CP-650R^2 August 2004

BCONT 2.2
A Computer Program for Calculating Bound→Continuum Transition Intensities for Diatomic Molecules

Robert J. Le Roy and Geoffrey T. Kraemer

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

Incomplete Draft Manual: Code & Manual "Under Construction"

University of Waterloo

Chemical Physics Research Report

BCONT 2.2
A Computer Program for Calculating Bound→Continuum Transition Intensities for Diatomic Molecules

Robert J. Le Roy and Geoffrey T. Kraemer

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 BCONT, which can solve the radial or one-dimensional Schröinger equation for bound and/or quasibound levels of any smooth single or double-minimum potential, and calculate photodissociation cross sections, ...

1 Introduction

This program is designed to calculate, and if desired, to fit to experimental data for bound → continuum transition intensities associated with any of:

Photodissociation, from a single initial level, a specified set of initial levels, or a thermal vibrational-rotational level population of some initial electronic state, into one or more final electronic states.

Spontaneous Emission, from a single initial level, a specified set of initial levels, or a thermal vibrational-rotational level population of some initial electronic state, into one or more final electronic states.

Predissociation from one or more levels of a specified initial electronic state into one or more final electronic states.

Fits performed using this program may be used to determine the repulsive portion of the final electronic state potential energy curve(s) and the associated transition moment or coupling function(s), and can simultaneously treat data for multiple isotopomers, as well as combinations of different types of observables (e.g., total and/or partial photodissociation cross sections, and/or branching ratios).

In the following, Section II describes more explicitly what calculations BCONT can perform, how the basic equation are solved, how the program functions, and outlines some of its options. Section III then states the input/output conventions, indicates the units assumed for the physical parameters of interest, and describes how to run the program on a UNIX 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 IV. Section V then describes the most significant differences between the current and earlier[1] versions of this program. Finally, the Appendix describes the structure of the program and the roles of its various subroutines, presents listings of illustrative sample data input files and of the resulting output, and indicates the CPU time those sample cases required.

The current version of the (extensively commented) source code for BCONT may be obtained by filling in a form accessed throught the "Computer Programs" link web page *http://leroy.uwaterloo.ca*. 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[2]. 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 March XX, 2001. Individuals currently utilizing older versions of this code [1] will likely find it desirable to obtain the current version. I would also appreciate having users inform me (e-mail: leroy@uwaterloo.ca) of any apparent errors or instabilities in the code, or of additional features which might appear desirable for future versions.

2 Properties Calculated by, and Capabilities of BCONT

The three types of properties which may be calculated using BCONT are schematically illustrated in Fig. 1, and described in the next three subsections.

2.1 Photodisociation or Continuum Absorption

The first of these properties is the photodissociation of a molecule in a discrete level with vibrational and rotational quantum numbers v and J, energy $E_{v,J}$ and unit normalized radial wavefunction $\psi_{v,J}(r)$, due to absorption of light of frequency ν, which drives a transition into a continuum level of final electronic state–s at energy $E = E_{v,J} + h\nu$ with radial wavefunction $\psi^{s}_{E,J'}(r)$. The photodissociation cross section for this process, in units [Å^2/molecule], is given by [3, 4, 5]:

$$\begin{aligned} \sigma_s(v,J;\nu) &= [8\,\pi^3\,\nu/3hc] \sum_{J'} \frac{S_J^{J'}}{2J+1} \left|\langle\psi_{E,J'}(r)|M_s(r)|\psi_{v,J}(r)\rangle\right|^2 \qquad (1) \\ &= 3.22696\times 10^{-4}\,\bar{\nu}\,\sqrt{\mu/K_s(E)}\,\sum_{J}^{\prime} \frac{S_J^{J'}}{2J+1} \left(\int_0^\infty \psi^*_{E,J'}(r)\,M_s(r)\,\psi_{v,J}(r)\,dr\right)^2 \end{aligned}$$

Here, $\bar{\nu} = \nu/c$ is the transition energy in cm^{-1}, $K_s(E) = [E - V_s(r=\infty)]$ is the asymptotic relative kinetic energy of the molecular fragments dissociating along the potential energy curve $V_s(r)$ of electronic state–s, in cm^{-1}, $S_J^{J'}$ is the usual Hönl-London rotational intensity factor [6], the reduced mass μ is in u, and the transition moment function $M_s(r)$ has units debye and incorporates the appropriate ratio of initial- to final-state electronic degeneracy factors. Note too that the second line of this expression assumes unit asymptotic normalization of the continuum radial wavefunction $\psi_{E,J'}(r)$ (see subsection II.X).

To facilitate comparisons with experiment, the quantity actually calculated by the program (and units assumed for data being fitted to) is the decadic molar extinction coefficient, in units $[\ell/\text{mole}\cdot\text{cm}]$:

$$\begin{aligned} \epsilon_s(v,J;\nu) &= N_A\,\sigma_s(v,J;\nu)\times 10^{-19}/\ln(10) \qquad (2) \\ &= 8.4397210\,\bar{\nu}\,\sqrt{\mu/K_s}\sum_{J'} \frac{S_J^{J'}}{2J+1} \left(\int_0^\infty \psi^*_{E,J'}(r)\,M_s(r)\,\psi_{v,J}(r)\,dr\right)^2 \end{aligned}$$

where N_A is Avogadro's number.

While Eq.(2) gives the absorption coefficient for transitions from the single initial-state level (v,J), the observable of interest is often the total absorption from a population of molecules in thermal equilibrium at some temperature T. This total thermal absorption coefficient may be written as

$$\epsilon_s(T;\nu) \;=\; \sum_v \sum_J F_{v,J}(T)\,\epsilon_s(v,j;\nu) \qquad (3)$$

where $F_{v,J}(T) = (2J+1)\,e^{-E_{v,J}/k_B T}/Q(T)$ is the fraction of the initial-state population in vibration-rotation level (v,J), and $Q(T) = \sum_v \sum_J (2J+1)\,e^{-E_{v,J}/k_B T}$ is the molecular partition function for levels of the initial electronic state.

Of course, Eqs. (2) and (3) are for partial absorption coefficients into a particular final electronic state, s. In many cases the natural observables would be the total absorption coefficients

$$\epsilon_{\rm tot}(v,J;\nu) \;=\; \sum_s \epsilon_s(v,J;\nu) \qquad (4)$$

$$\text{or} \qquad \epsilon_{\rm tot}(T;\nu) \;=\; \sum_s \epsilon_s(T;\nu) \qquad (5)$$

or a sum over some particular subset of these partial absorption coefficients, such as a sum over all final states with a common potential energy asymptote.

In any case, with an appropriate choice of input control parameter, BCONT will readily calculate any of these single-initial-state or thermal-population, partial or total cross sections.

2.2 Spontaneous Continuum Emission from (a) Discrete Level(s)

The second property considered herein is the Einstein coefficient for spontaneous emission of light of frequency ν from vibration-rotation level (v,J) with energy $E_{v,J}$ of the initial electronic state, into continuum levels with energy $E = E_{v,J} - h\nu$ of final electronic state–s. The rate coefficient for this process, in units $[\text{s}^{-1}/\text{cm}^{-1}]$ is given by:

$$\begin{aligned} A_s(v,J;\nu) &= \Big(64\,\pi^4\,\nu^3/3\,h\Big) \sum_{J'} \frac{S_J^{J'}}{2J+1} \left|\langle\psi_{E,J'}(r)|M_s(r)|\psi_{v,J}(r)\rangle\right|^2 \qquad (6) \\ &= 2.4313849\times 10^{-8}\,\bar{\nu}^3\,\sqrt{\mu/K_s}\sum_{J'} \frac{S_J^{J'}}{2J+1} \left(\int_0^\infty \psi^*_{E,J'}(r)\,M_s(r)\,\psi_{v,J}(r)\,dr\right)^2 \end{aligned}$$

This expression has essentially the same form as Eqs. (1) and (2), and the various variables have the same meaning. Once again, the summation over all spectroscopically allowed $J \rightarrow J'$ transitions cannot be experimentally broken down. However, the partial rates for emission into different final electronic states may or may not be experimentally separable. In the latter case, the observable would simply involve a sum over all participating final electronic states:

$$A_{\rm tot}(v,J;\nu) \;=\; \sum_s A_s(v,J;\nu) \tag{7}$$

Similarly, while the above expressions are for emission from a single (v,J) level, if the emission arises from a thermally equilibrated initial-state population, the total emission rate at a given frequency would be given by

$$A_{\rm tot}(T;\nu) \;=\; \sum_v \sum_J F_{v,J}(T)\, A_{\rm tot}(v,J;\nu) \tag{8}$$

where $F_{v,J}(T)$ is the fraction of the initial-state population in level (v,J), as defined above. Again, for an appropriate choice of input parameters BCONT will calculate any of these single or thermal initial-state, partial or total emission rates.

