

Contents

Foreword	v
----------------	---

Chapter 1. RMPROP-Version 2: A Computer Program for Quantum Mechanical Close Coupling Calculations for Inelastic Collisions

*M.J. Unekis, D.W. Schwenke, N. Mullaney Harvey,
and D.G. Truhlar*

Introduction	1
Close Coupling Theory	4
Coupled Channels Equations	4
Asymptotic Boundary Conditions	8
R Matrix Propagation Algorithm	10
Sector Adiabatic Basis Functions	10
Sector Propagation Matrix	11
Stepsize Determination	14
Propagation Across Sector Boundaries	16
Homogeneous and Inhomogeneous Options	18
Reduction of the Number of Closed Channels Propagated in the Large-r Region	19
Asymptotic Analysis and Reordering of Channels	22
Single and Multiple Energy Runs	25
Program Structure	27
Segmentation of Program and Flow Chart	27
Restart Options	27
Vectorization	30
Matrix Utilities	30
Errata	31
Known Bugs in Version 1.0	31
Known Typographical errors in Chapter Describing Version 1.0	32
Concluding Remarks	33
Appendix	35
Atom-Diatom Scattering	35
Diatom-Diatom Scattering	38
References	45

Chapter 1.

RMPROP-VERSION 2: A Computer Program for Quantum Mechanical Close Coupling Calculations for Inelastic Collisions

Michael J. Unekis,* David W. Schwenke,† Nancy Mullaney Harvey,*
and Donald G. Truhlar*

Department of Chemistry, Chemical Physics Program, and
Supercomputer Institute, University of Minnesota
Minneapolis, MN 55455-0431, U.S.A.

and

NASA Ames Research Center, Moffett Field, CA 94035, U.S.A.

1. Introduction

The most popular approach to the quantum mechanical treatment of inelastic collisions is the close coupling method, which converts the partial differential Schrödinger equation with scattering boundary conditions into a set of coupled ordinary differential equations with nonhomogeneous boundary conditions.¹⁻⁵ These equations may be solved by boundary value methods⁶ or propagation techniques^{3-5,7-14}, and the techniques may also be subdivided into so-called approximate solution approaches³⁻⁷ and invariant embedding (also called approximate-potential) approaches.⁸⁻¹⁴ The R matrix propagation algorithm¹¹⁻¹⁴ is a stable and efficient invariant embedding algorithm for the solution of the close coupling equations for molecular collisions, and the present chapter provides an introduction to the computer program RMPROP-VERSION 2 which employs this method for molecular collisions.

RMPROP is a program for the solution of the close coupling formulation of the Schrödinger equation as applied to atomic and molecular collisions. It obtains the solution by propagation of Wigner's derivative matrix, called the R matrix, from small to large values of the scattering radial coordinate. It

* University of Minnesota.

† University of Minnesota and NASA Ames Research Center.

requires information about the asymptotic states of the collision partners and their interaction potential function, and it yields the scattering (S) matrix or elements of this matrix. The elements of the scattering matrix are the state-to-state or channel-to-channel scattering amplitudes from which all physical observables of a fixed-energy collision process may be calculated by the researcher. In the current version of the program, calculations may be performed for multiple energies, multiple potential function approximations, or multiple stepsizes in a single run, or information may be saved and re-used to perform calculations at one or more additional energies at a later date. Additionally, if desired, the program may be stopped and continued at a later date to propagate the solution to a larger value of the scattering radial coordinate.

The user must supply a subprogram that gives matrix elements of the interaction potential in the basis of the close coupling problem as a function of the center-of-mass separation r of the collision partners. The basis functions are labelled by a full set of quantum numbers of the asymptotically separated collision partners, and each basis function defines a "channel". The program is distributed with a test suite which contains several examples of such potential subprograms and solutions of the resulting close coupling equations with various numbers of channels. The potential functions in the test suite may readily be modified to perform calculations with other potential energy functions for a wide variety of problems. In the examples in the test suite the close coupling basis consists of simultaneous eigenfunctions of the total angular momentum J , the orbital angular momentum of relative translational motion ℓ , and the noninteracting Hamiltonians of the separated collision partners. In the Appendix to this chapter we present expressions for matrix elements of the potential between these basis functions for atom-rigid diatom collisions and for vibrating, rotating diatom-diatom collisions, for cases where the interaction potential is expressed in terms of common angular functions.

Some limitations on the program are that rearrangement collisions cannot be treated, and at least one of the collision partners must be neutral. (Both of these limitations are due to the type of boundary conditions that are imposed on the solutions at large values of the propagation coordinate r .) However, the current version of RMPROP (Version 2.0) has been modified from the first distributed version (RMPROP Version 1.0 in MOTECC-91) so that it can now handle systems without strong repulsions between the collision partners at zero separation (such as electron-atom and electron-molecule systems) in which some or all of the individual channel orbital angular momentum quantum numbers are zero. (Adding this capability involved generalizing the type of boundary condition that can be enforced at small values of the propagation coordinate).

The program solves the close coupling equations by R matrix propagation, which means that the scattering radial coordinate r , which equals the distance between the centers of mass of the collision partners, is divided into sectors, the R matrix is propagated locally across each sector, by assuming the potential is locally constant, and the local R matrices are combined into a global R matrix, which is propagated towards large r . This algorithm has two major advantages: (i) it is very stable, and (ii) the stepsize, i.e., the sector width, may be increased in regions where the interaction potential is relatively constant, so that very few steps are required in the large- r region. The major computational steps are as follows. In each sector the close coupling equations in terms of the interaction potential between the collision partners are transformed to a local, sector-dependent basis by diagonalizing the interaction potential matrix at the center of the sector. The transformed equations are solved under the assumption that the potential function does not vary across the sector. These solutions to the transformed equations and the derivatives of these solutions with respect to the center-of-mass separation (or information equivalent to the solutions and their derivatives) are matched to the next sector at the boundary so that a sector R matrix propagating the solution through a sector and across one of its boundaries is defined. When the sector R matrices are combined, a global R matrix, which propagates the solutions from the strong-interaction region at small r to the asymptotic region at large r , is obtained. The solutions in the asymptotic region are matched to scattering boundary conditions from which an S matrix is calculated. The user should repeat the calculation to converge it with respect to increasing the size of the close coupling basis, decreasing the sector widths, and increasing the range of r over which the propagation is carried out, as well as with respect to any numerical parameters associated with the interaction potential function.

Much of the operation of the program and practical instructions for its use are given in an ASCII documentation file which is distributed with the code. This file is called the "on-line manual". The on-line manual also describes a test suite which may be used to check whether the program is performing correctly in a new environment or on a new machine. Input files for the test suite are distributed as part of the code. The purpose of the present article is to summarize the theoretical and algorithmic background of the code.

2. Close Coupling Theory

This section provides an overview of the dynamical theory underlying the computational procedures. As described in the introduction, RMPROP may be used to calculate scattering matrix elements or state-to-state transition probabilities for atom-molecule, molecule-molecule, ion-atom, ion-molecule, electron-atom, or electron-molecule collisions. The current version of the code has no limitations on the orbital angular momentum; it may be zero or nonzero in any channel for any of these types of collisions. The treatment of the problem involves the reduction of the time-independent Schrödinger equation for the system to a set of coupled ordinary differential equations, the *close coupling* or *coupled channels* equations, and the subsequent solution of these equations.

2.1. Coupled channels equations

In the treatment which follows, we consider the electronically elastic collision of two vibrating, rotating molecules A and B, since this is a general example in which both collision partners have internal structure. The close coupling equations for other collision problems also lead to Eqs. (10) and (12) below and may also be treated by RMPROP. The program is in Hartree atomic units, although the equations in this chapter are valid in any (consistent) set of unit.

All physical information concerned with a collision of molecule A and molecule B may be derived from the scattering wave function $\Psi_n(\mathbf{r}, \mathbf{x}, E)$ which is a solution of the Schrödinger equation

$$H\Psi_{n_0}(\mathbf{r}, \mathbf{x}, E) = E\Psi_{n_0}(\mathbf{r}, \mathbf{x}, E) \quad (1)$$

where the vector \mathbf{r} connects the center of mass of A and the center of mass of B, \mathbf{x} represents all internal coordinates of the collision partners, E is the total energy, n_0 denotes the collection of quantum numbers needed to uniquely identify an initial channel, and H is the system Hamiltonian:

$$H = \frac{-\hbar^2}{2\mu} \nabla_r^2 + H_{int}(\mathbf{x}) + V(\mathbf{x}, \mathbf{r}) \quad (2)$$

where μ is the reduced mass of relative translational motion, ∇_r^2 is the Laplacian with respect to \mathbf{r} , H_{int} is the "internal Hamiltonian", defined as the sum of the Hamiltonians of the isolated A and B molecules, and $V(\mathbf{x}, \mathbf{r})$ is the interaction potential function which vanishes in the large- r limit. It is convenient to separate ∇_r^2 into its radial and angular parts and to partition H_{int} into an "easily" diagonalized part \tilde{H}_{int} (which may be just H_{int}) and the remainder

ΔH_{int} . We then combine the angular portion of $\frac{-\hbar^2}{2\mu} \nabla^2$ with \bar{H}_{int} to form the primitive Hamiltonian, H^0 , so that Eq. (2) becomes

$$H = \frac{-\hbar^2}{2\mu} \left\{ \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) \right\} + H^0(\mathbf{x}, r) + \mathcal{U}(\mathbf{x}, r) \quad (3)$$

where

$$H^0 = \frac{\ell_r^2}{2\mu r^2} + \bar{H}_{int}(\mathbf{x}) \quad (4)$$

$$\mathcal{U}(\mathbf{x}, r) = V(\mathbf{x}, r) + \Delta H_{int}(\mathbf{x}) \quad (5)$$

and where $\frac{\ell_r^2}{2\mu r^2}$ is the quantum mechanical operator for the centrifugal potential of the relative translational motion of A with respect to B.

To determine Ψ_n , we expand it in terms of simultaneous matrix eigenvectors of \bar{H}_{int} and ℓ_r^2 defined by

$$\int d\hat{\mathbf{r}} \int d\mathbf{x} X_m^*(\mathbf{x}, \hat{\mathbf{r}}) \bar{H}_{int} X_n(\mathbf{x}, \hat{\mathbf{r}}) = \delta_{mn} \bar{E}_n \quad (6)$$

$$\ell_r^2 X_n(\mathbf{x}, \hat{\mathbf{r}}) = \hbar^2 \ell_n(\ell_n + 1) X_n(\mathbf{x}, \hat{\mathbf{r}}) \quad (7)$$

and

$$\int d\hat{\mathbf{r}} \int d\mathbf{x} X_m^*(\mathbf{x}, \hat{\mathbf{r}}) X_n(\mathbf{x}, \hat{\mathbf{r}}) = \delta_{mn} \quad (8)$$

where δ_{mn} is the Kronecker delta, and Eq. (8) is an orthonormality condition. In these equations, $\hat{\mathbf{r}}$ is the unit vector which has the same direction as \mathbf{r} , and n and m are collective quantum numbers. Each different set of such quantum numbers is a possible initial or final quantum state for the collision and is called a channel. In terms of the above close coupling basis, the wave function expansion is

$$\Psi_{n_0}(\mathbf{x}, r, E) = \frac{1}{r} \sum_{n=1}^N X_n(\mathbf{x}, \hat{\mathbf{r}}) f_{nn_0}(r, E) \quad (9)$$

where each term in this expansion is associated with a different channel. Substituting Eq. (9) into Eq. (1), multiplying through from the left by $rX^*(\mathbf{x}, \hat{\mathbf{r}})$, and integrating over \mathbf{x} and $\hat{\mathbf{r}}$ yields

$$\left\{ \frac{-\hbar^2}{2\mu} \frac{d^2}{dr^2} + \ell_m(\ell_m + 1) \frac{\hbar^2}{2\mu r^2} + \tilde{\epsilon}_m \right\} f_{m n_0}(r, E) + \sum_{n=1}^N V_{mn}(r) f_{m n_0}(r, E) = E f_{m n_0}(r, E), \quad m = 1, 2, \dots, N \quad (10)$$

where

$$V_{mn}(r) = \int d\mathbf{x} \int d\hat{\mathbf{r}} X_m^*(\mathbf{x}, \hat{\mathbf{r}}) V(\mathbf{x}, \hat{\mathbf{r}}) X_n(\mathbf{x}, \hat{\mathbf{r}}). \quad (11)$$

In practice, the $V(\mathbf{x}, \hat{\mathbf{r}})$ is often expanded in terms of a set of known angular functions in order to facilitate the evaluation of Eq. (11). The integration over the angles in Eq. (11) may, if desired, then be evaluated analytically and the integration over the other variables by numerical quadrature. In the appendix we present formal expressions for the matrix elements $V_{mn}(r)$ given an expansion of the potential in Legendre functions (for atom-diatom systems) and in Launay body-frame functions (for diatom-diatom systems). These are two examples, and subroutines to evaluate the $V_{mn}(r)$ for potentials in these forms are included in the test suite, but we emphasize that RMPROP may be applied to any problem for which $V_{mn}(r)$ is available and for which at least one of the two collision partners is neutral. Equations (10) are called the close coupling equations (or the coupled channels equations), and they may be written in the form

$$\frac{d^2}{dr^2} \mathbf{f}(r, E) = \mathbf{D}(r, E) \mathbf{f}(r, E) \quad (12)$$

where a bold symbol denotes a matrix (except for \mathbf{x} , which denotes a set of coordinates, and $\hat{\mathbf{r}}$, which denotes a unit vector). The elements of \mathbf{f} are the f_{mn} of Eq. (9), where the rows of \mathbf{f} refer to different channels in which the radial translational wave function is represented, and the columns of \mathbf{f} refer to different sets of initial quantum numbers. The size of N , the number of terms in Eq. (9), is found by allowing one term for each channel of interest, and including sufficient additional terms so that the expansion (9) will be an accurate representation of the wave function. For example, for diatom-diatom collisions the quantum numbers included in n or n_0 are v_1 and v_2 , which are the vibrational quantum numbers of the two molecules, j_1 and j_2 , which are the internal rotational quantum numbers of the two molecules, j_{12} , which is the

quantum number associated with the vector sum of the angular momenta associated with j_1 and j_2 , ℓ (which may be different for different n and is therefore called ℓ_n above), which is the orbital angular momentum for relative motion of the collision partners, J and M , which are the total angular momentum and its projection onto a laboratory-fixed Z-axis, parity, ζ , which is $(-1)^{\ell+L}$, and η , which is the symmetry operator for interchange of the two indistinguishable molecules. The quantum number η is relevant only if the colliding molecules are identical.¹⁵ It is also important that due to the triangle inequality which governs coupling of the internal angular momentum vectors (here j_{12}) and ℓ to obtain J , more than one channel may contribute to the cross section for transitions involving a particular quantum mechanical state of the asymptotically separated subsystems (section 3.7).

