report on

The Application of the Discrete Variable Representation to the S-matrix Version of the Kohn Variational Principle for Quantum Reactive Scattering Calculations

G. C. Groenenboom

Institute of Theoretical Chemistry,
University of Nijmegen,
Toernooiveld 1, 6525 ED Nijmegen,
The Netherlands
gerritg@theochem.ru.nl

November 1992,
last updated October 2006

This work was supported by NWO

Foreword

This report describes research done on the application of the Discrete Variable Representation to the S-matrix version of the Kohn variational principle for quantum reactive scattering calculations. An outline of this work has been given in three quarterly reports (in Dutch) for the Netherlands Organization for Scientific Research (NWO).

This report contains a detailed description of the formalism that has been developed and also some of mathematical background of the Discrete Variable Representation, iterative methods for solving linear equations, and the Kohn variational principle. A publication which also contains results of the application of the method to three-atom systems is being prepared (G. C. Groenenboom, D. T. Colbert, and W. H. Miller, publication in preparation).

The work has been done at the University of California at Berkeley, CA, U.S.A. in the theoretical chemistry goup of Prof. W. H. Miller, from April 1991 - June 1992.

I gratefully acknowledge the financial support from the Eindhoven University of Technology (until October 1991), the Netherlands Organization for Scientific Research (NWO) (April 1992 - March 1992) and the Department of Chemistry of the UCB (May 1992 - June 1992).

The work was done in close collaboration with Dr. D. T. Colbert who was also a postdoc in the Miller group at the same time. It is a pleasure to thank him, prof. Miller and all the other members of the Miller group for the many lively discussions and the unforgettable time they have given me during my stay in Berkeley.

Contents

1 Introduction **3**

2 Mathematical background **5**
2.1 The Krylov space 5
2.2 The Lanczos recursion formula 7
2.3 Orthogonal polynomials 8
2.4 Gaussian integration 8
2.5 The Discrete Variable Representation 9
2.6 Sinc-function DVR 11
2.7 Iterative methods for linear systems 13
2.8 Preconditioning and GMRES 15

3 Reactive Scattering **17**
3.1 S-matrix version of the Kohn variational principle 17
3.2 The reactive case 19
3.3 The DVR applied to the S-matrix Kohn method 22
3.4 Implementation of $M\vec{c}$ 23
3.5 Distorted waves 26

Chapter 1

Introduction

Elastic and inelastic scattering problems can be solved with well established time-independent propagation methods [1]. In such methods the wave function (or its log-derivative) is numerically propagated along a scattering coordinate. For (in)elastic scattering problems the scattering coordinate is simply the distance between the centers of mass of the collision fragments. If a chemical reaction occurs during the collision the scattering coordinate will be different for every possible arrangement. The only way to use a propagation method in the reactive case is by transforming to hyperspherical coordinates [2].

Alternatively one can use a variational method which gives full flexibility in the choice of the coordinate system. In 1988 Zhang, Chu, and Miller [3] developed an efficient and convenient formalism based on the S-matrix version of the Kohn variational principle. In this formalism the calculation basically comes down to choosing a (real) basis set to represent the wave function in the interaction region, computing matrix elements of the Hamiltonian, and solving linear equations of the following form:

$$(\hat{H} - E)\vec{x} = \vec{b} \tag{1.1}$$

The dimension of the Hamiltonian matrix increases rapidly with the number of coordinates and the total energy. If the Hamiltonian needs to be stored in the fast memory of the computer in order to solve the linear equations there will be a certain limit on the size of the problem that can be treated. Furthermore, the cost of computing the potential energy matrix elements is large, because it requires evaluating multi-dimensional quadratures. To solve similar problems in bound state calcuations people have succesfully employed the Discrete Variable Representation (DVR) [4, 5]. In this report we show how the DVR can be used in scattering calculations too.

In the DVR the potential energy matrix is diagonal and the price of computing the Hamiltonian matrix elements is neglectable. As a result the Hamiltonian is very sparse and iterative methods can be used to solve the linear equations and the memory limitations are virtually gone.

In an iterative method a solution is being sought in the space spanned by the vectors $\{\vec{b}, \hat{H}\vec{b}, \hat{H}^2\vec{b}, \ldots\}$. In Chapter 2 we start by discussing the properies of this so called Krylov space. Coincidentally this theory can also be used to introduce the theory of orthogonal polynomials and Gaussian integration, which forms the basis of the DVR. For a more thorough, and still very clear presentation of Krylov spaces we refer to the book on symmetric eigenvalue

problems by Parlett [6]. In our discussion of Gaussian integration we follow Stoer and Bulirsch [7].

In Chapter 2 we present the theory of DVR in a perhaps somewhat unusual way, in order to unify the "conventional DVR" with a special type of DVR that was recently introduced by Colbert and Miller [8]. This DVR is based on an equally spaced grid and does not refer to an equilibrium structure, so it could be advantageous for radial coordinates in scattering problems.

In Chapter 3 we introduce the Kohn variational principle and show how to apply the DVR to it. The exploitation of the sparsity of the Hamiltonian in the iterative methods is not completely trivial so we also give some details on an efficient algorithm to compute the matrix-vector product $\hat{H}\vec{x}$.

The final topic is the computation of distorted waves which help to decrease the size of the interaction region and thus the dimension of the Hamiltonian matrix. We show how the right boundary conditions can be enforced by running a standard inelastic scattering propagator (the renormalized Numerov method) twice for two slightly different potentials and taking the proper linear combination of the solutions.

