted, while if no BOB corrections are considered ($\text{IBOB} \leq 0$), omit all of READS #14–18.

#14. READ(5,*) MN1R, MN2R, PAD, QAD, NU1, NU2, QNA, NT1, NT2

MN1R & MN2R: The integer mass numbers of the *reference* isotopes of atoms 1 & 2 (or atoms A & B), respectively, whose masses M_a^{ref} appear in Eqs. (29) & (30) [58].

NU1 & NU2: For atoms $a = 1$ & 2 (or A & B), these are the orders N_{ad}^a of the polynomial expansions of Eq. (29). If either of them is set < 0 , neglect READS # 15 and/or 16, as appropriate, and omit that correction.

NTA & NTB: For atoms $a = 1$ & 2, these are the orders N_{na}^a of the polynomial expansions of Eq. (30). If either of them is set < 0 , neglect READS # 17 and/or 18, as appropriate, and omit that correction.

PAD, QAD & QNA: The (positive) integers p_{ad} , q_{ad} , and q_{na} , respectively, defining the polynomial expansion variables $y_p^{\text{eq}}(r)$ and $y_q^{\text{eq}}(r)$ for the potential energy and centrifugal BOB functions of Eq (29) and (30). One should normally set $\text{PAD} = p_{\text{ad}} \geq \{\text{the power of the leading inverse-power term contributing to the long-range potential } \text{PAD} = \text{MMLR}(1)\}$. If $\text{QAD} = 0$ and $\text{UaINF} = 0$ the adiabatic potential correction function is collapsed to a simple power series in $y_p(r)$.

For cases in which $\text{IBOB} > 0$,

#15. IF(NU1.GE.0) READ(5,*) U1INF, (U1(I), I= 0, NU1)

#16. IF(NU2.GE.0) READ(5,*) U2INF, (U2(I), I= 0, NU2)

#17. IF(NT1.GE.0) READ(5,*) T1INF, (T1(I), I= 0, NT1)

#18. IF(NT2.GE.0) READ(5,*) T2INF, (T2(I), I= 0, NT2)

UaINF & Ua(i): for $a = 1$ or 2, are the (real*8) expansion parameters defining the ‘adiabatic’ potential correction functions of Eq. (29): $\text{Ua}(j) = u_j^a$ and $\text{UaINF} = u_{\infty}^a$ all in units cm^{-1} .

TaINF & Ta(i): for $a = 1$ or 2, are the (real*8) expansion parameters defining the ‘non-adiabatic’ centrifugal potential correction functions of Eq. (30): all of $\text{Ta}(j) = t_j^a$ and $\text{TaINF} = t_{\infty}^a$ are dimensionless.

For a calculation involving only a single potential energy curve ($\text{NUMPOT} = 1$ in READ #1), now proceed directly to READ #19. However, if $\text{NUMPOT} = 2$ first repeat READS #5–18 to input the second potential function.

Now read the parameters controlling which levels are to be calculated, and what expectation values and or matrix elements are to be calculated (if any).

#19. READ(5,*) NLEV1, AUTO1, LCDC, LXPCT, NJM, JDJR, IWR, LPRWF

NLEV1: If > 0 , integer NLEV1 is a number of vibrational or vibrotational levels to be calculated. Their quantum number specifications are then input via READ #20.

If ≤ 0 , the program automatically finds all vibrational levels from $v = 0 - |\text{NLEV1}|$ associated with the rotational quantum number read in as IJ(1) (see below). If the input value of NLEV is very large and negative, the program will (attempt to) find *all* possible vibrational levels associated with the specified $J = \text{IJ}(1)$.

AUTO1: Integer $\text{AUTO1} > 0$ (normal option) causes the program to (attempt to) automatically generate realistic trial eigenvalues for all desired levels, so only their quantum number labels need be input via (READ #20a). If this fails, setting $\text{AUTO1} \leq 0$ will allow/require a trial energy $\text{GV}(i)$ to be input (via READ #20b) for each specified level (using the $\text{NLEV1} > 0$ option).

LCDC: If $\text{LCDC} > 0$, calculate the inertial rotational constant B_v and the first 6 centrifugal distortion constants $\{-D_v, H_v, L_v, \dots \text{etc.}\}$ for all of the levels specified by NLEV1 . These results are also written in a compact format to Channel-9.

LXPCT: An integer controlling what expectation values/matrix elements are to be calculated. For $\text{LXPCT} = 0$, no expectation values or matrix elements are calculated (in which case READs #21–26 are omitted). *Even* values of $|\text{LXPCT}| \neq 0$ cause the results to be written in compact form to channels–7 or 8 (as appropriate, see below), as well as to channel-6; *Odd* values of LXPCT yield only channel-6 output, and *negative* values of LXPCT cause most of the (relatively wordy) output to channel-6 to be suppressed.

$\text{LXPCT} = -1$ causes the eigenvalues and (if appropriate) quasibound level widths to be written compactly to channel-7, and no expectation values or matrix elements are calculated (so READs #21–26 are omitted).

$\text{LXPCT} = 1, 2$ or -2 causes calculation of expectation values of the kinetic energy and of positive powers of the distance variable specified by READ #21 plus either #22 or #23–26 (as appropriate, see below). Write results to channel-6 if $\text{LXPCT} = 1$ or 2; also write them (compactly) to channel-7 if $\text{LXPCT} = 2$, and write them only (compactly) to channel-7 if $\text{LXPCT} = -2$.

$|\text{LXPCT}| \geq 3$ invokes the calculation of matrix elements coupling levels of Potential-1 to each other (if $\text{NUMPOT} = 1$) or to levels of Potential-2 (if $\text{NUMPOT} = 2$), as specified by READs #27 & 28. Write results to channel-6 if $\text{LXPCT} > 0$ and (compactly) to channel-8 if $\text{LXPCT} = \pm 4$.

$|\text{LXPCT}| \geq 5$ also causes the component radial moments comprising the overall matrix elements to be written to channel-7, while still writing the overall matrix elements for selection-rule allowed transitions to channel-8. For $\text{LXPCT} = \pm 6$ write *only* the radial moment components, and omit the output to channel-8.

NJM & JDJR: If (integer) $\text{NJM} > 0$, for each (vibrational) level generated by the NLEV1 specification, automatically calculate eigenvalues (and if appropriate, expectation values and matrix elements) for all rotational sublevels J ranging from the input-specified (see below) $J = \text{IJ}(i)$ to a maximum of $J = \text{NJM}$ (or until that vibrational level energy predissociates above the potential barrier), with J increasing in steps of JDJR . e.g.: to automatically generate all possible rotational levels, set $\text{JDJR} = 1$, $\text{IJ}(i) = 0$ (or more strictly $= |\Omega|$) and NJM very large (e.g., $\text{NJM} = 999$).

IWR: An integer controlling the printout of diagnostics and calculation details inside `SCHRQ`.

If $\text{IWR} \neq 0$ print warning and error messages inside subroutine `SCHRQ`, as appropriate. Unless one is troubleshooting, normally set $\text{IWR} = -1$.

If $\text{IWR} \geq 1$ also print final eigenvalue and node count for every level determined.

If $\text{IWR} \geq 2$ also print end-of-range wave function amplitudes.

If $\text{IWR} \geq 3$ also print intermediate trial eigenvalues as the iterative convergence proceeds.

LPRWF: If $\text{LPRWF} > 0$ write to channel-6 the wavefunction at every $\{\text{LPRWF}\}^{th}$ mesh point.

If $\text{LPRWF} < 0$ write wavefunction compactly to channel-10 at every $|\text{LPRWF}|^{th}$ mesh point.

If $\text{LPRWF} = 0$, no wavefunction printout.

Read quantum numbers specifying which vibration-rotation levels (of Potential-1) are to be determined.

#20 a. IF(AUTO1.GT.0) READ(5,*) (IV(I),IJ(I),I= 1,max{1,|NLEV1|})

#20 b. IF(AUTO1.LE.0) READ(5,*) (IV(I),IJ(I),GV(I),I= 1,max{1,|NLEV1|})

IV(i) & IJ(i): For $NLEV1 > 0$ these are the vibrational [$v = IV(i)$] and rotational [$J = IJ(i)$] quantum numbers of the levels to be determined; if $NJM > IJ(i)$ the program also automatically calculates rotational levels for that $v = IV(i)$ with $J = IJ(i)$ to NJM in steps of $JDJR$.

For $NLEV1 \leq 0$, read one $\{IV(i), IJ(i)\}$ pair. The value of $IV(1)$ is ignored, but $J = IJ(1)$ is the rotational quantum number for which all vibrational levels up to $v = |NLEV1|$ are to be determined.

GV(i): If $AUTO1 \leq 0$, READ #20b is used in place of #20a, and $GV(i)$ is the trial energy read in for each level $v = IV(i)$, $J = IJ(i)$. This option presumes $NLEV1 > 0$.

If expectation values or matrix elements are to be calculated (i.e., if $LXPCT \neq 0$ or -1), READs #21-26 specify the desired arguments. However, if $LXPCT = 0$ or -1 the data input for this case is now finished.

#21. READ(5,*) MORDR, IRFN, DREF

MORDR: is an integer specifying the highest power of the chosen radial function or distance coordinate $RFN(r)$ whose expectation values or matrix elements are to be calculated (see Eq. (7)). The current program version is dimensioned for $MORDR \leq 20$. To calculate only Franck-Condon factors (when $|LXPCT| \geq 3$), input $MORDR = -1$.

IRFN & DREF: are integer and real variables, respectively, specifying the definition of the radial function or distance coordinate $RFN(r)$.

If $IRFN \leq -10$, RFN is generated by user-supplied code inserted at line #509-530 of the main program. In this case $DREF$ is a dummy variables, and READs #23-26 are omitted.

If $IRFN = -4$, then $RFN(r) = r$ and the matrix element operator is a polynomial in r of order $MORDR$, with coefficients $DM(i)$, pre-multiplying the derivative operator $\frac{d}{dr}$ which acts on the wavefunction for ‘Potential-2’.

If $IRFN = -3$, $RFN(r) = 1/r^3$.

If $IRFN = -2$, $RFN(r) = 1/r^2$.

If $IRFN = -1$, use a Dunham-type expansion coordinate $RFN(r) = (r - r_{dref})/r_{dref}$.

If $IRFN = 0$, the function $RFN(r) = r$, the distance coordinate itself.

If $IRFN = 1 - 9$, use the Šurkus-type variable $RFN(r) = y_p(r) = (r^p - r_{dref}^p)/(r^p + r_{dref}^p)$, with $p = IRFN$.

For $IRFN = -1$ or $1 - 9$, a positive (real) input value of $DREF$ specifies it as the reference distance r_{dref} for these cases (usually set $r_{dref} = r_e$). However, if the input value of $DREF \leq 0.0$, the program internally (iteratively) determines a value of r_{ref} such that the expectation value of $RFN(r)$ is identically zero for the first vibration-rotation level considered.

If $IRFN \geq 10$, $RFN(r)$ is a function defined by reading in and interpolating over (and extrapolating beyond) input values of some known radial function (e.g., a dipole or transition moment function). This reading and interpolation/extrapolation is performed by the same subroutine package PREPOT used for treating a numerical input potential (see READs #5-8). In this case $DREF$ is a dummy variable, READ #22 is omitted, and the code internally sets $MORDR = 1$, $DM(0) = 0.0$ and $DM(1) = 1.0$.

#22. IF(DABS(IRFN).LE.9) READ(5,*) (DM(J), J= 0,MORDR)

DM(j) : Coefficients of the power series in $\text{RFN}(r)$ defining the argument of the overall expectation values or matrix elements: $M(r) = \sum_{j=0}^{\text{MORDR}} \text{DM}(j) \times \text{RFN}(r)^j$.

If the expectation value or matrix element radial function argument is to be defined by interpolating over and extrapolating beyond a set of read-in points ($\text{IRFN} \geq 10$), use the same read sequence, options and procedures employed for treating a numerical input potential. Most input parameters here have definitions essentially equivalent to those associated with **READ** #5–8.