The quantum numbers J , M , η , and ζ are "good" quantum numbers, so that the matrix \mathbf{D} is block diagonal in them; furthermore, \mathbf{D} is independent of M . If we use simultaneous eigenfunctions of their respective operators as our eigenfunctions, we partially uncouple the close coupling equations into independent sub-blocks, and this can reduce the computational effort for solution. In particular, we show below (section 4.1) that some portions of this algorithm contain operations which scale asymptotically proportionately to N^3 where N is the size of the close coupling expansion. Therefore, in the limit of large problems when the asymptotic scaling is applicable, a reduction in N by a factor of two by the uncoupling results in a factor of eight savings in the time required by some portions of the program in each run. (If both decoupled problems are solved the net savings is a factor of 4). In the applications included in the test suite we do in fact use simultaneous eigenfunctions of the above operators, and furthermore we only consider the blocks specified by $\eta = \zeta = +1$ (the *program* does *not* have this restriction), while allowing J to vary according to the problem and doing only one value of M .

The elements of \mathbf{D} are given by

$$D_{mn}(r, E) = \frac{2\mu}{\hbar^2} V_{mn}(r) + \delta_{mn} \left[\frac{\ell_n(\ell_n + 1)}{r^2} - \bar{k}_n^2 \right] \quad (13)$$

where

$$\bar{k}_n^2 = \frac{2\mu}{\hbar^2} (E - \bar{\epsilon}_n) \quad (14)$$

where \mathbf{D} is real and symmetric, and k_n is called the primitive wave number. The channels are ordered so that $k_n^2 \geq k_m^2$ if and only if $n \geq m$.

2.2. Asymptotic boundary conditions

In order to treat the large- r boundary conditions on Eq. (12) we must define a transformation which diagonalizes \mathbf{D} at large r . (This is similar to the diagonalization at finite r used in the R matrix propagation technique, which we discuss further in section 3.1.) At any r we may define functions

$$Z_m(\mathbf{x}, r) = \sum_{n=1}^N U_{mn}(r) X_n(\mathbf{x}, \hat{\mathbf{r}}) \quad (15)$$

such that

$$\sum_{k,l=1}^N U_{kn}(r) D_{kl}(r, E) U_{lm}(r) = \delta_{mn} [\lambda_{mn}(r, E)]^2. \quad (16)$$

It is noteworthy that in Eq. (13), the total energy E appears only on the diagonal of \mathbf{D} , due to the Kronecker delta multiplying the term k^2 which contains E . Because the E appears only on the diagonal of \mathbf{D} , the eigenvalues λ_{mn}^2 depend on E , but the eigenvectors (which are the columns of \mathbf{U}) do not. (This property is used when performing calculations at multiple energies so that second and subsequent energies do not require as much work.) If we use the Z_m as new basis functions (which are called adiabatic basis functions,^{11,16,17} or sometimes "sector adiabatic" or "quasidiabatic") the expansion of the wave function becomes

$$\Psi_{n_0}(\mathbf{x}, r, E) = \frac{1}{r} \sum_{m=1}^N Z_m(\mathbf{x}, r) g_{mn_0}(r, E) \quad (17)$$

where the g_{mn} are related to the f_{mn} by the application of the unitary transformation \mathbf{U} of Eq. (15). The terms in Eq. (17), like the ones in Eq. (9), are called channels.

We now consider small- r and large- r boundary conditions on the coupled equations. At small r , the program has an option (section 3.5) to choose either homogeneous or inhomogeneous boundary conditions; the homogeneous option requires regular boundary conditions at the origin:

$$g_{mn} \underset{r \rightarrow 0}{\sim} 0, \quad 1 \leq n, m \leq N \quad (18)$$

and the inhomogeneous option allows nonvanishing g at the origin. At large r , there is more than one equivalent form of the boundary conditions to the close coupling equations. The program allows the choice of sine/cosine boundary conditions:

$$g_{mn} \underset{r \rightarrow \infty}{\sim} \begin{cases} \delta_{mn} \sin \left[|k_m(E)| r - \frac{\ell_m \pi}{2} \right] + a_{mn}(E) \cos \left[|k_m(E)| r - \frac{\ell_m \pi}{2} \right], & 1 \leq m \leq P^o \\ \delta_{mn} b \exp [|k_m(E)| r] + a_{mn}(E) \exp [- |k_m(E)| r], & P^o + 1 \leq m \leq N \end{cases} \quad (19)$$

or of matching to Ricatti-Bessel functions:

$$g_{mn} \underset{r \rightarrow \infty}{\sim} \begin{cases} \delta_{mn} j_{\ell_m} [|k_m(E)| r] - a_{mn}(E) n_{\ell_m} [|k_m(E)| r] & 1 \leq m \leq P^o \\ \delta_{mn} b \exp [|k_m(E)| r] + a_{mn}(E) \exp [- |k_m(E)| r] & P^o + 1 \leq m \leq N \end{cases} \quad (20)$$

where j_{ℓ} and n_{ℓ} are the Ricatti-Bessel functions:

$$\begin{aligned} j_{\ell}(x) &= \left(\frac{\pi}{2} \right)^{\frac{1}{2}} J_{\ell + \frac{1}{2}}(x) \\ n_{\ell}(x) &= (-1)^{\ell+1} \left(\frac{\pi x}{2} \right)^{\frac{1}{2}} J_{-\ell - \frac{1}{2}}(x) . \end{aligned} \quad (21)$$

In Eq. (19), the coefficient b is arbitrary, k_m is an asymptotic wave number defined by

$$|k_m| = \lim_{r \rightarrow \infty} |\lambda_{mn}(r, E)| \quad (22)$$

because k_m is imaginary if λ_{mn}^2 is negative, and where P^o is defined as the number of open channels in the basis, where an open channel is defined as one with positive k_m^2 . Channels which are not "open" are said to be "closed". If H_{mn} is the same as \bar{H}_{mn} , then k_m equals $|k_m|$ and the boundary conditions (19) and (20) apply to the f_{mn} as well as to the g_{mn} .

The final result of the calculation is the $P^o \times P^o$ unitary scattering matrix S defined by

$$S = [1 + iK(E)] [1 - iK(E)]^{-1} \quad (23a)$$

or the transition matrix defined by

$$\mathbf{T} = \mathbf{S} - \mathbf{1} \quad (23b)$$

where $\mathbf{1}$ is the unit matrix, $i^2 = -1$, and \mathbf{K} is the $P^o \times P^o$ reactance matrix with elements

$$K_{nm} = k_n^{1/2} a_{nm} k_m^{1/2}, \quad 1 \leq n, m \leq P^o. \quad (24)$$

Note that a_{nm} may be calculated using either (19) or (20), but if the calculation is converged the results will be the same from either equation.

One may then calculate all desired physical observables from the scattering matrix or transition matrix by standard formulas.^{18,19}

3. R Matrix Propagation Algorithm

The R matrix propagation algorithm was presented originally by Light and Walker¹¹ for collinear atom-diatom collisions, and it has been generalized by various workers, including those authors. Our own generalization,^{12-14,16,17,20,21} as used in RMPROP, is based directly on their original paper and is reviewed below.

3.1. Sector adiabatic basis functions

In R matrix propagation the close coupling equations expressed as in Eq. (12) are solved by subdividing the coordinate r into some number N_i of sectors, with sector midpoints $r_c^{(i)}$ and sector widths $h^{(i)}$ such that

$$r_c^{(i+1)} = r_c^{(i)} + \frac{[h^{(i+1)} + h^{(i)}]}{2} \quad (25)$$

In sector i , it is convenient to expand the wave function Ψ_n in terms of sector-dependent functions defined by

$$Z_m^{(i)}(\mathbf{x}, \hat{\mathbf{r}}) = \sum_{n=1}^N T_{nm}^{(i)} X_n(\mathbf{x}, \hat{\mathbf{r}}), \quad 1 \leq m \leq P^{(i)}. \quad (26)$$

The $N \times P^{(i)}$ rectangular matrix $\mathbf{T}^{(i)}$ is comprised of the first $P^{(i)}$ columns of the $N \times N$ matrix $\mathbf{U}^{(i)}$, where $\mathbf{U}^{(i)}$ diagonalizes \mathbf{D} at the center of sector i ,

$$\sum_{k, k'=1}^{\nu} U_{kn}^{(i)} D_{kk'}(r_c^{(i)}, E) U_{k'm}^{(i)} = \delta_{nm} [\lambda_{nn}^{(i)}(E)]^2 \quad (27)$$

and where the eigenvalues are ordered in ascending order.

The wave function Ψ_n is expressed in terms of the new functions $Z_m^{(i)}$ by

$$\Psi_n(\mathbf{x}, \mathbf{r}, E) = \frac{1}{r} \sum_{m=1}^{p^{(i)}} Z_m^{(i)}(\mathbf{x}, \hat{\mathbf{r}}) g_{mn}^{(i)}(r, E), \quad (28)$$

where the $g^{(i)}$ solve the equation (see also section 3.2 below)

$$\frac{d^2}{dr^2} \mathbf{g}^{(i)}(r, E) = \mathbf{L}^{(i)}(r, E) \mathbf{g}^{(i)}(r, E) \quad (29)$$

where

$$L_{nm}^{(i)}(r, E) = \sum_{k, k'=1}^N T_{kn}^{(i)} D_{kk'}(r, E) T_{k'm}^{(i)}, \quad 1 \leq n, m \leq P^{(i)} \quad (30)$$

so that the channels of Eq. (28) are uncoupled at the center of sector i . The new radial functions $g^{(i)}$ are related to the functions f by

$$g_{mn}^{(i)}(r, E) = \sum_{k=1}^N T_{km}^{(i)} f_{kn}(r, E), \quad 1 \leq m, n \leq P^{(i)} \quad (31)$$

3.2. Sector propagation matrix

We consider first the homogeneous small- r boundary conditions, Eq. (18). In principle we should have $r_c^{(i)} - \frac{\hbar^2}{2}$ equal to zero. However when both collision partners are atoms, molecules, or ions or for electron scattering when none of the ℓ_n in eq. (13) is zero, one finds that

$$V_{nn}(r) + \frac{\ell_n(\ell_n + 1)}{2\mu r^2} \gg E, \quad \text{all } n, \text{ and } r \ll \sigma, \quad (32)$$

where σ is some (small) distance at which the collision partners begin to repel strongly or have a high centrifugal potential. This implies that for decreasing r , all $f_{mn}(r)$, and therefore all $g_{mn}(r)$, decrease rapidly and are negligible for r less

than some finite nonzero value of r . When this is true we choose such a nonzero value of r to be $r_c^{(i)}$ and therefore avoid the work involved in propagating the solution to the equations over a region in which it is already known to be essentially zero. Equally important if the homogeneous boundary condition at small r is satisfied, it allows us to simplify the propagation, as discussed below Eq. (61).

When restriction (32) is not applicable, as in electron-atom scattering where the orbital angular momentum ℓ_n of the electron is zero, we start the integration at (or essentially at) the origin and we use inhomogeneous boundary conditions. (The inhomogeneous propagation mode may also be useful for certain ways of running the code in multiprocessor mode. Although we do not pursue this parallelization strategy in the present article, the inclusion of the inhomogeneous option makes version 2.0 of RMPROP "parallel ready".) It should be noted that the inhomogeneous option of the program will also give correct results for systems for which the homogeneous option is suitable for use, but it will carry out unnecessary work in such cases.

We begin the propagation with N channels, that is, with $P^{(i)} = N$. At large r , because of Eq. (19), f_{mn} for $m > P^n$ decays rapidly to zero. We can reduce the computational expense of solution by allowing $P^{(i)}$ to decrease in this region according to some given criterion.²¹ In order to simplify the algorithm, we also decrease the number of channels by no more than one per sector, that is $P^{(i+1)}$ must equal $P^{(i)}$ or $P^{(i)} - 1$. We also impose the constraint $P^{(i)} \geq P^n$ for all (i). We define the $2P^{(i)} \times 2P^{(i)}$ sector propagator $\mathbf{P}^{(i)}$ by

$$\mathbf{G}_L^{(i)}(E) = \mathbf{P}^{(i)}(E) \mathbf{G}_R^{(i)}(E), \quad (33)$$

where the $2P^{(i)} \times 2P^{(i)}$ matrix $\mathbf{G}^{(i)}$ is given by

$$\mathbf{G}^{(i)}(E) = \begin{pmatrix} \mathbf{g}^{(i)}(r, E) \\ \mathbf{g}'^{(i)}(r, E) \end{pmatrix} \quad (34)$$

where $\mathbf{g}^{(i)}(r, E)$ denotes $\frac{d\mathbf{g}^{(i)}}{dr}$, and the subscripts L and R denote a quantity evaluated at the left and right hand sides of a sector, respectively:

$$\mathbf{H}_L^{(i)} = \mathbf{H}_c^{(i)} - \frac{\mathbf{H}^{(i)}}{2}, \quad (35)$$

$$r_R^{(i)} = r_c^{(i)} + \frac{h^{(i)}}{2} \quad (36)$$

We partition the $\mathbf{P}^{(i)}(E)$ of Eq. (33) so that

$$\mathbf{P}^{(i)}(E) = \begin{pmatrix} \mathbf{P}_1^{(i)} & \mathbf{P}_2^{(i)} \\ \mathbf{P}_3^{(i)} & \mathbf{P}_4^{(i)} \end{pmatrix}, \quad (37)$$

and so that the matrices $\mathbf{P}^{(i)}(E)$ are all square.