Chapter 2

Mathematical background

2.1 The Krylov space

For a linear operator A and a vector $\vec{q}$ the Krylov space [6] of dimension m is defined as:

$$\mathcal{K}^m(A, \vec{q}) = \text{span}\{\vec{q}, A\vec{q}, \ldots, A^{m-1}\vec{q}\} \tag{2.1}$$

First we note that a diagonal shift does not affect the Krylov space:

$$\mathcal{K}^m(A + \lambda I, \vec{q}) = \mathcal{K}^m(A, \vec{q}) \tag{2.2}$$

Furthermore we have:

$$\text{if} \quad \vec{x} \in \mathcal{K}^m \quad \text{then} \quad A\vec{x} \in \mathcal{K}^{m+1} \tag{2.3}$$

To demonstrate another important property of $\mathcal{K}^m(A, \vec{q})$ we consider a Hermitian operator A with n eigenvalues and eigenfunctions:

$$A\vec{u}_i = \lambda_i \vec{u}_i \tag{2.4}$$

and we assume:

$$|\lambda_1| < |\lambda_2| < \ldots < |\lambda_n| \tag{2.5}$$

We can now find the eigenvector $\vec{u}_n$ by using the power method: If $\vec{q}$ is an arbitrary vector with non-zero component in the direction of $\vec{u}_n$ then $A^m\vec{q}$ will converge to $\vec{u}_m$ with increasing m. This can be shown by writing $\vec{q}$ as a linear combination of eigenvectors:

$$\vec{q} = \sum_{i=1}^{n} q_i \vec{u}_i \tag{2.6}$$

This gives:

$$A^m\vec{q} = \sum_{i=1}^{n} q_i \lambda_i^m \vec{u}_i \tag{2.7}$$

$$= \lambda_n^m \Big(\sum_{i=1}^{n-1} \left[\frac{\lambda_i}{\lambda_n}\right]^m q_i\vec{u}_i + q_n\vec{u}_n\Big) \tag{2.8}$$

Since $|\frac{\lambda_i}{\lambda_n}| < 1$ for $i = 1, \ldots, n-1$ the relative contribution of $\vec{u}_n$ will grow with increasing m. By applying the power method to the diagonally shifted matrix $A - \lambda_n I$ we will find the eigenvector $\vec{u}_1$ since the shift turns the smallest eigenvalue into the largest. Since the Krylov space is invariant under diagonal shifts (see Eq. 2.2) we can now understand that a Krylov space tends to span the eigenvectors of A starting from both ends of the spectrum.

Mathematically, if A has n non-degenerate eigenvalues the Krylov space will contain the n corresponding eigenvectors after n iterations, if the starting vector has non-zero components for all eigenvectors. In finite precision calculations round-off errors usually cause the Krylov space to eventually contain all eigenvectors, even if initially $\vec{q}$ is orthogonal to a certain eigenvector.

If A has degenerate eigenvalues then mathematically the Krylov space will contain only one component of each degenerate subspace. If, e.g. $\lambda_i = \lambda_j$ then the Krylov space will only contain the component $q_i\vec{u}_i + q_j\vec{u}_j$.

An orthonormal basis for a Krylov space

We can define an orthonormal basis V^n for a Krylov space of dimension n with the property:

$$\text{span}\{V^m\} = \mathcal{K}^m(A, \vec{q}), \quad m = 1, 2, \ldots, n \tag{2.9}$$

where

$$V^m = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m\} \tag{2.10}$$

and

$$(\vec{v}_i, \vec{v}_j) = \delta_{ij} \tag{2.11}$$

This basis can be generated using Gram-Schmidt orthogonalization:

$$\beta_1\vec{v}_1 = \vec{q} \tag{2.12}$$

$$\beta_{i+1}\vec{v}_{i+1} = A\vec{v}_i - \sum_{j=1}^{i}(\vec{v}_j, A\vec{v}_i)\vec{v}_j, \quad i = 1, \ldots, n-1 \tag{2.13}$$

where $\beta_i > 0$ is chosen such that $|\vec{v}_i| = 1$ for $i = 1, \ldots, n$.

We define $\tilde{A}$ to be the matrix representation of A in the basis V:

$$\tilde{A}_{ij} = (\vec{v}_i, A\vec{v}_j) \quad \text{or} \quad \tilde{A} = V^\dagger A V \tag{2.14}$$

Because of equation Eqs. (2.3) and (2.9) we have $A\vec{v}_j \in V^{j+1}$ and so:

$$\tilde{A}_{i,j} = 0 \quad \text{for} \quad i > j + 1 \tag{2.15}$$

In other words $\tilde{A}$ is an upper Hessenberg matrix (in this 4×4 example the dots represent non-zero elements):

$$\tilde{A} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \\ & & \cdot & \cdot \end{bmatrix} \tag{2.16}$$

2.2 The Lanczos recursion formula

From Eq. (2.15) is follows that if A is Hermitian, i.e. $\tilde{A}_{i,j} = \tilde{A}^{\star}_{j,i}$, then $\tilde{A}$ has to be tridiagonal and so Eq. (2.13) goes over into the following three term recursion formula:

$$\beta_{i+1}\vec{v}_{i+1} = A\vec{v}_i - \tilde{A}_{i,i}\vec{v}_i - \tilde{A}_{i-1,i}\vec{v}_{i-1} \tag{2.17}$$

By taking the scalar product from the left with $\vec{v}_{i+1}$ we get:

$$\beta_{i+1} = (\vec{v}_{i+1}, A\vec{v}_i) = \tilde{A}_{i+1,i} \tag{2.18}$$

So we finally get:

$$\begin{aligned} \beta_1\vec{v}_1 &= \vec{q} && (2.19)\\ \beta_{i+1}\vec{v}_{i+1} &= A\vec{v}_i - \alpha_i\vec{v}_i - \beta^{\star}_i\vec{v}_{i-1} && (2.20)\\ \alpha_i &= (\vec{v}_i, A\vec{v}_i) && (2.21) \end{aligned}$$

and $\tilde{A}$ is a tridiagonal matrix:

$$\tilde{A} = \begin{bmatrix} \alpha_1 & \beta^{\star}_2 & & \\ \beta_2 & \alpha_2 & \beta^{\star}_3 & \\ & \beta_3 & \alpha_3 & \ddots \\ & & \ddots & \ddots \end{bmatrix} \tag{2.22}$$

A direct application of the above formulas is the calculation of the eigenvalues of A (the bisection algorithm can be used to obtain the eigenvalues from the tridiagonal matrix $\tilde{A}$ [6, 7]). Due to round off errors the vectors $\vec{v}_i$ generated by the Lanczos recursion formulas can strongly deviate from their exact values. This introduces spurious eigenvalues and multiple copies of eigenvalues in the spectrum of $\tilde{A}$, but some eigenvalues will still be very accurate and an easy way exists to identify the correct eigenvalues [9].