2.3 Predissociation

The third type of property which can be calculated or fitted to using BCONT is isoenergetic predissociation from one or more discrete levels of a bound initial-state potential energy curve into the continuum of one or more final electronic states whose potential asymptotes lie below the energies of the (v,J) levels in question. The "Golden rule" expression for such predissociation rates is, in units s^{-1}

$$\begin{aligned} k_s(v,J) \;&=\; \Big(4\,\pi^2/h\Big)\, \big|\langle \psi_{E,J'}(r)|M_s(r)|\psi_{v,J}(r)\rangle\big|^2 \\ &=\; 9.1755539\times 10^{10}\,\sqrt{\mu/K_s}\,\times \left(\int_0^\infty \psi^*_{E,J'}(r)\, M_s(r)\, \psi_{v,J}(r)\, dr\right)^2 \end{aligned} \tag{9}$$

While most variables appearing herein have the same significance as in Eqs. (1)–(6), the function $M_s(r)$ does not. For the case of predissociation this is a operator with units of energy (in the present case, cm^{-1}) describing the coupling between the initial and final electronic states. The precise nature of this operator depends on the nature of the two electronic states [5]: in some cases it is a simple function of r, in some cases a function of r multiplied by J or $J(J+1)$, and in some cases it involves a differential operator. The first and third of these possibilities are allowed for in the present version of BCONT; details are presented in Section III.XX.

The uncertainty principle tells us that the predissociation rate for a given level is reflected in a Lorentzian broadening with a full width at half maximum of

$$\Gamma_{\rm FWHM} \;=\; \sum_s k_s(v,J)/2\,\pi\, c \tag{10}$$

where c is the speed of light. If the observable is the level width itself, that width depends on the sum over contributions from rates into all accessible final-state channels, as shown in Eq. (10). In some cases, however, analysis of the velocity or angular distribution of the dissociation products may also allow partial widths into individual final states or subsets of final states to be determined. BCONT can perform calculations for any of these possibilities.

Note too that in predissociation processes the molecular rotational angular momentum must be conserved, so $J' = J$ and there is no sum over J' in the resulting rate expression.

2.4 Application of BCONT

BCONT may be used in one of two modes. The first is simply to perform "forward" calculations of a property or set of properties using a given initial-state potential and one or more sets of final-state potentials and transition moment or coupling functions. The second mode is to perform a least-squares fit to experimental values of one or more properties for one or more isotopomers in order to optimize parameters defining the analytic expressions for the repulsive final-state potentials and transition moment functions. Such fits are of course non-linear, and hence require realistic initial trial values of the requisite potential and transition moment (or coupling) function parameters. In either mode, the property or properties considered may depend on one or several different final electronic states. However, as currently configured, the program will only consider one initial electronic state at a time, though it does allow for the possibility of the associated initial-state potential being slightly different for different molecular isotopomers. Similarly, in the present version of BCONT the fitting mode will only optimize parameters describing the final electronic state(s); the initial-state potential is assumed to be fixed. This is also not a fundamental linitation, and may be relaxed in a later release.

In the least-squares fitting mode, it is particularly important to realize the a variety of properties may be used in a single fit; e.g. partial and total absorption coefficients for multiple isotopomers, as well as pranching ratios for dissociation into different final-state channels. An evocative illustration of this capability is seen in a recent analysis of UV photodissociation data for HI and DI in which four different final-state potential curves and transition moment functions were determined from fits to two sets of thermal total absorption coefficients and five different sets of experimental branching ratio measurements [7].

3 Computational Methodologies

3.1 Solving the Radial Schrödinger Equation

The core of the program is concerned with solving the radial or (effective) one-dimensional Schrödinger equation to determine the discrete eigenvalues and eigenfunctions of the specified initial-state potential, and to calculate the radial eigenfunctions in continuum above the asymptote(s) of the final electronic state(s). In wither case thus involves solving the differential equation

$$-\frac{\hbar^2}{2\mu}\,\frac{d^2\Psi_{v,J}(R)}{dR^2} \;+\; V_J(R)\,\Psi_{v,J}(R) \;=\; E_{v,J}\,\Psi_{v,J}(R) \tag{11}$$

where μ is the effective or reduced mass of the system, J the rotational quantum number, and the effective one-dimensional potential $V_J(R)$ is a sum of the rotationless (electronic) potential 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 $\Omega =$ IOMEG 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 IOMEG > 99 (see the discussion of data input statements #16 and 24 in Section IV), this term becomes $[J^2 - 1/4]\,\hbar^2/2\mu R^2$. The program also defines the reduced mass appearing in Eq. (11) as Watson's "charge-modified reduced mass",[8] $\mu_W = (M_A\,M_B)/(M_A{+}M_B{-}m_e\,Q)$, where M_A and M_B are the atomic masses of the two atoms, m_e is the electron mass, and $Q =$ CHARGE (see input data statement #1) the ± integer net charge on the molecule (ion).

The core of the calculation is the solution of Eq. (11) to determine the eigenvalues $E_{v,J}$ and eigenfunctions $\Psi_{v,J}(R)$ of the potential $V_J(R)$. This is done in subroutine SCHRQ, which is based on the famous Cooley-Cashion-Zare routines SCHR,[9, 10, 11, 12, 13] but incorporates special features such as the ability to automatically locate and calculate the widths of "quasibound" or tunneling-predissociation levels.[14, 15, 16] These are metastable states which lie above the dissociation limit, but whose dissociation is inhibited by a potential energy barrier.

The accuracy of the eigenfunctions and eigenvalues obtained is largely determined by the size of the (fixed) radial mesh RH (read on line #4 of the data file) used in the numerical integration of Eq. (11). 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 15 to 30 mesh points between adjacent wavefunction nodes in the classically allowed region. An appropriate mesh size may be estimated using the particle-in-a-box expression

$$\mathrm{RH} \;=\; \pi/\left(\mathrm{NPN}\times[(\mu/16.85762908)\times\max\{E-V(R)\}]^{1/2}\right) \tag{12}$$

where NPN is the selected minimum number of mesh points per wavefunction node (say 20), $\max\{E-V(R)\}$ is the maximum of the local kinetic energy (in cm^{-1}) for the levels under consideration (in general it is ≈ the potential well depth), and the numerical factor is identified below in Section III. A value of NPN which is too small yields results which are unreliable, while too large a value may require excessive computational effort or cause array dimensions to be exceeded. Thus, while Eq. (12) is a useful guide, a careful user should always empirically vary RH to determine the largest value which yields results of the accuracy desired for their particular application.

In general, the outward and inward numerical integration must start at distances RMIN and RMAX (data file line #4), respectively, which 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 that does occur a smaller RMIN or larger RMAX value might be needed to give 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. $0.7-0.8$ 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 (data file line #4) and the largest distance consistent with the specified mesh and the internally-defined (see Section IV) 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 $R_{end}(v,J)$ is therefore determined for each level using the semiclassical result of Eq. (**??**) which shows that the wavefunction dies off exponentially in the classically forbidden region with an exponent of

$$-\sqrt{2\mu/\hbar^2}\int_{R_2(v,J)}^{R_{end}(v,J)}[V_J(R)-E_{v,J}]^{1/2}\,dR \tag{13}$$

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 $R_{end}(v,J)$ 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}. While apparently somewhat fussy, in calculations for levels spanning a wide range of energies use of this procedure can reduce the overall computation time by a factor of two.

3.2 Locating Quasibound Levels and Determining Their Widths

4 Defining the Initial-State Potential $V(R)$

The potential-generation package which reads necessary input and returns the potential array and associated parameters is controlled by subroutine PREPOT (for "PREpare POTential"). It uses subroutine package GENINT for interpolation/extrapolation over a set of read-in turning points, and subroutine POTGEN for generating a potential as an analytic function. Values of the necessary input parameters enter via read statements #5–15; for the 2–state case invoked by inputting NUMPOT = 2 , this block of input statements is used twice.

One may choose to define a potential either by a set of NTP turning points { XI(i) , YI(i) } read in data statement #8, or (if NTP ≤ 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. (11) is performed in a manner specified by the read-in parameter NUSE. For NUSE > 0 this involves the use of piecewise NUSE–point polynomials (typically NUSE = 8 or 10), while for NUSE ≤ 0 the interpolation uses a cubic spline function. If the range of numerical integration [RMAX, 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 XI(2), the potential is extrapolated inward with an exponential function fitted to the first three turning points. Similarly, if RMAX > XI(NTP −1) the potential for $R >$ XI(NTP −1) is extrapolated outward as either an exponential-type function or a (sum of) inverse-power terms, as specified by parameters ILR, NCN and CNN of read statement #6 (see Section V).

If the user wishes to define the initial-state potential by an analytic function, rather than by an array of points, the integer input parameter NTP of read statement #5 should be set ≤ 0 . The program then *skips* read statements #6–8 and proceeds instead to #9–15 (see Section V), where it reads values of the parameters defining the chosen analytic potential. The present version of the code allows for the following five families of potential functions.

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

$$V(R) \ = \ \mathcal{D}_e \left[n(R_e/R)^m - m(R_e/R)^n \right] / (m-n) \qquad (14)$$

(ii) The "Modified Lennard-Jones" (MLJ) function which combines a flexible form for the potential well with the correct inverse-power long-range behaviour:[17]

$$V(R) \ = \ \mathcal{D}_e \left[1 - (R_e/R)^n \, e^{\beta(z)\, z} \right]^2 \qquad (15)$$

where $\beta(z)$ is a power series in the variable $z = (R - R_e)/(R + R_e)$.