We use the first-order Magnus method^{11,22-24} for $\mathbf{P}^{(i)}(E)$. The $\mathbf{g}^{(i)}$ are the $2\mathbf{P}^{(i)}$ linearly independent solution vectors to equation (29). Since each column $\mathbf{g}^{(i)}$ of $\mathbf{G}^{(i)}$ is linearly independent, we can write a propagation equation similar to (33) for each of the $\mathbf{g}^{(i)}$, using the partitioned $\mathbf{P}^{(i)}(E)$ of equation (37):

$$\begin{pmatrix} \mathbf{g}_L^{(i)} \\ \mathbf{g}_R^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_1^{(i)} & \mathbf{P}_2^{(i)} \\ \mathbf{P}_3^{(i)} & \mathbf{P}_4^{(i)} \end{pmatrix} \begin{pmatrix} \mathbf{g}_R^{(i)} \\ \mathbf{g}_L^{(i)} \end{pmatrix}. \quad (38)$$

Following the work of Pechukas and Light,²² we use exponential operators to translate the $\mathbf{g}^{(i)}(r)$ and $\mathbf{g}'^{(i)}(r)$ across a sector of width $h^{(i)}$ centered at $r = r_c^{(i)}$, i.e.

$$\mathbf{P}^{(i)}(E) = \exp \left[\mathbf{B} \left(r_c^{(i)} - \frac{h^{(i)}}{2}, r_c^{(i)} + \frac{h^{(i)}}{2} \right) \right] \quad (39)$$

for which, to third order in the stepsize, \mathbf{B} is given by

$$\mathbf{B} = -h^{(i)} \begin{pmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{L}^{(i)}(r_c^{(i)}, E) & \mathbf{0} \end{pmatrix} - \frac{(h^{(i)})^3}{12} \begin{pmatrix} -\mathbf{L}^{(i)'}(r_c^{(i)}, E) & \mathbf{0} \\ \mathbf{0} & -\mathbf{L}^{(i)'}(r_c^{(i)}, E) \end{pmatrix} \quad (40)$$

+ Order $[(h^{(i)})^5]$

where

$$h^{(i)} = r_R^{(i)} - r_L^{(i)} \quad (41)$$

and where

$$\mathbf{L}^{(i)'}(r_c^{(i)}, E) = \left. \frac{d\mathbf{L}^{(i)}}{dr} \right|_{r=r_c^{(i)}(E)} \quad (42)$$

with the $L^{(i)}$ defined by Eq. (30). In the first Magnus approximation, the second and higher terms in (40) are neglected. This is tantamount to setting the derivative in (42) equal to zero so that the matrix $L^{(i)}$ is approximated by the diagonal matrix $[\lambda^{(i)}(E)]^2$ throughout sector (i). Therefore the first Magnus propagator would be the exact propagator when the interaction matrix is independent of r in sector (i), and for this reason it converges to the exact result as the sector width is decreased. Substituting Eq. (30) into Eqs. (39) and (40) and using Eq. (37), one obtains for the $P_1^{(i)}(E)$ using the first Magnus approximation:

$$[P_1^{(i)}(E)]_{nm} = [P_4^{(i)}(E)]_{nm} = \begin{cases} \delta_{nm} \cosh \left[-h^{(i)} |\lambda_{nn}^{(i)}(E)| \right], & \lambda_{nn}^{(i)2}(E) > 0 \\ \delta_{nm} \cos \left[-h^{(i)} |\lambda_{nn}^{(i)}(E)| \right], & \lambda_{nn}^{(i)2}(E) < 0 \end{cases} \quad (43)$$

$$[P_2^{(i)}(E)]_{nm} = \begin{cases} \delta_{nm} |\lambda_{nn}^{(i)}(E)|^{-1} \sinh \left[-h^{(i)} |\lambda_{nn}^{(i)}(E)| \right], & \lambda_{nn}^{(i)2}(E) > 0 \\ \delta_{nm} |\lambda_{nn}^{(i)}(E)|^{-1} \sin \left[-h^{(i)} |\lambda_{nn}^{(i)}(E)| \right], & \lambda_{nn}^{(i)2}(E) < 0 \end{cases} \quad (44)$$

$$[P_3^{(i)}(E)]_{nm} = \begin{cases} \lambda_{nn}^{(i)2}(E) (P_2^{(i)})_{nm}, & \lambda_{nn}^{(i)2}(E) > 0 \\ -|\lambda_{nn}^{(i)2}(E)| (P_2^{(i)})_{nm}, & \lambda_{nn}^{(i)2}(E) < 0 \end{cases} \quad (45)$$

3.3. Stepsize determination

Examination of Eq. (40) reveals that the error of the first Magnus propagator is proportional to the size of the (neglected) second term in the series, which is $(h^{(i)})^3 \frac{dL^{(i)}}{dr}$, so it is desirable to choose the stepsize to minimize the error. In order to do this one should choose $h^{(i)}$ so that the effect of the second- and higher-order terms is negligible. However, it is required to know $h^{(i)}$ before one may calculate $L^{(i)}$ for the sector with $r = r_c^{(i)}$, so that strictly speaking, second-order terms in a sector cannot be calculated before completing that sector. It is therefore necessary to *estimate* an error in order to decide on a stepsize. The estimated error in sector $(i+1)$ is given by

$$\text{error} \propto h^{(i+1)3} \sqrt{\left[\frac{1}{N} \sum_{j=1}^N \left(\frac{dD_{jj}^{(i)}}{dr} \right)^2 \right]} \quad (46)$$

Since one has not yet evaluated $D^{(i+1)}$ when computing the stepsize at sector $(i+1)$, one must estimate the numerical derivative using $D^{(i)}$ and $D^{(i-1)}$. This

translates into the following algorithm. First, a trial width is determined according to

$$h^{(trial)} = \min \left\{ \begin{array}{l} \epsilon \left[\sum_{j=1}^N \left(\frac{D_{jj}^{(i)} - D_{jj}^{(i)}}{r_c^{(i)} - r_c^{(i)}} \right)^2 \right]^{-\frac{1}{6}} \\ h_{\max} \end{array} \right. \quad (47a)$$

where ϵ and h_{\max} are input parameters to the program, and where h_{\max} may vary in different regions of the propagation according to other input to FORTRAN unit 5. Then the stepsize in sector (i-1) is given as

$$h^{(i+1)} = \max \left\{ \begin{array}{l} h_{\min} \\ h^{(trial)} \end{array} \right. \quad (47b)$$

where h_{\min} is another input parameter to the program and may be allowed to differ in the same regions of the propagation as h_{\max} .

Another approach may be used, and that is to use a constant stepsize throughout the propagation. This technique, however, negates one of the most useful features of the R matrix propagation algorithm, which is the ability to take large stepsizes in regions where the interaction potential is not sensitive to the center-of-mass separation of the collision partners, and which allows one to greatly reduce the number of steps taken in the asymptotic region of a calculation. In addition, care must be exercised that final results are converged with respect to the fixed stepsize, so that the total time necessary to obtain converged results with a fixed stepsize as compared to using variable stepsizes is much greater than just the ratio of the variable and fixed stepsizes themselves.

The program also allows using fixed stepsizes and using one value of ϵ in one interval and another value of ϵ in a different interval. (See the on-line manual.) For many problems, a fixed stepsize at small r followed by one or two regions with ϵ -controlled stepsizes at larger r is the most smoothly convergent and efficient procedure.

3.4. Propagation across sector boundaries

The continuity of the functions $g^{(i)}$ of Eq. (29) across sector boundaries may be expressed by

$$g_R^{(i-1)}(E) = T(i-1, i) g_L^{(i)}(E) \quad (48)$$

$$g_R^{(i-1)}(E) = T(i-1, i) g_A^{(i)}(E) \quad (49)$$

where the overlap matrix $T(i-1, i)$ is defined by

$$T_{nm}(i-1, i) = \sum_{k=1}^N T_{kn}^{(i-1)} T_{km}^{(i)} \quad 1 \leq n, m \leq P^{(i)} \quad (50)$$

Therefore, when $P^{(i-1)}$ does not equal $P^{(i)}$ only the upper left $P^{(i)} \times P^{(i)}$ portion of $g^{(i-1)}$ is used to calculate $g^{(i)}$. We now define the sector R matrix $r^{(i)}$ by

$$\begin{pmatrix} g_R^{(i-1)}(E) \\ g_R^{(i)}(E) \end{pmatrix} = \begin{pmatrix} r_1^{(i)}(E) & r_2^{(i)}(E) \\ r_3^{(i)}(E) & r_4^{(i)}(E) \end{pmatrix} \begin{pmatrix} g_R^{(i-1)}(E) \\ -g_R^{(i)}(E) \end{pmatrix} \quad (51)$$

$$r^{(i)}(E) = \begin{pmatrix} r_1^{(i)}(E) & r_2^{(i)}(E) \\ r_3^{(i)}(E) & r_4^{(i)}(E) \end{pmatrix} \quad (52)$$

where the $r_j^{(i)}$ matrices are $P^{(i)} \times P^{(i)}$. The $r_j^{(i)}$ are expressed in terms of the $P_j^{(i)}(E)$ and the $T(i-1, i)$ as follows:

$$r_1^{(i)}(E) = T(i-1, i) P_1^{(i)}(E) [P_3^{(i)}(E)]^{-1} [T(i-1, i)]^{-1} \quad (53)$$

$$r_2^{(i)}(E) = T(i-1, i) [P_3^{(i)}(E)]^{-1} \quad (54)$$

$$r_3^{(i)}(E) = [P_3^{(i)}(E)]^{-1} [T(i-1, i)]^{-1} \quad (55)$$

$$r_4^{(i)}(E) = [P_3^{(i)}(E)]^{-1} P_4^{(i)}(E) \quad (56)$$

It should be noted that $P_j^{(i)}$ is diagonal so that its inversion is not time-consuming.

The global R matrix $\mathbf{R}^{(i)}(E)$ which spans from the left-hand side of the first sector to the right-hand side of sector (i) is defined by

$$\begin{pmatrix} \mathbf{g}_L^{(i)}(E) \\ \mathbf{g}_R^{(i)}(E) \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^{(i)}(E) & \mathbf{R}_2^{(i)}(E) \\ \mathbf{R}_3^{(i)}(E) & \mathbf{R}_4^{(i)}(E) \end{pmatrix} \begin{pmatrix} \mathbf{g}_L^{(1)}(E) \\ -\mathbf{g}_R^{(1)}(E) \end{pmatrix} \quad (57)$$

where the $\mathbf{R}_j^{(i)}$ matrices are all square.

The equations for $\mathbf{R}^{(i)}$, $\mathbf{R}^{(i)}$, and $\mathbf{R}^{(i)}$ are:

$$\mathbf{R}_1^{(i)}(E) = \mathbf{R}_1^{(i-1)}(E) - \mathbf{R}_2^{(i-1)}(E)\mathbf{S}(i-1, i, E)\mathbf{R}_3^{(i-1)}(E) \quad (58a)$$

$$\mathbf{R}_2^{(i)}(E) = \mathbf{R}_2^{(i-1)}(E)\mathbf{S}(i-1, i, E)\mathbf{r}_2^{(i)}(E) \quad (58b)$$

$$\mathbf{R}_3^{(i)}(E) = \mathbf{r}_3^{(i)}(E)\mathbf{S}(i-1, i, E)\mathbf{R}_3^{(i-1)}(E) \quad (58c)$$

where

$$\mathbf{S}(i-1, i, E) = [\mathbf{R}_4^{(i-1)}(E) + \mathbf{r}_1^{(i)}(E)]^{-1}. \quad (58d)$$

The matrix $\mathbf{R}^{(i)}$ only depends on $\mathbf{R}^{(i-1)}$ and the $\mathbf{r}^{(i)}$, and not on the $\mathbf{R}^{(i-1)}$, $\mathbf{R}^{(i-1)}$, or $\mathbf{R}^{(i-1)}$:

$$\mathbf{R}_4^{(i)}(E) = \mathbf{r}_4^{(i)}(E) - \mathbf{r}_3^{(i)}(E) [\mathbf{R}_4^{(i-1)}(E) + \mathbf{r}_1^{(i)}(E)]^{-1} \mathbf{r}_2^{(i)}(E). \quad (58e)$$

The values of the four submatrices of $\mathbf{R}^{(1)}(E)$ in sector (1) are given by

$$\mathbf{R}_1^{(1)}(E) = \mathbf{P}_1^{(1)}(E)\mathbf{P}_3^{(1)}(E)^{-1} \quad (59a)$$

$$\mathbf{R}_2^{(1)}(E) = \mathbf{R}_3^{(1)}(E) = \mathbf{P}_3^{(1)}(E)^{-1} \quad (59b)$$

$$\mathbf{R}_4^{(1)}(E) = \mathbf{r}_4^{(1)}(E). \quad (59c)$$

The program however has an option to calculate the $\mathbf{R}^{(i)}$ treating the $\mathbf{r}^{(i)}$ implicitly instead of explicitly. This option is chosen by setting the value of the input variable NPRP equal to 2. When this option is chosen, the program does not explicitly store the sector r matrices $\mathbf{r}^{(i)}$, $\mathbf{r}^{(i)}$, $\mathbf{r}^{(i)}$, or $\mathbf{r}^{(i)}$ at each step and perform

the operations of Eq. (58e) on them to obtain the \mathbf{R}^{ψ} . Instead, the program uses temporary arrays, which contain intermediate values used in the determination of the $r^{(i)}$ to solve Eq. (58e) at each sector. The use of the temporary arrays means that one less matrix inversion, and several fewer matrix-vector multiplications are performed at each step, and that for "later date second energy runs" (Section 3.8) the $[\mathbf{T}(i-1, 1)]^{-1}$ need not be written to disk, since it is never computed, which will decrease storage requirements. However, in version 2.0 of RMPROP this option may only be used in conjunction with the homogeneous option (Section 3.5), where only the \mathbf{R}^{ψ} submatrix of the full matrix $\mathbf{R}^{(i)}(E)$ is propagated at each step.

3.5. Homogeneous and inhomogeneous options

The homogeneous and inhomogeneous options in the program refer to the selection of the small- r boundary conditions. To use inhomogeneous boundary conditions the logical variable, called LR4MT, is set false (see the on-line manual). When this variable is true the program propagates just the submatrix \mathbf{R}^{ψ} , and when it is false, the program propagates all four submatrices, \mathbf{R}^{ψ} , \mathbf{R}^{ψ} , \mathbf{R}^{ψ} , and \mathbf{R}^{ψ} of the global $\mathbf{R}^{(i)}(E)$ matrix. The choice of this option is determined by the physical conditions of the scattering problem, as described below.

