M. J. Unekis, D. W. Schwenke, N. M. Harvey, and D. G. Truhlar, in *Modern Techniques in Computational Chemistry: MOTECC-91*, edited by E. Clementi (ESCOM, Leiden, 1991), pp. 749-772.

[M. J. Unekis, D. W. Schwenke, N. M. Harvey, and D. G. Truhlar, Mod. Tech. Comput. Chem.: MOTECC-91, 749-72 (1991)]

Chapter 17.

RMPROP: 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, Minnesota 55455-0431, U.S.A.

and

NASA Ames Research Center, Moffett Field, California 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.[1-3] These equations may be solved by boundary value methods[4] or propagation techniques, and the latter may be subdivided into so-called approximate solution approaches[4-7] and invariant embedding (also called approximate-potential) approaches.[7-11] The R matrix propagation algorithm[11-14] 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 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 requires information

* University of Minnesota.

† University of Minnesota and NASA Ames Research Center.

Modern Techniques in Computational Chemistry: MOTECC-91

Edited by

Enrico Clementi

International Business Machines Corporation
Kingston, New York 12401, U.S.A.

ESCOM
Leiden 1991

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, from which all physical observables of a fixed-energy collision process may be calculated by the researcher. The program may also be used to calculate specific state-to-state cross sections if desired. 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 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, the orbital angular momentum of relative translational motion, and the noninteracting Hamiltonians of the separated collision partners.

Some limitations on the program are that rearrangement collisions cannot be treated, and at least one of the collision partners must be neutral. In addition, if one of the collision partners is an electron, so that there is no strong repulsive potential when the collision partners are at zero separation, the orbital angular momentum quantum number must be nonzero in all channels.

The program solves the close coupling equations by R matrix propagation,[11-14] 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, and the local R matrices are combined into a global R matrix, which is propagated towards large r. This algorithm has the advantage that 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 nformation equivalent to the solutions and their derivatives) are matched to the ext sector at the boundary so that a sector R matrix propagating the solution rough a sector and across one of its boundaries is defined. When the sector R atrices are combined, a global R matrix, which propagates the solutions from he strong-interaction region at small r to the asymptotic region at large r, is btained. The solutions in the asymptotic region are matched to scattering oundary conditions from which an S matrix is calculated. The user should epeat the calculation to converge it with respect to increasing the size of the close oupling basis, decreasing the sector widths, and increasing the distance at large r t which propagation is stopped, as well as with respect to any numerical parameers associated with the interaction potential function.

Much of the operation of the program and practical instructions for its use are iven in an on-line manual which is distributed with the code. (By "on-line nanual" we mean an ASCII documentation file.) The on-line manual also conains a test suite which may be used to check whether the program is performing orrectly in a new environment or on a new machine. 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 or molecule-molecule collisions, and may be used for electron-atom or electron-molecule collisions as well when the scattering electron has nonzero orbital angular momentum. In all cases, the underlying athematics is the same; 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* equations, and the subsequent solution of these equations.

2.1 Coupled channels equations

In the treatment which follows, we consider the collision of two molecules A and B, for greatest generality. The program is 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 $\Psi_{n_0}(\mathbf{r}, \mathbf{x}, E)$ which is a solution of the Schrödinger equation

$$H\Psi_{n_0} = E\Psi_{n_0}. \tag{1}$$

The vector **r** connects the center of mass of A and the center of mass of B, 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:

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

where μ is the reduced mass of relative translational motion, ∇_r^2 is the Laplacian with respect to **r**, H_{int} is the "internal Hamiltonian", defined as the sum of the Hamiltonians of the isolated A and B molecules, and V(**x**, **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 $\tilde{H}_{int}$ (this may be just H_{int}) and the remainder ΔH_{int}. We then combine the angular portion of $-(\hbar^2/2\mu)\nabla_r^2$ with $\tilde{H}_{int}$ to form the primitive Hamiltonian, H^0, so that equation (1) becomes

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

where

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

and

$$\vartheta(\mathbf{x}, r) = V(\mathbf{x}, r) + \Delta H_{int}(\mathbf{x}) \tag{5}$$

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

To determine Ψ_{n_0}, we expand it in terms of simultaneous matrix eigenvectors of $\tilde{H}_{int}$ and ℓ_r^2 defined by

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

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

and

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

ere δ_{mn} is the Kronecker delta, and equation (8) is an orthonormality condition. these equations, $\hat{r}$ is the unit vector which has the same direction as $\mathbf{r}$, and n m are collective quantum numbers identifying channels. In terms of the above e coupling basis, the wave function expansion is

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

ere each term in this expansion is associated with a different channel. Substiing equation (9) into equation (1), multiplying through by $rX^*(\mathbf{x}, \hat{\mathbf{r}})$, and inteating over $\mathbf{x}$ and $\hat{\mathbf{r}}$ yields

$$\left\{ \frac{-\hbar^2}{2\mu} \frac{d^2}{dr^2} + \ell_m(\ell_m + 1) \frac{\hbar^2}{2\mu r^2} + \tilde{\varepsilon}_m \right\} f_{mn_0} + \sum_{n=1}^{N} V_{mn}(r) f_{mn_0} \tag{10}$$
$$= E f_{mn_0}(r, E), \qquad m = 1, 2, \ldots N$$

here

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

quations (10) are called the close coupling equations (or the coupled channels equations), and they may be written in the form