(iii) The "Extended Morse Oscillator" (EMO) function:[18]

$$V(R) \ = \ \mathcal{D}_e \left[1 - e^{\beta(z)\,(R-R_e)} \right]^2 \qquad (16)$$

where $\beta(z)$ is again a power series in the variable $z = (R - R_e)/(R + R_e)$. Truncating this series at the constant term yields the familiar simple Morse potential. Another Morse-type function allowed by POTGEN is Hua's 4–parameter potential[19]

$$V(R) \ = \ \mathcal{D}_e \left([1 - e^{-b\,(R-R_e)}]/[1 - C\, e^{-b\,(R-R_e)}] \right)^2 \qquad (17)$$

(iv) Seto's modification[18] of Šurkus' GPEF potential[20] which incorporates the familiar Dunham,[21] Simons-Parr-Finlan,[22] and Ogilvie–Tipping[23] expansions as special cases invoked by different definitions of the expansion variable z (see discussion of *read* #9 in Section V):

$$V(R) \ = \ c_0 \, z^2 \left(1 + \sum_{m=1} c_m \, z^m \right) \qquad (18)$$

(v) The generalized HFD function defined in terms of the reduced distance variable $x = R/R_e$:[24, 25]

$$V(R) \ = \ A\, x^p \, e^{-\beta_0\, x - \beta_1\, x^2} + D(x) \sum_m C_m/R^m \qquad (19)$$

where $D(x)$ is a damping function which cuts off the inverse-power terms at short distances.

Alternatively, a user may introduce their own functional form by simply replacing subroutine POTGEN with their own potential routine. To retain consistency with the rest of the present code such user-prepared POTGEN subroutine should be:

CALL POTGEN(LNPT, N, VLIM, R, RM2, V, NCN)

The first argument parameter LNPT is an integer which specifies whether parameters defining a new potential are to be read in and printed ($\text{LNPT} > 0$), or whether additional potential points are to be calculated from the potential defined on a preceding call to this routine ($\text{LNPT} \leq 0$). In LEVEL's potential-generating routine PREPOT (which calls POTGEN) LNPT is fixed $= 1$.* The other input quantities are the integer N specifying the size of the radial mesh point array, the absolute energy VLIM at the potential asymptote, the N–point array of distances $\text{R}(i)$ (in Å) at which potential values are to be generated, and the squared inverse distance array $\text{RM2}(i) = 1/\text{R}(i)^2$. It returns the desired N–point potential array $\text{V}(i)$ (in units cm^{-1}), as well as the integer NCN which is the (positive) power of the asymptotically-dominant inverse-power term in that potential energy function. Under the option in which the program automatically searches for many or all vibrational levels of a given potential (when NLEV1 is large and negative, see read #16), NCN is used in a near-dissociation theory[26, 27, 28] algorithm to estimate the number and energies of missing levels. If the chosen 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., $\text{NCN} = 99$).

5 Defining the Final-State Potential(s)

6 Defining the Transition Monent or Coupling Function(s)

* While not used here, LNPT is retained in the calling sequences of GENINT and POTGEN to facilitate the use of these subroutines in other programs.

7 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. (**??**) used for calculating the Einstein coefficients of Eq. (6), defined in terms of the expansion parameters DM(i) of read statement $^{\#}$19 (see Section IV), is assumed to be in debye (where $1\,\text{au} = 2.54174776\,\text{debye} = 8.4783579 \times 10^{-30}\,\text{C}\cdot\text{m}$). Note, however, that in the $\text{IRFN} \geq 2$ option for generating the radial function defining the matrix element argument (see read $^{\#}$18) by numerical interpolation over a set of read-in points, 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. In addition, while a set of read-in turning points used to define the potential may be in any convenient units, appropriate conversion factors must be read (see read $^{\#}$7) which may convert them to Å and cm^{-1}.

The values of the physical constants appear in the program in two places. The first is the factor $2\mu/\hbar^2 = \mu/16.85762908\,[\text{cm}^{-1}\,\text{Å}^2]$ (where μ is in amu) appearing in the radial Schrödinger equation of Eq. (11). The second is in the collections of terms defining the numerical factor in Eq. (6) used in calculating the Einstein coefficient for the rate of spontaneous emission. These constants are based on the 1998 compilation of Ref.[29], while the atomic isotope masses tabulated in subroutine MASSES were taken from the compilation of Ref.[30].

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 (of the PREPOT package). In the former, NDIMR (currently 20001) 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 $^{\#}$16). The two array size parameters set inside the PREPOT package are NTPMX (currently 600) which is the maximum number of potential (or radial function, for $\text{IRFN} > 1$) turning points which may be read in, and MAXSP (in SPLINT) which is the number of spline coefficients required when interpolating over the read-in function values, and should be set equal to $4\times$ 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 read statements $^{\#}$5 & 16) writes a condensed output file to one or more of channels 7–10. Those executing the program in a UNIX operating system environment may wish to create and store in the system or user's 'bin' directory a shell named (say) 'rbc', such as that shown here:

```
# UNIX shell 'rbc' to execute the compiled version of program BCONT named
#   bc.x, which is stored in the user's directory /upath/ with input data
#   file  $1.5  and 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
```

which 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.

8 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,*) TITLE
 #2      READ(5,*) AN1, AN2, CHARGE, NISTP, NFS, NSETS, FITIT
         DO  iso= 1, NISTP
 #3          READ(5,*) MN1(iso), MN2(iso)
 #4a         IF((AN1.LE.0).OR.(AN1.GT.109)) READ(5,*) NAME1(iso), MASS1(iso)
 #4b         IF((AN2.LE.0).OR.(AN2.GT.109)) READ(5,*) NAME2(iso), MASS2(iso)
             ENDDO
 #5      READ(5,*) RH, RMIN, RMAX, OVRCRT
 #6      READ(5,*) IWRSCH, IWROVR
 #7      IF(FITIT.GT.0) READ(5,*) IROUND, LPDER, UCUTOFF, DFACT, LPRINT
c  Loop over the NSETS different types of observables to be considered
         DO  iset= 1, NSETS
 #8          READ(5,*) INFO(iset)
 #9          READ(5,*) ISOT(iset), BOLTZ(iset), DTYPE(iset), IFRPW(iset), PQR(iset)
             IF(NFS.GT.1) THEN
                 IF(DTYPE(iset).EQ.1) THEN
 #10                 READ(5,*) (CN(iset,ifs), ifs= 1,NFS)
                   ELSEIF(DTYPE(iset).EQ.2) THEN
 #11                 READ(5,*) (CN(iset,ifs), ifs= 1,NFS)
 #12                 READ(5,*) (CD(iset,ifs), ifs= 1,NFS)
                   ENDIF
                 ENDIF

c ... for calculating or fitting to predissociation rates or level widths ....
             IF(IFRPW(iset).EQ.0) THEN
 #13             READ(5,*) NVJ(iset)
                 IF(FITIT.LE.0) THEN
                     IF(NVJ(iset).GT.0) THEN
 #14                     READ(5,*) (VFIX(ivj), JFIX(ivj), ivj= 1, NVJ(iset))
                       ELSE
 #15                     READ(5,*) V1ST(iset), VMAX(iset), J1ST(iset), JMAX(iset)
                       ENDIF
                   ELSEIF(FITIT.GT.0) THEN
 #16                 READ(5,*) PUNITS(iset)
                     DO  ivj= 1, NVJ(iset)
 #17                     READ(5,*) VFIX(ivj), JFIX(ivj), OBS(ivj,iset), UNC(ifr,iset)
                         ENDDO
                   ENDIF
                 ENDIF

c ... for calculating or fitting to intensities or branching ratios ...
             IF(IFRPW(iset).NE.0) THEN
 #18             IF(BOLTZ(iset).GT.0) READ(5,*) TEMP(iset), VMAX(iset), NJ(iset)
 #19             IF(BOLTZ(iset).LE.0) READ(5,*) V1ST(iset), J1ST(iset)
 #20             IF(FITIT.LE.0) READ(5,*) NFREQ(iset), FREQ1, DFREQ
                 IF(FITIT.GT.0) THEN
 #21                 READ(5,*) NFREQ(iset), FREQYN
                     DO  ifr= 1, NFREQ(iset)
 #22                     READ(5,*) FREQ(ifr,iset), OBS(ifr,iset), UNC(ifr,iset)
                         ENDDO
                     ENDIF
             ENDDO

c ... If fitting to multiple data sets ... allow for scaling

          IF((FITIT.GT.0).AND.(NSETS.GT.1)) THEN
 #23          READ(5,*) (SF(iset), iset=2, NSETS)
 #24          READ(5,*) (SCALE(iset), iset=2, NSETS)
              ENDDO