Consider the situation where all channels are strongly closed at the starting point of the propagation, that is, $[\lambda_{nn}^{(i)}]^2 > \frac{2\mu E}{\hbar^2}$ for all n . In this case, Eqs. (43) and (45) show that both $[\mathbf{P}_1^{(i)}(E)]_{nn}$ and $[\mathbf{P}_3^{(i)}(E)]_{nn}$ become exponentially large:

$$[\mathbf{P}_1^{(i)}(E)]_{nn} \underset{\lambda_{nn}^{(i)}(E) \rightarrow \infty}{\approx} \exp [|\lambda_{nn}^{(i)}(E)| h^{(i)}] , \quad (60)$$

$$[\mathbf{P}_3^{(i)}(E)]_{nn} \underset{\lambda_{nn}^{(i)}(E) \rightarrow \infty}{\approx} - |\lambda_{nn}^{(i)}(E)| \exp [|\lambda_{nn}^{(i)}(E)| h^{(i)}] . \quad (61)$$

Combining Eqs. (60) and (61) with Eq. (59b) shows that $\mathbf{R}_2^{(i)}$ and $\mathbf{R}_4^{(i)}$ are approximately zero under these conditions. If $\mathbf{R}_2^{(i)}$ and $\mathbf{R}_4^{(i)}$ are negligible, then Eqs. (58) show that \mathbf{R}^{ψ} and \mathbf{R}^{ψ} remain small throughout the entire propagation. It will be seen from Eq. (87b) below that when $\mathbf{R}_2^{(i)}$ and $\mathbf{R}_4^{(i)}$ are negligible, all scattering information (a_{mn} , $1 \leq m, n \leq P^{(i)}$) can be determined solely from $\mathbf{R}_1^{(i)}$ and thus only this submatrix needs to be included in the propagation. Solution of the equations under this assumption is called the homogeneous option. The inhomogeneous option, as its name implies, involves solution of

the equations subject to inhomogeneous small- r boundary conditions and subsequent propagation of the full $\mathbf{R}^{(l)}(E)$ matrix.

The homogeneous option is applicable to atom-atom, atom-molecule, molecule-molecule, ion-atom, and ion-molecule scattering, i.e., for systems in which the interaction potential is steeply repulsive at the origin, since Eqs. (60)-(61) may be easily satisfied if the propagation is begun sufficiently far enough inside the repulsive potential. The inhomogeneous option must be used for electron scattering problems in which one or more of the channels had a zero value of l , because for such systems the potential tends to negative infinity for small r , and so the requirement that $[\lambda_{lm}^{(l)}]^2 \gg \frac{2\mu E}{\hbar^2}$ is not fulfilled.

The inhomogeneous option is formally applicable to heavy-particle systems as well, and will give the same scattering matrix as the homogeneous option, but use of this option for these systems is computationally inefficient. The use of the inhomogeneous option requires that the full $\mathbf{R}^{(l)}(E)$ matrix be propagated rather than the \mathbf{R}^{ψ} , which increases the memory required by the program. Furthermore, matrix manipulations involving the \mathbf{R}^{ψ} , \mathbf{R}^{ψ} , and \mathbf{R}^{ψ} are required, which slows execution of the program.

The program contains several other options and approximations (see the discussions after equations (59a)-(59c) and in Section 3.6) which decrease the memory and CPU requirements even further, but which may only be used with the homogeneous option.

3.6. Reduction of the number of closed channels propagated in the large- r region

As explained in the derivation of the close coupling equations, the expansion of the wave function often includes a number of terms which are energetically inaccessible to the system, over some or all of the r range. These are needed for convergence at small r , but when closed at large separations, they do little more than slow down the computation. In this section we consider the conditions under which the number of channels propagated can be reduced from P to P^o , where P^o is the number of open channels in the asymptotically large- r region, without introducing errors in the reactance matrix \mathbf{K} or scattering matrix \mathbf{S} . In practice we only drop one channel per sector, and we only attempt to drop channels in the large- r region. We consider the case where only $\mathbf{R}^{\psi}(E)$ is propagated, and we allow dropping channels only with the constraint that $P^{(l)}(E) \geq P^o(E)$. Before proceeding, it will be convenient to introduce the fol-

lowing convention for partitioning a $P \times P$ matrix into quadrants according to whether open and/or closed channels are linked there:

$$Y = \begin{pmatrix} Y^{oo} & Y^{oc} \\ Y^{co} & Y^{cc} \end{pmatrix} \quad (62)$$

where the superscript oo identifies the $P^o \times P^o$ submatrix containing Y_{mn} for $1 \leq m \leq P^o$ and $1 \leq n \leq P^o$, i.e., the matrix elements linking open channels to open channels, the superscript oc identifies the $P^o \times (P - P^o)$ submatrix containing Y_{mn} for $1 \leq m \leq P^o$ and $P^o < n \leq P$, i.e., the matrix elements linking open channels to closed channels, the superscript co identifies the $(P - P^o) \times P^o$ submatrix containing the Y_{mn} for $P^o < m \leq P$ and $1 \leq n \leq P^o$, and the superscript cc identifies the $(P - P^o) \times (P - P^o)$ submatrix containing Y_{mn} for $P^o < m \leq P$ and $P^o < n \leq P$, i.e., the matrix elements linking closed channels to closed channels. To justify dropping the $P - P^o$ closed channels from propagation in sector $(i - 1)$, we must show that to accurately obtain $R^{(i)oo}(E)$ from $R^{(i-1)oo}(E)$ does not require any information involving the $P - P^o$ closed channels. (We only require the oo submatrix of $R^{(i)}(E)$ because the submatrices involving closed channels are not required in the asymptotic analysis to obtain the reactance matrix, when the reactance matrix is derived from $R^{(i)}(E)$ and not from the full $R^{(i)}(E)$). Using equation (58) and the notation for partitioned matrices, $R^{(i)oo}(E)$ can be written as

$$R_4^{(i)oo}(E) = r_4^{(i)oo}(E) - r_3^{(i)oo}(E) [S^{oo}(i-1, i, E)r_2^{(i)oo}(E) + S^{oc}(i-1, i, E)r_2^{(i)co}(E)] - r_3^{(i)oc}(E) [S^{co}(i-1, i, E)r_2^{(i)oo}(E) + S^{cc}(i-1, i, E)r_2^{(i)co}(E)] \quad (63)$$

In our calculations both $T(i-1, i)$ and $T^{-1}(i-1, i)$ tend to a unit matrix in the large- r limit, (provided that $H = H_{int}$), so equation (53) reduces to

$$\lim_{r_L^{(i)} \rightarrow \infty} r_1^{(i)} = P_1^{(i)}(E) [P_3^{(i)}(E)]^{-1}, \quad (64)$$

i.e., it tends to a diagonal matrix. Additionally, it can be shown¹³ that $R^{(i)}(E)$ tends to a block-diagonal matrix in the large- r limit. Therefore, for large r ,

$$\lim_{r_L^{(i)} \rightarrow \infty} S(i-1, i)^{oo} = [R_4^{(i-1)oo}(E) + r_1^{(i)oo}(E)]^{-1}, \quad (65)$$

$$\lim_{r_L^{(i)} \rightarrow \infty} S(i-1, i)^{oc} = 0^{oc}, \quad (66)$$

$$\lim_{r_L^{(i)} \rightarrow \infty} S(i-1, i)^{co} = \mathbf{0}^{co} , \quad (67)$$

$$\lim_{r_L^{(i)} \rightarrow \infty} S(i-1, i)^{cc} = [\mathbf{R}_4^{(i-1)cc}(E) + \mathbf{r}_1^{(i)cc}(E)]^{-1} . \quad (68)$$

Substituting these into (63) yields

$$\begin{aligned} \lim_{r_L^{(i)} \rightarrow \infty} \mathbf{R}_4^{(i)oo}(E) &= \mathbf{r}_4^{(i)oo}(E) - \mathbf{r}_3^{(i)oo}(E) \mathbf{S}^{oo}(i-1, i, E) \mathbf{r}_2^{(i)oo}(E) \\ &\quad - \mathbf{r}_3^{(i)oc}(E) \mathbf{S}^{cc}(i-1, i, E) \mathbf{r}_2^{(i)co}(E) \end{aligned} \quad (69)$$

Since, at asymptotically large r , $\mathbf{T}(i-1, i)$ approaches a unit matrix, Eqs. (54) and (55) show that

$$\lim_{r_L^{(i)} \rightarrow \infty} \mathbf{r}_2^{(i)}(E) = \mathbf{r}_3^{(i)}(E) . \quad (70)$$

Therefore, if

$$\lim_{r_L^{(i)} \rightarrow \infty} \mathbf{r}_2^{(i)co}(E) = \mathbf{0}^{co} , \quad (71)$$

then Eq. (69) shows that the calculation of $\mathbf{R}_4^{(i)oo}(E)$ is independent of information about the $P - P^o$ closed channels. As a result, if all of the elements of $\mathbf{r}_2^{(i)}(E)$ which involve a particular closed channel are sufficiently small, that channel can be dropped from propagation and the $\mathbf{R}_4^{(i+1)oo}(E)$, and therefore the reactance matrix will still be accurately calculated.

The way the program implements the consequences of the above argument is as follows. If $r^{(i)}$ is less than some input value read in by the program, or if $P^{(i)}$ has already been reduced to P^o , $P^{(i+1)}$ is set to $P^{(i)}$. Otherwise the program checks to see whether $|(r^{(i)})_{m,m}| \leq \text{EPSRED}$ and $|(r^{(i)})_{m,m}| \leq \text{EPSRED}$, where EPSRED is a variable set by the user (a typical value would be 1.0×10^{-3}). If both inequalities are satisfied, then $P^{(i+1)}$ is set to $P^{(i)} - 1$; otherwise, $P^{(i+1)} = P^{(i)}$.

It should however be stressed that the current version of the program will seek to drop channels from the propagation as described above *only* in the case that just $\mathbf{R}^{(i)}(E)$ is propagated, i.e., only for the homogeneous option. In the case that the full matrix $\mathbf{R}^{(i)}(E)$ is propagated, the inhomogeneous option, then the number of channels propagated will remain the same in all sectors. However,

any channels which are strongly closed asymptotically should be neglected when performing the asymptotic analysis in order to extract the reactance matrix. As will be discussed in more detail below, the program will check for strongly closed channels before performing the asymptotic analysis and neglect them, regardless of whether the full $\mathbf{R}^{(i)}(E)$ matrix or just the $\mathbf{R}_i^{(i)}(E)$ submatrix is propagated, and regardless of the number of channels $P^{(i)}$ propagated within any given sector i .

3.7. Asymptotic analysis and reordering of channels

Before applying the boundary conditions of Eqs. (19) or (20) to $\mathbf{g}^{(i)}$ it is sometimes convenient to reorder the channels in $\mathbf{g}^{(i)}$, and sometimes it becomes necessary to make linear combinations of the channels in $\mathbf{g}^{(i)}$. It is convenient to reorder the channels if $V_{mn}(r)$ falls off faster than r^{-2} . This occurs because for large r it is approximately true that

$$D_{nm}(r, E) = \delta_{nm} \left[\frac{\ell_n(\ell_n + 1)}{r^2} - \bar{k}_n^2(E) \right], \quad (72)$$

and, depending on r and r' , it may be possible that $D_{nn}(r, E) > D_{mm}(r, E)$ and $D_{nn}(r', E) < D_{mm}(r', E)$. Since the subprogram which calculates $T(i)$ and $[\lambda^{(i)}]^2$ orders the eigenvalues from lowest to highest, the relative positions of channels n and m may change in \mathbf{g} when going from r to r' .

It is necessary to make new linear combinations of the channels if there exist degenerate channels, i.e., channels with $[\lambda_n^{(i)}]^2 = [\lambda_m^{(i)}]^2$ and $n \neq m$. This is because the matrix diagonalization routine will mix those states. Degenerate

channels can occur in the calculations at very large r where the term $\frac{\ell(\ell+1)}{(2\mu r^2)}$ is negligible since there may be channels with the same k_n but different values of ℓ_n . (Additional accidental degeneracies would occur if one used the harmonic oscillator and rigid-rotor approximations to calculate asymptotic energies, but the use of these is not recommended.) In order to sort out these effects, we make the transformation to new radial functions $\mathbf{h}^{(i)}$ defined by

$$g_{mn}^{(i)}(r, E) = \sum_{\ell=1}^N \sum_{k=1}^{P^{(i)}} T_{\ell m}^{(i)} U_{\ell k}^o h_{kn}^{(i)}(r, E), \quad 1 \leq m \leq P^{(i)}, \quad 1 \leq n \leq P^{(i)} \quad (73)$$

where U^o diagonalizes H_{im} in the X_n basis, and has the channels in some fixed order that does not mix degenerate channels. This equation can be written in matrix notation as

$$\mathbf{g}^{(i)}(r, E) = [\mathbf{T}^{(i)}]^T \mathbf{U}^o \mathbf{h}^{(i)}(r, E) . \quad (74)$$

We then define a new global R matrix satisfying

$$\begin{pmatrix} \tilde{\mathbf{h}}_L^{(i)} \\ \tilde{\mathbf{h}}_R^{(i)} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}}_1^{(i)} & \tilde{\mathbf{R}}_2^{(i)} \\ \tilde{\mathbf{R}}_3^{(i)} & \tilde{\mathbf{R}}_4^{(i)} \end{pmatrix} \begin{pmatrix} \mathbf{h}_L^{(i)} \\ -\mathbf{h}_R^{(i)} \end{pmatrix} \quad (75)$$

where

$$\tilde{\mathbf{R}}_1^{(i)} = (\mathbf{T}^{(i)T} \mathbf{U}^o)^{-1} \mathbf{R}_1^{(i)} [\mathbf{T}^{(i)}]^T \mathbf{U}^o , \quad (76)$$

$$\tilde{\mathbf{R}}_2^{(i)} = (\mathbf{T}^{(i)T} \mathbf{U}^o)^{-1} \mathbf{R}_2^{(i)} [\mathbf{T}^{(i)}]^T \mathbf{U}^o , \quad (77)$$

$$\tilde{\mathbf{R}}_3^{(i)} = (\mathbf{T}^{(i)T} \mathbf{U}^o)^{-1} \mathbf{R}_3^{(i)} [\mathbf{T}^{(i)}]^T \mathbf{U}^o , \quad (78)$$

$$\tilde{\mathbf{R}}_4^{(i)} = (\mathbf{T}^{(i)T} \mathbf{U}^o)^{-1} \mathbf{R}_4^{(i)} [\mathbf{T}^{(i)}]^T \mathbf{U}^o . \quad (79)$$