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

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_0} of equation (9), where the rows of $\mathbf{f}$ (labelled by "n" there) refer to different channels in which the radial translational wave function is represented, and the columns of $\mathbf{f}$ (labelled by "n_0" there) refer to different sets of initial quantum numbers. The size of N, the number of terms in equation (9), is found by allowing one term for each channel of interest, and including sufficient additional terms so that the expansion (9) will be a mathematically accurate representation of the wave function. For the present treatment (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 $(-1)^{j_1+j_2+\ell}$, 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]

The quantum numbers J, M, η, and ζ are "good" quantum numbers, so that the matrix **D** is block diagonal in them. 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 the applications included in the test suite we do in fact follow this procedure, and furthermore we only consider the blocks specified by $\eta = \zeta = +1$ (the program does *not* have this restriction), while allowing J and M to vary according to the problem.

The elements of **D** are given by

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

where

$$\tilde{k}_n^2 = \frac{2\mu}{\hbar^2}(E - \tilde{\mathcal{E}}_n) \tag{14}$$

where **D** is real and symmetric, and $\tilde{k}_n$ is called the primitive wave number. The channels are ordered so that $\tilde{k}_n^2 > \tilde{k}_m^2$ if and only if $n > m$.

2.2. Asymptotic boundary conditions

In order to treat the large-r boundary conditions on equation (12) we must define a transformation which diagonalizes **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

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

such that

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

It is noteworthy that in equation (13), the total energy E appears only on the diagonal of **D**, due to the Kronecker delta multiplying the term $\tilde{k}_n^2$ which contains E. Because the E appears only on the diagonal of **D**, the eigenvalues λ_{mn}^2 depend on E, but the eigenvectors (which are the columns of **U**) do not. (This property is i when performing calculations at multiple energies so that second and subse- nt energies do not require as much work.) If we use the Z_m as new basis func- s (which are called adiabatic basis functions[11,16,17]), the expansion of the wave ction becomes

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

ere the g_{mn_0} are related to the f_{mn_0} by the application of the unitary transforma- n U of equation (15). The terms in equation (17), like the ones in equation (9), called channels. At small r, the program requires regular boundary conditions the origin:

$$g_{mn} \underset{r \to 0}{\sim} 0 \qquad 1 \leq n, m \leq N \tag{18}$$

t large r, there is more than one equivalent form of the boundary conditions to e close coupling equations. The program allows the choice of sine/cosine oundary conditions:

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

or of matching to Ricatti-Bessel functions:

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

where j_{ℓ_m} and h_{ℓ_m} are the Ricatti-Bessel functions:

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

The coefficient b in (19) is arbitrary, k_n is an asymptotic wave number defined by

$$k_n = \lim_{r \to \infty} |\lambda_{nn}(r, E)|, \tag{22}$$

and P^o is defined as the number of open channels in the basis, where an open channel is defined as one with positive k_n^2. Channels which are not "open" are said to be “closed”. If H_{int} is the same as $\tilde{H}_{int}$, than k_n equals $|\tilde{k}_n|$ 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

$$\mathbf{S} = [\mathbf{1} + i\mathbf{K}(E)]\,[\mathbf{1} - i\mathbf{K}(E)]^{-1} \tag{23a}$$

or the transition matrix defined by

$$\mathbf{T} = \mathbf{S} - \mathbf{1} \tag{23b}$$

where **1** is the unit matrix, $i^2 = -1$, and **K** is the real symmetric $P^o \times P^o$ reactance matrix with elements

$$\mathbf{K}_{nm} = k_n^{1/2} a_{nm} k_m^{1/2} \qquad 1 \leq n, m \leq P^o \tag{24}$$

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.

3. R Matrix Propagation Algorithm

The R matrix propagation algorithm was presented originally by Light and Walker[11] for colinear atom-diatom collisions, and it has been generalized by various workers, including those authors. Our own generalization,[12-14,16-19] 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 equation (12) are solved by subdividing the coordinate r into some number N_s of sectors, with sector midpoints $r_c^{(i)}$ and sector widths $h^{(i)}$ such that

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

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

$$Z_m^{(i)}(x, \hat{r}) = \sum_{n=1}^{N} T_{nm}^{(i)} X_m(x, \hat{r}) \qquad 1 \leq m \leq P^{(i)} \tag{26}$$

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

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

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

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

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

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

ere

$$L_{nm}^{(i)}(r, E) = \sum_{k,k'=1}^{N} \mathbf{T}_{kn}^{(i)} D_{kk'}(r, E) \mathbf{T}_{k'm}^{(i)} \qquad 1 \le n, m \le P^{(i)} \tag{30}$$

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

$$g_{mn}^{(i)}(r, E) = \sum_{k=1}^{N} \mathbf{T}_{km}^{(i)} f_{kn}(r, E) \qquad 1 \le m, n \le P^{(i)} \tag{31}$$

3.2. Sector propagation matrix

We consider first the small-r boundary conditions, equation (18). In principle we should have $r_c^{(1)} - (h^{(1)}/2)$ equal to zero. However for atom-molecule and molecule-molecule collisions or for electron scattering when none of the ℓ_n in eq. (13) is zero, one finds that

$$V_{nn}(r) + \frac{\ell_n(\ell_n + 1)}{2\mu r^2} >> E \qquad \text{all} \quad n, r << \sigma \tag{32}$$

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. We choose such a nonzero value of r to be $r_c^{(1)}$ 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 the homogeneous boundary condition at small r allows us to simplify the propagation, as discussed below eq. (57).