c  Read initial-state potential energy function specifications

 #25     READ(5,*) NTPI, LPPOTI, OMEGA(0), VLIMI
         IF(NTP.GT.0) THEN
 #26         READ(5,*) NUSEI, IR2I, ILRI, NCNI, CNNI
 #27         READ(5,*) RFACTI, EFACTI, VSHIFTI
 #28         READ(5,*) (XI(I), YI(I),I= 1,NTPI)
           ELSE
 #29         READ(5,*) IPOTL, MPAR, NPAR, NVARB, IBOB, DSCM, REQ
 #30         IF(NVARB.GT.0) READ(5,*) (PARM(I), I=1,NVARB)
             IF(IBOB.GT.0) THEN
 #31             READ(5,*) RMN1, RMN2, NC1, NC2, NG1, NG2, RX
 #32             IF(NC1.GE.0) READ(5,*) (CA1(I), I=0,NC1)
 #33             IF(NC2.GE.0) READ(5,*) (CA2(I), I=0,NC2)
 #34             IF(NG1.GE.0) READ(5,*) (GA1(I), I=0,NG1)
 #35             IF(NG2.GE.0) READ(5,*) (GA2(I), I=0,NG2)
                 ENDIF
           ENDIF

c  Read final-state potential energy function specifications

 #36      READ(5,*) REXFS, REXTMF, LPFS, LPTMF
          DO  ifs= 1, NFS
 #37          READ(5,*) FSTYPE(ifs), OMEGA(ifs), NFSPRM(ifs), VLIMF(ifs), XCOORD(ifs)
 #38          READ(5,*) (FSPRM(j,ifs), j=1, NFSPRM(ifs))
 #39          IF(FITIT.GT.0) READ(5,*) (FSVAR(j,ifs), j= 1, NFSPRM(ifs))
              IF(FSTYPE(ifs).EQ.2) THEN
 #40              READ(5,*) NTPFS(ifs)
 #41              READ(5,*) NUSEF, IR2F, ILRF, NCNF, CNNF
 #42              READ(5,*) RFACTF, VFACTF, VSHIFTF(ifs)
 #43              READ(5,*) (RTPF(i), VTPF(i), i=1, NTPFS(ifs))
                  ENDIF
 #44          READ(5,*) GFS(ifs), TMFTYP(ifs), OTMF(ifs)
 #45          READ(5,*) (TMFPRM(m,ifs), m=0, OTMF(ifs))
 #46          IF(FITIT.GT.0) READ(5,*) (TMFVAR(m,ifs), m=0, OTMF(ifs))
              IF(TMFTYP(ifs).EQ.0) THEN
 #47              READ(5,*) NPTMF, TMFLIM
 #48              READ(5,*) NUSETMF, ILRTMF, NCNTMF, CNNTMF
 #49              READ(5,*) RFACTMF, MFACTMF
 #50              READ(5,*) (Xi(i), Yi(i), i=1, NPTMF)
                  ENDIF
              ENDDO
```

Definitions and Descriptions of Input File Data

A. Identify the Physical Species and General Control Parameters

Read a text title or description for the calculation.

#1. `READ(5,*) TITLE`

TITLE : a line of text of up to 75 characters describing the particular problem, enclosed in single quotes: e.g., 'Title of Problem'

Read integers identifying the molecule or system.

#2. `READ(5,*) AN1, AN2, CHARGE, NISTP, NFS, NSETS, FITIT`

AN1 & AN2 : (integer) atomic numbers of the atoms/particles #1 & 2 forming the molecule.

CHARGE : ± integer for the total charge on the molecule. Used to generate Watson's charge-modified reduce mass for molecular ions:[8] $\mu = M_A\, M_B/(M_A + M_B - m_e \times \mathrm{CHARGE})$, where m_e is the electron mass.

NISTP : (integer) number of distinct isotopomers to be considered in the simulation; > 1 when simultaneously fitting to separate data sets for different isotopomers of a given species.

NFS : the (integer) number of different final electronic state potentials contributing to the property calculation(s)

NSETS : the (integer) number of distinct sets of properties to be calculated (in a pure "forward" calculation), or the number of distinct sets of experimental data to be simultaneously treated in a fit [e.g., thermal absorption coefficient at different temperatures and/or for different isotopomers, and/or photodissociation branching ratios, and/or ... etc.].

FITIT : (integer) specifying whether to perform forward calculation(s) from a set of known potentials (FITIT$\leq$ 0), or to fit to set(s) of input data to optimize final-state potential function and/or transition moment or coupling function parameters.

Loop over the NISTP isotopomers, reading the mass numbers MN1 and MN2 of the two atoms.

#3. `READ(5,*) MN1(isot), MN2(isot)`

MN1 & MN2 : integer mass numbers of the atoms/particles #1 & 2 forming the isotth molecular isotopomer. For a normal stable atomic isotope, its mass is 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 returned.

In the special case when AN1 and/or AN2 is either ≤ 0 or > 109 , read in a two-character alphanumeric name for that particle and its mass (in amu). This alows the treatment of non-molecular model systems or of exotic species such as muonium or positronium "molecules".

#4.a `IF((AN1.LE.0).OR.(AN1.GT.109)) READ(5,*) NAME1(isot), MASS1(isot)`
#4.b `IF((AN2.LE.0).OR.(AN2.GT.109)) READ(5,*) NAME2(isot), MASS2(isot)`

NAME1 *or* **NAME2 :** a two-character alphanumeric name for the particle whose mass is being read, enclosed in single quotes, as in $'$mu$'$.

MASS1 *or* **MASS2 :** the masses (in amu) of the particle in question.

Read real numbers defining the range and mesh of the numerical integration (all in Å), and a numerical convergence criterion.

#5 `READ(5,*) RH, RMIN, RMAX, OVRCRT`

RH : the numerical integration mesh size; see discussion associated with Eq. (12) in Section II.A.

RMIN & RMAX : the inner and outer limits, respectively, of the range of numerical integration (see discussion of Section II). As zeroth order estimates one may set RMIN$\approx$ 0.6×(potential inner wall position) and RMAX very large (say, = 99 Å). Internally RMAX is reset to the smaller of this read-in value and the largest distance allowed by RMIN, RH and the array dimension NDIMR (see Section III).

OVRCRT : relative amplitude convergence criterion for quantal continuum wavefunctions (typically ca. 1×10^{-6}).

Read parameters to control internal print level - used for troubleshooting.

#6. `READ(5,*) IWRSCH, IWROVR`

IWRSCH : Integer to control print level in calculations of discrete initial state levels and wavefunctions. If < 0 print only errors/warnings [recommended normal setting]; if $= 0$ no printing; if ≥ 1 also print final eigenvalues & node count; if ≥ 2 also end-of-range wavefunction amplitudes; if ≥ 3 also intermediate trial energies during eigenvalue convergence.

IWROVR : Integer to control print level in continuum wavefunction/overlap integral calculation. If < 0 print only errors/warnings [recommended normal setting]; if $= 0$ no printing; if ≥ 1 also print overlap integrals; if $geq2$ also print final converged amplitudes; if $geq3$ also show all converging amplitudes.

If performing a fit to experimental data, read parameters governing features of the fit and associated printout.

#7. `IF(FITIT.GT.0) READ(5,*) IROUND, LPDER, UCUTOFF, DFACT, LPRINT`

IROUND : Setting (integer) IROUND $\neq 0$ causes the "sequential rounding and refitting" procedure of Ref.[31] to be implemented, with each parameter being rounded at the |IROUND|'th significant digit of its uncertainty. If IROUND > 0 the sequential rounding is applied in turn to the remaining free parameter with the largest relative uncertainty; if IROUND < 0 the rounding to proceeds sequentially from the last parameter to the first. If IROUND $= 0$ the fit simply stops after full convergence, with no parameter rounding; this last option saves considerable computation time, and should normally be chosen except for a "final" fit to obtain parameters for publication.

LPDER : an integer set > 0 to cause the partial derivative array generated in each fitting cycle to be printed to channel-10. Normally (except when trouble-shooting) set LPDER $= 0$.

UCUTOFF : a real number allowing one to selectively omit some of the input data from the analysis without creating a new data file; read-in data with experimental uncertainties ¿ UCUTOFF are ignored in the least-squares fit.

DFACT : a real scaling factor used in calculating derivatives-by-differences in least-squares partial derivative subroutine DYIDPJ. Its use is under development. Normally users should set DFACT $= 1.0$.

LPRINT : an integer specifying the level of printing inside the core least-squares subroutine package NLLSSRR. Setting it $= 0$ yields no internal printout except for convergence failure warning messages; this is the recommended 'normal' choice when there are no problems with the fits. If LPRINT < 0 print converged unrounded parameters; if LPRINT ≥ 1 also print converged rounded parameters (when IROUND $\neq 0$); if LPRINT ≥ 2 also print parameter changes on each rounding step; if LPRINT ≥ 3 also report parameter convergence criterion satisfied; if LPRINT ≥ 4 also print convergence test on each fitting cycle; if LPRINT ≥ 5 also print parameter changes & uncertainties on each fitting cycle.

B. Specify the Properties to be Calculated or Fitted To

Now, loop over the NSETS cases specifying the different (sets of) properties to be calculated, or different sets of data to be included in the fit. For each case, use read #8 and 9, plus whichever combination of reads #10-22 are appropriate or necessary for that case, plus #23 and 24.

#8. `READ(5,*) INFO(iset)`

INFO : a one-line name or description for this case, consisting of up to 70 characters, enclosed in single quotes: e.g., 'name for data set'

#9. `READ(5,*) ISOT(iset), BOLTZ(iset), DTYPE(iset), IFRPW(iset), PQR(iset)`

ISOT : integer ($1 \leq$ ISOT $\leq$ NISTP) identifying the isotopomer for this data set or property.

BOLTZ : integer > 0 if property calculation requires a Boltzman weighted sum over initial-state $\{v, j\}$ levels; BOLTZ ≤ 0 if property is for a single initial-state $\{v, J\}$ level (e.g., predissociation, or absorption or emission from a single $\{v, J\}$ level).