The matrix \mathbf{a} whose elements appear in Eq. (19) is determined by

$$\mathbf{a}(E) = \lim_{i \rightarrow \infty} \mathbf{a}^{(i)}(E) \quad (80)$$

where

$$\mathbf{a}^{(i)}(E) = [-\mathbf{F}(E) + \mathbf{M}^{(i)}(E) \mathbf{H}^{(i)}(E)]^{-1} [\mathbf{B}^{(i)}(E) + \mathbf{M}^{(i)}(E) \mathbf{G}^{(i)}(E)] \Delta . \quad (81)$$

$$F_{mn}^{(i)}(E) = \delta_{mn} \begin{cases} \cos \left[|k_m(E)| r_R^{(i)} - \frac{\ell_m \pi}{2} \right] , & 1 \leq m \leq P^o , \\ -n_{r_m} \left[|k_m(E)| r_R^{(i)} \right] , & 1 \leq m \leq P^o , \\ \exp \left[-|k_m(E)| r_R^{(i)} \right] , & P^o < m \leq P^{(i)} , \end{cases} \quad (82)$$

$$B_{mn}^{(i)}(E) = \delta_{mn} \begin{cases} \sin \left[|k_m(E)| r_R^{(i)} - \frac{\ell_m \pi}{2} \right], & 1 \leq m \leq P^o, \\ j_{\ell_m} \left[|k_m(E)| r_R^{(i)} \right], & 1 \leq m \leq P^o, \\ \exp \left[|k_m(E)| r_R^{(i)} \right], & P^o < m \leq P^{(i)}, \end{cases} \quad (83)$$

$$H_{mn}^{(i)}(E) = |k_m| \delta_{mn} \begin{cases} \sin \left[|k_m(E)| r_R^{(i)} - \frac{\ell_m \pi}{2} \right], & 1 \leq m \leq P^o, \\ n_{\ell_m} \left[|k_m(E)| r_R^{(i)} \right], & 1 \leq m \leq P^o, \\ \exp \left[-|k_m(E)| r_R^{(i)} \right], & P^o < m \leq P^{(i)}, \end{cases} \quad (84)$$

$$G_{mn}^{(i)}(E) = |k_m(E)| \delta_{mn} \begin{cases} \cos \left[|k_m(E)| r_R^{(i)} - \frac{\ell_m \pi}{2} \right], & 1 \leq m \leq P^o, \\ j_{\ell_m} \left[|k_m(E)| r_R^{(i)} \right], & 1 \leq m \leq P^o, \\ \exp \left[|k_m(E)| r_R^{(i)} \right], & P^o < m \leq P^{(i)}, \end{cases} \quad (85)$$

$$\Delta_{nm} = \delta_{nm} \begin{cases} 1, & 1 \leq m \leq P^o \\ b, & P^o \leq m \leq P^{(i)}, \end{cases} \quad (86a)$$

$$\mathbf{M}^{(i)}(E) = \bar{\mathbf{R}}_4^{(i)}(E) + \bar{\mathbf{R}}_3^{(i)}(E) \mathbf{x}^{(1)}(E) [1 - \bar{\mathbf{R}}_1^{(i)}(E) \mathbf{x}^{(1)}(E)]^{-1} \bar{\mathbf{R}}_2^{(i)}(E), \quad (86b)$$

and

$$x_{mn}^{(1)}(E) = \begin{cases} |k_m(r_L^{(1)}, E)| \cos [|k_m(r_L^{(1)}, E)| r_L^{(1)}] / \sin [|k_m(r_L^{(1)}, E)| r_L^{(1)}], & m \text{ open} \\ |k_m(r_L^{(1)}, E)|, & m \text{ closed.} \end{cases} \quad (86c)$$

If only $\mathbf{R}^{\psi}(E)$ is propagated, then $\mathbf{M}^{(i)}(E) = \mathbf{R}^{\psi}(E)$.

The program will drop strongly closed channels from the asymptotic analysis according to the following prescription. We first discuss the homogeneous case, where only $\mathbf{R}^{\psi}(E)$ is propagated, and then the inhomogeneous case, where the full $\mathbf{R}^{\omega}(E)$ is propagated.

For the homogeneous case, the program considers the off-diagonal matrix elements of the $\mathbf{R}^{\psi}(E)$. If it is true that for a given $m > P^o$, $(\mathbf{R}^{\psi})_{mm} = (\mathbf{R}^{\psi})_{mm} = 0$ for

all $n \leq P^o$, channel m is not required in the calculation of a_{nm}^{ω} , $1 \leq n, m \leq P^o$. This program determines the smallest $n \geq P^o$, called P_q^{ω} , such that $|(R^{\omega})_{nm}| \leq \text{EPSDR}$ and $|(R^{\omega})_{nm}| \leq \text{EPSDR}$ for all $n \leq P^o$, where EPSDR is some small number (a typical value is 1.0×10^{-3}), and then uses the upper left $P_q^{\omega} \times P_q^{\omega}$ subblock of $R^{\omega}(E)$ to calculate \mathbf{a}^{ω} by Eq. (81).

For the inhomogeneous case, the full matrix $M^{\omega}(E)$, is used to calculate \mathbf{a}^{ω} , so that the test is performed on $M^{\omega}(E)$. It should be noted that the program tests the $M^{\omega}(E)$ matrix itself and not the intermediate matrices used to construct it in Eq. (86b). As in the homogeneous case, the program considers the off diagonal elements of the matrix $M^{\omega}(E)$. If it is true that for $n > P^o$, $M_{nm}^{\omega} = M_{nm}^{\omega} = 0$ for all $n \leq P^o$, channel m is not required in the calculation of a_{nm}^{ω} , $1 \leq n, m \leq P^o$. The program determines the smallest $n \geq P^o$, called P_q^{ω} , such that $M_{nm}^{\omega} \leq \text{EPSDR}$ and $M_{nm}^{\omega} \leq \text{EPSDR}$ for all $n \leq P^o$, where EPSDR is some small number (a typical value is 1.0×10^{-3}), and then uses the upper left $P_q^{\omega} \times P_q^{\omega}$ subblock of $M^{\omega}(E)$ to calculate \mathbf{a}^{ω} by Eq. (81).

3.8. Single/multiple energy runs

Inspection of Eq. (13) shows that the total energy E appears only as a multiple of the unit matrix, so that the matrices T^{ω} are independent of the total energy, and the eigenvalues $\lambda_{nn}^{\omega}(E)$ at a new energy E_{new} are easily related to those at the old energy E_{old} :

$$[\lambda_{nn}^{(i)^2}(E_{\text{new}})] = [\lambda_{nn}^{(i)^2}(E_{\text{old}})] + \frac{2\mu(E_{\text{old}} - E_{\text{new}})}{\hbar^2} \quad (87)$$

This fact may be used to save computer time for multiple-energy runs by reusing the λ^{ω} and $T(i-1, i)$. Saving these variables makes the calculation of the $V_{mm}(r)$, T^{ω} , and $T(i-1, i)$ and depending on the value of NPROP, $T(i-1, i)^{-1}$ unnecessary for second and subsequent energies and affords significant reductions in computer time. The possibility of these savings is one of the many attractive characteristics of the present algorithm. The drawback, however, is a corresponding increase in storage requirements to save the temporary values in these arrays. There are two ways in which the present program implements the second-energy calculations. In the first, which is associated with the logical variable LTYPE2 = .FALSE. in the program, the calculations for a given energy are taken to completion before the calculation for the second energy begins, and to do this as efficiently as possible requires the storage of the $P^{\omega} \times P^{\omega}$ matrices $T(i-1, i)$ and $[T(i-1, i)]^{-1}$ for each sector over which the solutions are propagated, and storage of information about the

original total energy, angular momentum, and basis set. Since the total number of sectors can number in the hundreds, this method requires a great deal of storage when $P^{(i)}$ is large; we therefore do not recommend the first method for routine usage. Because the propagation at the first energy must be taken to completion before the propagation at the second energy may be begun, the second energy runs performed in this fashion are referred to as "later date" second-energy runs. These runs are governed by the value of the input variable IROWS read in from FORTRAN unit 5. As it is generally more efficient to do LTYPE2 = .TRUE. (simultaneous) second-energy runs, only a brief description of the later-date second-energy-run option will be given. In order to prepare for a later-date second-energy run one must set IROWS = 1. Information needed for a restart run will be written to FORTRAN units 8 and 14. In order to do a later-date run using this information, one sets the variable IROWS = -1 and the variable NE (also found in FORTRAN UNIT 5) = 2 (later-date second energy runs can only do one second energy). The energies are read in from FORTRAN unit 5; if IROWS = -1 the first energy read in will be ignored.

The second method, which is associated with the logical variable LTYPE2 = .TRUE. in the program, is to propagate all energies together, that is, the global R matrix for sector (i) is calculated for all of the energies before the global R matrix for sector (i + 1) is calculated for any of the energies. If there are fewer energies than sectors (which is almost always the case) this will decrease the storage requirements, since in this case the $P^{(i)} \times P^{(i)}$ matrices $T(i-1, i)$ and $[T(i-1, i)]^{-1}$ must only be stored for the sector currently being propagated. The second method is also the one used to enable simultaneous propagation of solutions for different size basis sets. In order to do this type of multiple energy run, all of the energy values for which computations are to be performed must be known in advance. The input parameter NE in FORTRAN unit 5 should be set to the negative of the number of different energies at which solutions are to be propagated, and the input array E(NE) should contain these energy values. For example, if it were desired to run three energies simultaneously, then NE should be set to -3, and there should be three different energies supplied. It should also be noted that the value of E required in the input file is the *total* energy, which includes the translational energy of the initial state and the internal energies, including zero point energy, of the collision partners in that initial state. Further details on values of input parameters required by the two types of multiple energy runs are given in the discussion of multiple energy runs and INPUT/OUTPUT in the on-line manual.

4. Program Structure

This section of the chapter gives an overview of the program structure and describes program flow in a typical run. In addition, it explains some of the options allowing "multiple runs" during a single execution.

4.1. Segmentation of program and flow chart

Roughly speaking, the calculations may be separated into six components. These are listed in Table I below. It should be noted that components 5 (subroutine GNSCAT) and 6 ("everything else") take up the least time during a typical run: subroutine GNSCAT is called at most several times during a run, and the sundries in component 6 contain few large matrix operations. The two most computationally costly tasks are usually component 1, the calculation of the potential function matrix elements, and component 2, the assembly and diagonalization of the D matrix. Component 1 involves the computation of approximately $\frac{1}{2}N^2$ multi-dimensional integrals, and component 2 contains matrix operations which become proportional to N^3 as N grows large. Components 3 and 4 also contain some N^3 matrix operations, but are still not as expensive as 2.

Due to the computational expense of performing runs with large N, it is advisable to take advantage of the feature of the R matrix propagation method which allows one to perform runs at a second energy by re-using some of the matrices from the first energy. The justification for this is discussed in Section 3.8, where we describe how we do this in the program.

Figure 1 is a flow chart of the program. The roman numerals I-V in the flow chart correspond to the components in the Table. The multiple-potential loop in the flow chart encompasses the large loop from component I to the final check for more sectors, but was excluded for clarity.

4.2. Restart options

The program contains a restart option: depending on the values of the input variables described below, the program can write restart information to disk after every ISAVE sectors, where ISAVE is a variable read in by the program, and in a subsequent execution read in this restart information to continue the propagation from the point at which the restart file was written.

The restart option serves a twofold purpose. It is useful in case of a system crash or shutdown in the middle of a run, because if the restart file is intact, the run can be continued from the point of the last restart file and so less

Table I. Components of the Program.

Component Number	Subroutine name(s)	Purpose
I	POT	Calculation $V_{nm}^{(i)}$ at $r_i^{(i)}$
II	RMPROP, RS	Assembly and diagonalization of $D(i)$ to obtain $\lambda_{nm}^{(i)}$ and $T^{(i)}$
III	TAUMTS	Calculation of $T(i-1, i)$ and/or $[T(i-1, i)]^{-1}$
IV	RPROP	Calculation of $R^{(i)}$ (inhomogeneous option) or R^p (homogeneous option) and $P^{(i+1)}$
V	GNSCAT	Calculation of S from $R^{(i)}$ or R^p
VI	...	Everything else

computer time will have been lost. It is also handy in case of batch queue limits for CPU time, in which case a run may be performed in blocks of ISAVE sectors (or a multiple thereof).

In order to use the restart option, one must modify the input variables IREST and ISAVE in FORTRAN unit 5. One should initially perform a run with IREST = 0 and with ISAVE = 10 (for example). IREST tells the program that this run is NOT itself a restart run, and so it should not look for the restart files. ISAVE = 10 means that at sectors 10, 20, 30, etc., restart information will be written to disk alternating between FORTRAN units 17 and 18. The reason that restart information is written to *two* disk files is so that there will always be one set of "good" restart information available: if a system crash occurs while restart information is being written to one disk file then the program may still be restarted (once the system is back up) from the other disk file. It should also be noted that for runs with large numbers of channels that these files may be quite large; the user should beware of excessive I/O charges or of filling up a disk with these files.