We begin the propagation with N channels, that is, with $P^{(1)} = N$. At large r, because of equation (19), f_{mn} for $m > P^o$ 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.[19] 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 $P^{(i)} \geq P^o$ for all (i). We define the $2P^{(i)} \times 2P^{(i)}$ sector propagator $\mathbf{P}^{(i)}$ by

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

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

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

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

$$r_L^{(i)} = r_c^{(i)} - \frac{h^{(i)}}{2}, \tag{35}$$

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

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

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

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

We use the first-order Magnus method[11,20-22] for $\mathbf{O}^{(i)}(E)$. The $\mathbf{g}^{(i)}$ are the $2P^{(i)}$ linearly independent solution vectors to equation (29). Since each column $g^{(i)}$ 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):

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

Following the work of Pechukas and Light,[20] 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 $r = r_c^{(i)}$:

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

r which, to third order in the stepsize, **B** is given by

$$\mathbf{B} = -h^{(i)}\begin{pmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{L}^{(i)}\left(r_c^{(i)}, E\right) & \mathbf{0} \end{pmatrix} - \frac{\left(h^{(i)}\right)^3}{12}\begin{pmatrix} -\mathbf{L}'^{(i)}\left(r_c^{(i)}, E\right) & \mathbf{0} \\ \mathbf{0} & -\mathbf{L}'^{(i)}\left(r_c^{(i)}, E\right) \end{pmatrix} + \text{Order}\left[\left(h^{(i)}\right)^5\right] \tag{40}$$

here

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

nd where

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

vith the $\mathbf{L}^{(i)}$ defined by equation (30). In the first Magnus approximation, the econd and higher terms in (40) are neglected. This is tantamount to setting the lerivative in (42) equal to zero so that the matrix $\mathbf{L}^{(i)}$ 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 equation (30) into equations (39) and (40) and using equation (37), one obtains for the $\mathbf{P}_j^{(i)}(E)$ using the first Magnus approximation:

$$\left[\mathbf{P}_1^{(i)}(E)\right]_{nm} = \left[\mathbf{P}_4^{(i)}(E)\right]_{nm} = \begin{cases} d_{nm}\cosh\left[-h^{(i)}\left|l_{nn}^{(i)}(E)\right|\right], & l_{nn}^{(i)^2}(E) > 0 \\ d_{nm}\cos\left[-h^{(i)}\left|l_{nn}^{(i)}(E)\right|\right], & l_{nn}^{(i)^2}(E) < 0 \end{cases} \tag{43}$$

$$\left[\mathbf{P}_2^{(i)}(E)\right]_{nm} = \begin{cases} d_{nm}\left|l_{nn}^{(i)}(E)\right|^{-1}\sinh\left[-h^{(i)}\left|l_{nn}^{(i)}(E)\right|\right], & l_{nn}^{(i)^2}(E) > 0 \\ d_{nm}\left|l_{nn}^{(i)}(E)\right|^{-1}\sin\left[-h^{(i)}\left|l_{nn}^{(i)}(E)\right|\right], & l_{nn}^{(i)^2}(E) < 0 \end{cases} \tag{44}$$

$$\left[\mathbf{P}_3^{(i)}(E)\right]_{nm} = \begin{cases} \lambda_{nn}^{(i)^2}(E)\left(\mathbf{P}_2^{(i)}\right)_{nm}, & \lambda_{nn}^{(i)^2}(E) > 0 \\ -\lambda_{nn}^{(i)^2}(E)\left(\mathbf{P}_2^{(i)}\right)_{nm}, & \lambda_{nn}^{(i)^2}(E) < 0 \end{cases} \tag{45}$$

3.3. Stepsize determination

Examination of equation (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

$$\left(h^{(i)}\right)^3 \frac{d\mathbf{L}^{(i)}}{dr},$$

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 $\mathbf{L}^{(i)}$ 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

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

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

$$h^{(i+i)} = \min \begin{cases} \mathcal{E}\left[\frac{1}{N} \sum_{j=1}^{N} \left(D_{jj}^{(i)} - D_{jj}^{(i-1)} \right)^2 \right]^{-1/6} \Big/ \left(r_c^{(i)} - r_c^{(i-1)} \right) \\ h_{\max} \end{cases} \tag{47}$$

where $\mathcal{E}$ and $h_{\max}$ are input parameters to the program.

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 $\mathcal{E}$ in one interval and another in another. (See the on-line manual.) For many problems, a ed stepsize at small r followed by one or two regions with Ɛ-controlled stepsizes the most stable and efficient procedure.