The Lanczos recursion formulas also form the basis of the SYMMLQ algorithm [10] for solving the set of linear equations $A\vec{x} = \vec{b}$. This application is discussed in section 2.7.

If A is not Hermitian we can still use Eq. (2.13) to turn A into an upper Hessenberg matrix. This is the basis of the GMRES algorithm for solving an arbitrary set of linear equations (see section 2.8).

The advantage of all these algorithms is that the matrix A only enters the maxtrix vector product $A\vec{v}_i$. This means that any sparsity or special structure in A can be fully exploited. The Lanczos recursion based algorithms have the additional advantage that only three columns of the matrix V^m need to be stored simultaneously.

Coincidentally the theory presented here can also be used tho introduce the theory of orthogonal polynomials, which we will need to define Gaussian quadratures. The latter form the basis of the Discrete Variable Representation (DVR). In the next section we will discuss orthogonal polynomials.

2.3 Orthogonal polynomials

For a given weight function $w(x) \geq 0$ on the interval $[a, b]$ (where a and b may be $\pm\infty$) we define the scalar product $(\vec{f}, \vec{g})$ between the real functions $f(x)$ and $g(x)$ as:

$$(\vec{f}, \vec{g}) = \int_a^b w(x) f(x) g(x) dx \tag{2.23}$$

For our purposes it will be convenient to use weight functions normalized such that:

$$\int_a^b w(x) dx = 1 \tag{2.24}$$

Now consider the Krylov space of the x-operator for the starting function $p_0(x) = 1$:

$$\Pi_j = \{1, x, x^2, \ldots, x^{j-1}\} \tag{2.25}$$

Since x is a real symmetric operator $(\vec{f}, x\vec{g}) = (\vec{g}, x\vec{f})$ we can use the Lanczos recursion formulas to generate the orthogonal polynomials $\{p_0(x), p_1(x), \ldots\}$ corresponding to a certain weight function. For example, the weight function $\pi^{-\frac{1}{2}} e^{-x^2}$ on the interval $[-\infty, +\infty]$ will give the Hermite polynonials and the constant function on the interval $[-1, 1]$ will give the Legendre polynonimals.

Note that we included the normalization in the Lanczos recursion formulas so we have $(p_i, p_j) = \delta_{ij}$. This differs from Stoer and Bulirsch [7, section 3.6] who give the recursion formulas that give a normalization consistent with Abramowitz and Stegun [11, Chap. 22].

2.4 Gaussian integration

Consider the integral

$$I[f] = \int_a^b w(x) f(x) dx \tag{2.26}$$

where $w(x)$ is a weight function on the interval $[a, b]$. With the definitions of the previous section this integral is just the following scalar product:

$$I[f] = \int_a^b w(x) p_0(x) f(x) dx = (\vec{p_0}, \vec{f}) \tag{2.27}$$

An n-point quadrature approximation to the integral is defined by n points $\{x_1, \ldots, x_n\}$ and weights $\{w_1, \ldots, w_n\}$:

$$\tilde{I}[f] = \sum_{i=1}^{n} w_i f(x_i) \tag{2.28}$$

The idea of Gaussian integration is that since there are $2n$ parameters in the quadrature it is possible to make the integration exact for the first $2n$ orthogonal polynomials (and, since the quadrature formula is linear, any linear combination thereof):

$$I[p_j] = (\vec{p_0}, \vec{p_j}) = \delta_{0,j} \quad \text{for} \quad j = 0, \ldots, 2n-1 \tag{2.29}$$

For an arbitrary set of quadrature points we can make the integration exact for the first n orthogonal polynomials by choosing the weights as the solution of the following set of linear equations:

$$\begin{bmatrix} p_0(x_1) & \dots & p_0(x_n) \\ p_1(x_1) & \dots & p_1(x_n) \\ \vdots & & \vdots \\ p_{n-1}(x_1) & \dots & p_{n-1}(x_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.30}$$

It turns out that in order to make the quadrature exact up to polynomials of degree $2n-1$ we have to take as the quadrature points the zeros of $p_n(x)$. To show this we first note that:

$$\text{span}\{\vec{p}_0, \dots, \vec{p}_{2n-1}\} = \text{span}\{\vec{p}_0, \dots, \vec{p}_n, x\vec{p}_n, x^2\vec{p}_n, \dots, x^{n-1}\vec{p}_n\} \tag{2.31}$$

So we can write every $f(x) \in \Pi_{2n-1}$ as:

$$f(x) = r(x) + q(x)p_n(x) \quad \text{where} \quad r, q \in \Pi_{n-1} \tag{2.32}$$

and for the integral we get:

$$I[f] = (\vec{p}_0, \vec{r}) + (\vec{q}, \vec{p_n}) \tag{2.33}$$

We have already shown that the quadrature is exact for $(\vec{p}_0, \vec{r})$. The second term the quadrature will give zero because $p_n(x_i) = 0, \quad i = 1, \dots, n$. But this is exactly what it should be since $\vec{q} \in \Pi_{n-1}$ and so $(\vec{q}, \vec{p}_n) = 0$.