In order to restart a run from an existing disk restart file, there are two things to be done. First, the value of IREST in the FORTRAN unit 5 for the run to be restarted should be set to 17 or to 18. A nonzero value of IREST on input tells the program that it should look for restart information, and the value of 17 or 18 tells the program which FORTRAN unit the restart information should be read from (if IREST is nonzero and yet not 17 or 18, the program

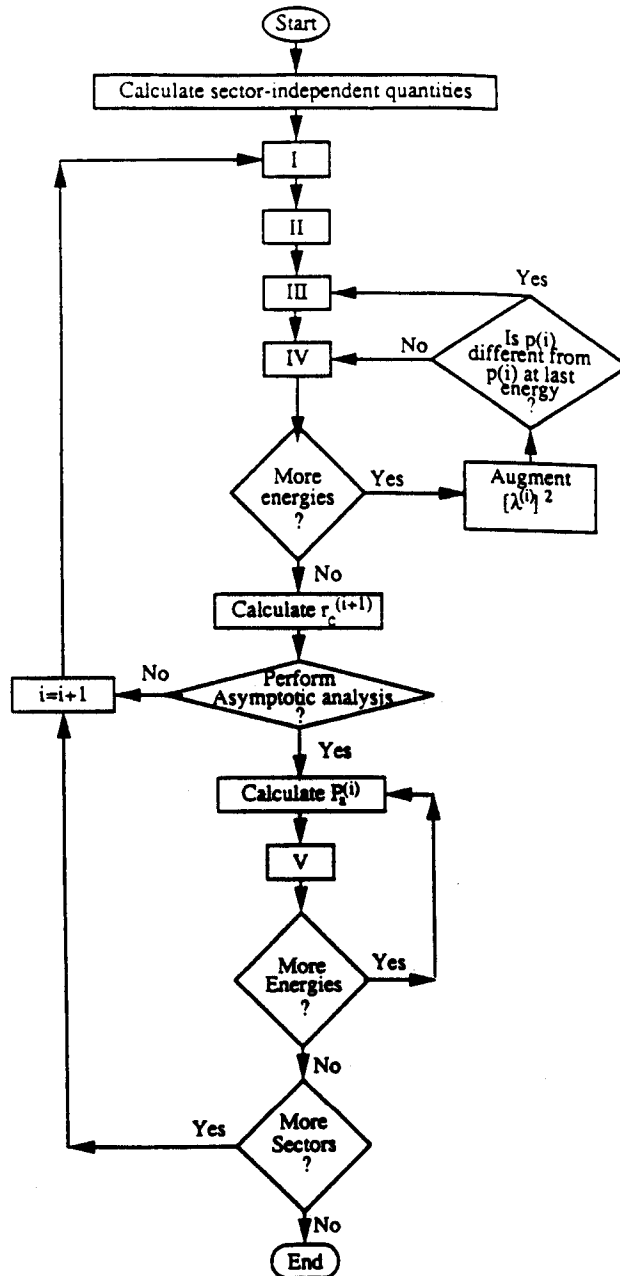


Figure 1. Flow Chart of the RMPROP computer program.

will seek for restart information of FORTRAN unit 17 by default). Secondly, one should be sure that the restart run is actually a continuation of the one which wrote the disk file. If the current job is smaller (less channels, or smaller basis set) than the original, then the final results will be meaningless; if the current job is larger, the program will terminate with an END-OF-RECORD error message. Also, if a run is begun with inhomogeneous boundary conditions (i.e., the full $\mathbf{R}^{(i)}(E)$ is propagated), then the output to FORTRAN units 17 and 18 will reflect this. The homogeneous/inhomogeneous option cannot be changed during a run.

5. Vectorization

RMPROP is a very efficient program on a vector pipeline computer. Most of the work is in vectorizable loops. Further discussion of this point is found in previous papers.^{14,25,26} In particular, extensive use is made of highly efficient FORTRAN program libraries, including LAPACK for general architectures and SCILIB for use on Cray systems. For further information, see section 9 of this chapter and sections 6 and 16 of the on-line manual.

Because of this high efficiency, RMPROP can be used to solve very large problems. For example, we have reported calculations with up to 1358 channels²⁷ for a diatom-diatom scattering problem with long-range dipole-dipole coupling. In unpublished work²⁸ we have completed calculations on the same problem with up to 2472 channels.

6. Matrix Utilities

In version 1.0 of RMPROP we used the FORTRAN subroutine RS from the EISPACK subroutine library to handle the diagonalization step in Eq. (88), i.e.

$$\sum_{k,k'=1}^N U_{kn}^{(i)} D_{kk'}(r_c^{(i)}, E) U_{k'm}^{(i)} = \delta_{nm} [\lambda_{nm}^{(i)}(E)]^2. \quad (88)$$

This subroutine was included with the program, or was available on Cray computer systems, by accessing the SCILIB scientific subroutine library from Cray Research, Inc. In the current version of the program we use instead an adaptation of the FORTRAN routine EVVRSP and its subsidiary routines supplied to us by Stephen Elbert from Iowa State University, which are distributed as part of version 2.0 of RMPROP with his permission.

Another change made to the matrix utilities in version 2.0 of RMPROP is to the libraries used for matrix-matrix multiplication, and for the solution of linear equations of the form $Ax = B$, where A and B are full matrices and x is a vector. Version 1.0 of RMPROP used the FORTRAN routine MXMA adapted from Cray Research's scientific subroutine library (SCILIB) for matrix-matrix multiplication, and the FORTRAN routine LUSOLV for linear equation solutions and inversion of a matrix. In the current version of RMPROP we use the FORTRAN subroutines from the LAPACK subroutine library²⁹ to perform these tasks. We use DGEMM for matrix-matrix multiplication, and we use DGETRF with DGETRS for the linear equation solution and DGETRI for the matrix inversion.

The use of the LAPACK subroutine library increases both the portability of the code and its execution rate. This is true for two reasons. First, source codes for the LAPACK subroutine library are available by anonymous electronic mail from netlib@ornl.gov, which means that standard FORTRAN code is available. Second, much of this library consists of the Basic Linear Algebra Subroutines (BLAS), which are available in machine-optimized library modules on both Cray-2 and IBM RS/6000 architectures, as well as many other computers. Use of the library versions of the routines wherever possible further improves the performance of the program (for more information, see Section 6 and 16 of the on-line manual).

7. Errata

The following is a list of known bugs in version 1.0 of RMPROP, followed by a list of known typographical errors in the book chapter, "RMPROP: A Computer Program for Quantum Mechanical Close Coupling Calculations for Inelastic Collisions", published in *Modern Techniques in Computational Chemistry: MOTECC-91* edited by Enrico Clementi.

7.1. Known bugs in version 1.0

For LPR (4) = .TRUE., the legend written to FORTRAN unit 6 states that the accompanying matrix contains the "ELEMENTS OF PACKED INTERACTION MATRIX". This is inaccurate. It should print instead "ELEMENTS OF UNPACKED INTERACTION MATRIX". Note that this bug has been corrected in version 2.0 of RMPROP, the on-line manual now specifies whether the interaction matrix is in unpacked form.

At the end of a run, the FORTRAN source code contains an instruction to close FORTRAN unit 6. This resulted in an error message on the IBM 3090. This statement has been deleted in version 2.0 of RMPROP.

In the subroutine stpot.f, the index for the DO LOOP labeled 6 in subroutine PREPOT was mistyped as NVIB instead of NVIBX. This routine is used in the test runs hfhfstr1, hfhfstr2, hfhfstr3, and hfhfstr4. This error will affect the channel reordering according to energy in cases where the two channels whose indices are to be exchanged have different values for v_1 and v_2 , but did not affect any of the results in the test suite. This bug has been corrected in version 2.0 of RMPROP.

The eigenvectors written out with the eigenphases generated by LGS (15) were computed incorrectly. This bug has been corrected in version 2.0 of RMPROP.

The legend for the execution time for the eigensystem analysis generated by LPR (1) has been changed to reflect the new matrix utilities present in version 2.0 of RMPROP.

The call to SUBROUTINE HEADER has been removed from PROGRAM RMPROP, so that it is only called by SUBROUTINE PREPOT for each test run. As a result, the information printed out by SUBROUTINE HEADER will only appear once in FORTRAN unit 15 (the same information had been printed twice).

If the input logical variable LJRI is true, and the variables EPSMAG and EPSPH are both nonzero, a message concerning testing convergence of scattering matrix elements for convergence with respect to RMAX is printed out. This message is irrelevant, because if LJRI is true, no propagation is performed. Printing of the message has been suppressed in this case in version 2.0 of RMPROP.

7.2. Known typographical errors in chapter describing version 1.0

In Eq. (61) in the book chapter for version 1.0 of RMPROP, the last term in the first line of the equation reads $r^{j\infty}$; it should read $r^{j\infty}$. This error has been corrected in the present chapter, where the same information is contained in Eq. (63).

In the discussion of second-energy runs, the book chapter for version 1.0 of RMPROP states that in order to do a later-date second energy run, the input variable NE for that later-date second energy run should be -2. This is incorrect,

and inconsistent with the input data for un hfhadr2 in the test suite which illustrates a later-date second energy run. The input value of NE should be 2.

The discussion of variable stepsize determination was over-simplified in the book chapter for version 1.0 of RMPROP. The discussion in the current chapter is complete.

The relation of eigenvalues of the interaction matrix at a new energy to those at an old energy has been clarified and corrected. This was Eq. (88) of section 3.8 in the book chapter for version 1.0 of RMPROP and is Eq. (87) of the current chapter.

The book chapter for version 1.0 of RMPROP did not mention that the scattering matrix is complex.

The discussion of channel reordering in section 2.1 of the book chapter for version 1.0 of RMPROP read "The channels are ordered so that $k_n^2 > k_m^2$ if and only if $n > m$ ". This has been changed to read "The channels are ordered so that $k_n^2 \geq k_m^2$ if and only if $n \geq m$." in the current chapter.

In the book chapter for version 1.0 of RMPROP, Eqs. (77)-(80) governing the transformation of the $R^{(j)}$ were incorrect. These have been corrected in the current chapter, and are now Eqs. (76)-(79).

In the book chapter for version 1.0 of RMPROP, the program flow chart contained a typographical error. Step III of the program was labeled as executing twice in succession. This is incorrect. This error has been corrected in the present chapter.

8. Concluding Remarks

RMPROP is a very general and efficient R matrix propagation code for solution of the close coupling equations for molecular collisions. Over the years, various earlier versions have been run successfully on a wide variety of computers. For the present version (version 2.0) we have successfully run a test suite on an IBM RS/6000 Model 550 computer under the AIX operating system Version 3.2, and on a Cray-2 computer under UNICOS, version 6.1, and this test suite is distributed with the code.

9. Acknowledgments

This work was supported in part by the National Science Foundation, Minnesota Supercomputer Institute, and Cray Research University Grant Program. We are grateful to Professor Stephen Elbert of Iowa State University for supplying us with the EVVRSP eigensystem analysis routines and for giving us permission to distribute these routines with version 2.0 of RMPROP.

Appendix 1A.

In this section we consider the expansion of a general interaction potential involving two collision partners in a set of angular basis functions, and we consider the subsequent evaluation of matrix elements of the form of the $V_{nm}(r)$ of Eq. (11) given such an expansion. This expansion greatly simplifies the evaluation because it makes the angular contribution to the $V_{nm}(r)$ an analytical expression. Furthermore, for many systems it is convenient to evaluate these angular terms once prior to the beginning of the propagation and to use them without recalculation at each sector. We consider as examples both the system of an atom and a rigid diatomic molecule, for which the potential is expanded in Legendre polynomials, and the system of two identical vibrating, rotating heteronuclear molecules, for which the interaction potential is expanded in Launay³³ body-frame functions. We consider the atom-diatom case first, and then the diatom-diatom system. It should be noted that programs utilizing these formulas are included in the test suite distributed with the program. For more details on the program themselves, see the on-line manual distributed with the program.

A.1. Atom-diatom scattering

For the collision of an atom (assumed structureless) with a rigid diatomic molecule, the system Hamiltonian is

$$H(\hat{R}, \mathbf{r}) = T(\mathbf{r}) + H_{rot}(\hat{R}) + V(\hat{R}, \mathbf{r}) \quad (A1)$$

where \hat{R} denotes the orientation of the diatom in relation to space-fixed axes, \mathbf{r} is the vector from the incident atom to the center of mass of the diatom, $V(\hat{R}, \mathbf{r})$ is the interaction potential, and $T(\mathbf{r})$ is the kinetic energy operator for relative motion in three dimensions:

$$T(\mathbf{r}) = -\frac{\hbar^2}{2\mu} \nabla_r^2 \quad (A2)$$

where μ is the reduced mass for relative translational motion of the collision partners, and where ∇_r^2 is the Laplacian with respect to \mathbf{r} . As in Eqs. (3) and (4) above we separate ∇_r^2 into its radial and angular terms

$$T(r) = -\frac{\hbar^2}{2\mu} \left[\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) \right] + \frac{\ell_r^2}{2\mu r^2} \quad (A3)$$

where the orbital angular momentum operator ℓ_r^2 contains all the angular dependence of $T(r)$.

The eigenvalues of the operator ℓ_r^2 are the spherical harmonics $Y_{\ell m}(\hat{r})$ where the ℓ is the orbital angular momentum quantum number, and the projection of the orbital angular momentum onto the z axis is $m\hbar$. The eigenvalue equation relating the ℓ_r^2 operator and the $Y_{\ell m}(\hat{r})$ is

$$\ell_r^2 Y_{\ell m} = \ell(\ell + 1)\hbar^2 Y_{\ell m} \quad (A4)$$

We define $\hat{H}_{int} = H_{rot}$, where

$$H_{rot} = \hat{J}^2/2I, \quad (A5)$$

where \hat{J}^2 is the quantum mechanical operator for the square of the diatomic rotational angular momentum, I is the diatomic molecule's moment of inertia, and where there is no potential contribution to the asymptotic Hamiltonian for a rigid rotor. The eigenfunctions of the rotational Hamiltonian (A5) are the spherical harmonics $Y_{jm}(\hat{R})$ which satisfy the Schrödinger equation

$$H_{rot}(\hat{R})Y_{jm}(\hat{R}) = \epsilon_j Y_{jm}(\hat{R}) \quad (A6)$$

with the eigenvalues

$$\epsilon_j = \frac{\hbar^2}{2I} j(j + 1) \quad (A7)$$

It is convenient to couple the orbital angular momentum eigenfunctions and the internal angular momentum eigenfunctions to give eigenfunctions of the total angular momentum, which is a vector sum of the orbital and rotational angular momenta

$$\vec{J} = \vec{j} + \vec{\ell} \quad (A8)$$

where in the above expression the arrow above a variable denotes that it is a vector quantity. The coupled eigenfunctions are

$$Y_{j\ell}^{JM}(\hat{R}, \hat{r}) = \sum_{m_j} \sum_{m_\ell} (j\ell m_j m_\ell | j\ell JM) Y_{j m_j} Y_{\ell m_\ell} \quad (\text{A9})$$

where $(j\ell m_j m_\ell | j\ell JM)$ is a Clebsch-Gordan coefficient.³⁰

Given the basis functions of (A9) our goal is to find the elements of the potential coupling matrix

$$V_{j\ell j'\ell'}^J(r) = \int d\hat{R} \int d\hat{r} (Y_{j\ell}^{JM}(\hat{R}, r))^* V(\hat{R}, r) Y_{j'\ell'}^{JM}(\hat{R}, \hat{r}) \quad (\text{A10})$$

which are formally independent of M because the space is isotropic.