4. *Propagation across sector boundaries*

e continuity of the functions $\mathbf{g}^{(i)}$ of equation (29) across sector boundaries may expressed by

$$\mathbf{g}_R^{(i-1)}(E) = \boldsymbol{\tau}(i-1, i)\mathbf{g}_L^{(i)}(E) \tag{48}$$

$$\mathbf{g'}_R^{(i-1)}(E) = \boldsymbol{\tau}(i-1, i)\mathbf{g'}_L^{(i)}(E) \tag{49}$$

here the overlap matrix $\boldsymbol{\tau}(i-1, i)$ is defined by

$$\tau_{nm}(i-1, i) = \sum_{k=1}^{N} T_{kn}^{(i-1)} T_{km}^{(i)} \qquad 1 \le n, m \le P^{(i)} \tag{50}$$

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

$$\begin{pmatrix} \mathbf{g}_R^{(i-1)}(E) \\ \mathbf{g}_R^{(i)}(E) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1^{(i)}(E) & \mathbf{r}_2^{(i)}(E) \\ \mathbf{r}_3^{(i)}(E) & \mathbf{r}_4^{(i)}(E) \end{pmatrix} \begin{pmatrix} \mathbf{g'}_R^{(i-1)}(E) \\ -\mathbf{g'}_R^{(i)}(E) \end{pmatrix} \tag{51}$$

$$\mathbf{r}^{(i)}(E) = \begin{pmatrix} \mathbf{r}_1^{(i)}(E) & \mathbf{r}_2^{(i)}(E) \\ \mathbf{r}_3^{(i)}(E) & \mathbf{r}_4^{(i)}(E) \end{pmatrix} \tag{52}$$

where the $\mathbf{r}_j^{(i)}$ matrices are $P^{(i)} \times P^{(i)}$. The $\mathbf{r}_j^{(i)}$ are expressed in terms of the $\mathbf{P}_j^{(i)}(E)$ and the $\boldsymbol{\tau}(i-1, i)$ as follows:

$$\mathbf{r}_1^{(i)}(E) = \boldsymbol{\tau}(i-1, i)\mathbf{P}_1^{(i)}(E)\left[\mathbf{P}_3^{(i)}(E)\right]^{-1}[\boldsymbol{\tau}(i-1, i)]^{-1} \tag{53}$$

$$\mathbf{r}_2^{(i)}(E) = \boldsymbol{\tau}(i-1, i)\left[\mathbf{P}_3^{(i)}(E)\right]^{-1} \tag{54}$$

$$\mathbf{r}_3^{(i)}(E) = \left[\mathbf{P}_3^{(i)}(E)\right]^{-1}[\boldsymbol{\tau}(i-1, i)]^{-1} \tag{55}$$

$$\mathbf{r}_4^{(i)}(E) = \left[\mathbf{P}_3^{(i)}(E)\right]^{-1}\mathbf{P}_4^{(i)}(E) \tag{56}$$

It should be noted that $\mathbf{P}_3^{(i)}$ is diagonal so that its inversion is not time-consuming.

The global R matrix which spans from the left-hand side of the first sector to the right-hand side of sector (i) is defined by

$$\begin{pmatrix} g_L^{(1)}(E) \\ g_R^{(i)}(E) \end{pmatrix} = \begin{pmatrix} R_1^{(i)}(E) & R_2^{(i)}(E) \\ R_3^{(i)}(E) & R_4^{(i)}(E) \end{pmatrix} \begin{pmatrix} g'^{(1)}_L(E) \\ -g'^{(i)}_R(E) \end{pmatrix} \tag{57}$$

where the $\mathbf{R}_j^{(i)}$ matrices are all square. It can be shown that if

$$\left[\lambda_{nn}^{(1)}\right]^2 >> \frac{2\mu E}{\hbar^2}$$

for all n, then $\mathbf{R}_2^{(i)}$ and $\mathbf{R}_3^{(i)}$ are approximately zero and that all scattering information (a_{mn}, $1 \leq m, n \leq P^o$) can be determined solely from $\mathbf{R}_4^{(N_s)}$. (In the present version of the program we impose this condition as a requirement, which may be easily satisifed if the propagation is begun sufficiently far enough inside the repulsive potential barrier. This poses no restriction for atom-atom, atom-molecule, and molecule-molecule scattering, but it does mean that the code is inapplicable, as already stated, to electron scattering problems in which one or more channels has a zero value for ℓ. We note for completeness though that we encountered no difficulty in treating such cases with earlier, more general versions of the code.[16,18]) The matrix $\mathbf{R}_4^{(i)}$ only depends on $\mathbf{R}_4^{(i-1)}$ and the $\mathbf{r}_j^{(i)}$:

$$\mathbf{R}_4^{(i)}(E) = \mathbf{r}_4^{(i)}(E) - \mathbf{r}_3^{(i)}(E)\left[\mathbf{R}_4^{(i-1)}(E) + \mathbf{r}_1^{(i)}(E)\right]^{-1}\mathbf{r}_2^{(i)}(E) \tag{58}$$

and

$$\mathbf{R}_4^{(1)}(E) = \mathbf{r}_4^{(1)}(E) \tag{59}$$

The program however has an option, chosen by setting the value of the input variable NPROP equal to 2, which allows it to calculate the $\mathbf{R}_4^{(i)}$ treating the $\mathbf{r}_j^{(i)}$ implicitly instead of explicitly. If this option is chosen, then one less matrix inversion must be performed at each step, and for "later date second energy runs" (Section 3.7) the $[\tau(i-1, i)]^{-1}$ need not be written to disk, which decreases storage requirements.