DTYPE : integer $= 1$ if the property concerns a transition into one or more final states (absorption or emission intensities, or predissociation rates); $= 2$ if this property is a ratio of intensities to different states (a branching ratio).

IFRPW : An integer to further specify the property being calculated – it is the integer power of the frequency to be used in the property calculation: IFRPW $= 0$ for predissociation, $= 1$ for absorption coefficients, and $= 3$ for spontaneous emission. For model calculations, set IFRPW $= -1$ to calculate absorption coefficients assuming a constant frequency factor, and $= -3$ to calculate emission intensities assuming a constant frequency factor.

PQR : integer $= 1$ to cause P, Q & R branch intensities to be calculated, weighted by Hönl-London factors, and summed over; PQR $= 0$ invokes the "Q–branch" approximation of fixing $J' = J''$.

If the property for case–iset is an intensity or sum of intensities into *more* than one (NFS > 1) final electronic state (DTYPE=1), input weights in read #10 and skip reads #11 & 12; if it is a ratio of intensities (DTYPE=2), skip #10 and read weights defining the branching ratio in #11 & 12. NOTE that if NFS=1 these three READ's are redundant, and hence are skipped.

#10. `READ(5,*) (CN(iset,ifs), ifs= 1,NFS)`

CN : the integer weights for the final-state intensities added to give values of this property: $I_{tot} = \text{CN}(1)\times I_1 + \text{CN}(2)\times I_2 + ... + \text{CN}(\text{NFS})\times I_{\text{NFS}}$, where I_i is the transition intensity into final-state–i. For a single final state (NFS $= 1$), read CN(1) $= 1$.

If this property is a ratio of intensities (DTYPE=2), read (integer) coefficients specifying the combination of calculated final-state intensities.

#11. `READ(5,*) (CN(iset,ifs), ifs= 1,NFS)`
#12. `READ(5,*) (CD(iset,ifs), ifs= 1,NFS)`

CN & DN : The integer numerator and denominator weights specifying the particular intensity branching ratio: $I_{tot} = \frac{\text{CN}(1)\times I_1 + \text{CN}(2)\times I_2 + ... + \text{CN}(\text{NFS})\times I_{\text{NFS}}}{\text{CD}(1)\times I_1 + \text{CD}(2)\times I_2 + ... + \text{CD}(\text{NFS})\times I_{\text{NFS}}}$, where I_i is the transition intensity into final-state–i.

If the property is predissociation from levels of the initial state, use READs #13 – 17.

#13. `READ(5,*) NVJ(iset)`

NVJ : is the number of initial-state $\{v, J\}$ levels for which predissociation rates are to be calculated. For a "forward" calculation (FITIT ≤ 0), NVJ ≤ 0 causes calculations to be performed for all possible $\{v, J\}$ within specified ranges (see read #15).

For a foward calculation (not fitting), use either read #14 or 15 to specify the levels whose predissociation rates are to be calculated, and skip reads #16 & 17; for a fit, skip reads #14 & 15 and use reads #16 & 17 to input the data to be fitted.

If NVJ($iset$) > 0 read NVJ pairs of quantum numbers $\{v, J\}$ specifying the levels for which predissociation rates are to be calculated.

#14. `IF(NVJ(iset).GT.0) READ(5,*) (VFIX(ivj), JFIX(ivj), ivj= 1,NVJ(iset))`

Alternately, if NVJ($iset$) ≤ 0 read in lower and upper bounds on v and J and calculate predissociation rates for all $\{v, J\}$ such that V1ST($iset$) $\leq v \leq$ VMAX($iset$) and J1ST($iset$) $\leq J \leq$ JMAX($iset$) .

#15. `IF(NVJ(iset).LE.0) READ(5,*) V1ST(iset), VMAX(iset), J1ST(iset), JMAX(iset)`

For a fit to predissociation data, skip reads #14 & 15 and read, first the units of the input data, and then the experimental data and their uncertainties, one case per line of input.

#16. `READ(5,*) PUNITS(iset)`

PUNITS : When fitting, it specifies the units for the input experimental predissociation rates: PUNITS > 0 for widths (FWHM) in cm^{-1}, $= 0$ for lifetimes in seconds, and < 0 for rates in s^{-1}.

#17. `READ(5,*) (VFIX(ivj), JFIX(ivj), OBS(ivj,iset), UNC(ivj,iset), ivj= 1,NVJ(iset))`

VFIX & JFIX : are the v and J quantum numbers of the predissociating level.

OBS & UNC : are the observed predissociation rates or widths (in units specified by PUNITS) and their uncertainties (in the same units).

If the property is absorption or emission intensities, or branching ratios, allow for two cases.
(i) For absorption or emission from a thermal initial-state population (BOLTZ >0) ...

#18. `IF(BOLTZ(iset).GT.0) READ(5,*) TEMP(iset), VMAX(iset), NJ(iset)`

TEMP is the Kelvin temperature characterizing the thermal initial-state population.

VMAX : an upper bound v value cutoff for the thermal sum over initial-state vibrational levels.

NJ : specifies how the sum over a thermal rotational population is to be done. For NJ $=0$ perform all calculations with J(initial)$=0$; for NJ <0 perform a direct sum from $J=0$ to a maximum of $J=|\mathrm{NJ}|$, subject to a cutoff when 99.9% of the population for that v is accounted for; for NJ >0 sum over contributions from NJ average J values in NJ equally weighted segments of the rotational population for each v, as per Ref.[3].

(ii) For absorption or emission from a particular (v,J) level, (BOLTZ ≤ 0), ...

#19. `IF(BOLTZ(iset).LE.0) READ(5,*) V1ST(iset), J1ST(iset)`

V1ST & J1ST are the v and J quantum numbers of the level whose absorption or emission coefficients are to be calculated (or fitted to).

For a forward (non-fitting) calculation, specify the frequencies at which intensities (or branching ratios) are to be calculated.

#20. `IF(FITIT.LE.0) READ(5,*) NFREQ(iset), FREQ1, DFREQ`

NFREQ is the number of transition energies for the simulation.

FREQ1 & DFREQ : Perform calculations at the NFREQ transition energies (in cm^{-1}) $\bar{\nu}(i) = \mathrm{FREQ1} + (i-1) \times \mathrm{DFREQ}$.

For a fit (FITIT >0), read in the experimental intensities or branching ratios, and their incertainties.

#21. `READ(5,*) NFREQ(iset), FREQYN`

NFREQ is the number of experimental data in this set.

FREQYN is an integer: if >0 it specifies that the ordinate values of the input data are transition energies in cm^{-1}, while FREQYN ≤ 0 indicates that those read-in values are wavelengths in nm (program uses cm^{-1} internally).

Now loop over the NFREQ frequencies, for ifr $= 1 -$ NFREQ(*iset*) , reading one entry per line.

#22. `READ(5,*) (FREQ(ifr,iset), OBS(ifr,iset), UNC(ifr,iset), ifr= 1, NFREQ(iset))`

FREQ is the transition energy in cm^{-1} (if FREQYN >0) or wavelength in nm (if FREQYN ≤ 0) for this datum.

OBS is the experimental transition intensity of branching ratio, in units [ℓ/mole $\cdot$ cm] for absorption, and s^{-1} for emission.

UNC is the uncertainty in the experimental value, in the same units.

End of loop over the NSETS *properties to be predicted or input data sets to be fitted to!*

When performing fits to multiple data sets, one may wish/need to apply a global scaling to experimental data or predictions for certain data subsets to allow for (say) uncertainties in concentration measurements for different isotopomers. Read these statements if FITIT >0 and NSETS ¿ 1 ; otherwise ignore them.

#23. `READ(5,*) (SF(iset), iset= 2, NSETS)`
#24. `READ(5,*) (SCALE(iset), iset= 2, NSETS)`

SF : a multiplicative scaling factor applied to the experimental data for data set *iset* to make it consistent with that for data set–1.

SCALE : an integer flag to specify whether scaling factor SF(*iset*) is to be held fixed (when SCALE $=0$) or varied (when SCALE >0) in the fit.

C. Specify the Initial-State Potential

Some combination of the next 11 *read* statements defines the initial (bound-state) potential energy function. A particular case always starts with *read* #25, but then uses *either reads* #26–28 for numerical interpolation over a set of input turning points, *or* (some of) *reads* #29–35 for the case of an analytic potential function. This segment of the input is identical to the potential specification section for program LEVEL [32].

#25. `READ(5,*) NTPI, LPPOTI, OMEGA(0), VLIMI`

NTPI : An integer which is set ≤ 0 to generate an analytic potential using POTGEN; in this case, *skip over reads* #26–28 and go directly to read #29. If NTP > 0, it is the number of turning point pairs to be input via *read* #28.

LPPOTI : controls printing of the potential array (normal setting $=0$ yields no printing). If LPPOTI >0 write to standard output (channel–6) the potential and its first 2 derivatives-by-differences at every LPPOTIth mesh point; it is sometimes useful to do this when troubleshooting. Setting LPPOTI <0 writes the resulting potential in condensed format to channel–8 at every |LPPOTI|th mesh point; this may be useful if (for example) one wants to input this potential into a plotting routine.

OMEGA(0) : The (integer) total electronic angulat momentum projection quantum number (Λ or Ω) for the initial electronic state (required for detailed rotational intensity weighting).

VLIMI : The absolute energy (in cm^{-1}) of the initial-state potential asymptote. This sets the absolute energy scale used in the calculation.

For a pointwise potential (if NTPI >0), need to specify how the interpolation is to be done, and since RMAX often lies well outside the range of the input turning points, must also specify how the potential is to be extrapolated at large R. *Skip* over this down to *read* #29 if NTPI ≤ 0.