- #23. **READ**(5,*) **NRFN**, **RFLIM**
- #24. **READ**(5,*) **NUSEF**, **ILRF**, **NCNF**, **CNNF**
- #25. **READ**(5,*) **RFACTF**, **MFACTF**
- #26. **READ**(5,*) (**XIF**(I), **YIF**(I), I= 1,**NRFN**)

NRFN is the number of known function values $\{\text{XIF}(i), \text{YIF}(i)\}$ to be read in,

RFLIM is the limiting asymptotic value imposed when extrapolating beyond the range of the input values, and

NUSEF specifies how the interpolation is to be performed, while **ILRF**, **NCNF** and **CNNF** define the manner in which it extrapolates to large r (see comments for **READ** #6).

RFACTF & **MFACTF** convert units of input distances $\text{XIF}(i)$ and ordinates $\text{YIF}(i)$, respectively, to Å and whatever units are required for the expectation value/matrix element argument $M(r)$ (debye, for a dipole or transition moment function).

For matrix element calculations ($|\text{LXPCT}| \geq 3$), couple each level of Potential-1, generated as specified by **READS** #19 & 20, to all rotation levels of the **NLEV2** vibrational levels $v = \text{IV2}(i)$ allowed by the rotational selection rules $\Delta J = \text{J2DL}$ to J2DU in steps of J2DD (e.g., for P & R transitions: $\text{J2DL} = -1$, $\text{J2DU} = +1$ & $\text{J2DD} = +2$). If **NUMPOT** = 2 these are levels of Potential-2 and no constraints are imposed, but if **NUMPOT** = 1 the matrix elements couple levels of Potential-1 to one another, and to avoid redundancy the program considers only emission from (rotational sublevels of) these **NLEV2** vibrational levels into *lower* (v'', J'') levels generated as per **READS** #18 & 19. Integer **AUTO2** > 0 causes **LEVEL** to automatically generate trial eigenvalues for all desired levels (preferred option), so only their vibrational quantum number labels need be input (**READ** #28a). If this fails, setting **AUTO2** ≤ 0 will require a trial pure vibrational energy $\text{GV2}(i)$ to be read in (**READ** #28b) for each specified level.

- #27. **READ**(5,*) **NLEV2**, **AUTO2**, **J2DL**, **J2DU**, **J2DD**
- #28 a. **IF**(**NLEV2**.GT.0) **READ**(5,*) (**IV2**(I), I= 1,**NLEV2**)
- #28 b. **IF**(**NLEV2**.GT.0) **READ**(5,*) (**IV2**(I), **GV2**(I), I= 1,**NLEV2**)

5 Program Updates

This section briefly describes changes in the program's capabilities since Version 8.0 (April 2007). I am grateful to users for making a number of suggestions that have been incorporated into the program and/or its documentation.

Updates New to Version 8.2 (08 May 2014)

- For EMO, MLR or DELR analytic potentials, the distinction between N_S and N_L that allowed for different exponent polynomial orders for $r > r_e$ and $r \leq r_e$ has been removed, since experience has shown that equivalent or better behaviour was achieved by introducing the displaced expansion centre $\text{RREF} = r_{\text{ref}}$ (where usually $r_{\text{ref}} > r_e$) and (for the MLR) by allowing for $q \neq p$ in the exponent polynomial coefficient function [34, 46, 35, 42].
- The MLR exponent coefficient function now may optionally be represented by a Pashov-type [59, 60] cubic spline function [39]. [NOT fully tested!]

- The inverse-power potential tails of the MLR, DELR and HPP potentials have been generalized to allow the (optional) inclusion of ‘damping functions’, as per Eqs. (20)–(22). In addition to improving long-range behaviour, this leads to improved very short-range behaviour and sometimes also to more compact (i.e., smaller `Nbeta`) potential function expressions [35, 42].
- The above changes required a substantial modification of the input data lines governing analytic potentials with introduction of parameters `PPAR`, `QPAR`, `APSE` and `Nbeta` in `READ #9`, and the introduction of new `READS #10` and `11` that include the new parameters `RREF`, `rhoAB`, `IDF` and `IDSTT`.

Acknowledgements

I am pleased to acknowledge the foresight and stimulation provided by the late Professor George Burns of the University of Toronto who instigated my work on this type of computational research tool back in 1966, and the grace and financial backing provided by the late Professor Richard Bernstein, then of the University of Wisconsin, Madison, who patiently supported my meanderings in this area. I must also acknowledge the inspiration and assistance provided by R.N. Zare’s seminal 1964 Franck-Condon intensity factor paper and the associated reports [1, 2, 3]. I thank J.Y. Seto for assistance in developing the initial version of level-finder subroutine `ALF`, and J.Y. Seto and G.T. Kraemer for assistance in developing the atomic mass database subroutine `MASSSES`. I am also grateful to many users for comments which have identified errors and suggested useful extensions of the code, with particular thanks to R.W. Field, P.F. Bernath, A.J. Ross, A.S. Dickinson and G.C. McBane. My work in this area his work has been supported generously over many years by the Natural Sciences and Engineering Research Council of Canada.

References

- [1] R. N. Zare and J. K. Cashion, *The IBM Share Program D2 NU SCHR 1072 for Solution of the Schrödinger Radial Equation*, by J.W. Cooley: *Necessary and Useful Modifications for its Use on an IBM 7090*, University of California Lawrence Radiation Laboratory Report UCRL-10881 (1963).
- [2] R. N. Zare, University of California Lawrence Radiation Laboratory Report UCRL-10925 (1963).
- [3] R. N. Zare, *J. Chem. Phys.* **40**, 1934 (1964).
- [4] a) R. J. Le Roy, *LEVEL 1.0*, University of Wisconsin Theoretical Chemistry Institute Report WIS-TCI-429G (1971). b) R. J. Le Roy, *LEVEL 2.0* University of Waterloo Chemical Physics Research Report CP-58 (1976). c) *ibid*, *LEVEL 3.0*, CP-110 (1979). d) *ibid*, *LEVEL 4.0*, CP-230 (1983). e) *ibid*, *LEVEL 5.0*, CP-330 (1991). f) *ibid*, *LEVEL 5.1*, CP-330R (1992). g) *ibid*, *LEVEL 5.2* CP-330R² (1993). h) *ibid*, *LEVEL 5.3*, CP-330R³ (1994). i) *ibid*, *LEVEL 6.0*, CP-555 (1995). j) *ibid*, *LEVEL 6.1*, CP-555R (1996) k) *ibid*, *LEVEL 7.0 – 7.7*, CP-642 – 642R³, CP-655 & CP-661 (2000 – 2005) l) *ibid* *LEVEL 8.0*, CP-663 (2007 – 2014).
- [5] R. J. Le Roy, *LEVEL 8.2: A Computer Program for Solving the Radial Schrödinger Equation for Bound and Quasibound Levels*, University of Waterloo Chemical Physics Research Report CP-663 (2014); see <http://leroy.uwaterloo.ca/programs/>.
- [6] J. Watson, *J. Mol. Spectrosc.* **80**, 411 (1980).
- [7] J. W. Cooley, *Math. Computations* **15**, 363 (1961).
- [8] J. Cashion, *J. Chem. Phys.* **39**, 1872 (1963).
- [9] R. J. Le Roy and R. B. Bernstein, *J. Chem. Phys.* **54**, 5114 (1971).
- [10] R. J. Le Roy and W.-K. Liu, *J. Chem. Phys.* **69**, 3622 (1978).
- [11] J. N. L. Connor and A. D. Smith, *Mol. Phys.* **43**, 397 (1981).
- [12] Y. Huang and R. J. Le Roy, *J. Chem. Phys.* **119**, 7398 (2003), erratum: *ibid* **126** 169904 (2007).
- [13] K. Smith, *The Calculation of Atomic Collision Processes* (Wiley-Interscience, New York, 1971), chapter 4.
- [14] M. S. Child, *Semiclassical Mechanics with Molecular Applications* (Clarendon Press, Oxford, 1991).

- [15] P.-O. Löwdin, *An Elementary Iteration-Variation Procedure for Solving the Schrödinger Equation*, Technical Note No. 11, Quantum Chemistry Group, Uppsala University, Uppsala, Sweden (1958).
- [16] J.-S. Cho and R. Le Roy, in *70th International Symposium on Molecular Spectroscopy* (University of Illinois, Urbana-Champaign, Illinois, 2014), paper MG02.
- [17] M. Hehenberger, P. Froelich, and E. Brändas, *J. Chem. Phys.* **65**, 4559 (1976).
- [18] M. Hehenberger, *J. Chem. Phys.* **67**, 1710 (1977).
- [19] R. J. Le Roy, *Comp. Phys. Comm.* **52**, 383 (1989).
- [20] R. J. Le Roy and G. T. Kraemer, *BCONT 2.2: A Computer Program for Calculating Bound→Continuum Transition Intensities for Diatomic Molecules*, University of Waterloo Chemical Physics Research Report CP-650R² (2004); see <http://leroy.uwaterloo.ca/programs/>.
- [21] G. Herzberg, *Spectra of Diatomic Molecules* (Van Nostrand, New York, 1950).
- [22] J. Tellinghuisen, *J. Mol. Spectrosc.* **122**, 455 (1987).
- [23] J. M. Hutson, *J. Phys. B: At. Mol. Phys.* **14**, 851 (1981).
- [24] P. F. Bernath, *Spectra of Atoms and Molecules*, 2nd ed. (Oxford University Press, New York, 2005).
- [25] A. Hansson and J. K. G. Watson, *J. Mol. Spectrosc.* **233**, 169 (2005).
- [26] J. Y. Seto, *Direct Fitting of Analytic Potential Functions to Diatomic Molecule Spectroscopic Data*, M.Sc. Thesis, Department of Chemistry, University of Waterloo (2000).
- [27] A. A. Šurkus, R. J. Rakauskas, and A. B. Bolotin, *Chem. Phys. Lett.* **105**, 291 (1984).
- [28] J. L. Dunham, *Phys. Rev.* **41**, 721 (1932).
- [29] G. Simons, R. G. Parr, and J. M. Finlan, *J. Chem. Phys.* **59**, 3229 (1973).
- [30] J. F. Ogilvie, *Proc. Roy. Soc. (London)* **A 378**, 287 (1981).
- [31] E. G. Lee, J. Y. Seto, T. Hirao, P. F. Bernath, and R. J. Le Roy, *J. Mol. Spectrosc.* **194**, 197 (1999), GeO paper.
- [32] J. Y. Seto, Z. Morbi, F. Charron, S. K. Lee, P. F. Bernath, and R. J. Le Roy, *J. Chem. Phys.* **110**, 11756 (1999).
- [33] Y. Huang, *Determining Analytical Potential Energy Functions of Diatomic Molecules by Direct Fitting*, M.Sc. Thesis, Department of Chemistry, University of Waterloo (2001).
- [34] R. Le Roy, N. Dattani, J. Coxon, A. Ross, P. Crozet, and C. Linton, *J. Chem. Phys.* **131**, 204309 (2009).
- [35] R. Le Roy, C. Haugen, J. Tao, and H. Li, *Mol. Phys.* **109**, 435 (2011).
- [36] W. Hua, *Phys. Rev. A* **42**, 2524 (1990).
- [37] R. Le Roy and R. Henderson, *Mol. Phys.* **105**, 663 (2007).
- [38] P. Hajigeorgiou and R. Le Roy, *J. Chem. Phys.* **112**, 3949 (2000).
- [39] J. Tao, R. Le Roy, and A. Pashov, in *65th Ohio State University International Symposium on Molecular Spectroscopy* (Columbus, Ohio, 2010), p. paper MG07.
- [40] R. Le Roy, Y. Huang, and C. Jary, *J. Chem. Phys.* **125**, 164310 (2006).
- [41] T. Yukiya, N. Nishimiya, Y. Samajima, K. Yamaguchi, M. Suzuki, C. Boone, I. Ozier, and R. J. Le Roy, *J. Mol. Spectrosc.* **283**, 32 (2013).
- [42] R. Henderson, A. Shayesteh, J. Tao, C. Haugen, P. Bernath, and R. Le Roy, *J. Phys. Chem. A* **117**, 13373 (2013).
- [43] C. Douketis, J. M. Hutson, B. J. Orr, and G. Scoles, *Mol. Phys.* **52**, 763 (1984).
- [44] K. T. Tang and J. P. Toennies, *J. Chem. Phys.* **80**, 3726 (1984).
- [45] W. Gunton, M. Semczuk, N. Dattani, and K. Madison, *Phys. Rev. A* **88**, 062510 (2013).
- [46] N. Dattani and R. Le Roy, *J. Mol. Spectrosc.* **268**, 199 (2011).