Evaluation of the matrix elements may be performed analytically under the following conditions. Let the interaction potential $V(\hat{R}, r)$ be expanded² in Legendre polynomials $P_\lambda(\hat{R} \cdot \hat{r})$ where the radial expansion coefficients are denoted by $v_\lambda(r)$:

$$V(\hat{R}, r) = \sum_{\lambda} v_\lambda(r) P_\lambda(\hat{R} \cdot \hat{r}) \quad (\text{A11})$$

When the interaction potential is expanded in this way, the matrix elements of (A10) become

$$V_{j\ell j'\ell'}^J(r) = \sum_{\lambda=0}^{\lambda_{\max}} v_\lambda(r) g_\lambda(j\ell, j'\ell'; J) \quad (\text{A12})$$

where the $g_\lambda(j\ell, j'\ell'; J)$ are known as Percival-Seaton³¹ coefficients, for which analytical expressions are available. Note that even though the coupled angular momentum eigenfunctions of (A9) are complex, the elements of the potential coupling matrix (A10) are real, and they reduce to zero asymptotically because the $v_\lambda(r)$ do. The sum in (A12) terminates at $\lambda_{\max} = \min(j + j', \ell + \ell')$ because the $g_\lambda(j\ell, j'\ell'; J)$ are zero for $\lambda > j + j'$ or $\lambda > \ell + \ell'$. For homonuclear diatomics, since the potential is invariant with respect to interchanging the diatomic nuclei, only even values of λ need be used in (A11). Furthermore, even though there may exist several values of ℓ_n for a channel n with angular momentum j which satisfy the vector relation (A8), the only channels which couple to one another are those with identical parity $(-1)^{j+\ell}$. Therefore, for homonuclear diatomic species, Δj is always even, and so the ℓ values in the basis set will all be even at even J and will be odd at odd J .

A.2. Diatom-diatom scattering

For the collision of two identical vibrating, rotating heteronuclear molecules, we divide the Hamiltonian into a sum of heteronuclear diatomic terms and many-body terms, so that $H_{int} = H_{int}^{(1)} + H_{int}^{(2)}$ and

$$H_{int} = H_{int}^{(1)} + H_{int}^{(2)} \quad (A13)$$

$$H_{int}^{(i)} = -\frac{\hbar^2}{2\mu_m} \left[\frac{1}{R_i^2} \frac{\partial}{\partial R_i} \left(R_i^2 \frac{\partial}{\partial R_i} \right) \right] + \frac{j_{R_i}^2}{2\mu_m R_i^2} + V_{vib}(R_i), \quad i = 1, 2, \quad (A14)$$

where μ_m is the reduced mass of one of the molecules, R_i is the bond length of the i^{th} molecule, j_{R_i} is the quantum mechanical operator for the square of the rotational angular momentum of the i^{th} molecule, and V_{vib} is the vibrational potential. We have used a number of different functions for V_{vib} ; references for these potentials are included in the on-line manual. We choose for the basis functions $X_n(\mathbf{x}, \hat{\mathbf{r}})$ the following¹⁵:

$$X_n(\mathbf{x}, \hat{\mathbf{r}}) = [2(1 + \delta_{v_1 v_2} \delta_{j_1 j_2})]^{-1/2} [\Phi_\alpha(\mathbf{x}, \hat{\mathbf{r}}) + \eta(-1)^{j_1 + j_2 + j_{12} + \ell} \Phi_\alpha(\mathbf{x}, \hat{\mathbf{r}})], \quad (A15)$$

where

$$\Phi_\alpha(\mathbf{x}, \hat{\mathbf{r}}) = (R_1 R_2)^{-1} \chi_{v_1 j_1}(R_1) \chi_{v_2 j_2}(R_2) \Theta_{j_1 j_2 j_{12} \ell}^{JM}(\hat{R}_1, \hat{R}_2, \hat{\mathbf{r}}), \quad (A16)$$

where

$$\Theta_{j_1 j_2 j_{12} \ell}^{JM}(\hat{R}_1, \hat{R}_2, \hat{\mathbf{r}}) = \sum_{\substack{m_1 m_2 \\ m_{12} m_\ell}} (j_1 m_1 j_2 m_2 | j_{12} j_{12} m_{12}) (j_{12} m_{12} \ell m_\ell | j_{12} \ell JM) \\ \times Y_{j_1 m_1}(\hat{R}_1) Y_{j_2 m_2}(\hat{R}_2) Y_{\ell m_\ell}(\hat{\mathbf{r}}) \quad (A17)$$

and where χ_{v_j} is a vibrational eigenfunction of

$$\left[-\frac{\hbar^2}{2\mu_m} \frac{d^2}{dR^2} + \frac{j(j+1)\hbar^2}{2\mu_m R^2} + V_{vib}(R) \right] \chi_{v_j}(R) = \epsilon_{v_j} \chi_{v_j}(R). \quad (A18)$$

In the above equations, the terms of the form $(\dots \dots \dots | \dots \dots \dots)$ are Clebsch-Gordan coefficients, the $Y_{\ell m}$ are spherical harmonics, and \hat{R}_i is a unit vector giving the orientation of the i^{th} monomer in a laboratory-fixed reference frame.

The $X_n(\mathbf{x}, \hat{r})$ of Eq. (A15) is an expression for a wave function which is invariant with respect to exchange of the two molecules, which are assumed in this example to be indistinguishable from one another. Therefore, the subscript n is a collective index for the set of quantum numbers $\nu_1, j_1, \nu_2, j_2, j_{12}, \ell, J, M$, and η , where η is the eigenvalue for interchange of the two molecules, and can have the value $+$ or $-$. The right hand side of Eq. (A15) is the sum of two wave functions where the two molecules are formally distinguished from one another, plus a normalization factor, where the $\delta_{\nu_1\nu_2}$ and $\delta_{j_1j_2}$ allow for the case where the quantum numbers of the two monomers are identical. The subscript α on the right hand side represents the quantum numbers $\nu_1, j_1, \nu_2, j_2, j_{12}, \ell, J$, and M , where molecule 1 is known to have vibrational and rotational quantum numbers ν_1 and j_1 , and molecule 2 is known to have vibrational and rotational quantum numbers ν_2 and j_2 , and $\bar{\alpha}$ is the same as α except that the quantum numbers of the two molecules are exchanged.

For the applications in the test suite, involving two HF molecules, the $\chi_{\nu j}$ are obtained by solving Eq. (A4) by the linear variational method using a basis of harmonic oscillator functions. The monomer eigenenergies $\epsilon_{\nu j}$ may be obtained by this method as well but for the test cases they are derived from experimental spectroscopic parameters.³² The values of scattering matrix elements must be converged with respect to the number of harmonic oscillator functions. Also, Eq. (A13) implies that

$$\epsilon_n = \bar{\epsilon}_n = \epsilon_{\nu_1 j_1} + \epsilon_{\nu_2 j_2} \quad (\text{A19})$$

In order to evaluate the matrix elements of Eq. (11) it is first necessary to express the V_{nn} , where n now means the same as in Eq. (A15), in terms of matrix elements between the Φ_α :

$$\begin{aligned} V_{nn}(r) &= [4(1 + \delta_{\nu_1\nu_2} \delta_{j_1j_2})(1 + \delta_{\nu_1\nu_2} \delta_{j_1j_2})]^{-1/2} \\ &\times [V_{\alpha\alpha}(r) + \eta(-1)^{j_1+j_2+j_{12}+\ell} V_{\bar{\alpha}\alpha}(r) \\ &+ \eta'(-1)^{j_1+j_2+j_{12}+\ell} V_{\alpha\bar{\alpha}}(r) + \eta\eta'(-1)^{j_{12}+j_{12}} V_{\bar{\alpha}\bar{\alpha}}(r)] \end{aligned} \quad (\text{A20})$$

where the first term on the right hand side is a normalization factor (note that δ_{j_1} and δ_{v_1} for identical quantum numbers must be included for both the initial and final states), where

$$V_{\alpha\alpha'}(r) = \int d\mathbf{x} \int d\hat{r} \Phi_{\alpha}^*(\mathbf{x}, \hat{r}) V(\mathbf{x}, r) \Phi_{\alpha'}(\mathbf{x}, \hat{r}) \quad (\text{A21})$$

and where the last term only includes $(-1)^{j_1+j_2}$ because $V_{nn'}$ is diagonal in $j_1 + j_2 + \ell$. Since the potential energy must be independent of how the individual monomers are labeled, $V_{nn'}$ is also diagonal in η . Therefore we may assign $\eta = \eta'$ in Eq. (A20), and we find that

$$V_{\alpha\alpha'}(r) + \eta(-1)^{j_1+j_2+j_{12}+\ell} V_{\bar{\alpha}\alpha'}(r) = \eta(-1)^{j_1+j_2+j_{12}+\ell} V_{\alpha\bar{\alpha}'}(r) + (-1)^{j_{12}+j_{12}+\ell} V_{\alpha\alpha'}(r) \quad (\text{A22})$$

and so

$$V_{nn'}(r) = \delta_{nn'} [(1 + \delta_{v_1 v_2} \delta_{j_1 j_2}) (1 + \delta_{v_1' v_2'} \delta_{j_1' j_2'})]^{-1/2} \times [V_{\alpha\alpha'}(r) + \eta(-1)^{j_1+j_2+j_{12}+\ell} V_{\bar{\alpha}\alpha'}(r)] \quad (\text{A23})$$

using the factor of 2 introduced in Eq. (A22) to cancel the 4 in the normalization term.

The determination of $V_{\alpha\alpha'}$ in Eq. (A23) requires the evaluation of the eight dimensional integral in Eq. (A21). In order to do this we expand the potential $V(\mathbf{x}, r)$ in Launay *body-frame* functions^{30,33,34}:

$$V(\mathbf{x}, r) = \sum_{q_1 q_2 \mu} v_{q_1 q_2 \mu}(R_1, R_2, r) Y_{q_1 q_2 \mu}(\hat{r}_1, \hat{r}_2) \quad (\text{A24})$$

where

$$Y_{q_1 q_2 \mu}(\hat{r}_1, \hat{r}_2) = \frac{4\pi}{[2(1 + \delta_{\mu 0})]^{1/2}} [Y_{q_1 \mu}(\hat{r}_1) Y_{q_2 -\mu}(\hat{r}_2) + Y_{q_1 -\mu}(\hat{r}_1) Y_{q_2 \mu}(\hat{r}_2)] \quad (\text{A25})$$

and where \hat{r}_i is in the same direction as \hat{R}_i , but is given in the body-fixed reference frame where the z axis is in the direction of \hat{r} . Substituting the above into Eq. (A21) yields

$$V_{\alpha\alpha'}(r) = \sum_{q_1 q_2 \mu} B_{\beta\beta'}^{q_1 q_2 \mu} C_{\gamma\gamma'}^{q_1 q_2 \mu}(r) \quad (\text{A26})$$

where

$$B_{\beta\beta'}^{q_1 q_2 \mu} = \int d\hat{R}_1 \int d\hat{R}_2 \int d\hat{r} \theta_{j_1 j_2 j_1 2}^{JM*}(\hat{R}_1, \hat{R}_2, \hat{r}) Y_{q_1 q_2 \mu}(\hat{r}_1, \hat{r}_2) \theta_{j_1' j_2' j_1' 2}^{J'M'}(\hat{R}_1, \hat{R}_2, \hat{r}) \quad (\text{A27})$$

$$C_{\gamma\gamma'}^{q_1 q_2 \mu}(r) = \int dR_1 \int dR_2 \chi_{v_1 j_1}^*(R_1) \chi_{v_2 j_2}^*(R_2) v_{q_1 q_2 \mu}(R_1, R_2, r) \chi_{v_1' j_1'}(R_1) \chi_{v_2' j_2'}(R_2) \quad (\text{A28})$$

and where β and γ are composite indices. β stands for the quantum numbers j_1, j_2, j_{12}, J and M and γ stands for the quantum numbers j_1, j_2, v_1 and v_2 . We discuss Eq. (A27) first and then Eq. (A28).