We have now established that an n-point Gaussian quadrature is exact for polynomials up to degree $2n-1$. This means that the overlap (the scalar product) of two polynomials of degree $n-1$ can be calculated exactly by a discrete sum:

$$(\vec{p}_i, \vec{p}_j) = \sum_{k=1}^{n} w_k p_i(x_k) p_j(x_k), \quad i, j = 0, \dots, n-1 \tag{2.34}$$

The quadrature is also exact for the matrix elements of the x-operator:

$$(\vec{p}_i, x\vec{p}_j) = \sum_{k=1}^{n} w_k p_i(x_k) x_k p_j(x_k), \quad i, j = 0, \dots, n-1 \tag{2.35}$$

We will use those two properties in the following section.

2.5 The Discrete Variable Representation

We will now apply the theory presented so far to a bound-state problem in quantum mechanics. First, we want to eliminate the weight function from the definition of the scalar product so we absorb the square root of the weight into the functions:

$$\phi_i(x) = \sqrt{w(x)}\; p_{i-1}(x), \quad i = 1, \dots, n \tag{2.36}$$

For later reference we will call this the Variational Basis set. The Gaussian quadrature for these functions, which is again exact for the overlap of these functions, has adjusted weights w'_k:

$$(\phi_i, \phi_j) \equiv \int \phi_i(x)\phi_j(x)dx = \delta_{ij} = \sum_{k=1}^{n} w'_k \phi_i(x_k)\phi_j(x_k) \tag{2.37}$$

So

$$w'_k = \frac{w_k}{w(x_k)} \tag{2.38}$$

We now define the discrete representation of $\phi_i(x)$ as $\vec{u}_i$:

$$(\vec{u}_i)_k \equiv U_{ki} = \sqrt{w'_k}\ \phi_i(x_k) \tag{2.39}$$

So U is a $n \times n$ unitary matrix:

$$(\phi_i, \phi_j) = (\vec{u}_i, \vec{u}_j) = \sum_{k=1}^{n} U_{ki} U_{kj} = \delta_{ij} \tag{2.40}$$

We will now define a new basis set which we will label by DVR and which is related to the old basis set by a unitary transformation:

$$\phi_k^{\mathrm{DVR}}(x) = \sum_{i=1}^{n} U_{ki} \phi_i(x) \tag{2.41}$$

The function $\phi_k^{\mathrm{DVR}}(x)$ is zero at all the quadrature points, except x_k:

$$\begin{aligned} \phi_k^{\mathrm{DVR}}(x_j) &= \sum_i U_{ki} \phi_i(x_j) & (2.42)\\ &= \frac{1}{\sqrt{w'_j}} \sum_i U_{ki} U_{ji} = \frac{1}{\sqrt{w'_j}} \delta_{kj} & (2.43) \end{aligned}$$

where we used the fact that the transpose of a unitary matrix is also unitary (Eq. 2.40 gives the scalar product between columns of U, here we have a scalar product between rows). The idea of the DVR approximation is to use this basis set together with the quadrature approximation for the *potential energy matrix elements*. The resulting potential energy matrix is diagonal with the values of the potential at the quadrature points on the diagonal:

$$\begin{aligned} V_{ij}^{\mathrm{DVR}} &= \sum_{k=1}^{n} w'_k \phi_i^{\mathrm{DVR}}(x_k) V(x_k) \phi_j^{\mathrm{DVR}}(x_k) & (2.44)\\ &= \sum_{k=1}^{n} \delta_{ik} V(x_k) \delta_{jk} = V(x_i) \delta_{ij} & (2.45) \end{aligned}$$

The kinetic energy matrix in the DVR basis is related to the kinetic energy matrix in the polynomial basis by a unitary transformation:

$$\begin{aligned} T_{kl}^{\mathrm{DVR}} &= (\phi_k^{\mathrm{DVR}}, \hat{T} \phi_l^{\mathrm{DVR}}) & (2.46)\\ &= \sum_{ij} U_{ki} (\phi_i, \hat{T} \phi_j) U_{lj} & (2.47) \end{aligned}$$

or in matrix notation:

$$T^{\mathrm{DVR}} = U T^{\mathrm{VBR}} U^T \tag{2.48}$$

where VBR stands for Variational Basis set Representation.

It is interesting to note that in the DVR-basis the $\hat{x}$ operator is a diagonal matrix with the quadrature points x_k on the diagonal:

$$X_{ij}^{\mathrm{DVR}} = (\phi_i^{\mathrm{DVR}}, \hat{x}\phi_j^{\mathrm{DVR}}) = x_i \delta_{i,j} \tag{2.49}$$

But since the quadrature is exact for matrix elements of the x operator (see Eq. 2.35) we could actually define the DVR-functions as the eigenfunctions of the $\hat{x}$ operator (in the n dimensional Variational Basis set) and the quadrature points as the corresponding eigenvalues. This is actually how in practice the quadrature points are found: one just diagonalizes the tridiagonal matrix corresponding to the representation of the $\hat{x}$ operator in the basis of orthogonal polynomials. Furthermore, the quadrature weights can be found with the following recipe: write the eigenfunctions of the $\hat{x}$ operator (the DVR-functions) in the basis of orthogonal polynomials; the square of the first component of ϕ_k^{DVR} is the weight of point k:

$$\begin{aligned} |(\phi_1, \phi_k^{\mathrm{DVR}})|^2 &= |\sum_l w_l' \phi_1(x_l) \phi_k^{\mathrm{DVR}}(x_l)|^2 && (2.50) \\ &= |w_k' \sqrt{w(x_k)}\ \frac{1}{\sqrt{w_k'}}|^2 && (2.51) \\ &= w_k && (2.52) \end{aligned}$$

2.6 Sinc-function DVR

We will now derive in an alternative fashion a DVR that was first introduced by Colbert and Miller [8]. To show the similarity between this DVR and a DVR that is based on a set of orthogonal polynomials, we will first define a set of DVR-functions and show that we have a quadrature that is exact for the scalar product between these functions. It is most convenient to define the sinc-function basis set $\{|n>, n = -\infty, \ldots, \infty\}$ in frequency (ω) space and then use a Fourier transform to get the x-representation of the basis set.