- [47] M. Semczuk, X. Li, W. Gunton, M. Haw, N. Dattani, J. Witz, A. Mills, D. Jones, and K. Madison, Phys. Rev. A **87**, 052505 (2013).
- [48] J. Hepburn, G. Scoles, and R. Penco, Chem. Phys. Lett. **36**, 451 (1975).
- [49] C. Douketis, G. Scoles, S. Marchetti, M. Zen, and A. J. Thakkar, J. Chem. Phys. **76**, 3057 (1982).
- [50] C. Samuelis, E. Tiesinga, T. Laue, M. Elbs, H. Knöckel, and E. Tiemann, Phys. Rev. A **63**, 012710 (2000).
- [51] R. J. Le Roy, in *Molecular Spectroscopy*, edited by R. N. Barrow, D. A. Long, and D. J. Millen (Chemical Society of London, London, 1973), Vol. 1, Specialist Periodical Report 3, pp. 113–176.
- [52] R. J. Le Roy, J. Chem. Phys. **73**, 6003 (1980).
- [53] R. J. Le Roy, J. Chem. Phys. **101**, 10217 (1994).
- [54] R. J. Le Roy and Y. Huang, J. Mol. Struct. (Theochem) **591**, 175 (2002).
- [55] P. Mohr, B. Taylor, and D. Newell, Rev. Mod. Phys. **84**, 1527 (2012).
- [56] M. Wang, G. Audi, A. Wapstra, F. Kondev, M. MacCormick, X. Xu, and B. Pfeiffer, Chin. Phys. C **36**, 1603 (2012).
- [57] J. Y. Seto, R. J. Le Roy, J. Vergès, and C. Amiot, J. Chem. Phys. **113**, 3067 (2000).
- [58] R. J. Le Roy, J. Mol. Spectrosc. **194**, 189 (1999).
- [59] A. Pashov, W. Jastrzębski, and P. Kowalczyk, Comp. Phys. Comm. **128**, 622 (2000).
- [60] A. Pashov, W. Jastrzębski, and P. Kowalczyk, J. Chem. Phys. **113**, 6624 (2000).
- [61] S. D. Peyerimhoff and R. J. Buenker, Chem. Phys. **57**, 279 (1981).
- [62] R. J. Le Roy, D. R. T. Appadoo, K. Anderson, A. Shayesteh, I. E. Gordon, and P. F. Bernath, J. Chem. Phys. **123**, 204304 (2005).
- [63] D. Barrow, M. Slaman, and R. A. Aziz, J. Chem. Phys. **91**, 6348 (1991).
- [64] O. Allard, A. Pashov, H. Knöckel, and E. Tiemann, Phys. Rev. A **66**, 042503:1 (2002).

Appendix: Program Structure, Illustrative Cases, Sample data and output files, and timing

A. Program Structure

The present section lists the names and outlines the functions of the various subroutines used by **LEVEL**, and indicates their hierarchy. In particular, the level of indentation in this list indicates which subroutines call which others; unless stated otherwise, each subroutine is called exclusively by the immediately preceding routine having one lower level of indentation.

LEVEL: The main program which reads input data and instructions, and calls the potential preparation, eigenfunction determination, and overlap integral routines.

MASSSES: A data subroutine containing accurate atomic masses and other properties of all stable atomic isotopes. For normal cases, its presence obviates the need for a user to look up and type precise particle masses into the input data file. Also called by POTGEN for to define mass-scaling of BOB functions.

ALF: For any smooth single-minimum, shelf-state or double-minimum potential, ALF (Automatic Level Finder) uses multiple calls to subroutine SCHRQ (see below) to determine the vibrational energies of all levels from $v = 0$ up to some maximum v specified by its input parameter $KVMAX = |NLEV1|$. If fewer than the specified number of levels are found, warnings are printed.

SCECOR: used first-order semiclassical estimates of $(v + 1/2)$ and $dG(v)/dv$, and sometimes more brute force methods, to generate a trial eigenvalue for the ‘next’ level of interest.

SCHRQ: Solves the Schrödinger equation to determine the eigenvalue and (unit normalized) eigenfunction of the vibrational level lying closest to the input trial energy.

QBOUND: For quasibound levels (those lying above the potential asymptote, but behind a potential barrier), applies the Airy function boundary condition at the third turning point to initiate the inward integration of the wave function for such levels [9, 10].

WIDTH: Calculates the tunneling predissociation lifetime or width of a quasibound level [10, 11, 12].

LEVQAD: Called by WIDTH to evaluate the near-turning-point contributions to the semiclassical quadratures over the potential well and the barrier to tunneling which are required for calculating the predissociation rate.

CDJOEL: Calculates the diatomic molecule centrifugal distortion constants of Eq.(7). The required input is the effective (centrifugally-distorted, if appropriate) radial potential, and the eigenvalue and eigenfunction of the level in question (as calculated by SCHRQ).

LEVXPC: Calculates the desired diagonal expectation values of powers of the specified distance coordinate or (interpolated) radial function $RFN(r)$ (see READ #19 in Section 4).

MATXEL: Calculates the desired off-diagonal matrix elements of powers of the specified radial function or distance coordinate, and the radiative lifetime or Einstein emission coefficient \mathcal{A} [s⁻¹].

PREPOT: The subroutine which oversees reading of the parameters defining the potential and generating the required potential array at the N distances specified by the input distance array.

GENINT: Uses piecewise polynomial or cubic spline functions to interpolate over a set of read-in turning points to yield the potential array at the equally-spaced radial mesh required by SCHRQ. If necessary it extrapolates beyond range of the given points with analytic functions (see options for parameter ILR of READ #6).

PLYINTRP: Performs piecewise polynomial interpolation on a given array of points and returns the function value and (if desired) its derivatives at a specified point. It is used by **GENINT** to interpolate for the potential for cases with **NUSE** > 0. This routine is also used by **SPLINT** to define the first derivatives at the inner and outer ends of the interpolation region, $R = \text{XI}(1)$ and $\text{XI}(\text{NTP})$, respectively.

SPLINT: Calls subroutine **SPLINE** to generate the $4 \times (\text{NTP} - 1)$ coefficients required to define the cubic spline through the **NTP** given turning points, and then uses these coefficients to generate the desired interpolated function values at the specified integration mesh points.

SPLINE: Generates the coefficients defining the cubic spline function through the given **NTP** turning points.

POTGEN: For the various analytic potential function specified by input parameter **IPOTL** (see **READ** #9), reads in the parameters defining those functions and generates the desired array of potential function values on the specified radial grid. If **BOB** correction functions are to be included, also reads in the parameters defining them and incorporates them into the effective adiabatic potential used by **SCHRQ**.

DAMPF: Generates the damping functions of Eqs. (21) and (22) utilized in defining the tails of the **MLR**, **DELR** and **HPP** potential energy functions.

AF3X3LEV: Performs the 3×3 diagonalization used to define the long-range tail of the **MLR** potential function for the $c1\ ^3\Sigma_g^+$ state of Li_2 [46].

LKcoef: Asen Pashov's subroutine to generate the array of **Nbeta** \times **Nbeta** coefficient that define his spline 'basis functions' [59].

Scalc: A function subroutine to generate the $S_m(x)$ 'basis functions' used by Pashov to generate values of a spline function.

ludcmp: A subroutine used by **LKcoef**.

B. Illustrative Cases

The running time for this program will depend entirely on the complexity of the calculation being performed and the type of computer being used; CPU requirements may range from a fraction of a second to a couple of minutes. This Appendix presents sample data files and the resulting output for representative cases which illustrate some of the types of problems to which the program may be applied. On a somewhat aged UNIX workstation, the 4 examples of **Case 1** require 0.125 s of CPU time, the five examples comprising **Case 2** require 1.28 s, and the "production run" calculation of **Case 3** requires 23.71 s.

Case 1: This data file (see Appendix C) consists of five separate data sets which illustrate a variety of the capabilities of the program, including the fact that it can treat several independent problems in a single run by simply putting the input data for several cases into the same file. Parts of the resulting output are listed in Appendix D.

- a) For a Lennard-Jones(12,6) potential, find all the vibrational levels and calculate the associated values of the centrifugal distortion constants. This is a model system with fictitious particle masses, so those masses and the chosen particle names 'L1' and 'J2' are input by **READ** #2.
- b) For the same simple model Lennard-Jones(12,6) potential of Case 1.b), calculate all possible infrared matrix elements involving levels with $v \leq 2$ and $J(\text{upper}) \leq 1$.
- c) For the same model Lennard-Jones(12,6) potential of Case 1.b), locate all (four) vibrational levels of the centrifugally-distorted potential associated with $J = 18$. This demonstrates that the procedure for automatically finding all vibrational levels works for a potential with a barrier

(here, centrifugal in origin) protruding above its dissociation limit. This example invokes the highest print level inside SCHRQ (by setting `IWR = 3`, see `READ #17`) in order to illustrate the progress of the iterative eigenvalue convergence procedure. In this case convergence details for each level are presented twice (though only one iteration is required the second time) because of a quirk of the internal program logic.

- d) To illustrate the ability to determine the eigenvalues of a double-minimum potential, find the eigenvalues for vibrational levels $v = 56 - 80$ of an *ab initio* potential for the ${}^3\Pi_g$ state of Cl_2 [61], and calculate expectation values of various powers of r . The lowest of these levels lie in the region where all levels belong to the deeper outer well, while the highest lies above the barrier separating the two wells. The output expectation values show that as the energy increases, the wave function amplitude hops back and forth between the two wells. This case illustrates the ability of the program, and in particular, its automatic level-finder subroutine `ALF`, to automatically find arbitrary levels of a double minimum potential.

Case 2. Illustrative input data files are provided for eigenvalue calculations using a number of the analytic potential energy functions described in §2.6. (Appendix E).

- a) To illustrate an application to an **Extended Morse Oscillator (EMO) potential**, this case calculates the eigenvalues of ${}^{109}\text{AgD}$ from the ‘ $\text{EMO}_3(7)$ ’ potential determined in Ref. [62] from a combined-isotopologue analysis that treated ${}^{107}\text{AgH}$ as the ‘reference isotopologue’.
- b) To illustrate use of one of the **Morse/Long-Range (MLR) potential** form and the incorporation of Born-Oppenheimer breakdown (BOB) contributions to the radial and centrifugal potentials, this case calculates properties of levels of the ground state of ${}^{24}\text{MgD}$ using the analytic **MLR potential** and BOB correction functions determined from a combined isotopologue analysis of data for the MgH system in which ${}^{24}\text{MgH}$ was the “reference isotopologue” (see Ref. [42]). Note that in such cases, calculation of the rotational constants is based on a centrifugal potential which includes the BOB correction function of Eq. (30).
- c) This is an example of a calculation performed using the **DELR potential** form that was introduced in Ref. [12]. This is not precisely the same DELR potential function reported there, since the current version of the code no longer allows the exponent polynomial in the DELR form to have different orders for $r \leq r_e$ and $r > r_e$. However, it still provides an excellent representation of the data for emission into the $B\,{}^1\Pi_u$ state of Li_2 .
- d) This calculation for the ground $X\,{}^1\Sigma^+$ state of ${}^{20}\text{Ne}^{84}\text{Kr}$ is performed using the **generalized HFD function** reported by Aziz *et al.* [63].
- e) This calculation is performed using the “**X-expansion or HPP**” **polynomial potential** for the $X\,{}^1\Sigma_g^+$ state of Ca_2 reported by Allard *et al.* [64]. Note that the fact that the 18-digit polynomial expansion coefficients reported in Ref. [64] are rounded to (approximately) 13 digits by our program appears to have no significant effect on the results.