3.5. 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. These are needed for convergence at small r, but 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°, where P° is the number of open channels in the asymptotically large-r region, without introducing errors in the reactance matrix **K** or scattering matrix 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 $\mathbf{R}_4^{(i)}(E)$ is propagated, and we allow dropping channels only with the constraint

t $P^{(i)}(E) \geq P^{o}(E)$. Before proceeding, it will be convenient to introduce the fol-
ing convention for partitioning a P × P matrix into quadrants according to
ther open and/or closed channels are linked there:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}^{oo} & \mathbf{Y}^{oc} \\ \mathbf{Y}^{co} & \mathbf{Y}^{cc} \end{pmatrix} \tag{60}$$

re the superscript oo identifies the $P^{o} \times P^{o}$ submatrix containing Y_{mn} for
m $\leq P^{o}$ and $1 \leq n \leq P^{o}$, i.e., the matrix elements linking open channels to
n channels, the superscript oc identifies the $P^{o} \times (P - P^{o})$ submatrix containing
for $1 \leq m$, P^{o} and $P^{o} \leq n \leq P$, i.e., the matrix elements linking open channels
closed channels, the superscript co identifies the $(P - P^{9}) \times P^{o}$ submatrix con-
ning the Y_{mn} for $P^{o} \leq m \leq P$ and $1 \leq n \leq P^{o}$, and the superscript cc identifies
$(P - P^{o}) \times (P - P^{o})$ submatrix containing Y_{mn} for $P^{o} \leq m \leq P$ and $P^{o} \leq n \leq P$,
. the matrix elements linking closed channels to closed channels. To justify
opping the $P - P^{o}$ closed channels from propagation in sector (i − 1), we must
ow that to accurately obtain $\mathbf{R}_4^{(i)oo}(E)$ from $\mathbf{R}_4^{(i-1)oo}(E)$ does not require any infor-
ation involving the $P - P^{o}$ closed channels. (We only require the oo submatrix
$\mathbf{R}_4^{(i)}(E)$ because the submatrices involving closed channels are not required in
e asymptotic analysis to obtain the reactance matrix). Using equation (58) and
e notation for partitioned matrices, $\mathbf{R}_4^{(i)oo}(E)$ can be written as

$$\begin{aligned} {}_4^{(i)oo}(E) = \mathbf{r}_4^{(i)oo}(E) - \mathbf{r}_3^{(i)oo}(E)\left[\mathbf{S}^{oo}(i-1, i, E)\mathbf{r}_2^{(i)oo}(E) + \mathbf{S}^{oc}(i-1, i, E)\mathbf{r}_2^{(i)oc}\right] \\ - \mathbf{r}_3^{(i)oc}(E)\left[\mathbf{S}^{co}(i-1, i, E)\mathbf{r}_2^{(i)oo}(E) + \mathbf{S}^{cc}(i-1, i, E)\mathbf{r}_2^{(i)co}\right] \end{aligned} \tag{61}$$

where

$$\mathbf{S}(i-1, i, E) = \left[\mathbf{R}_4^{(i-1)}(E) + \mathbf{r}_1^{(i)}(E)\right]^{-1} \tag{62}$$

In our calculations both $\tau(i-1, i)$ and $\tau^{-1}(i-1, i)$ tend to a unit matrix in the large-r limit, so equation (53) reduces to

$$\lim_{r_L^{(i)} \to \infty} \mathbf{r}_1^{(i)} = \mathbf{P}_1^{(i)}(E)\left[\mathbf{P}_3^{(i)}(E)\right]^{-1} \tag{63}$$

i.e., it tends to a diagonal matrix. Additionally, it can be shown[13] that $\mathbf{R}_4^{(i)}(E)$ tends to a block-diagonal matrix in the large-r limit. Therefore, for large r,

$$\lim_{r_L^{(i)} \to \infty} \mathbf{S}(i-1, i)^{oo} = \left[\mathbf{R}_4^{(i-1)oo}(E) + \mathbf{r}_1^{(i)oo}(E)\right]^{-1} \tag{64}$$

$$\lim_{r_L^{(i)} \to \infty} \mathbf{S}(i-1, i)^{oc} = \mathbf{0}^{oc} \tag{65}$$

$$\lim_{r_L^{(i)} \to \infty} \mathbf{S}(i-1, i)^{co} = \mathbf{0}^{co} \tag{66}$$

$$\lim_{r_L^{(i)}\to\infty} S(i-1,i)^{cc} = \left[\mathbf{R}_4^{(i-1)cc}(E) + \mathbf{r}_1^{(i)cc}(E)\right]^{-1} \tag{67}$$

Substituting these into (61) yields

$$\begin{aligned}\lim_{r_L^{(i)}\to\infty} \mathbf{R}_4^{(i)oo}(E) = \mathbf{r}_4^{(i)oo}(E) &- \mathbf{r}_3^{(i)oo}(E)\mathbf{S}^{oo}(i-1,i,E)\mathbf{r}_2^{(i)oo}(E)\\ &- \mathbf{r}_3^{(i)oc}(E)\mathbf{S}^{cc}(i-1,i,E)\mathbf{r}_2^{(i)co}(E)\end{aligned} \tag{68}$$