In ω-space we have a Fourier basis on the interval $-\omega_{\max} \ldots \omega_{\max}$:

$$\phi_n(\omega) =< \omega|n >= \begin{cases} \frac{1}{\sqrt{2\omega_{\max}}} e^{in\pi \frac{\omega}{\omega_{\max}}} & \text{for } |\omega| \leq \omega_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{2.53}$$

After Fourier transformation we obtain a basis that is complete for all band-limited functions, i.e. functions that have no frequency components above a certain frequency $\omega_{\max}$:

$$\begin{aligned} \phi_n(x) =< x|n > &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\omega x} < \omega|n > d\omega && (2.54) \\ &= \frac{1}{2\sqrt{\pi\omega_{\max}}} \int_{-\omega_{\max}}^{\omega_{\max}} e^{-i\omega x} e^{in\pi \frac{\omega}{\omega_{\max}}} d\omega && (2.55) \\ &= \sqrt{\frac{\omega_{\max}}{\pi}} \mathrm{sinc}(\omega_{\max} x - n\pi) && (2.56) \end{aligned}$$

Where

$$\mathrm{sinc}(x) \equiv \frac{\sin(x)}{x} \tag{2.57}$$

So with

$$\omega_{\max} = \frac{2\pi}{\lambda} = \frac{\pi}{\Delta} \tag{2.58}$$

we get:

$$\phi_n(x) = \frac{1}{\sqrt{\Delta}}\text{sinc}[\pi(\frac{x}{\Delta} - n)] \tag{2.59}$$

Since $\text{sinc}(n\pi) = \delta_{n,0}$ it is easy to verify that the quadrature $x_i = i\Delta, \quad i = -\infty, \ldots, +\infty$ and weights $w_i = \Delta$ is exact for the overlap integrals:

$$\begin{aligned} < n|m > &= \int \phi_n(x)\phi_m(x)dx & (2.60)\\ &= \sum_{i=-\infty}^{\infty} \Delta \frac{\text{sinc}[\pi(i-n)]}{\sqrt{\Delta}} \frac{\text{sinc}[\pi(i-m)]}{\sqrt{\Delta}} & (2.61)\\ &= \sum_{i=-\infty}^{\infty} \delta_{i,n}\delta_{i,m} = \delta_{n,m} & (2.62) \end{aligned}$$

To calculate matrix elements of the $\frac{\partial}{\partial x}$ and $\frac{\partial^2}{\partial x^2}$ operators we can use the fact that the derivatives of band limited functions are also band limited, i.e. if

$$f(x) = \int_{-\infty}^{\infty} h(\omega)e^{i\omega x}d\omega \quad \text{with} \quad h(\omega) = 0 \quad \text{for} \quad |\omega| > \omega_{\max} \tag{2.63}$$

then

$$\begin{aligned} \frac{\partial^n}{\partial x^n} f(x) &= \int_{-\infty}^{\infty} h(\omega)(i\omega)^n e^{i\omega x}d\omega & (2.64)\\ &= \int_{-\infty}^{\infty} g(\omega)e^{i\omega x}d\omega & (2.65) \end{aligned}$$

and so the derivatives are also band limited, with the same bandwidth:

$$g(\omega) = h(\omega)(i\omega)^n = 0 \quad \text{for} \quad |\omega| > \omega_{\max} \tag{2.66}$$

This means that we can use the quadrature to evaluate the matrix elements:

$$\begin{aligned} < n|\frac{\partial^k}{\partial x^k}|m > &= \int_{-\infty}^{\infty} dx \frac{1}{\Delta}\text{sinc}[\pi(\frac{x}{\Delta} - n)]\frac{\partial^k}{\partial x^k}\text{sinc}[\pi(\frac{x}{\Delta} - m)] & (2.67)\\ &= \sum_{i=-\infty}^{\infty} \text{sinc}[\pi(i-n)]\frac{\partial^k}{\partial x^k}\text{sinc}\,[\pi(\frac{x}{\Delta} - m)]\Big|_{x=i\Delta} & (2.68)\\ &= \frac{\partial^k}{\partial x^k}\text{sinc}\,(\pi\frac{x}{\Delta})\Big|_{x=(n-m)\Delta} & (2.69) \end{aligned}$$

This gives for $k = 1$:

$$< n|\frac{\partial}{\partial x}|m >= \begin{cases} 0 & n = m \\ \frac{1}{\Delta}\frac{(-1)^{n-m}}{n-m} & n \neq m \end{cases} \tag{2.70}$$

and for $k = 2$ we get:

$$< n|\frac{\partial^2}{\partial x^2}|m >= \begin{cases} -\frac{1}{3}\frac{\pi^2}{\Delta^2} & n = m \\ -\frac{2}{\Delta^2}\frac{(-1)^{n-m}}{(n-m)^2} & n \neq m \end{cases} \tag{2.71}$$

Note that for the diagonal term we need the second derivative of the sinc function at $x = 0$, which is most easily done by using Taylor series:

$$\frac{\partial^2}{\partial x^2}\left.\frac{\sin(x)}{x}\right|_{x=0} = \frac{\partial^2}{\partial x^2}\left.\frac{x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \ldots}{x}\right|_{x=0} = -\frac{1}{3} \tag{2.72}$$