Case 3. This is a data set for an illustrative line-list “production run” for the case of **two pointwise potentials** (`NUMPOT = 2` in `READ #1` and `NTP > 0` in `READ #5`) which generates the predicted transition energies and Einstein emission coefficients for some 267 555 lines in the $B({}^3\Pi_{0u}^+) - X({}^1\Sigma_g^+)$ spectrum of Br_2 . This case again exploits the capability of the `NLEV1 < 0` option for automatically finding the first $|\text{NLEV1}| + 1$ levels of a given potential. It also illustrates a use of the `NJM > 0` option for finding many (or all) rotational sublevels for each v , and of the application of rotational selection rules in calculating transitions between two different electronic states. The input data file is listed in Appendix F, parts of the channel-6 output in Appendix G and a portion of the associated channel-8 output is in Appendix H.

The largest piece of output for this case consists of a listing the transition frequencies and Einstein \mathcal{A} -coefficients for some 267 555 transitions (yielding a line-list of that length) written to channel-8. A small segment of this output file is shown in Appendix H. Since the associated “standard” output to channel-6 is also moderately lengthy (some 3221 lines), the listing of it presented in Appendix G is also considerably truncated. However, this output illustrates some of the (usually not serious) warning messages that may be generated by the program. In particular, for $v' = 34$, $J' = 100 - 103$ of the truncated listing of Appendix G, warnings occur for quasibound levels for which **RMAX** is smaller than the outermost turning point at which the program was attempting to apply the Airy function boundary condition (see Section 2.B). This warning is printed once each iteration as **SCHRQ** converges on the associated eigenvalue; as it indicates, in this case the Airy function boundary condition is replaced by use of the WKB wave function initialization of Eq. (3). The second type of warning (“... **so tunneling calculation uses...**”) is printed following convergence on the eigenvalue for such a level, as a reminder that the width calculation for this case uses an approximation estimate of the portion of the exponent integral from the end of the range to the actual outermost turning point. However, when this situation arises the associated tunneling level widths are usually extremely small, and although the resulting predicted width may be slightly in error the calculated eigenvalue is usually quite accurate.

A different type of problem gives rise to the lines beginning “**SCHRQ has a convergence problem...**” and “**CAUTION for ...**” seen here for $v = 34$, $J = 127$. This is the highest quasibound level for this v ; it lies very close to the centrifugal barrier maximum, and after 30 iterations from each of two separate starting points, the Airy function boundary condition was unable to achieve full convergence to **EPS** for this particular level. In spite of this problem, however, the last eigenvalue change of “**DE= 1.63D-03**” cm^{-1} is considerably smaller than the width (**FWHM** = 0.90 cm^{-1}) of this tunneling predissociation level, so the lack of full convergence has negligible effect on the real accuracy of the resulting eigenvalue.

Another type of warning message is that seen for $v = 40$ when $J = 93$ (“... **find onee turn point: ...**”). It appears when the automatic search for ever higher rotational sublevels goes past the top of the (centrifugal) barrier maximum. In this case the code makes additional tries to place a level marginally below that barrier, a step which sometimes fails (as it did for in this case) and sometimes succeeds.

C. Input data file for Cases 1a) – 1d)

Note that entries after the “%” sign on each line of the input data file are comments identifying the variables, and are ignored by the program.

```

0 30 0 30 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'L1' 30.34373256D0 % NAME1 MASS1
'J2' 30.34373256D0 % NAME2 MASS2
'Case 1.a: For a model L.J.(12,6) get all vib. levels and their CDCs'
0.0020 0.6 20. 1.d-6 % RH RMIN RMAX EPS
0 0 0 0.D0 % NTP LPPOT IOMEG VLIM
1 12 6 0 0 0 % IPOTL PPAR QPAR APSE Nbeta IBOB
1000.d0 1.d0 1.d0 % DSCM REQ Rref
-999 1 1 -1 0 1 -1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0 % IV(1) IJ(1)

0 30 0 30 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'L1' 30.34373256D0 % NAME1 MASS1
'J2' 30.34373256D0 % NAME2 MASS2
'Case 1.b: For same model L.J.(12,6) get radial matrix elements for v < 3'
0.0020 0.6 20. 1.d-6 % RH RMIN RMAX EPS
0 0 0 0.D0 % NTP LPPOT IOMEG VLIM
1 12 6 0 0 0 % IPOTL PPAR QPAR APSE Nbeta IBOB
1000.d0 1.d0 1.d0 % DSCM REQ Rref
-2 1 0 5 1 1 -1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0 % IV(1) IJ(1)
4 -1 1.D0 % MORDR IRFN RREF
1.d0 -0.2d0 0.03d0 -0.004d0 0.0005d0 % {DM(j)}
3 1 -1 +1 2 % NLEV2 AUTO2 J2DL J2DU J2DD
0 1 2 % IV2(1) IV2(2) IV2(3)

-1 0 -1 0 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'L1' 30.34373256D0 % NAME1 MASS1
'J2' 30.34373256D0 % NAME2 MASS2
'Case 1.c: For the same L.J.(12,6), find levels of potential with a barrier'
0.0020 0.6 20. 1.d-6 % RH RMIN RMAX EPS
0 0 0 0.D0 % NTP LPPOT IOMEG VLIM
1 12 6 0 0 0 % IPOTL PPAR QPAR APSE Nbeta IBOB
1000.d0 1.d0 1.d0 % DSCM REQ Rref
-999 1 0 -1 0 1 3 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 18 % IV(1) IJ(1)

17 35 17 35 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 1.d: Levels of a pointwise double-minimum potential for 3(PI)g Cl2'
0.001 1.6 10. 1.d-06 % RH RMIN RMAX EPS
29 0 0 95440.D0 % NTP LPPOT IOMEG VLIM
0 0 2 1 0.D5 % NUSE IR2 ILR NCN CNN
0.5291772108D0 8065.5444D0 0.d0 % RFACT EFAC VSHIFT
3.34 10.608 3.40 9.683 3.43 9.221 3.49 8.758
3.57 8.295 3.66 8.132 3.76 8.051 3.86 8.132 3.94 8.268
4.03 8.350 4.17 8.595 4.26 8.704 4.31 8.758 4.46 8.432
4.56 8.134 4.66 7.887 4.80 7.615 4.91 7.425 5.00 7.343
5.17 7.125 5.37 6.962 5.51 6.908 5.66 6.880 5.80 6.908
5.94 6.989 6.00 7.044 6.20 7.125 6.40 7.261 6.60 7.425
22 1 0 1 0 1 -1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
56 0 57 0 58 0 59 0 60 0 61 0 62 0 63 0 64 0 65 0 66 0 67 0
68 0 69 0 70 0 72 0 72 0 73 0 74 0 75 0 76 0 77 0 78 0 80 0
3 0 0.D0 % MORDR IRFN RREF
1.d0 -2.d-1 3.d-2 -4.d-3 % {DM(j)}
```


D. Channel-6 Output file for Cases 1 a) – 1 d)

```

Case 1.a: For a model L.J.(12,6) get all vib. levels and their CDCs
=====
Generate ZMU= 15.17186628000(u) & BZ= 8.999999997D-01((1/cm-1)(1/Ang**2))
      from atomic masses: 30.34373256000 & 30.34373256000(u)
Integrate from RMIN= 0.600 to RMAX= 20.00 with mesh RH= 0.002000(Angst)

Potential #1 for L1( 30)-J2( 30)
=====
State has OMEGA= 0 and energy asymptote: Y(lim)= 0.0000(cm-1)

Lennard-Jones(12, 6) potential with De= 1000.000(cm-1) Re = 1.000000(A)
-----
Calculate properties of the single potential described above
Potential-1 uses inner boundary condition of zero value at RMIN

Eigenvalue convergence criterion is EPS= 1.0D-06(cm-1)
Airy function at 3-rd turning point is quasibound outer boundary condition

State-1 electronic angular momentum OMEGA= 0
      yields centrifugal potential [J*(J+1) - 0.00]/r**2
For J= 0, find the first 400 levels of Potential-1 with VLIM= 0.0000

ALF finds the highest calculated level is E(v= 6)=-0.77249285

E(v= 0, J= 0)= -811.519 Bv= 1.0540426 -Dv= -4.1381D-05 Hv= -6.5120D-09
      Lv= -1.9007D-12 Mv= -7.2230D-16 Nv= -3.1926D-19 Ov= -1.5585D-22
E(v= 1, J= 0)= -507.170 Bv= 0.9332450 -Dv= -6.0794D-05 Hv= -1.4820D-08
      Lv= -6.5533D-12 Mv= -3.7208D-15 Nv= -2.4359D-18 Ov= -1.7523D-21
E(v= 2, J= 0)= -287.838 Bv= 0.8013476 -Dv= -9.2177D-05 Hv= -3.5187D-08
      Lv= -2.3947D-11 Mv= -2.0879D-14 Nv= -2.1041D-17 Ov= -2.3384D-20
E(v= 3, J= 0)= -141.503 Bv= 0.6566553 -Dv= -1.4627D-04 Hv= -9.2303D-08
      Lv= -1.0432D-10 Mv= -1.5291D-13 Nv= -2.6179D-16 Ov= -4.9802D-19
E(v= 4, J= 0)= -54.890 Bv= 0.4974391 -Dv= -2.4978D-04 Hv= -2.9569D-07
      Lv= -6.5212D-10 Mv= -1.9135D-12 Nv= -6.6507D-15 Ov= -2.5885D-17
E(v= 5, J= 0)= -13.337 Bv= 0.3221808 -Dv= -4.9470D-04 Hv= -1.4933D-06
      Lv= -9.1824D-09 Mv= -7.7465D-11 Nv= -7.8195D-13 Ov= -8.8619D-15
E(v= 6, J= 0)= -0.7724928 Bv= 1.2900D-01 -Dv= -1.6478D-03 Hv= -3.8715D-05
      Lv= -2.0424D-06 Mv= -1.4479D-07 Nv= -1.1986D-08 Ov= -1.0942D-09

Find 7 Potential-1 vibrational levels with J= 0
      v E(v) v E(v) v E(v) v E(v)
-----
0 -811.5192 2 -287.8379 4 -54.8902 6 -0.7725
1 -507.1696 3 -141.5034 5 -13.3368

An n= 6 N-D theory extrapolation from last 2 levels implies vD = 6.631
=====

```

```

Case 1.b: For same model L.J.(12,6) get radial matrix elements for v < 3
=====
Generate ZMU= 15.17186628000(u) & BZ= 8.999999997D-01((1/cm-1)(1/Ang**2))
      from atomic masses: 30.34373256000 & 30.34373256000(u)
Integrate from RMIN= 0.600 to RMAX= 20.00 with mesh RH= 0.002000(Angst)

Potential #1 for L1( 30)-J2( 30)
=====
State has OMEGA= 0 and energy asymptote: Y(lim)= 0.0000(cm-1)

Lennard-Jones(12, 6) potential with De= 1000.000(cm-1) Re = 1.000000(A)
-----
Calculate properties of the single potential described above
Potential-1 uses inner boundary condition of zero value at RMIN

Eigenvalue convergence criterion is EPS= 1.0D-06(cm-1)
Airy function at 3-rd turning point is quasibound outer boundary condition

State-1 electronic angular momentum OMEGA= 0
      yields centrifugal potential [J*(J+1) - 0.00]/r**2
For J= 0, find the first 3 levels of Potential-1 with VLIM= 0.0000
      and automatically increment J in steps of 1 to a maximum value of 1

Matrix element argument expansion variable is: X = (r - DREF)/DREF
      where reference length is held fixed at DREF = 1.0000000000(Angstroms)
Coefficients of expansion for radial matrix element/expectation value argument:
      1.000000D+00 -2.000000D-01 3.000000D-02 -4.000000D-03 5.000000D-04

Using the rotational selection rule: delta(J)= -1 to 1 with increment 2
      calculate matrix elements for coupling to the 3 vibrational levels of
      Potential-2: v = 0 1 2
-----
Coupling E(v= 0, J= 1)= -809.4112 to E(v= 0, J= 0)= -811.5192
      Moment matrix elements: <X** 0>= 0.9999997392 <X** 1>= 0.0315663983
      <X** 2>= 0.0043724231 <X** 3>= 0.0004274505 <X** 4>= 0.0000681864
FCF= 1.0000D+00 <M>= 9.93816D-01 d(E)= -2.11 A(Einst)= 9.6706D-07 s-1
+++++
For vibrational level v = 0 of Potential-1
      J E J E J E J E
-----
0 -811.519 1 -809.411
+++++
Coupling E(v= 1, J= 0)= -507.1696 to E(v= 0, J= 1)= -809.4112
      Moment matrix elements: <X** 0>= 0.0007038706 <X** 1>= 0.0555644965
      <X** 2>= 0.0061799007 <X** 3>= 0.0011668795 <X** 4>= 0.0002051809
FCF= 4.9543D-07 <M>=-1.02282D-02 d(E)= -302.24 A(Einst)= 9.0586D-04 s-1