#26. `READ(5,*) NUSEI, IR2I, ILRI, NCNI, CNNI`

NUSEI : Specifies how the interpolation is to be done. If NUSEI >0 use NUSEI–point piecewise polynomials; if NUSEI ≤ 0 perform cubic spline interpolation. For highly precise and smooth input points, such as those generated from an RKR calculation, NUSEI $=8$, 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 (NUSEI $=4$ or 6) or splines (NUSEI ≤ 0) are usually best.

IR2I : For very steep repulsive potential walls, better interpolation is often attained by actually interpolating over $R^2 \times V(R)$; setting IR2I >0 causes this to be done (normally recommended). The option of turning this off (setting IR2I ≤ 0) and interpolating over $V(R)$ itself may be used to obtain an estimate of the effect of "interpolation noise" on the final results.

ILRI : Specifies how to extrapolation from the last input turning points to RMAX. For a long extrapolation one of ILRI $= -1$, 0 or 1 is usually most appropriate; however, if the input points extend to fairly near dissociation one should set ILRI ≥ 2 and specify the theoretically appropriate value of NCNI (≥ 1), and if it is available, also input an estimate of CNNI (see below).

For ILRI <0, fit the last 3 points to: VLIMI $-A \times \exp[-b(R-R_o)^2]$

For ILRI $=0$, fit the last 3 points to: VLIMI $-A \times R^p \times \exp[-b\,R]$.

For ILRI $=1$, fit the last 2 points to: VLIMI $-A/R^B$.

For ILRI $=2$ or 3, respectively, fit the last 2 or 3 points to a sum of 2 or 3 inverse-power terms, with powers differing by 2: VLIMI $-\sum_{m=0}^{\mathrm{ILRI}-1} C_{\mathrm{NCNI}+2m}/R^{\mathrm{NCNI}+2m}$.

For ILRI ≥ 4, fit last few points to a sum of ILRI inverse-power terms, with powers differing by 1: $V(R) = \mathrm{VLIMI} - \sum_{m=0}^{\mathrm{ILRI}-1} C_{\mathrm{NCNI}+m}/R^{\mathrm{NCN}+m}$.

NCNI : For inverse-power potential extrapolation with ILRI ≥ 2, NCNI (>0) specifies the limiting inverse-power behaviour: $V(R) \propto \mathrm{VLIMI} - \mathrm{CNNI}/R^{\mathrm{NCNI}}$. Otherwise (for ILRI ≤ 1) it is a dummy input variable.

CNNI : For inverse-power potential extrapolation with ILRI ≥ 2, setting CNNI $\neq 0$ causes the leading inverse-power coefficient to be fixed at the read-in value CNNI $= C_{\mathrm{NCNI}}$ [cm^{-1} Å$^{\mathrm{NCNI}}$] rather than be determined from a fit the outermost turning points.

The input turning points may come from *ab initio* or other sources and may need to be converted to the units (cm^{-1} and Å) used inside the program.

#27. `READ(5,*) RFACTI, EFACTI, VSHIFTI`

RFACTI & EFACTI : are the factors required to convert units of the NTPI input turning point distances XI(i) and energies YI(i) to Å and cm^{-1}, respectively. If no conversion is required, read in factors of 1.D0.

VSHIFTI : An energy shift (in cm^{-1}) added to the input potential points to make them consistent with VLIMI. This addresses the fact that input *ab initio* or RKR turning points may be expressed relative to an energy zero inconsistent with the user-defined VLIMI value.

Read in the actual turning points.

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

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

If the input potential is defined by an analytic function (the NTPI ≤ 0 case) using the POTGEN subroutine supplied, the parameters characterizing it are input via some or all of read statements #29–35. For a user's POTGEN function the input parameters (if any) are up to them, but the calling sequence must match that expected by PREPOT (see discussion of Section II.E).

#29. `READ(5,*) IPOTL, MPAR, NPAR, NVARB, IBOB, DSCM, REQ`

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

IPOTL = 1 generates a Lennard-Jones(MPAR, NPAR) function.

IPOTL = 2 generates an MLJ potential:[17, 33] $V(R) = \mathcal{D}_e[1 - (R_e/R)^{\rm NPAR}\, e^{\beta(z)\,z}]^2$ where $z = (R - R_e)/(R + R_e)$ and $\beta(z)$ is a power series in z.
For MPAR ≥ 0 polynomial $\beta(z)$ has order NVARB -1 and NVARB coefficients PARM(i). For MPAR < 0 this polynomial has order NVARB -4 and is constrained to yield a limiting long-range inverse-power potential coefficient of $C_{\rm NCN}$ = PARM(NVARB-2) by the switching function $f_S(R) = 1/[e^{\alpha_S(R-R_S)} + 1]$ where α_S = PARM(NVARB $-$ 1) and R_S = PARM(NVARB) (see Ref.[17]).

IPOTL = 3 normally generates either a simple Morse (when NVARB = 1) or an "Extended Morse Oscillator" (EMO) potential[18, 34, 35] (when NVARB > 1): $V(R) = \mathcal{D}_e[1 - e^{\beta(z)\,(R-R_e)}]^2$ where $z = (R - R_e)/(R + R_e)$ and $\beta(z)$ is a power series in z of order NVARB -1 with the NVARB coefficients PARM(i).

- If MPAR = -1, generate Wei Hua's[19] 4–parameter Morse-like potential $V(R) = \mathcal{D}_e\left([1 - e^{-b\,(R-R_e)}]/[1 - C\, e^{-b\,(R-R_e)}]\right)^2$ where b = PARM(1) and C = PARM(2).
- If MPAR = -2, generate the Coxon-Hajigeorgiou[36] "Generalized Morse Oscillator" with exponent parameter β a power series expansion in $(R - R_e)$.

IPOTL = 4 uses Seto's modification[18] of Šurkus' GPEF potential[20] $V(R) = c_0\, z^2[1 + \sum_{m=1} c_m\, z^m]$ where $z = (R^{\rm NPAR} - {R_e}^{\rm NPAR})/(a_S\, R^{\rm NPAR} + b_S\, {R_e}^{\rm NPAR})$ with c_0 read in as c_0 = DSCM, the c_i ($i \geq 1$) as the first NVARB -2 input parameters PARM(i), a_S = PARM(NVARB -1) and b_S = PARM(NVARB). *Note* that Šurkus' case of NPAR < 0 is accommodated by Seto's identity[18] $z(-{\rm NPAR},\, a_S,\, b_S) = z({\rm NPAR},\, -b_S,\, -a_S)$.

- Dunham expansions are generated by setting NPAR = 1, $\alpha_S = 0$ & $b_S = 1$.
- SPF expansions are generated by setting NPAR = 1, $\alpha_S = 1$ & $b_S = 0$.
- Ogilvie–Tipping expansions are generated by setting NPAR = 1, $\alpha_S = b_S = 0.5$.

IPOTL = 5 generates generalized HFD(NPAR, 6, 8, 10, 12, 14) type potentials:
$V(R) = A\, x^{\rm PARM(5)}\, e^{-\beta\, x - {\rm PARM(4)}\, x^2} + D(x)[\, C_{\rm NPAR}/R^{\rm NPAR} + C_6/R^6 + C_8/R^8 + ...\,]$
where $x = R/R_e$, A and β are defined (internally) to yield the input values of $\mathcal{D}_e$ = DSCM and R_e = REQ, the inverse-power coefficients $C_{\rm NPAR}$, C_6, C_8, ... etc. are input as dimensionless parameters ($\overline{C}_n = C_n/\mathcal{D}_e\, {R_e}^n$) PARM(i) for i = 6, 7, ... etc., and the damping function is defined as $D(x) = \exp\{-{\rm PARM(1)}\, [{\rm PARM(2)}/x - 1]^{\rm PARM(3)}\}$ for $x \leq$ PARM(2) and $D(x) = 1$ for $x >$ PARM(2). Note: (i) if no low-power (NPAR < 6) inverse-power term is required, simply set PARM(6) = 0.0, and (ii) necessarily NVARB $\geq$

7 ; NVARB $=7$ causes the potential to have $1/R^{\rm NCN}$ and $1/R^6$ terms; to include the additional inverse powers (8, 10, 12, up to 14), simply increase NVARB and read additional dimensionless constants PARM(i) $= \overline{C}_{2(i-4)} = C_{2(i-4)}/\mathcal{D}_e\,(R_e)^{2(i-4)}$.

MPAR & NPAR : Integers used to characterize particular potential forms (see above).

NVARB : The (integer) number of (real) parameters PARM(i) to be read in to define a particular potential.

IBOB : An integer to specify whether (for IBOB >0) or not (for IBOB ≤ 0) atomic-mass-dependent Born-Oppenheimer breakdown correction terms are to be included in the potential and/or centrifugal $1/R^2$ functions.

DSCM : Normally (except for the IPOTL $=4$ case, above), the potential well depth $\mathcal{D}_e$ in cm^{-1}.

REQ : The equilibrium distance R_e in Å.

#30. `IF(NVARB.GT.0) READ(5,*) (PARM(I),I=1,NVARB)`

PARM(i) : The NVARB (real*8) parameters read in to define the selected analytic potential.

If atomic-mass-dependent Born-Oppenheimer breakdown (B-O-B) corrections are to be incorporated into the potential and/or centrifugal $1/R^2$ functions (IBOB >0), use read statement #31 and one or more (as needed) of #32–35. In the POTGEN subroutine supplied these are additive corrections expressed as power series in $z = (R-R_e)/(R+R_e)$, starting with the constant term, with the isotope dependence convention defined as in Ref.[37]. If B-O-B corrections are not considered (IBOB ≤ 0), *omit* reads #31–35.

#31. `READ(5,*) RMN1, RMN2, NC1, NC2, NG1, NG2, RX`

RMN1 & RMN2 : The integer mass numbers of the reference isotopes of atoms 1 & 2, respectively, defining the B-O-B correction functions (see Ref.[37]).

NC1 & NC2 : For atoms $i=1$ & 2, the numbers of terms in the 'adiabatic' potential correction functions: $[(M_i^\alpha - M_i^r)/M_i^\alpha]\sum_{m=0}^{{\rm NC}i-1} u_m^i\, z^m$, where M_i^α is the mass of the actual atomic isotope considered (IMNi from read #1) and M_i^r the mass of the reference isotope (mass number RMNi) in terms of which the correction is defined.[35, 37] Set them <0 to omit such corrections.

NG1 & NG2 : For atoms $i=1$ & 2, the number of terms in the 'non-adiabatic' correction functions $q_i^\alpha = [M_i^r/M_i^\alpha]\sum_{m=0}^{{\rm NG}i-1} q_m^i\, z^m$ incorporated into the centrifugal potential $(\hbar^2/2\mu)\,J(J+1)\,[1+q_1^\alpha(z)+q_2^\alpha(z)]/R^2$, with M_i^α and M_i^r defined as above.[35, 37] Set them ≤ 0 to omit such corrections.

RX : A real variable which may be set $\neq 0$ to invoke a legacy form[36] of the centrifugal correction function expressed using terms of the form $[(R-R_x)^m - (R_e-R_x)^m]$. For the normal case (above),[35, 37] read in RX $= R_x = 0$.