If we have the boundary condition $\psi(0) = 0$ we can use (for $n = 1, 2, \ldots, \infty$):

$$\phi_n^-(x) =< x|n^- > = \phi_n(x) - \phi_n(-x) \tag{2.73}$$

$$= \frac{1}{\sqrt{\Delta}}\left\{\text{sinc}[\pi(\frac{x}{\Delta} - n)] - \text{sinc}[\pi(\frac{x}{\Delta} + n)]\right\} \tag{2.74}$$

This basis is orthonormal on the range $0 \ldots \infty$ and:

$$\phi_n^-(x_i) = \frac{1}{\sqrt{\Delta}}(\delta_{i,n} - \delta_{i,-n}) = \frac{1}{\sqrt{\Delta}}\delta_{i,n}, \quad (i, n > 0) \tag{2.75}$$

So we have the quadrature points $x_i = i\Delta, \quad i = 1, 2, \ldots, \infty$ and weights $w_i = \Delta$.

For the matrix elements of the $\frac{\partial}{\partial x}$ operator we can not use the quadrature since:

$$\frac{\partial}{\partial x}\phi_n^-(x)|_{x=0} \neq 0 \tag{2.76}$$

but for $\frac{\partial^2}{\partial x^2}$ we get:

$$< n^-|\frac{\partial^2}{\partial x^2}|m^- > = \sum_{k=1}^{\infty}\{\text{sinc}[\pi(k-n)] - \text{sinc}[\pi(k+n)]\}\frac{\partial^2}{\partial x^2}\phi_m^-(x_k) \tag{2.77}$$

$$= \frac{\partial^2}{\partial x^2}\{\text{sinc}[\pi(\frac{x}{\Delta} - m)] - \text{sinc}[\pi(\frac{x}{\Delta} + m)]\}_{x=n\Delta} \tag{2.78}$$

So:

$$< n^-|\frac{\partial^2}{\partial x^2}|m^- >= \begin{cases} -\frac{1}{3}\frac{\pi^2}{\Delta^2} + \frac{2}{\Delta^2}\frac{(-1)^{n+m}}{(n+m)^2} & n = m \\ -\frac{2}{\Delta^2}\left[\frac{(-1)^{n-m}}{(n-m)^2} - \frac{(-1)^{n+m}}{(n+m)^2}\right] & n \neq m \end{cases} \tag{2.79}$$

2.7 Iterative methods for linear systems

We are interested in finding a vector $\vec{x}$ that makes the length of the residual $|\vec{r}|$:

$$\vec{r} = \vec{b} - A\vec{x} \tag{2.80}$$

smaller than a certain threshold (A is an $n \times n$ matrix). In an iterative method we try a solution of the form:

$$\vec{x}_m = \sum_{i=1}^{m} c_i\vec{v}_i = V_m\vec{c} \tag{2.81}$$

Typically m is increased by 1 every iteration until convergence.

Substituting Eq. (2.81) into Eq. (2.80) gives:

$$\vec{r} = \vec{b} - AV_m\vec{c}_m \tag{2.82}$$

This is a set of n equations with m unknowns. One way of finding the coefficients $\vec{c}$ is by requiring $V_m^T\vec{r}$, i.e. the *projection* of the residual onto span$\{V_m\}$ to be zero. This gives the following $m \times m$ system:

$$V_m^T A V_m \vec{c} = V_m^T \vec{b} \tag{2.83}$$

If A is Hermitian and we generate V_m using the Lanczos recursion this becomes a tridiagonal system. This is the basis of the SYMMLQ algorithm of Paige and Saunders ([10]). In this algorithm LQ decomposition (L is a lower triangular matrix and Q an orthogonal matrix) is used to solve the tridiagonal system. The implementation is such that every iteration both the LQ decomposition and a solution vector x_m are updated and the V_m matrix need not be stored. Only three vectors of working space are needed. It is important to note that the vectors $\vec{v}_i$ generated by the Lanczos recusion can strongly deviate from their exact values due to round-off errors. This can actually cause the vectors $\vec{v}_i$ to become linearly dependent. This effect is well understood and it does not invalidate the final solution $\vec{x}$. However, it does increase the number of iterations needed to get a certain accuracy.

In the SYMMLQ algorithm it is important to use the right hand side $\vec{b}$ as the starting vector for the Lanczos recursion. If a good initial guess $\vec{x}_0$ for the solution $\vec{x}$ is known one can substitute:

$$\vec{x} = \vec{x}_0 + \vec{y} \tag{2.84}$$

and solve for $\vec{y}$:

$$A\vec{y} = \vec{b} - A\vec{x}_0 = \vec{b}' \tag{2.85}$$

where $\vec{b}'$ is used as the starting vector.

The SYMMLQ algorithm does *not* produce the solution $\vec{x}$ that for a given number of iterations gives the smallest possible residual. This can be easily understood since it is the *projection* of the residual that is being minimized, and not the residual itself. The latter can be achieved by the MINRES algorithm:

From Eq. (2.80) it is clear that the miminal residual is the distance between the vector $\vec{b}$ and a plane W spanned by the vectors $\{A\vec{v}_1, \ldots, A\vec{v}_m\}$. So an equation for the minimal residual solution can be found by requireing $\vec{r}$ to be orthogonal to the plane W:

$$\begin{aligned} (AV_m)^T\vec{r} &= 0 & (2.86)\\ V_m^T A^2 V_m \vec{c}'_m &= V_m^T A\vec{b} & (2.87) \end{aligned}$$

where the minimal residual solution is:

$$\vec{x}'_m = V_m\vec{c}'_m \tag{2.88}$$

Paige and Saunders [10] also describe an efficient implementation for the solution of these equations (the MINRES algorithm). They report that MINRES seems to be a little less accurate then SYMMLQ for ill conditioned problems but sometimes requires fewer itrations.