```

```

+++++
Coupling E(v= 1, J= 1)= -505.3034 to E(v= 0, J= 0)= -811.5192
Moment matrix elements: <X** 0>= -0.0007036644 <X** 1>= 0.0554599642
<X** 2>= 0.0061672593 <X** 3>= 0.0011650604 <X** 4>= 0.0002048749
FCF= 4.9514D-07 <M>=-1.16152D-02 d(E)= -306.22 A(Einst)= 4.0496D-04 s-1
+++++
Coupling E(v= 1, J= 1)= -505.3034 to E(v= 0, J= 2)= -805.1964
Moment matrix elements: <X** 0>= 0.0014096830 <X** 1>= 0.0556429498
<X** 2>= 0.0062065908 <X** 3>= 0.0011721096 <X** 4>= 0.0002064159
FCF= 1.9872D-06 <M>=-9.53729D-03 d(E)= -299.89 A(Einst)= 5.1293D-04 s-1
+++++
Coupling E(v= 1, J= 1)= -505.3034 to E(v= 1, J= 0)= -507.1696
Moment matrix elements: <X** 0>= 0.9999989173 <X** 1>= 0.1072309514
<X** 2>= 0.0227481313 <X** 3>= 0.0044941830 <X** 4>= 0.0010331538
FCF= 1.0000D+00 <M>= 9.79218D-01 d(E)= -1.87 A(Einst)= 6.5155D-07 s-1
+++++
For vibrational level v = 1 of Potential-1
J E J E J E J E J E
0 -507.170 1 -505.303
+++++
Coupling E(v= 2, J= 0)= -287.8379 to E(v= 0, J= 1)= -809.4112
Moment matrix elements: <X** 0>= -0.0001445686 <X** 1>= -0.0146868141
<X** 2>= 0.0021494006 <X** 3>= 0.0006270858 <X** 4>= 0.0002038462
FCF= 2.0900D-08 <M>= 2.85487D-03 d(E)= -521.57 A(Einst)= 3.6268D-04 s-1
+++++
Coupling E(v= 2, J= 0)= -287.8379 to E(v= 1, J= 1)= -505.3034
Moment matrix elements: <X** 0>= 0.0012364336 <X** 1>= 0.0838602021
<X** 2>= 0.0215270661 <X** 3>= 0.0065378301 <X** 4>= 0.0019424342
FCF= 1.5288D-06 <M>=-1.49150D-02 d(E)= -217.47 A(Einst)= 7.1749D-04 s-1
+++++
Coupling E(v= 2, J= 1)= -286.2356 to E(v= 0, J= 0)= -811.5192
Moment matrix elements: <X** 0>= 0.0001452962 <X** 1>= -0.0147714152
<X** 2>= 0.0021295981 <X** 3>= 0.0006226650 <X** 4>= 0.0002028174
FCF= 2.1111D-08 <M>= 3.16108D-03 d(E)= -525.28 A(Einst)= 1.5140D-04 s-1
+++++
Coupling E(v= 2, J= 1)= -286.2356 to E(v= 0, J= 2)= -805.1964
Moment matrix elements: <X** 0>= -0.0002889435 <X** 1>= -0.0146592358
<X** 2>= 0.0021551720 <X** 3>= 0.0006307829 <X** 4>= 0.0002051064
FCF= 8.3488D-08 <M>= 2.70514D-03 d(E)= -518.96 A(Einst)= 2.1384D-04 s-1
+++++
Coupling E(v= 2, J= 1)= -286.2356 to E(v= 1, J= 0)= -507.1696
Moment matrix elements: <X** 0>= -0.0012355622 <X** 1>= 0.0834456954
<X** 2>= 0.0214393061 <X** 3>= 0.0065170119 <X** 4>= 0.0019370216
FCF= 1.5266D-06 <M>=-1.73066D-02 d(E)= -220.93 A(Einst)= 3.3767D-04 s-1
+++++
Coupling E(v= 2, J= 1)= -286.2356 to E(v= 1, J= 2)= -501.5723
Moment matrix elements: <X** 0>= 0.0024778195 <X** 1>= 0.0841137737
<X** 2>= 0.0216303217 <X** 3>= 0.0065743346 <X** 4>= 0.0019558701
FCF= 6.1396D-06 <M>=-1.37213D-02 d(E)= -215.34 A(Einst)= 3.9306D-04 s-1
+++++
Coupling E(v= 2, J= 1)= -286.2356 to E(v= 2, J= 0)= -287.8379
Moment matrix elements: <X** 0>= 0.9999970447 <X** 1>= 0.2089561346
<X** 2>= 0.0661950523 <X** 3>= 0.0214010286 <X** 4>= 0.0074711130
FCF= 9.9999D-01 <M>= 9.60110D-01 d(E)= -1.60 A(Einst)= 3.9644D-07 s-1
+++++
For vibrational level v = 2 of Potential-1
J E J E J E J E J E
0 -287.838 1 -286.236
+++++
Find 3 Potential-1 vibrational levels with J= 0
v E(v) v E(v) v E(v) v E(v)
0 -811.5192 1 -507.1696 2 -287.8379
An n= 6 N-D theory extrapolation from last 2 levels implies vD = 6.812
=====

```

Case 1.c: For the same L.J.(12,6), find levels of potential with a barrier

```

=====
Generate ZMU= 15.17186628000(u) & BZ= 8.999999997D-01((1/cm-1)(1/Ang**2))
from atomic masses: 30.34373256000 & 30.34373256000(u)
Integrate from RMIN= 0.600 to RMAX= 20.00 with mesh RH= 0.002000(Angst)
Potential #1 for L1( 0)-J2( 0)
=====
State has OMEGA= 0 and energy asymptote: Y(lim)= 0.0000(cm-1)
Lennard-Jones(12, 6) potential with De= 1000.000(cm-1) Re = 1.000000(A)
Calculate properties of the single potential described above
Potential-1 uses inner boundary condition of zero value at RMIN
Eigenvalue convergence criterion is EPS= 1.0D-06(cm-1)
Airy function at 3-rd turning point is quasibound outer boundary condition
State-1 electronic angular momentum OMEGA= 0
yields centrifugal potential [J*(J+1) - 0.00]/r**2
For J= 18, find the first 400 levels of Potential-1 with VLIM= 0.0000
Solve for v= 0 J= 18 ETRIAL= -4.6535526D+02 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
1 -4.6535526D+02 4.87D-03 1.37D+02 9.86D+00 258 1.11 8.3D-27 2.5D-13 2.54 1 178
2 -4.5549824D+02 -4.11D-04 1.72D+02 -6.65D-01 260 1.12 1.0D-26 3.6D-13 2.54 1 177
3 -4.5616312D+02 -2.49D-06 1.70D+02 -4.08D-03 260 1.12 9.9D-27 3.5D-13 2.54 1 177

```

```

4 -4.5616720D+02 -9.36D-11 1.70D+02 -1.53D-07 260 1.12 9.9D-27 3.5D-13 2.54 1 177
-----
E(v= 0,J= 18)= -456.1672 4 Iter R(M)= 1.12 WF(NBEG= 52)/WF(M)= 9.9D-27
              INNER= 0 WF(NEND= 799)/WF(M)= 3.5D-13
Single well ICOR= 0: E(v= 0,J= 18)= -456.17 v(SC)= 0.002 dGdv= 308.355
              (vD-v)= 5.0609 E(next)= -2.0473D+02

Solve for v= 1 J= 18 ETRIAL= -2.0472865D+02 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 -2.0472865D+02 5.76D-03 1.62D+02 9.86D+00 328 1.25 -1.7D-26 3.9D-13 3.20 1 164
2 -1.9487134D+02 -7.70D-04 2.30D+02 -9.31D-01 332 1.26 -2.2D-26 4.8D-13 3.23 1 164
3 -1.9580231D+02 -1.04D-05 2.17D+02 -1.33D-02 331 1.26 -2.1D-26 4.5D-13 3.23 1 164
4 -1.9581562D+02 -1.98D-09 2.17D+02 -2.54D-06 331 1.26 -2.1D-26 4.5D-13 3.23 1 164
5 -1.9581563D+02 -8.60D-16 2.17D+02 -1.10D-12 331 1.26 -2.1D-26 4.5D-13 3.23 1 164
-----
E(v= 1,J= 18)= -195.8156 5 Iter R(M)= 1.26 WF(NBEG= 51)/WF(M)= -2.1D-26
              INNER= 0 WF(NEND= 1066)/WF(M)= 4.5D-13
Single well ICOR= 0: E(v= 1,J= 18)= -195.82 v(SC)= 1.003 dGdv= 213.103
              (vD-v)= 3.6578 E(next)= -3.5663D+01

Solve for v= 2 J= 18 ETRIAL= -3.5662990D+01 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 -3.5662990D+01 6.66D-03 1.74D+02 1.06D+01 414 1.43 2.0D-26 7.2D-13 4.94 1 159
2 -2.5053068D+01 -1.48D-03 3.06D+02 -1.35D+00 424 1.45 3.0D-26 7.3D-13 5.33 1 158
3 -2.6398794D+01 -4.76D-05 2.74D+02 -4.83D-02 422 1.44 2.8D-26 5.1D-13 5.33 1 158
4 -2.6447106D+01 -5.30D-08 2.73D+02 -5.39D-05 422 1.44 2.8D-26 5.1D-13 5.33 1 158
5 -2.6447160D+01 -7.96D-14 2.73D+02 -8.10D-11 422 1.44 2.8D-26 5.1D-13 5.33 1 158
-----
E(v= 2,J= 18)= -26.4472 5 Iter R(M)= 1.44 WF(NBEG= 50)/WF(M)= 2.8D-26
              INNER= 0 WF(NEND= 1780)/WF(M)= 5.1D-13
Single well ICOR= 0: E(v= 2,J= 18)= -26.45 v(SC)= 2.002 dGdv= 126.808
              (vD-v)= 2.1401 E(next)= 5.0337D+01

Solve for v= 3 J= 18 ETRIAL= 5.0336902D+01 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 5.0336902D+01 6.91D-03 1.97D+02 9.74D+00 550 1.70 -1.5D-26 1.0D-01 2.58 1 156
2 6.0080718D+01 -2.58D-03 5.09D+02 -1.41D+00 607 1.81 -2.8D-26 4.6D-01 2.24 1 156
3 5.8672116D+01 -2.43D-04 3.96D+02 -1.70D-01 595 1.79 -2.3D-26 3.8D-01 2.29 1 156
4 5.8501795D+01 -8.68D-06 3.87D+02 -6.23D-03 594 1.79 -2.3D-26 3.7D-01 2.30 1 156
5 5.8495566D+01 -2.91D-07 3.87D+02 -2.09D-04 594 1.79 -2.3D-26 3.7D-01 2.30 1 156
6 5.8495357D+01 -9.70D-09 3.87D+02 -6.96D-06 594 1.79 -2.3D-26 3.7D-01 2.30 1 156
7 5.8495350D+01 -3.23D-10 3.87D+02 -2.32D-07 594 1.79 -2.3D-26 3.7D-01 2.30 1 156
-----
E(v= 3,J= 18)= 58.4954 7 Iter R(M)= 1.79 WF(NBEG= 51)/WF(M)= -2.3D-26
              INNER= 0 WF(NEND= 849)/WF(M)= 3.7D-01
Lifetime= 5.035D-12(s) Width= 1.054D+00 dG/dv= 41.60 V(max)= 64.01
Single well ICOR= 0: E(v= 3,J= 18)= 58.50 v(SC)= 2.983 dGdv= 44.289
              (vD-v)= 0.2492 E(next)= 1.3932D+01
Find highest level of this potential is E(v= 3)= 5.8495350262D+01
ALF finds the highest calculated level is E(v= 3)= 58.495350

Solve for v= 0 J= 18 ETRIAL= -4.5616720D+02 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 -4.5616720D+02 -9.36D-11 1.70D+02 -1.53D-07 260 1.12 9.9D-27 3.5D-13 2.54 1 177
-----
E(v= 0,J= 18)= -456.1672 1 Iter R(M)= 1.12 WF(NBEG= 52)/WF(M)= 9.9D-27
              INNER= 0 WF(NEND= 799)/WF(M)= 3.5D-13

Solve for v= 1 J= 18 ETRIAL= -1.9581563D+02 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 -1.9581563D+02 -8.60D-16 2.17D+02 -1.10D-12 331 1.26 -2.1D-26 4.5D-13 3.23 1 164
-----
E(v= 1,J= 18)= -195.8156 1 Iter R(M)= 1.26 WF(NBEG= 51)/WF(M)= -2.1D-26
              INNER= 0 WF(NEND= 1066)/WF(M)= 4.5D-13

Solve for v= 2 J= 18 ETRIAL= -2.6447160D+01 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 -2.6447160D+01 -7.94D-14 2.73D+02 -8.08D-11 422 1.44 2.8D-26 7.2D-13 5.27 1 158
-----
E(v= 2,J= 18)= -26.4472 1 Iter R(M)= 1.44 WF(NBEG= 50)/WF(M)= 2.8D-26
              INNER= 0 WF(NEND= 1780)/WF(M)= 7.2D-13

Solve for v= 3 J= 18 ETRIAL= 5.8495350D+01 INNER= 0 WF(1st) WF(NEND)
ITER ETRIAL F(E) DF(E) D(E) M R(M) /WF(M) /WF(M) R(NEND) NBEG ITP1
-----
1 5.8495350D+01 -3.23D-10 3.87D+02 -2.32D-07 594 1.79 -2.3D-26 3.7D-01 2.30 1 156
-----
E(v= 3,J= 18)= 58.4954 1 Iter R(M)= 1.79 WF(NBEG= 51)/WF(M)= -2.3D-26
              INNER= 0 WF(NEND= 849)/WF(M)= 3.7D-01
Lifetime= 5.035D-12(s) Width= 1.054D+00 dG/dv= 41.60 V(max)= 64.01

Find 4 Potential-1 vibrational levels with J= 18
v E(v) v E(v) v E(v) v E(v)
-----
0 -456.1672 1 -195.8156 2 -26.4472 3 58.4954
=====