Since, at asymptotically large r, **T**(i −1, i) approaches a unit matrix, equations (54) and (55) show that

$$\lim_{r_L^{(i)}\to\infty} \mathbf{r}_2^{(i)}(E) = \mathbf{r}_3^{(i)}(E)^T \tag{69}$$

Therefore, if

$$\lim_{r_L^{(i)}\to\infty} \mathbf{r}_2^{(i)co}(E) = \mathbf{0}^{co}, \tag{70}$$

then equations (68) and (69) show that the calculation of $\mathbf{R}_4^{(i)oo}(E)$ is independent of information about the P – P° closed channels. Consequently, 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 $r_c^{(i)}$ is less than some input value read in by the program, or if $P^{(i)}$ has already been reduced to P°, $P^{(i+1)}$ is set to $P^{(i)}$. Otherwise the program checks to see whether

$$\left|\left(\mathbf{r}_2^{(i)}\right)_{nP^{(i)}}\right| \le \text{EPSRED} \quad \text{and} \quad \left|\left(\mathbf{r}_2^{(i)}\right)_{P^{(i)}n}\right| \le \text{EPSRED},$$

where EPSRED is a variable set by the user (a typical value would be 1.0×10^{-3}). If both inequalities are satisfied, than $P^{(i+1)}$ is set to $P^{(i)} - 1$; otherwise, $P^{(i+1)} = P^{(i)}$.

3.6. Asymptotic reordering of channels

Before applying the boundary conditions of equations (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

$$D_{nm}(r, E) = \delta_{nm}\left[\frac{\ell_n(\ell_n + 1)}{r^2} - \tilde{k}^2_n(E)\right] \tag{71}$$

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

is necessary to make new linear combinations of the channels if there exist generate channels, i.e., channels with $[\lambda^{(i)}_{nn}]^2 = [\lambda^{(i)}_{mm}]^2$ and $n \neq m$. This is because matrix diagonalization routine will mix those states. Degenerate channels can cur in the calculations at very large r where the term $[\ell(\ell + 1)]/(2\mu r^2)$ is neglile since there are channels with the same $\underset{\sim}{k}_n$ but different values of ℓ_n. (Addinal accidental degeneracies would occur if one used the harmonic oscillator and id-rotor approximations to calculate asymptotic energies, but the use of these is t recommended.) In order to sort out these effects, we make the transformation new radial functions $\mathbf{h}^{(i)}$ defined by

$$g^{(i)}_{mn}(r, E) = \sum_{\ell=1}^{N}\sum_{k=1}^{P^{(i)}} T^{(i)}_{\ell m} U^o_{\ell k} h^{(i)}_{kn}(r, E),\ 1 \le m \le P^{(i)},\ 1 \le n \le P^{(i)} \tag{72}$$

where U^o diagonalizes H_{int} 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

$$\mathbf{g}^{(i)}(r, E) = [\mathbf{T}(i)]^T \mathbf{U}^o \mathbf{h}^{(i)}(r, E) \tag{73}$$

We then define a new global R matrix satisfying

$$\begin{pmatrix} \mathbf{h}^{(1)}_L \\ \mathbf{h}^{(i)}_R \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{R}}^{(i)}_1 & \tilde{\mathbf{R}}^{(i)}_2 \\ \tilde{\mathbf{R}}^{(i)}_3 & \tilde{\mathbf{R}}^{(i)}_4 \end{pmatrix} \begin{pmatrix} \mathbf{h}'^{(1)}_L \\ -\mathbf{h}'^{(i)}_R \end{pmatrix} \tag{74}$$

where

$$\tilde{\mathbf{R}}^{(i)}_1 = (\mathbf{T}^{(i)}\mathbf{U}^o)^{-1}\tilde{\mathbf{R}}^{(i)}_1[\mathbf{T}^{(i)}[^T\mathbf{U}^o \tag{75}$$

$$\tilde{\mathbf{R}}^{(i)}_2 = (\mathbf{T}^{(i)}\mathbf{U}^o)^{-1}\tilde{\mathbf{R}}^{(i)}_2[\mathbf{T}^{(i)}]^T\mathbf{U}^o \tag{76}$$

$$\tilde{\mathbf{R}}^{(i)}_3 = (\mathbf{T}^{(i)}\mathbf{U}^o)^{-1}\tilde{\mathbf{R}}^{(i)}_3[\mathbf{T}^{(i)}]^T\mathbf{U}^o \tag{77}$$

$$\tilde{\mathbf{R}}^{(i)}_4 = (\mathbf{T}^{(i)}\mathbf{U}^o)^{-1}\tilde{\mathbf{R}}^{(i)}_4[\mathbf{T}^{(i)}]^T\mathbf{U}^o \tag{78}$$