Eq. (A27) which determines the $B_{\beta\beta'}$, is a six-dimensional integral which may be evaluated analytically and which is independent of sector (independent of r), so that the $B_{\beta\beta'}$ need only be evaluated once at the beginning of a run and then stored. We evaluate these terms in the following way. Define the simultaneous eigenfunctions $\theta_{j_1 j_2 j_1 2}^{JM, \Omega}$ of the operators with eigenvalues $j_1(j_1 + 1), j_2(j_2 + 1), j_{12}(j_{12} + 1), J(J + 1), M$ and Ω where Ω is the projection of J on the body fixed z axis:

$$\begin{aligned} \bar{\theta}_{j_1 j_2 j_1 2}^{JM, \Omega}(\hat{r}_1, \hat{r}_2, \hat{r}) &= \sum_{m_1 m_2} (j_1 m_1 j_2 m_2 | j_{12} j_{12} \Omega) Y_{j_1 m_1}(\hat{r}_1) \\ &\times Y_{j_2 m_2}(\hat{r}_2) \left[\frac{(2J + 1)}{4\pi} \right]^{1/2} D_{M\Omega}^{(J)}(\theta, \phi) \end{aligned} \quad (\text{A29})$$

where $D_{M\Omega}^{(J)}$ is a rotation matrix³⁰, and θ and ϕ are the inclination azimuthal angles of \hat{r} . The integral of an expansion element of the potential between the body-frame basis functions is similar to (A27):

$$\bar{B}_{\beta\beta'}^{q_1 q_2 \mu} = \int d\hat{r}_1 \int d\hat{r}_2 \bar{\theta}_{j_1 j_2 j_1 2}^{JM*}(\hat{r}_1, \hat{r}_2, \hat{r}) Y_{q_1 q_2 \mu}(\hat{r}_1, \hat{r}_2) \bar{\theta}_{j_1' j_2' j_1' 2}^{J'M'} \Omega'(\hat{r}_1, \hat{r}_2, \hat{r}) \quad (\text{A30})$$

and is given explicitly by³³

$$\begin{aligned}
\bar{B}_{\beta\alpha\beta'}^{q_1 q_2 \mu} &= \delta_{JJ'} \delta_{MM'} \delta_{\Omega\Omega'} [2(1 + \delta_{\mu 0})]^{-1/2} [(2j_1 + 1)(2j_2 + 1)(2j_{12} + 1)(2q_1 + 1) \\
&\quad (2q_2 + 1)(2j'_1 + 1)(2j'_2 + 1)(2j'_{12} + 1)]^{1/2} (-1)^{j'_1 + j'_{12} q_1 + \Omega} \\
&\quad \times \begin{pmatrix} j_1 & q_1 & j'_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} j_2 & q_2 & j'_2 \\ 0 & 0 & 0 \end{pmatrix} \sum_{q_{12}} (q_{12} + 1) \begin{pmatrix} j'_{12} & q_{12} & j_{12} \\ -\Omega & 0 & \Omega \end{pmatrix} \\
&\quad \times \begin{pmatrix} q_{12} & q_1 & q_2 \\ 0 & \mu & -\mu \end{pmatrix} \left\{ \begin{matrix} j_{12} & q_{12} & j_{12} \\ j_1 & q_1 & j_1 \\ j_2 & q_2 & j_2 \end{matrix} \right\} [1 + (-1)^{q_1 q_2 q_{12}}]
\end{aligned} \tag{A31}$$

where $\left\{ \begin{matrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{matrix} \right\}$ is a 9-j symbol.³⁰ The body-frame functions used above may be transformed into the laboratory-frame basis functions used to expand the system wave function by³³

$$\theta_{j_1 j_2 j_{12}}^{JM}(\hat{R}_1, \hat{R}_2, \hat{r}) = \sum_{\Omega} (2\ell + 1)^{1/2} \begin{pmatrix} \ell & j_{12} & J \\ 0 & \Omega & -\Omega \end{pmatrix} (-1)^{\Omega} \bar{\theta}_{j_1 j_2 j_{12}}^{JM}(\Omega, \hat{r}_1, \hat{r}_2, \hat{r}) \tag{A32}$$

where $\left(\begin{matrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{matrix} \right)$ is a 3-symbol³⁰. Similarly, the $B_{\beta\alpha\beta'}$ which we require in Eq. (A27) are related to the body-frame integrals of Eq. (A30) by

$$\begin{aligned}
B_{\beta\alpha\beta'}^{q_1 q_2 \mu} &= (2\ell + 1)(2\ell' + 1) \sum_{\Omega\Omega'} \begin{pmatrix} \ell & j_{12} & J \\ 0 & \Omega & -\Omega \end{pmatrix} \begin{pmatrix} \ell' & j'_{12} & J' \\ 0 & \Omega' & -\Omega' \end{pmatrix} \\
&\quad (-1)^{\Omega + \Omega'} B_{\beta\alpha\beta'}^{q_1 q_2 \mu}
\end{aligned} \tag{A33}$$

Combining Eqs. (A31) and (A33) gives

$$\begin{aligned}
B_{\beta\alpha\beta'}^{q_1 q_2 \mu} &= \delta_{JJ'} \delta_{MM'} (-1)^{j_{12} + j'_{12} + q_1 + q_2 + J} [2(1 + \delta_{\mu 0})]^{-1/2} [(2j_1 + 1)(2j_2 + 1)(2j_{12} + 1) \\
&\quad \times (2j'_1 + 1)(2j'_2 + 1)(2j'_{12} + 1)(2\ell + 1)(2\ell' + 1)(2q_1 + 1)(2q_2 + 1)]^{1/2} \begin{pmatrix} j_1 & q_1 & j'_1 \\ 0 & 0 & 0 \end{pmatrix} \\
&\quad \times \begin{pmatrix} j_2 & q_2 & j'_2 \\ 0 & 0 & 0 \end{pmatrix} \sum_{q_{12}} (q_{12} + 1) \begin{pmatrix} \ell & \ell' & q_{12} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} q_{12} & q_1 & q_2 \\ 0 & \mu & -\mu \end{pmatrix} \\
&\quad \left\{ \begin{matrix} \ell & \ell' & q_{12} \\ j'_{12} j_{12} & J & \end{matrix} \right\} \left\{ \begin{matrix} j'_{12} & q_{12} & j_{12} \\ j'_1 & q_1 & j_1 \\ j'_2 & q_2 & j_2 \end{matrix} \right\} \times [1 + (-1)^{q_1 + q_2 + q_{12}}]
\end{aligned} \tag{A34}$$

where $\left\{ \begin{matrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{matrix} \right\}$ is a 6-j symbol.³⁰ For the special case of $J = 0$, Eq. (A34) may be simplified to

$$\begin{aligned}
B_{\beta\ell\beta'\ell'}^{q_1q_2\mu} &= \delta_{j_12}\delta_{j'_12}\delta_{\mu0}(1+\delta_{\mu0})^{-1/2}[2(2j_1+1)(2j_2+1)(2q_1+1)(2q_2+1)(2\ell+1) \\
&\times (2j'_1+1)(2j'_2+1)(2\ell'+1)]^{1/2}(-1)^{j_1+j_2+q_1+q_2} \begin{pmatrix} j_1 & q_1 & j'_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} j_2 & q_2 & j'_2 \\ 0 & 0 & 0 \end{pmatrix} \\
&\times \sum_m \begin{pmatrix} \ell & j_1 & j_2 \\ 0 & m & -m \end{pmatrix} \begin{pmatrix} \ell' & j'_1 & j'_2 \\ 0 & -\mu-m & \mu+m \end{pmatrix} \\
&\begin{pmatrix} j'_1 & q_1 & j_1 \\ -\mu-m & \mu & m \end{pmatrix} \begin{pmatrix} j'_2 & q_2 & j_2 \\ \mu+m & -\mu & -m \end{pmatrix}. \tag{A35}
\end{aligned}$$

Since a 3-j symbol is a re-phased and re-normalized Clebsch-Gordan coefficient, and since Eq. (A35) has no 6-j or 9-j symbols, we can also express Eq. (A35) in terms of Clebsch-Gordan coefficients as

$$\begin{aligned}
B_{\beta\ell\beta'\ell'}^{q_1q_2\mu} &= \delta_{j_12}\delta_{j'_12}\delta_{\mu0}(-1)^{j_1+j_2+\ell'+q_1+q_2+\mu}[(2j'_2+1)(2j_2+1)(1+\delta_{\mu0})]^{-1/2} \\
&\times [2(2\ell+1)(2\ell'+1)(2q_1+1)(2q_2+1)]^{1/2}(j_10q_10|j_1q_1j'_10)(j_20q_20|j_2q_2j'_20) \\
&\times \sum_m (\ell0j_1m|\ell j_1j_2m)(\ell'0j'_1-\mu-m|\ell'j'_1j'_2-\mu-m)(j'_1-\mu-mq_1\mu|j'_1q_1j_1-m) \\
&\times (j'_2\mu+mq_2-\mu|j'_2q_2j_2m) \tag{A36}
\end{aligned}$$

Equation (A28) is a two-dimensional integral which must be performed at every sector for every unique pair $\gamma\gamma'$ of vibrational-rotational quantum numbers, and for every set of $q_1q_2\mu$ for which $B_{\beta\ell\beta'\ell'}^{q_1q_2\mu}$ is nonzero. The $C_{\gamma\gamma'}$ of Eq. (A28) are calculated for our test runs by a quadrature scheme³⁵ which uses "Gauss-ground-state" nodes. The values of the final scattering matrix elements of interest must be converged with respect to the number of points per vibrational coordinate used when performing this quadrature.

In order to evaluate (A28) entirely, however, it is also necessary to determine the $v_{q_1q_2\mu}$, which are given by

$$\begin{aligned}
v_{q_1q_2\mu}(R_1, R_2, r) &= \frac{1}{4\pi} \int_{-1}^1 [1 - \cos^2(\phi_1 - \phi_2)]^{-1/2} d[\cos(\phi_1 - \phi_2)] \int_{-1}^1 d(\cos \theta_1) \\
&\times \int_{-1}^1 d(\cos \theta_2) Y_{q_1q_2\mu}^*(\hat{r}_1, \hat{r}_2) V(\mathbf{x}, r) \tag{A37}
\end{aligned}$$

where θ_i and ϕ_i are the inclination and azimuthal angles of \hat{r}_i respectively. In the test suit included with the program, the calculation of (A37) has been performed differently for different HF-HF potentials. The modified Alexander-DePristo (MAD) potential includes the $v_{q_1q_2\mu}$ explicitly as parameters, as does the Schwenke-Truhlar (ST) potential. For these two surfaces, the eval-

uation of (A37) is not necessary. For other surfaces in the test suite where the integral is evaluated, integration over the θ_1 is performed using N_q -point Gauss-Legendre quadrature, and the integration over the $(\phi_1 - \phi_2)$ is performed using N_q -point Gauss-Chebyshev quadrature, where the order N_q of the quadrature was taken to be $q_{\max} + 1$, with q_{\max} the maximum value of q_1 or q_2 in the $v_{q_1 q_2}$. Therefore, the number of $v_{q_1 q_2}$ terms to be evaluated, which determines the maximum values of q_1 and q_2 used in the expansion of the potential (A24), is one of the parameters with respect to which calculated transition probabilities must be converged. It is important to note when performing the expansion above that the 3-dimensional integral in Eq. (A37) involves repeated evaluation of the $V(x, r)$, which is very time consuming, but *not* the basis set and so is independent of the number of channels in the close-coupling expansion, Eq. (10).

References

1. H.S.W. Massey, *Rev. Mod. Phys.* **28**, 199 (1956).
2. A.M. Arthurs and A. Dalgarno, *Proc. Roy. Soc., London* **A256**, 540 (1960).
3. L.L. Barnes, N.F. Lane, and C.C. Lin, *Phys. Rev.* **137**, A388 (1965).
4. A.C. Allison, *J. Comput. Phys.* **6**, 378 (1970).
5. W.A. Lester, Jr., in *Advances in Quantum Chemistry*, Vol. 9, edited by P.-O. Löwdin, Academic Press, New York, 1975, pp. 199-214.
6. J.Z.H. Zhang, D.J. Kouri, K. Haug, D.W. Schwenke, Y. Shima, and D.G. Truhlar, *J. Chem. Phys.* **88**, 2492 (1988).
7. D. Secrest, in *Atom-Molecule Collision Theory*, edited by R.B. Bernstein, Plenum, New York, 1979, pp. 265-299.
8. A. Degasperis, *Il Nuovo Cimento* [10] **34**, 1667 (1964).
9. B.R. Johnson and D. Secrest, *J. Math. Phys.* **7**, 2187 (1966).
10. M.E. Riley and A. Kuppermann, *Chem. Phys. Lett.* **1**, 537 (1968).
11. J.C. Light and R.B. Walker, *J. Chem. Phys.* **65**, 4272 (1976).
12. D.G. Truhlar, N.M. Harvey, K. Onda, and M.A. Brandt, in *Algorithms and Computer Codes for Atomic and Molecular Scattering Theory*, Vol. 1, edited by L. Thomas, National Resource for Computation in Chemistry, Lawrence Berkeley Laboratory, Berkeley, CA, 1979, pp. 220-289.
13. N.M. Harvey, Ph.D. dissertation, University of Minnesota, Minneapolis, 1979.
14. D.W. Schwenke and D.G. Truhlar, in *Supercomputer Applications*, edited by R.W. Numrich, Plenum, New York, 1985, pp. 215-254.
15. M.H. Alexander and A.E. DePristo, *J. Chem. Phys.* **66**, 2166 (1977).
16. N.A. Mullaney and D.G. Truhlar, *Chem. Phys. Lett.* **58**, 512 (1978).
17. N.A. Mullaney and D.G. Truhlar, *Chem. Phys.* **39**, 91 (1979).
18. J.M. Blatt and L.C. Biedernarn, *Rev. Mod. Phys.* **24**, 258 (1972).
19. D.G. Truhlar, C.A. Mead, and M.A. Brandt, *Adv. Chem. Phys.* **33**, 295 (1975).
20. D.G. Truhlar and N.A. Mullaney, *J. Chem. Phys.* **68**, 1574 (1978).
21. D.W. Schwenke, D. Thirumalai, D.G. Truhlar, and M.E. Coltrin, *J. Chem. Phys.* **78**, 3078 (1983).
22. P. Pechukas and J.C. Light, *J. Chem. Phys.* **44**, 3897 (1966).
23. J.C. Light, in *Methods in Computational Physics*, Vol. 10, edited by B. Alder, S. Fernbach, and M. Rotenberg, Academic Press, New York, 1971, pp. 111-141.
24. B.C. Garrett, M.J. Redmon, D.G. Truhlar, and C.F. Melius, *J. Chem. Phys.* **74**, 412 (1981).
25. D.W. Schwenke and D.G. Truhlar, in *Supercomputer Simulations in Chemistry*, edited by M. Dupuis, Springer-Verlag, Berlin, 1986, pp. 165-197.
26. D.W. Schwenke and D.G. Truhlar, *Theor. Chim. Acta* **72**, 1 (1987).

27. D.W. Schwenke and D.G. Truhlar, *J. Chem. Phys.* **88**, 4800 (1988).
28. M.J. Unekis, D.G. Truhlar, and D.W. Schwenke, unpublished.
29. E. Anderson, Z. Bai, C. Bichof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostouchov, D. Sorensen, *Lapack User's Guide*, SIAM, Philadelphia (1992).
30. A.R. Edmonds, *Angular Momentum in Quantum Mechanics*, Princeton University Press, Princeton, NJ (1960).
31. I.C. Percival and M.J. Seaton, *Proc. Camb. Phil. Soc.* **53**, 654 (1957).
32. D.U. Webb and K.N. Rao, *J. Mol. Spectroscopy* **28**, 121 (1968).
33. J.M. Launay, *J. Phys. B.* **10**, 3665 (1977).
34. G. Gioumoussis and C.F. Curtiss, *J. Math. Phys.* **2**, 96 (1961).
35. D.W. Schwenke and D.G. Truhlar, *Comp. Phys. Comm.* **34**, 57 (1984).