```
#32. IF(NC1.GT.0) READ(5,*) (CA1(I),I=1,NC1)
#33. IF(NC2.GT.0) READ(5,*) (CA2(I),I=1,NC2)
#34. IF(NG1.GT.0) READ(5,*) (GA1(I),I=1,NG1)
#35. IF(NG2.GT.0) READ(5,*) (GA2(I),I=1,NG2)
```

CA1(i) & CA2(i) : The (real*8) expansion parameters defining the 'adiabatic' potential correction functions described above: CA1(m) $= u_m^1$ and CA2(m) $= u_m^2$, in units cm^{-1}.

CA1(i) & CA2(i) : The (real*8) expansion parameters defining the centrifugal potential correction functions described above: GA1(m) $= q_m^1$ and GA2(m) $= q_m^2$ (dimensionless).

For a calculation involving only a single isotopomer (NISTP $=1$ in read #1), simply continue on *read* #35. However, if NISTP >1 , repeat read statements #24–34 for each isotopomer. This is needed because for some small–μ species, isotopomer-dependent initial-state potentials are known, and hence should be used.

D. Characterize the Final-State Potentials and Transition Moment Functions

#36. `READ(5,*) REXFS, REXTMF, LPFS, LPTMF`

REXFS : reference distance about which final-state potentials are expanded (see Section X.X).

REXTMS : reference distance about which transition moment functions are expanded (see Section X.X).

LPFS : Controls (optional) printing of the final-state potentials. For LPFS > 0 print every LPSFth point of each potential array to channel–9; otherwise no printout.

LPTMF : Controls (optional) printing of the transition moment function arrays. For LPTMF > 0 print every LPTMFth point of each array to channel–9; otherwise no printout.

Now loop over the NFS final electronic states, using *reads* #36 $- -49$ to input the parameters specifying the potential and transition moment function for each, and identifying which parameters (if any) are to be freed in a fit.

D. (i) Begin by specifying the basic potential type for each case.

#37. `READ(5,*) FSTYPE(ifs), OMEGA(ifs), NFSPRM(ifs), VLIMF(ifs), XCOORD(ifs)`

FSTYPE specifies the type of final-state potential.

- FSTYPE$=1$ for a repulsive exponential $V_f(r) = \mathrm{VLIMF} + A_1\,\exp\{-(R - \mathrm{REXFS})(A_2 + A_3\,z + A_4\,z^2 + ...)\}$, where the $\{A_i\}$ are read in as parameters $\mathrm{FSPRM}(i, ifs)$.
- FSTYPE$=3$ for an "Extended Morse Oscillator" $V_f(r) = \mathrm{VLIMF} + A_1\,\left[\exp\{-(R - A_2)(A_3 + A_4\,z + A_5\,z^2 + ...)\} - 1\right]^2 - A_1$, where the $\{A_i\}$ are read in as parameters $\mathrm{FSPRM}(i, ifs)$.
- FSTYPE$=2$ for a potential defined by a read-in set of turning points with a repulsive exponential attached to the two innermost points. The attached exponential wall has the form $V_f(r) = X_1 + X_2\,\exp\{-(R - \mathrm{REXFS})[A_1 + A_2\,z + A_3\,z^2 + ...]\}$, where X_1 and X_2 are determined by the fit to the two innermost turning points, and the $\{A_i\}$ are read-in as parameters $\mathrm{FSPRM}(i, ifs)$.

OMEGA : the integer total electronic angular momentum progection quantum number for final state *ifs*. Required when calculating Hönl-London factors needed for P/Q/R intensity calculation for the option PQR(*iset*)$=1$ (see *read* #8).

NFSPRM : is the number of parameters {FSPRM(*i,ifs*)} to be read in for a given final-state potential. Current dimensioning allows NFSPRM ≤ 6 .

VLIMF is the absolute energy (in cm^{-1}) at the asymptote of this potential.

XCOORD specifies the nature of the exponent expansion coordinate in the above potential forms.
- XCOORD$= p\,(p{=}1-9)$ defines $z = (R^p - \mathrm{REXFS}^p)/(R^p + \mathrm{REXFS}^p)$;
- XCOORD$=10$ defines $z = (R{-}\mathrm{REXFS})/R$ • XCOORD$=11$ defines $z = (R{-}\mathrm{REXFS})/\mathrm{REXFS}$

#38. `READ(5,*) (FSPRM(j,ifs), j= 1,NFSPRM(ifs))`

FSPRM(j,ifs) : are the NFSPRM parameters A_1, A_2, ... etc., characterizing this potential.

If performing a fit to experimental data (FITIT > 0 in *read* #1), now specify which potential parameters are free in the fit.

#39. `IF(FITIT.GT.0) READ(5,*) (FSVAR(j,ifs), j= 1, NFSPRM(ifs))`

FSVAR : an integer for each of the NFSPRM potential parameters input on the preceeding line. FSVAR(j, *ifs*) $= 1$ if parameter–j of state–*ifs* is to vary in the fit; otherwise set it $= 0$.

For a final-state potential defined partly by read-in turning points (FSTYPE = 2), read in the number of turning points NTPFS(*isf*) and parameters specifying how the interpolation over them and extrapolation beyond them is to be performed. The definitions of these parameters are precisely the same as those for the case of a pointwise initial-state potential (see reads #25 – 28), and are not repeated here.