The matrix **a** whose elements appear in equation (19) is determined by

$$\mathbf{a}(E) = \lim_{i \to \infty} \mathbf{a}^{(i)}(E) \tag{79}$$

where

$$\mathbf{a}^{(i)}(E) = \left[-\mathbf{F}(E) + \tilde{\mathbf{R}}_4^{(i)}(E)\mathbf{H}^{(i)}(E) \right]^{-1} \left[\mathbf{B}^{(i)}(E) + \tilde{\mathbf{R}}_4^{(i)}(E)\mathbf{G}^{(i)}(E) \right] \Delta \tag{80}$$

and

$$F_{mn}^{(i)}(E) = \delta_{mn} \begin{cases} \cos\left[k_m(E) r_R^{(i)} - \dfrac{\ell_m \pi}{2} \right] & 1 \le m \le P^o \\ \exp\left[-k_m(E) r_R^{(i)} \right] & P^o < m \le P^{(i)} \end{cases} \tag{81}$$

$$B_{mn}^{(i)}(E) = \delta_{mn} \begin{cases} \sin\left[k_m(E) r_R^{(i)} - \dfrac{\ell_m \pi}{2} \right] & 1 \le m \le P^o \\ \exp\left[k_m(E) r_R^{(i)} \right] & P^o < m \le P^{(i)} \end{cases} \tag{82}$$

$$H_{mn}^{(i)}(E) = k_m \delta_{mn} \begin{cases} \sin\left[k_m(E) r_R^{(i)} - \dfrac{\ell_m \pi}{2} \right] & 1 \le m \le P^o \\ \exp\left[k_m(E) r_R^{(i)} \right] & P^o < m \le P^{(i)} \end{cases} \tag{83}$$

$$G_{mn}^{(i)}(E) = k_m \delta_{mn} \begin{cases} \cos\left[k_m(E) r_R^{(i)} - \dfrac{\ell_m \pi}{2} \right] & 1 \le m \le P^o \\ \exp\left[-k_m(E) r_R^{(i)} \right] & P^o < m \le P^{(i)} \end{cases} \tag{84}$$

$$\Delta_{nm} = \delta_{nm} \begin{cases} 1 & 1 \le m \le P^o \\ b & P^o \le m \le P^{(i)} \end{cases} \tag{85}$$

If it is true that for a given $M > P^o$, $(\mathbf{R}_4^{(i)})_{nm} = (\mathbf{R}_4^{(i)})_{mn} = 0$ for all $n \le P^o$, channel m is not required in the calculation of $a_{nm}^{(i)}$, $1 \le n$, $m \le P^o$. This program determines the smallest $m \ge P^9$, called $P_a^{(i)}$, such that

$$\left| \left(\mathbf{R}_4^{(i)} \right)_{nm} \right| \le \text{EPSDR} \quad \text{and} \quad \left| \left(\mathbf{R}_4^{(i)} \right)_{mn} \right| \le \text{EPSDR for all } n \le P^o,$$

where EPSDR is some small number (a typical value is 1.0×10^{-3}), and then uses the upper left $P_a^{(i)} \times P_a^{(i)}$ subblock of $\mathbf{R}_4^{(i)}(E)$ to calculate $\mathbf{a}^{(i)}$.

.7. *Single/multiple energy runs*

nvestigation of equation (13) shows that the total energy E appears only as a ultiple of the unit matrix, so that the matrices $\mathbf{T}^{(i)}$ are independent of the total nergy, and the eigenvalues $\lambda_{nn}^{(i)2}(E)$ at a new energy are easily related to those at he old energy:

$$\left[\lambda_{nn}^{(i)^2}(E)\right] = \left[\lambda_{nn}^{(i)^2}(E')\right] + \frac{2\mu(E' - E)}{\hbar^2} \tag{86}$$

his fact may be used to save computer time for multiple-energy runs by reusing he $\lambda^{(i)}$ and $\tau(i-1,i)$. Saving these variables makes the calculation of the $_{nm}(r)$, $\mathbf{T}^{(i)}$, and $\tau(i-1,i)$ unnecessary for second and subsequent energies and ffords significant reductions in computer time. The possibility of these savings is ne of the many attractive characteristics of the present algorithm. The drawback, owever, is a corresponding increase in storage requirements to save the temporary values in these arrays. There are two ways in which the present program mplements 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 $P^{(i)} \times P^{(i)}$ matrices $\tau(i-1,i)$ and $[\tau(i-1,i)]^{-1}$ 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 $\tau(i-1,i)$ and $[\tau(i-1,i)]^{-1}$

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 in 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 $N^2/2$ 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 3.7, 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

Table I. Components of the Program.

Component Number	Subroutine name(s)	Purpose
I	POT	Calculation $V_{nm}^{(i)}$ at $r_c^{(i)}$
II	RMPROP, RS	Assembly and diagonalization of $\mathbf{D}^{(i)}$ to obtain $\lambda_{nn}^{(i)}$ and $\boldsymbol{\tau}^{(i)}$
III	TAUMTS	Calculation of $\boldsymbol{\tau}(i-1, i)$ and/or $[\mathbf{T}(i-1, i)]^{-1}$
IV	RPROP	Calculation $\mathbf{R}_4^{(i)}$ and $P^{(i+1)}$
V	GNSCAT	Calculation of S from $\mathbf{R}_4^{(i)}$
VI	...	Everything else

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 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.