```

Case 1.d: Levels of a pointwise double-minimum potential for 3(PI)g Cl2

Generate ZMU= 17.48442634000(u) & BZ= 1.037181808D+00((1/cm-1)(1/Ang**2))
from atomic masses: 34.96885268000 & 34.96885268000(u)

Integrate from RMIN= 1.600 to RMAX= 10.00 with mesh RH= 0.001000(Angst)

Potential #1 for Cl(35)-Cl(35)

=====

State has OMEGA= 0 and energy asymptote: Y(lim)= 95440.0000(cm-1)

Perform cubic spline interpolation over the 29 input points

To make input points Y(i) consistent with Y(lim), add Y(shift)= 0.0000

Scale input points: (distance)* 5.291772108D-01 & (energy)* 8.065544400D+03

to get required internal units [Angstroms & cm-1 for potentials]

r(i)	Y(i)	r(i)	Y(i)	r(i)	Y(i)
3.340000	10.60800000	4.170000	8.59500000	5.370000	6.96200000
3.400000	9.68300000	4.260000	8.70400000	5.510000	6.90800000
3.430000	9.22100000	4.310000	8.75800000	5.660000	6.88000000
3.490000	8.75800000	4.460000	8.43200000	5.800000	6.90800000
3.570000	8.29500000	4.560000	8.13400000	5.940000	6.98900000
3.660000	8.13200000	4.660000	7.88700000	6.000000	7.04400000
3.760000	8.05100000	4.800000	7.61500000	6.200000	7.12500000
3.860000	8.13200000	4.910000	7.42500000	6.400000	7.26100000
3.940000	8.26800000	5.000000	7.34300000	6.600000	7.42500000
4.030000	8.35000000	5.170000	7.12500000		

Extrapolate to X .le. 1.7992 with

Y= -5091495.587 +5.609801D+06 * exp(-4.542065D-02*X)

Extrapolate to X .GE. 3.3867 using

Y= 95440.0000 - [1.128770D+05/X**1 +1.377826D+05/X**3]

Calculate properties of the single potential described above

Potential-1 uses inner boundary condition of zero value at RMIN

Eigenvalue convergence criterion is EPS= 1.0D-06(cm-1)

Airy function at 3-rd turning point is quasibound outer boundary condition

State-1 electronic angular momentum OMEGA= 0

yields centrifugal potential [J*(J+1) - 0.00]/r**2

Solve for the 22 vibration-rotation levels of Potential-1:

(v,J) = (56, 0) (57, 0) (58, 0) (59, 0) (60, 0) (61, 0)

(62, 0) (63, 0) (64, 0) (65, 0) (66, 0) (67, 0)

(68, 0) (69, 0) (70, 0) (72, 0) (72, 0) (73, 0)

(74, 0) (75, 0) (76, 0) (77, 0)

Matrix element arguments are powers of the distance r (in Angstroms)

Coefficients of expansion for radial matrix element/expectation value argument:

1.000000D+00 -2.000000D-01 3.000000D-02 -4.000000D-03

Find 2 potential minima: Vmin= 64929.256 55490.947

at mesh points R = 1.99400 2.99500

Find 2 potential maxima: Vmax= 70645.436 84013.345

at mesh points R = 2.28400 10.00000

E(v= 56, J= 0)= 68214.438 <M(r)>= 0.4721236994 <KE>= 4580.196

<X** 1>= 3.63045127 <X** 2>= 13.66002892 <X** 3>= 52.89672853

E(v= 57, J= 0)= 68387.998 <M(r)>= 0.4695211055 <KE>= 4617.063

<X** 1>= 3.64732349 <X** 2>= 13.79516352 <X** 3>= 53.71727533

E(v= 58, J= 0)= 68476.426 <M(r)>= 0.6824612177 <KE>= 1651.393

<X** 1>= 2.04384519 <X** 2>= 4.18845602 <X** 3>= 8.60585634

E(v= 59, J= 0)= 68559.695 <M(r)>= 0.4669189761 <KE>= 4652.829

<X** 1>= 3.66408683 <X** 2>= 13.93033264 <X** 3>= 54.54340917

E(v= 60, J= 0)= 68729.539 <M(r)>= 0.4643045290 <KE>= 4686.656

<X** 1>= 3.68083020 <X** 2>= 14.06617696 <X** 3>= 55.37868492

E(v= 61, J= 0)= 68897.547 <M(r)>= 0.4617011736 <KE>= 4718.810

<X** 1>= 3.69738156 <X** 2>= 14.20154715 <X** 3>= 56.21723207

E(v= 62, J= 0)= 69063.700 <M(r)>= 0.4591313914 <KE>= 4749.186

<X** 1>= 3.71357576 <X** 2>= 14.33535705 <X** 3>= 57.05354180

E(v= 63, J= 0)= 69214.948 <M(r)>= 0.6781183397 <KE>= 1985.878

<X** 1>= 2.07712475 <X** 2>= 4.36259568 <X** 3>= 9.33364510

E(v= 64, J= 0)= 69228.156 <M(r)>= 0.4588457942 <KE>= 4751.128

<X** 1>= 3.71260668 <X** 2>= 14.36531493 <X** 3>= 57.39807961

E(v= 65, J= 0)= 69390.488 <M(r)>= 0.4540380379 <KE>= 4803.188

<X** 1>= 3.668128392 <X** 2>= 14.60094615 <X** 3>= 58.73339038

E(v= 66, J= 0)= 69551.030 <M(r)>= 0.4516314964 <KE>= 4825.287

<X** 1>= 3.75994565 <X** 2>= 14.72704336 <X** 3>= 59.54766857

E(v= 67, J= 0)= 69709.530 <M(r)>= 0.4496202586 <KE>= 4837.877

<X** 1>= 3.77158473 <X** 2>= 14.83463802 <X** 3>= 60.27548393

E(v= 68, J= 0)= 69864.123 <M(r)>= 0.4631813287 <KE>= 4633.896

<X** 1>= 3.66813651 <X** 2>= 14.22170337 <X** 3>= 57.46061741

E(v= 69, J= 0)= 69900.180 <M(r)>= 0.6579325788 <KE>= 2352.457

<X** 1>= 2.22764740 <X** 2>= 5.25807538 <X** 3>= 13.57005061

E(v= 70, J= 0)= 70022.038 <M(r)>= 0.4464976859 <KE>= 4867.346

<X** 1>= 3.78784876 <X** 2>= 15.00980661 <X** 3>= 61.55668989

E(v= 72, J= 0)= 70321.487 <M(r)>= 0.4485338715 <KE>= 4776.940

<X** 1>= 3.76631221 <X** 2>= 14.93793365 <X** 3>= 61.58542409

E(v= 72, J= 0)= 70321.487 <M(r)>= 0.4485338715 <KE>= 4776.940

```

<X** 1>= 3.76631221 <X** 2>= 14.93793365 <X** 3>= 61.58542409
-----
E(v= 73, J= 0)= 70453.823 <M(r)>= 0.5032628690 <KE>= 4057.676
<X** 1>= 3.36247828 <X** 2>= 12.39895889 <X** 3>= 49.05256070
-----
E(v= 74, J= 0)= 70524.503 <M(r)>= 0.5646387149 <KE>= 3567.119
<X** 1>= 2.91080360 <X** 2>= 9.57364570 <X** 3>= 35.10248406
-----
E(v= 75, J= 0)= 70634.089 <M(r)>= 0.4560303262 <KE>= 4812.997
<X** 1>= 3.70350160 <X** 2>= 14.63177900 <X** 3>= 60.55568119
-----
E(v= 76, J= 0)= 70771.143 <M(r)>= 0.4453926382 <KE>= 4880.577
<X** 1>= 3.77812257 <X** 2>= 15.13371060 <X** 3>= 63.24854142
-----
E(v= 77, J= 0)= 70905.689 <M(r)>= 0.4521314033 <KE>= 4783.371
<X** 1>= 3.72580681 <X** 2>= 14.82855323 <X** 3>= 61.89095769
=====