```
#40. READ(5,*) NTPFS(ifs)
#41. READ(5,*) NUSEF, IR2F, ILRF, NCNF, CNNF
#42. READ(5,*) RFACTF, VFACTF, VSHIFTF(ifs)
#43. READ(5,*) (RTPF(i), VTPF(i), i=1, NTPFS(ifs))
```

D. (ii) Now specify the nature of the transition moment function for this case.

```
#44. READ(5,*) GFS(ifs), TMFTYP(ifs), OTMF(ifs)
```

GFS : is a positive integer giving the electronic degeneracy of the transition into this final state[38].

TMFTYPE : an integer specifying the nature of the transition moment or coupling function in the bound→continuum overlap integrals. For TMFTYP ≥ 0 it is a power series expansion in a radial coordinate ztmf(R) defined as follows:

- For TMFTYP = 0 the expansion coordinate is defined by interpolating over a read-in array of points. This allows use of an *ab initio* transition moment function, in which case one would normally set OTMF = 1 and the two associated expansion parameters (see *read* #42) as TMFPRM(*m,ifs*) = 0.0 and 1.0 for $m = 0$ and 1. itself.
- For TMFTYP = $p = 1 - 9$ the coordinate is $(R^p - \text{REXTMF}^p)/(R^p + \text{REXTMF}^p)$.
- For TMFTYP = 10 the coordinate is $(R - \text{REXTMF})/R$.
- For TMFTYP = 11 the coordinate is $(R - \text{REXTMF})/\text{REXTMF}$.
- For TMFTYP = 12 the coordinate is $1/R^2$.
- For TMFTYP = 13 the expansion coordinate is the distance coordinate R
- For TMFTYP < 0 the coupling operator is a sum of |TMFTYP| terms involving radial partial derivatives, each with the form: $-(\hbar^2/2\mu)\left\{\frac{dW(R)}{dR} + 2\,W(R)\,\frac{d}{dR}\right\}$, where $W(R)$ is the Lorentzian function $W(R) = a/[4\,a^2 + (R - R_c)^2]$. If TMFTYP $= -1$ input (and fit to, if desired) the two parameters a and R_c . If TMFTYP $= -2$ input (and fit to, if desired) two sets of Lorentzian parameters: a_1, $R_{c,1}$, a_2 and $R_{c,2}$. This case was introduced to allow treatment of non-adiabatic predissociation between two adiabatic potentials with an avoided crossing at R_c [39].

OTMF is an integer specifying the order of the power series in the variable ztmf. For TMFTYP < 0 it is a dummy variable.

```
#45. READ(5,*) (TMFPRM(m,ifs), m= 0, OTMF(ifs))
```

TMFPRM : the (real value) coefficients of the power series expansion for the transition moment or coupling function. For TMFTYP < 0 they are the 2 (or 4) Lorentzian parameters mentioned above.

If performing a fit to experimental data (FITIT > 0 in *read* #1), must now specify which transition moment/coupling finction parameters are free in the fit.

```
#46. IF(FITIT.GT.0) READ(5,*) (TMFVAR(m,ifs), m= 0, OTMF(ifs))
```

TMFVAR : an integer for each of the OTMF + 1 parameters input on the preceeding line. TMFVAR(m, *if*
1 if parameter–m of state–*ifs* is to vary in the fit; otherwise set it = 0 .

For a transition moment or coupling function defined by an array of read-in points (TMFTYP = 0), read in the number of turning points NPTMF and parameters specifying how the interpolation over them and extrapolation beyond them is to be performed. The definitions of these parameters are precisely the same as those for the case of a pointwise initial-state potential (see reads #22 – 25), and are not repeated here.

```
#47. READ(5,*) NPTMF, TMFLIM
#48. READ(5,*) NUSETMF, ILRTMF, NCNTMF, CNNTMF
#49. READ(5,*) RFACTMF, MFACTMF
#50. READ(5,*) (Xi(i), Yi(i), i=1, NPTMF)
```

End of the loop specifying the forms of the potential and transition moment functions for each of the NFS *final states.*

End of channel-5 input data file.

Acknowledgements

We are pleased to acknowledge the foresight and stimulation provided by Professor George Burns of the University of Toronto who instigated the work of RJL on this type of computational research tool back in 1966. We also acknowledge the inspiration and assistance provided by R.N. Zare's seminal 1964 Franck-Condon intensity factor paper and the associated reports[9, 10, 11]. We are also grateful to Jenning Y. Seto for assistance in developing the level-finder subroutine ALF. This work has been supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] a) R. J. Le Roy, *BCONT 1.0* University of Waterloo Chemical Physics Research Report CP-XX (19YY). b) *ibid, BCONT 1.1*, CP-XXX (19YY). c) *ibid, BCONT 1.3*, CP-XXX (19YY). e) *ibid, BCONT 1.4*, CP-329R^3 (1993).

[2] R. J. Le Roy and G. T. Kraemer, *BCONT 2.2. Computer Program for Calculating Absorption Coefficients, Emission Intensities or (Golden Rule) Predissociation Rates*, University of Waterloo Chemical Physics Research Report CP-650R^2 (2004). The source code and manual for this program may be obtained from "Computer Programs" link at *http://leroy.uwaterloo.ca*.

[3] R. J. Le Roy, R. G. Macdonald, and G. Burns, J. Chem. Phys. **65**, 1485 (1976).

[4] J. Tellinghuisen, in *Photodissociation and Photoionization*, Vol. 60 of *Adv. Chem. Phys.*, edited by K. P. Lawley (John Wiley & Sons Ltd., New York, 1985), pp. 299–369.

[5] H. Lefebvre-Brion and R. W. Field, *Perturbations in the Spectra of Diatomic Molecules* (Academic Press, New York, 1986).

[6] G. Herzberg, *Spectra of Diatomic Molecules* (Van Nostrand, Toronto, 1950).

[7] R. J. Le Roy, G. T. Kraemer, and S. Manzhos, J. Chem. Phys. **117**, 9353 (2002).

[8] J. K. G. Watson, J. Mol. Spectrosc. **80**, 411 (1980).

[9] 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).

[10] R. N. Zare, University of California Lawrence Radiation laboratory Report UCRL-10925 (1963).

[11] R. N. Zare, J. Chem. Phys. **40**, 1934 (1964).

[12] J. W. Cooley, Math. Computations **15**, 363 (1961).

[13] J. Cashion, J. Chem. Phys. **39**, 1872 (1963).

[14] R. J. Le Roy and R. B. Bernstein, J. Chem. Phys. **54**, 5114 (1971).

[15] R. J. Le Roy and W.-K. Liu, J. Chem. Phys. **69**, 3622 (1978).

[16] J. N. L. Connor and A. D. Smith, Mol. Phys. **43**, 397 (1981).

[17] P. G. Hajigeorgiou and R. J. Le Roy, J. Chem. Phys. **112**, 3949 (2000).

[18] J. Y. Seto, *Direct Fitting of Analytic Potential Functions to Diatomic Molecule Spectroscopic Data*, M.Sc. Thesis, Department of Chemistry, University of Waterloo (2000).

[19] W. Hua, Phys. Rev. A **42**, 2524 (1990).

[20] A. A. Šurkus, R. J. Rakauskas, and A. B. Bolotin, Chem. Phys. Lett. **105**, 291 (1984).

[21] J. L. Dunham, Phys. Rev. **41**, 721 (1932).

[22] G. Simons, R. G. Parr, and J. M. Finlan, J. Chem. Phys. **59**, 3229 (1973).

[23] J. F. Ogilvie, Proc. Roy. Soc. (London) **A 378**, 287 (1981).

[24] J. Hepburn, G. Scoles, and R. Penco, Chem. Phys. Lett. **36**, 451 (1975).

[25] C. Douketis *et al.*, J. Chem. Phys. **76**, 3057 (1982).

[26] R. J. Le Roy, in *Molecular Spectroscopy*, edited by R. Barrow, D. A. Long, and D. J. Millen (Chemical Society of London, London, 1973), Vol. 1, specialist periodical report 3, pp. 113–176.

[27] R. J. Le Roy, J. Chem. Phys. **73**, 6003 (1980).

[28] R. J. Le Roy, J. Chem. Phys. **101**, 10217 (1994).

[29] P. J. Mohr and B. N. Taylor, Rev. Mod. Phys. **72**, 351 (2000).

[30] G. Audi and A. H. Wapstra, Nucl. Phys. **A565**, 1 (1993).

[31] R. J. Le Roy, J. Mol. Spectrosc. **191**, 223 (1998).

[32] R. J. Le Roy, *LEVEL 7.5: A Computer Program for Solving the Radial Schrödinger Equation for Bound and Quasibound Levels*, University of Waterloo Chemical Physics Research Report CP-655 (2002). The source code and manual for this program may be obtained from the "Computer Programs" link at *http://leroy.uwaterloo.ca*.

[33] J. Y. Seto, R. J. Le Roy, J. Vergès, and C. Amiot, J. Chem. Phys. **113**, 3067 (2000).

[34] E. G. Lee *et al.*, J. Mol. Spectrosc. **194**, 197 (1999).

[35] J. Y. Seto *et al.*, J. Chem. Phys. **110**, 11756 (1999).

[36] J. A. Coxon and P. G. Hajigeorgiou, J. Mol. Spectrosc. **150**, 1 (1991).

[37] R. J. Le Roy, J. Mol. Spectrosc. **194**, 189 (1999).

[38] E. E. Whiting *et al.*, J. Mol. Spectrosc. **80**, 249 (1980).

[39] A. D. Bandrauk and M. S. Child, Mol. Phys. **19**, 95 (1970).