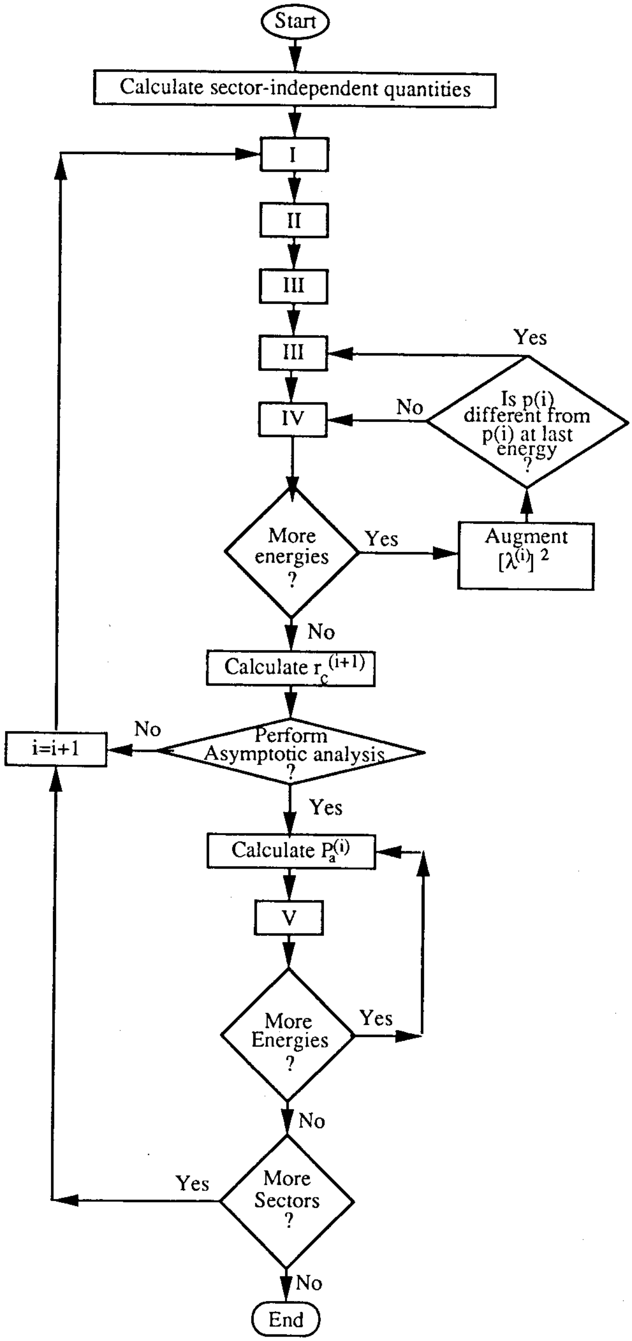
Start
Calculate sector-independent quantities
I
II
III
III
Yes
IV
No
Is p(i) different from p(i) at last energy ?
More energies ?
Yes
Augment $[\lambda^{(i)}]^2$
No
Calculate $r_c^{(i+1)}$
No
i=i+1
Perform Asymptotic analysis ?
Yes
Calculate $P_a^{(i)}$
V
More Energies ?
Yes
No
Yes
More Sectors ?
No
End

Figure 1. Flow Chart of the RMPROP computer program.

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 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.

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.[14,23,24]

Because of this high efficiency, RMPROP can be used to solve very large problems. For example, we have reported calculations with up to 1358 channels[25] for a diatom-diatom scattering problem with long-range dipole-dipole coupling. In unpublished work[26] we have completed calculations on the same problem with up to 1897 channels.

6. 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, first distributed version (version 1), we have made a formal test suite and have successfully run it on an IBM 3090 600J computer under the AIX operating system and on a Cray-2 computer under UNICOS, version 5.1.

7. Acknowledgments

This work was supported in part by the National Science Foundation, Minnesota Supercomputer Institute, and Cray Research, Inc.

8. 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. W.A. Lester, Jr., in *Advances in Quantum Chemistry*, Vol. 9, edited by P.-O. Löwdin, Academic Press, New York, 1975, pp. 199-214.
4. J.Z.H. Zhang, D.J. Kouri, K. Haug, D.W. Schwenke, Y. Shima, and D.G. Truhlar, J. Chem. Phys. **88**, 2492 (1988).
5. L.L. Barnes, N.F. Lane, and C.C. Lin, Phys. Rev. **137**, A388 (1965).
6. A.C. Allison, J. Comput. Phys. **6**, 378 (1970).
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. D.G. Truhlar and N.A. Mullaney, J. Chem. Phys. **68**, 1574 (1978).
19. D.W. Schwenke, D. Thirumalai, D.G. Truhlar, and M.E. Coltrin, J. Chem. Phys. **78**, 3078 (1983).
20. P. Pechukas and J.C. Light, J. Chem. Phys. **44**, 3897 (1966).
21. 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.
22. B.C. Garrett, M.J. Redmon, D.G. Truhlar, and C.F. Melius, J. Chem. Phys. **74**, 412 (1981).
23. D.W. Schwenke and D.G. Truhlar, in *Supercomputer Simulations in Chemistry*, edited by M. Dupuis, Springer-Verlag, Berlin, 1986, pp. 165-197.
24. D.W. Schwenke and D.G. Truhlar, Theor. Chim. Acta **72**, 1 (1987).
25. D.W. Schwenke and D.G. Truhlar, J. Chem. Phys. **88**, 4800 (1988).
26. M.J. Unekis, D.G. Truhlar, and D.W. Schwenke, unpublished.