2.8 Preconditioning and GMRES

Preconditioning is a way to reduce the number of iterations required. The idea is to multiply the linear system from the left with the inverse of a preconditioner P and try to solve:

$$P^{-1}A\vec{x} = P^{-1}\vec{b} \tag{2.89}$$

Here $P^{-1}A$ is never constructed but every iteration we calculate $\vec{y} = P^{-1}A\vec{x}$ in two steps:

$$\vec{z} = A\vec{x} \tag{2.90}$$

$$P\vec{y} = \vec{z} \tag{2.91}$$

Clearly, if $P = A$ one iteration will give the exact solution. However solving Eq. (2.91) would be just as hard as the original problem. So the idea is to try to find as a preconditioner a good approximation to A for which Eq. (2.91) can be easily solved, but which still reduces the number of iterations as much as possible.

In general we can not use the SYMMLQ algorithm to solve the preconditioned system, since $P^{-1}A$ need not be Hermitian, even if both P and A are. This is why preconditioning is used in combination with the GMRES algorithm, which can handle arbitrary matrices.

The GMRES algorithm

In section 2.1 we have shown that in order to generate a Gram-Schmidt orthogonalized basis in a Krylov space of a non-Hermitian matrix we need to project every new vector $A\vec{v}_m$ against all previous vectors in V_m. This is exactly what is being done in the Generalized Minimal Residual method. Obviously the memory requirements for GMRES are larger than for SYMMLQ since every iteration one needs to store one additional vector. Furthermore the overhead for the Gram-Schmidt process increases every iteration.

The algorithm is rather straightforward. Again $\vec{x}$ is written as:

$$\vec{x} = V_m\vec{c} \tag{2.92}$$

and the residual is:

$$\vec{r} = \vec{b} - AV_m\vec{c} \tag{2.93}$$

Since V_n (where n is the dimension of A) is a complete orthonormal basis, $|\vec{r}|$ can be minimized by minimizing $|V_n^T\vec{r}|$. This yields the following $(n \times m)$ least squares problem:

$$V_n^T AV_m\vec{c} = V_n^T\vec{b} \tag{2.94}$$

As shown in section 2.1 $H_m = V_n^T AV_m$ is an upper Hessenberg matrix. If we take $\vec{v}_1 = \vec{b}$ we get the following $(m+1) \times m$ least squares problem $[(\vec{e}_1)_i = \delta_{i,1}]$:

$$H_m\vec{c} = \vec{e}_1|\vec{b}| \tag{2.95}$$

A least squares problem is best solved by QR decomposition:

$$Q_m^T H_m\vec{c} = R_m\vec{e}_1|\vec{b}| \tag{2.96}$$

Q_m^T can be written as a product of 2×2 rotation matrices:

$$Q_m^T = Q_{m-1,m}^T Q_{m-2,m}^T \dots Q_{2,1}^T \tag{2.97}$$

where every iteration $Q_{m-1,m}$ is chosen to eliminate the off-diagonal element of H_m at position $(m+1, m)$. Every iteration the residual is available as the last component of $R\vec{e}_1|\vec{b}|$. Upon convergence the solution x can be calculated from Eq. (2.92), which is no problem since V_m had to be stored anyway.

Note that since in GMRES the Gram-Schmidt orthogonalization is done explicitly the set V_m is almost perfectly orthonormal, in contrast to what happens in the SYMMLQ algorithm.

If V_m becomes too big to be stored before convergence is reached, the algorithm can be restarted, as shown in Eqs. (2.84) and (2.85). However this often gives a huge increase in the total number of iterations needed. However the algorithms should eventually converge since the residual is bound to decrease every iteration.

Chapter 3

Reactive Scattering

3.1 S-matrix version of the Kohn variational principle

In this section we will introduce the S-KVP (see e.g.[3, 12]) for an elastic scattering problem. Following Zhang et al. we will adopt the convention that the wave function in the bra symbol is not complex conjugated, i.e.:

$$< f|g >= \int_0^\infty f(r)g(r)dr \tag{3.1}$$

First we will show a variational principle for a bound state (L^2) wave function. Consider the integral:

$$I =< \psi|\hat{H} - E|\psi > \tag{3.2}$$

where $\hat{H}$ is a one dimensional Hamiltonian:

$$\hat{H} = \hat{T} + \hat{V} = -\frac{\hbar^2}{2\mu}\frac{\partial^2}{\partial r^2} + V(r) \tag{3.3}$$

and

$$\psi(0) = 0 \tag{3.4}$$

A first order variation in ψ ($\psi \to \psi + \delta\psi$) gives (keeping only terms up to the first order in $\delta\psi$):

$$\delta I =< \delta\psi|\hat{H} - E|\psi > + < \psi|\hat{H} - E|\delta\psi >= 2 < \delta\psi|\hat{H} - E|\psi > \tag{3.5}$$

If δI is zero for all possible variations $\delta\psi$ we get:

$$(\hat{H} - E)|\psi >= 0 \tag{3.6}$$

In this derivation we used:

$$< \psi|\hat{H} - E|\delta\psi >=< \delta\psi|\hat{H} - E|\psi > \tag{3.7}$$

For the multiplicative (the potential energy) terms in $\hat{H}$ this evidently holds. For the kinetic energy term in $\hat{H}$ it holds if the surface term vanishes:

$$\int_0^\infty fg''dr = [fg' - f'g]_0^\infty + \int_0^\infty f''gdr \tag{3.8}$$

This will always be the case if the wave function goes to zero at large r. Of course this does not hold for scattering wave functions.