```

E. Input file for Cases 2a) – 2e)

Note that entries after the “%” sign on each line of the input data file are comments identifying the variables, and are ignored by the program.

```

47 109 1 2 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 2.a: X-state {109}AgD using EM03(7,7) potential [JCP 123, 204304(2005)]'
0.0010 0.8 30.0 1.d-6 % RH RMIN RMAX EPS
0 0 0 19250.d0 0.d0 % NTP LPPOT OMEGA VLM
3 3 3 0 7 1 % IPOTL PPAR QPAR APSE Nbeta IBOB
19250.d0 1.6179162d0 0.d0 % DSCM REQ Rref
1.54358095D+00 3.73860D-02 1.66424D-01 9.8030D-02 1.7089D-01 6.0200D-02
1.4000D-01 2.2400D-01
107 1 3 3 -1 3 3 -1 2 % MN1R MN2R PAD QAD NU1 NU2 QNA NT1 NT2
0.0D+0 0.0D+0 1.175D+01 1.756D+01 -1.270D+01 % U2INF UA2(0) UA2(1) ...
0.0D+0 0.0D+0 1.5D-04 9.3D-04 % T2INF TA2(0) TA2(1) ...
-55 1 2 -1 0 1 -1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0 % IV(1) IJ(1)

12 24 1 2 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 2.b: Properties of MgD from final recommended 2013 MLR potential for MgH'
0.0005d0 0.50 99.d0 1.d-05 % RH RMIN RMAX EPS
0 0 0 0.0d0 % NTP LPPOT IOMEG VLIM
4 5 4 -1 12 1 % IPOTL PPAR QPAR APSE Nbeta IBOB
11104.25d0 1.7296854D0 2.74d0 % DSCM REQ Rref
3 0.81 -2 1 % NCMM rhoAB IVSR IDSTT
6 2.77550D+05
8 3.45490D+06
10 4.61400D+07
1.170475460D+00 1.08015790D+00 2.67329710D+00 2.483590D+00
7.401300000D-01 1.91470000D-01 6.03830000D-01 -2.48730D+00
-7.665300000D+00 -5.7310000D+00 2.83900000D+00 6.05400D+00
2.400000000D+00 % PARM(i=0,13)
24 1 6 4 3 13 4 -1 7 % MN1R MN2R PAD QAD NU1 NU2 QNA NT1 NT2
0.d0 1.310D+00 4.180D+00 2.670D+00 4.0D+00 % U1inf, U1i(i)
0.d+00 -1.5183D+01 3.7524D+01 -1.1607D+01 3.2200D+01 6.0570D+01 2.2950D+02
-1.5610D+03 -2.0130D+03 1.2530D+04 5.3600D+03 -4.9740D+04 1.0500D+04
7.110D+04 -4.70D+04 % U2inf, U2i(i)
0.d0 0.000D+00 7.266D-04 2.700D-04 9.110D-04 2.970D-03 -1.900D-03
-7.200D-03 2.300D-02 % T2inf, T2i(i)
-99 1 1 -1 0 1 1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0 % IV(1) IJ(1)

3 7 3 7 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 2.c: Modified Huang/Le Roy DELR(8) for Li2(B) [JCP 119, 7398 (2003)]'
0.005 1.5 60. 1.d-8 % RH RMIN RMAX EPS
-1 0 1 0.d0 % NTP LPPOT IOMEG VLIM
5 3 3 -1 8 1 % IPOTL PPAR QPAR APSE Nbeta IBOB
2986.600d0 2.935961d0 3.6d0 % DSCM REQ Rref
4 0.54d0 -0 1 % NCMM rhoAB IVSR IDSTT
3 -1.788d5 6 6.97586d6 8 1.378d8 10 3.445d9 % MMCM(j) CMM(j)
1.0585149D+00 3.70878D-01 2.77660D-01 1.1109D-01 -1.3076D-01 -2.805D-01
-8.69D-02 3.29D-01 1.61D-01
7 7 3 3 2 2 3 -1 -1 % MN1R MN2R PAD QAD NU1 NU2 QNA NT1 NT2
1.05578d0 2.52D-01 -3.29D+00 1.40D+00 % U1INF U1(0) U1(1)
1.05578d0 2.52D-01 -3.29D+00 1.40D+00 % U2INF U2(0) U2(1)
-99 1 2 0 0 1 -1 0 % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0 % IV(1) IJ(1)

10 20 36 84 0 1 % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 2.d: Aziz HFD-B(6,8,10,12) for {20}Ne{84}Kr [JCP 91, 6348 (1989)]'
0.010 2.0 99. 1.d-08 % RH RMIN RMAX EPS
-1 00 0 0.d0 49.75d0 % NTP LPPOT IOMEG VLIM
6 0 0 0 0 0 % IPOTL PPAR QPAR APSE Nbeta IBOB
49.75d0 3.621d0 0.d0 % DSCM REQ Rref
4 0.d0 -2 1 % NCMM rhoAB IVSR IDSTT
6 1.16162375d0 8 0.39191564d0
10 0.35284381d0 12 0.29724436d0
1.d0 0.9365d0 2.d0 0.71720676d0 0.d0 % MMLR(i) CMM(i) for i=1,2
% alpha_i (i=1-3) beta2 gamma

```

```

-99 1 2 0 0 1 -1 0          % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRW
0 0                          % IV(i) IJ(i)

20 40 20 40 0 1          % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 2.e: "Tiemann-type" potential for Ca2(X) [Phys.Rev. A66, 042503(2002)]'
0.0005 3.0 80. 1.d-08      % RH RMIN RMAX EPS
-1 0 0 1102.096077d0       % NTP LPPOT IOMEG V LIM
8 1 1 -1 20 0             % IPOTL PPAR QPAR APSE Nbeta IBOB
1102.096077d0 4.277277d0 0.d0 % DSCM REQ Rref
2 0.d0 -2 1              % NCMM rhoAB IVSR IDSTT
6 -1.074d7 8 -2.4505d8    % MMLR(j) CMM(j) j=1,NCMM
0.00042747d0
-0.254083092764773077d01 0.379611002601149221d04 0.382070302022495241d03
-0.274390396954679318d04 -0.322736334190800926d04 0.363113805693018548d03
0.634370542189755270d04 -0.740151835960846893d04 -0.190738913003729067d05
0.542347392433017594d05 0.440392304373011066d05 -0.155387944954526116d06
-0.836628381353236182d05 0.213831067083156871d06 0.155922449222826835d06
-0.156260872999483137d06 -0.146711120959219668d06 0.277542999772230869d05
0.712908015579339117d05 -0.126115550408998979d04
-0.5929d0 3.66d0 10.d0    % b R{inn} R{out}
-99 1 2 0 0 1 -1 0        % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0                      % IV(i) IJ(i)

```

F. Input file for Case 3

Note that entries after the “%” sign on each line of the input data file are comments identifying the variables, and are ignored by the program.

```

35 79 35 79 0 2          % IAN1 IMN1 IAN2 IMN2 CHARGE NUMPOT
'Case 3: Predict emission for B-X Br2 based on Gerstenkorn (1987) constants'
0.0015 1.85 15.0 1.d-4    % RH RMIN RMAX EPS
123 0 0 19742.072d0       % NTP LPPOT OMEGA V LIM
10 0 4 5 1.8D+05          % NUSE IR2 ILR NCN CNN
1.D0 1.D0 15902.4802d0     % RFACT EFAC VSHIFT & B-state turning points
2.32483352887883 3834.348717 2.32492725139916 3831.646866
2.32505070817846 3828.090070 2.32520875281731 3823.540508
2.32540641247184 3817.856430 2.32564885907544 3810.893306
..... skip 56 lines listing 112 more turning points .....
6.87678817820732 3817.856430 7.25492791418183 3823.540508
7.69918940093639 3828.090070 8.22965127226713 3831.646866
8.87579778305112 3834.348717

47 0 0 16056.926D0        % NTP2 LPPOT2 OMEGA2 V LIM2
10 0 0 6 0.D0            % NUSE2 IR22 ILR2 NCN2 CNN2
1.D0 1.D0 0.d0           % RFACT EFAC VSHIFT2
2.05649830399176 4483.356304 2.06283295581950 4189.629584
2.06951602200031 3893.539190 2.07659093472317 3595.104297
2.08411094486653 3294.343327 2.09214251282019 2991.273984
..... skip 18 lines listing 36 more turning points .....
2.56851940726203 3294.343327 2.58514008764203 3595.104297
2.60140908192717 3893.539190 2.61738272682198 4189.629584
2.63310768612594 4483.356304

-40 1 0 -4 130 1 -1 0    % NLEV1 AUTO1 LCDC LXPCT NJM JDJR IWR LPRWF
0 0                      % IV(i) IJ(i)
1 0 1.D0                % MORDR IRFN DREF
-0.219 0.265            % DM(0) DM(1)
15 1 -1 +1 2            % NLEV2 AUTO2 J2DL J2DU J2DD
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 % {IV2(i)}

```

G. Channel-6 Output file for Case 3

```

Case 3: Predict emission for B-X Br2 based on Gerstenkorn (1987) constants
=====
Generate ZMU= 39.45916880000(u) & BZ= 2.340730616D+00((1/cm-1)(1/Ang**2))
from atomic masses: 78.91833760000 & 78.91833760000(u)
Integrate from RMIN= 1.850 to RMAX= 15.00 with mesh RH= 0.001500(Angst)

Potential #1 for Br( 79)-Br( 79)
=====
State has OMEGA= 0 and energy asymptote: Y(lim)= 19742.0720(cm-1)
Perform 10-point piecewise polynomial interpolation over 123 input points
Beyond read-in points extrapolate to limiting asymptotic behaviour:
Y(r) = Y(lim) - ( 0.1800000D+06)/r** 5
To make input points Y(i) consistent with Y(lim), add Y(shift)= 15902.4802
Scale input points: (distance)* 1.000000000D+00 & (energy)* 1.000000000D+00
to get required internal units [Angstroms & cm-1 for potentials]
r(i) Y(i) r(i) Y(i) r(i) Y(i)
-----

```

```

2.32483353 3834.3487 2.42191370 1696.7527 3.22507343 1821.0440
2.32492725 3831.6469 2.43001839 1568.5519 3.26089292 1941.3789
..... skip 37 lines of this listing .....
2.40751394 1941.3789 3.15509529 1568.5519 8.22965127 3831.6469
2.41443578 1821.0440 3.18985642 1696.7527 8.87579778 3834.3487
-----
Extrapolate to X .le. 2.3249 with
Y= 14763.263 +3.552602D+09 * exp(-5.797858D+00*X)
Function for X .GE. 8.230 generated by 4-point inverse-power interpolation
with leading term 1/r**5 relative to dissociation limit YLIM= 19742.072
and (dimensionless) leading coefficient fixed as C5= 180000.00
-----
Get matrix elements between levels of Potential-1 (above) & Potential-2 (below)
-----
For Potential #2:
=====
State has OMEGA= 0 and energy asymptote: Y(lim)= 16056.9260(cm-1)
Perform 10-point piecewise polynomial interpolation over 47 input points
To make input points Y(i) consistent with Y(lim), add Y(shift)= 0.0000
Scale input points: (distance)* 1.000000000D+00 & (energy)* 1.000000000D+00
to get required internal units [Angstroms & cm-1 for potentials]
-----
r(i) Y(i) r(i) Y(i) r(i) Y(i)
-----
2.05649830 4483.3563 2.18626063 646.2909 2.42013527 966.1727
2.06283296 4189.6296 2.19810575 485.5309 2.43259119 1125.2912
2.06951602 3893.5392 2.21248391 324.2269 2.44438269 1283.8592
2.07659093 3595.1043 2.22316187 227.1840 2.45563605 1441.8751
2.08411094 3294.3433 2.23177061 162.3804 2.46644318 1599.3372
2.09214251 2991.2740 2.24253421 97.4903 2.47687343 1756.2436
2.10077036 2685.9133 2.25851483 32.5137 2.49680783 2068.3827
2.11010533 2378.2776 2.28102606 0.0000 2.51575473 2378.2776
2.12029713 2068.3827 2.30436129 32.5137 2.53392724 2685.9133
2.13155647 1756.2436 2.32199230 97.4903 2.55147764 2991.2740
2.13767849 1599.3372 2.33440900 162.3804 2.56851941 3294.3433
2.14419655 1441.8751 2.34467357 227.1840 2.58514009 3595.1043
2.15117974 1283.8592 2.35784040 324.2269 2.60140908 3893.5392
2.15871971 1125.2912 2.37638051 485.5309 2.61738273 4189.6296
2.16694249 966.1727 2.39240504 646.2909 2.63310769 4483.3563
2.17602973 806.5054 2.40683303 806.5054
-----
Extrapolate to X .le. 2.0628 with
Y= -2292.367 +1.200354D+10 * exp(-6.996051D+00*X)
Function for X .GE. 2.6174 generated as
Y= 16056.9260 - ( 1.657906D+06 ) * r** 22.046446 * exp{-( 9.991684*r)}
-----
Potential-1 uses inner boundary condition of zero value at RMIN
Eigenvalue convergence criterion is EPS= 1.0D-04(cm-1)
Airy function at 3-rd turning point is quasibound outer boundary condition
State-1 electronic angular momentum OMEGA= 0
yields centrifugal potential [J*(J+1) - 0.00]/r**2
For J= 0, find the first 41 levels of Potential-1 with VLIM=*****
and automatically increment J in steps of 1 to a maximum value of 999
Matrix element arguments are powers of the distance r (in Angstroms)
Coefficients of expansion for radial matrix element/expectation value argument:
-2.190000D-01 2.650000D-01
Potential-2 uses inner boundary condition of zero value at RMIN
Using the rotational selection rule: delta(J)= -1 to 1 with increment 2
calculate matrix elements for coupling to the 15 vibrational levels of
Potential-2: v = 0 1 2 3 4 5 6 7 8 9 10 11 12 13
14
State-2 electronic angular momentum OMEGA= 0
yields centrifugal potential [J*(J+1) - 0.00]/r**2
-----
..... skip 2727 lines summarizing results for v=0-33 .....
-----
*** For J=100 E= 19743.24 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=100 E= 19743.22 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=100 E= 19743.22 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 99.17 for R > R(max)= 15.00
*** For J=101 E= 19748.69 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=101 E= 19748.67 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=101 E= 19748.67 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 100.17 for R > R(max)= 15.00
*** For J=102 E= 19754.16 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=102 E= 19754.15 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=102 E= 19754.15 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 101.18 for R > R(max)= 15.00
*** For J=103 E= 19759.66 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=103 E= 19759.65 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J=103 E= 19759.65 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 102.19 for R > R(max)= 15.00
For J=127 ETRY= 19891.7867 > VMAX= 19891.7746 find onee turn point: R= 2.37
*** SCHRQ has a convergence problem, so for IT=10 cut DE= -3.51D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=20 cut DE= -3.39D-03 in HALF

```

```

*** SCHRQ has a convergence problem, so for IT=26 cut DE= -3.39D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=28 cut DE= -3.39D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=29 cut DE= 1.70D-03 in HALF
*** CAUTION for v= 34 J=127 SCHRQ doesn't converge by ITER=30 DE= 1.01D-03
*** SCHRQ has a convergence problem, so for IT= 7 cut DE= 1.80D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=27 cut DE= -3.39D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=28 cut DE= 1.70D-03 in HALF
*** SCHRQ has a convergence problem, so for IT=30 cut DE= -3.52D-03 in HALF
*** CAUTION for v= -1 J=127 SCHRQ doesn't converge by ITER=30 DE=-1.76D-03

```

For vibrational level		v = 34		of Potential-1							
J	E	J	E	J	E	J	E	J	E	J	E
0	19443.803	26	19466.021	52	19529.969	78	19632.179	104	19765.173		
1	19443.867	27	19467.723	53	19533.224	79	19636.793	105	19770.719		
2	19443.994	28	19469.486	54	19536.537	80	19641.452	106	19776.284		
3	19444.185	29	19471.312	55	19539.905	81	19646.158	107	19781.868		
4	19444.439	30	19473.198	56	19543.330	82	19650.908	108	19787.469		
5	19444.756	31	19475.147	57	19546.811	83	19655.702	109	19793.083		
6	19445.138	32	19477.157	58	19550.347	84	19660.539	110	19798.709		
7	19445.582	33	19479.228	59	19553.939	85	19665.420	111	19804.345		
8	19446.090	34	19481.360	60	19557.585	86	19670.343	112	19809.988		
9	19446.662	35	19483.553	61	19561.286	87	19675.307	113	19815.634		
10	19447.297	36	19485.806	62	19565.042	88	19680.311	114	19821.282		
11	19447.995	37	19488.120	63	19568.851	89	19685.356	115	19826.927		
12	19448.756	38	19490.495	64	19572.714	90	19690.440	116	19832.565		
13	19449.581	39	19492.930	65	19576.629	91	19695.561	117	19838.191		
14	19450.468	40	19495.425	66	19580.598	92	19700.721	118	19843.802		
15	19451.419	41	19497.979	67	19584.619	93	19705.916	119	19849.390		
16	19452.433	42	19500.594	68	19588.692	94	19711.148	120	19854.948		
17	19453.510	43	19503.267	69	19592.816	95	19716.413	121	19860.467		
18	19454.650	44	19506.000	70	19596.992	96	19721.713	122	19865.936		
19	19455.852	45	19508.792	71	19601.218	97	19727.044	123	19871.337		
20	19457.117	46	19511.643	72	19605.494	98	19732.407	124	19876.643		
21	19458.445	47	19514.552	73	19609.820	99	19737.800	125	19881.820		
22	19459.836	48	19517.520	74	19614.196	100	19743.222	126	19886.844		
23	19461.289	49	19520.546	75	19618.620	101	19748.672				
24	19462.804	50	19523.629	76	19623.092	102	19754.148				
25	19464.381	51	19526.770	77	19627.612	103	19759.649				

```

..... skip 230 lines summarizing results for v= 35-39 .....
*** For J= 74 E= 19742.97 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 74 E= 19742.96 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 74 E= 19742.96 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 72.87 for R > R(max)= 15.00
*** For J= 75 E= 19746.03 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 75 E= 19746.02 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 75 E= 19746.02 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 73.89 for R > R(max)= 15.00
*** For J= 76 E= 19749.10 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 76 E= 19749.09 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 76 E= 19749.09 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 74.90 for R > R(max)= 15.00
*** For J= 77 E= 19752.19 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 77 E= 19752.18 R(3-rd) > RMAX & E < V(N) so try WKB B.C. @ RMAX
*** For J= 77 E= 19752.18 R(3-rd) beyond range so tunneling calculation uses
pure centrifugal potential with J(app)= 75.92 for R > R(max)= 15.00
For J= 93 ETRY= 19800.6818 > VMAX= 19800.4705 find onee turn point: R= 2.35
For J= 93 ETRY= 19800.6892 > VMAX= 19800.4705 find onee turn point: R= 2.35
For J= 93 ETRY= 19800.6892 > VMAX= 19800.4705 find onee turn point: R= 2.35
For J= 93 ETRY= 19800.6892 > VMAX= 19800.4705 find onee turn point: R= 2.35

```

For vibrational level		v = 40		of Potential-1							
J	E	J	E	J	E	J	E	J	E	J	E
0	19616.873	19	19626.163	38	19652.625	57	19694.708	76	19749.092		
1	19616.922	20	19627.136	39	19654.470	58	19697.304	77	19752.181		
2	19617.020	21	19628.157	40	19656.358	59	19699.935	78	19755.283		
3	19617.168	22	19629.225	41	19658.289	60	19702.600	79	19758.395		
4	19617.364	23	19630.341	42	19660.263	61	19705.297	80	19761.516		
5	19617.609	24	19631.503	43	19662.279	62	19708.027	81	19764.642		
6	19617.904	25	19632.713	44	19664.337	63	19710.788	82	19767.771		
7	19618.247	26	19633.970	45	19666.437	64	19713.579	83	19770.899		
8	19618.639	27	19635.273	46	19668.577	65	19716.401	84	19774.022		
9	19619.081	28	19636.623	47	19670.759	66	19719.250	85	19777.138		
10	19619.571	29	19638.019	48	19672.980	67	19722.128	86	19780.239		
11	19620.109	30	19639.461	49	19675.242	68	19725.033	87	19783.320		
12	19620.697	31	19640.949	50	19677.543	69	19727.963	88	19786.373		
13	19621.333	32	19642.482	51	19679.882	70	19730.918	89	19789.385		
14	19622.017	33	19644.061	52	19682.260	71	19733.896	90	19792.337		
15	19622.750	34	19645.685	53	19684.676	72	19736.896	91	19795.203		
16	19623.531	35	19647.353	54	19687.130	73	19739.918	92	19797.968		
17	19624.360	36	19649.066	55	19689.620	74	19742.958				
18	19625.238	37	19650.824	56	19692.146	75	19746.017				

```
Find 41 Potential-1 vibrational levels with J= 0
```

v	E(v)	v	E(v)	v	E(v)	v	E(v)
0	15985.8099	11	17599.2328	22	18743.5515	33	19404.8742
1	16150.1163	12	17723.5255	23	18822.9413	34	19443.8033
2	16311.0672	13	17843.8623	24	18898.2750	35	19479.6566
3	16468.6043	14	17960.1998	25	18969.6004	36	19512.5503
4	16622.6712	15	18072.4992	26	19036.9760	37	19542.6009
5	16773.2113	16	18180.7266	27	19100.4706	38	19569.9259
6	16920.1678	17	18284.8542	28	19160.1627	39	19594.6434

An n= 5 N-D theory extrapolation from last 2 levels implies $v_D = 59.899$

Case 3: Predict emission for B-X Br₂ based on Gerstenkorn (1987) constants

dJ(J")	Band		E(lower)	E(2)-E(1)	A(Einstein)	F-C Factor	<v'j' M v"j">
	v'	v"					
P(1)	0 -	0	162.54	-15823.27	7.51655D-05	3.26301D-10	7.77793D-06
P(1)	0 -	1	485.69	-15500.12	1.98479D-03	9.03473D-09	4.12243D-05
P(1)	0 -	2	806.67	-15179.14	2.56291D-02	1.22429D-07	1.52860D-04
P(1)	0 -	3	1125.45	-14860.36	2.15688D-01	1.08216D-06	4.57792D-04
P(1)	0 -	4	1442.03	-14543.78	1.33030D+00	7.01609D-06	1.17424D-03
P(1)	0 -	5	1756.40	-14229.41	6.41089D+00	3.55737D-05	2.66365D-03
P(1)	0 -	6	2068.54	-13917.27	2.51332D+01	1.46864D-04	5.45245D-03
P(1)	0 -	7	2378.43	-13607.38	8.24026D+01	5.07537D-04	1.02119D-02
P(1)	0 -	8	2686.06	-13299.75	2.30523D+02	1.49801D-03	1.76763D-02
P(1)	0 -	9	2991.42	-12994.39	5.58669D+02	3.83406D-03	2.84933D-02
P(1)	0 -	10	3294.49	-12691.32	1.18684D+03	8.61075D-03	4.30264D-02
P(1)	0 -	11	3595.25	-12390.56	2.23108D+03	1.71302D-02	6.11533D-02
P(1)	0 -	12	3893.69	-12092.12	3.73965D+03	3.04191D-02	8.21222D-02
P(1)	0 -	13	4189.78	-11796.03	5.62413D+03	4.85201D-02	1.04526D-01
..... Remove 267521 lines							
R(91)	40 -	5	2427.17	-17370.80	4.55940D+01	3.22713D-04	-7.46819D-03
P(93)	40 -	5	2456.75	-17341.21	4.45961D+01	3.13842D-04	-7.36500D-03
R(91)	40 -	6	2736.53	-17061.44	1.63941D+01	1.20264D-04	4.60057D-03
P(93)	40 -	6	2765.99	-17031.98	1.74563D+01	1.27340D-04	4.73392D-03
R(91)	40 -	7	3043.63	-16754.34	8.11570D+01	6.22525D-04	1.05187D-02
P(93)	40 -	7	3072.96	-16725.00	8.14359D+01	6.21159D-04	1.05076D-02
R(91)	40 -	8	3348.45	-16449.52	6.05804D+00	4.87388D-05	2.95411D-03
P(93)	40 -	8	3377.66	-16420.31	5.45704D+00	4.36756D-05	2.79608D-03
R(91)	40 -	9	3650.98	-16146.99	4.46439D+01	3.71936D-04	-8.24583D-03
P(93)	40 -	9	3680.06	-16117.91	4.60071D+01	3.81203D-04	-8.34820D-03
R(91)	40 -	10	3951.19	-15846.78	4.35551D+01	3.79843D-04	-8.37720D-03
P(93)	40 -	10	3980.15	-15817.82	4.27258D+01	3.70610D-04	-8.27501D-03
R(91)	40 -	11	4249.08	-15548.89	4.25958D+00	3.84497D-05	2.69541D-03
P(93)	40 -	11	4277.91	-15520.06	4.87948D+00	4.38321D-05	2.87733D-03
R(91)	40 -	12	4544.62	-15253.35	5.56357D+01	5.28869D-04	1.00258D-02
P(93)	40 -	12	4573.32	-15224.64	5.60365D+01	5.29892D-04	1.00359D-02
R(91)	40 -	13	4837.80	-14960.17	6.82655D+00	6.82839D-05	3.61565D-03
P(93)	40 -	13	4866.37	-14931.60	6.15591D+00	6.12800D-05	3.42476D-03
R(91)	40 -	14	5128.60	-14669.37	2.97279D+01	3.08674D-04	-7.77062D-03
P(93)	40 -	14	5157.04	-14640.93	3.09159D+01	3.19393D-04	-7.90463D-03