To derive a variational principle for a scattering wave function we start with a trial wave function with S-matrix boundary conditions:

$$\tilde{\psi}(r) = -u_0(r) + u_1(r)\tilde{S} + \sum_i \phi_i(r)c_i \tag{3.9}$$

The first term is an incoming wave of unit flux asymptotically, i.e. at large r:

$$u_0(r) = v^{-\frac{1}{2}} e^{-ikr} \tag{3.10}$$

where $v = \hbar k/\mu$ and $k = \sqrt{2\mu E}\ /\hbar$. The factor $v^{-\frac{1}{2}}$ normalizes the incoming flux to unity. At $r = 0$, $u_0(r)$ has to be regular. This can be achieved with a cutoff function:

$$u_0(r) = v^{-\frac{1}{2}} e^{-ikr} f(r) \tag{3.11}$$

where $f(0) = 0$ and for $r \to \infty$, $f(r) \to 1$.

The second term consists of an outgoing wave:

$$u_1(r) = u_0(r)^\star \tag{3.12}$$

and the trial 1×1 S-matrix ($\tilde{S}$).

The first two terms of Eq. (3.9) are the analytic solution of Schrödinger equation at large r (where $V(r) = 0$) The third term is a linear combination of L^2 basis functions which gives the flexibility of the trial wave function in the interaction region.

For $\delta\tilde{\psi}(r)$ we have:

$$\delta\tilde{\psi}(r) = u_1(r)\delta\tilde{S} + \sum_i \phi_i(r)\delta c_i \tag{3.13}$$

So for this scattering trial wave function the first order variation in the integral (Eq. 3.2) is (using Eq. 3.8):

$$\delta < \tilde{\psi}|\hat{H} - E|\tilde{\psi} >= 2 < \tilde{\psi}|\hat{H} - E|\tilde{\psi} > -\frac{\hbar}{i}\delta\tilde{S} \tag{3.14}$$

Defining Γ:

$$\Gamma = \tilde{S} + \frac{i}{\hbar} < \tilde{\psi}|\hat{H} - E|\tilde{\psi} > \tag{3.15}$$

we get:

$$\delta\Gamma = \frac{2i}{\hbar} < \delta\tilde{\psi}|\hat{H} - E|\tilde{\psi} > \tag{3.16}$$

so the variational condition is:

$$\delta\Gamma = 0 \tag{3.17}$$

If the trial wave function $\tilde{\psi}$ is the exact solution of $(\hat{H} - E)|\psi >= 0$ then the S-matrix $S = \tilde{S} = \Gamma$. However if $\tilde{\psi}$ and $\tilde{S}$ are off to the first order we still have

$$S = \Gamma \tag{3.18}$$

Figure 3.1: Jacobi coordinates

correct to the first order since $\delta\Gamma = 0$. Eqs. (3.17) and (3.18) are usually summerized as:

$$S = \text{ext}\left[\tilde{S} + \frac{i}{\hbar} < \tilde{\psi}|\hat{H} - E|\tilde{\psi} >\right] \tag{3.19}$$

We can obtain an algebraic equation for the S-matrix by plugging the trial wave function (Eq. 3.9) into Eq. (3.15), setting the derivatives with respect to $\tilde{S}$ and c_i equal to zero (Eq. 3.17) and plugging the formal solution of the resulting set of linear equations into Eq. (3.18). We will not reproduce the formulas for the elastic scattering case [3], but go directly to the three atom (3D, J=0) reactive scattering case.

3.2 The reactive case

Jacobi coordinates

First we introduce the coordinate system used to describe the asymptotic regions. For a three atom system there are three atom-diatom arrangements, A+BC, B+CA, and C+AB. The Jacobi coordinates corresponding to an arrangement are labeled by the separate atom (γ =A, B, or C) and consist of the distance between the atom and the center of mass of the diatom (R_γ), the diatom bond length (r_γ) and an angle (θ_γ) as shown in Fig. (3.1).

The Hamiltonian

The Hamiltonian expressed in Jacobi coordinates of one of the arrangements is:

$$\hat{H} = \hat{T}_{R_\gamma}\frac{\hbar^2}{2\mu_\gamma R_\gamma}\hat{j}^2 + \Delta\hat{V}_\gamma + \hat{H}_{\text{int},\gamma} \tag{3.20}$$

where

$$\hat{T}_{R_\gamma} = -\frac{\hbar^2}{2\mu_\gamma}\frac{1}{R_\gamma}\frac{\partial^2}{\partial R_\gamma^2}R_\gamma \tag{3.21}$$

$$\Delta\hat{V}_\gamma = V_\gamma(R_\gamma, r_\gamma, \theta_\gamma) - V_\gamma(r_\gamma) \tag{3.22}$$

$$V_\gamma(r_\gamma) = \lim_{R_\gamma \to \infty} V(R_\gamma, r_\gamma, \theta_\gamma) \tag{3.23}$$

$$\hat{H}_{\mathrm{int},\gamma} = -\frac{\hbar^2}{2m_\gamma}\frac{1}{r_\gamma}\frac{\partial^2}{\partial r_\gamma^2} r_\gamma + \frac{\hbar^2}{2m_\gamma r_\gamma^2}\hat{j}^2 + V_\gamma(r_\gamma) \tag{3.24}$$

Note that the Euler angles specifying the orientation of the coordinate system never show up here because they can be integrated out analytically and for the $J = 0$ that leaves no resulting terms in the Hamiltonian.

The trial wavefunction

First we will define the channel eigenfunctions, i.e. the ro-vibrational eigenfunctions of the diatom. The channel eigenfunctions, labeled by the arrangement (γ), the vibrational quantum number v and the rotational quantum number j, are defined by the following equation:

$$\left[\hat{H}_{\mathrm{int},\gamma} - \epsilon_{\gamma v j}\right] \phi_{\gamma v j}(r_\gamma, \theta_\gamma) = 0 \tag{3.25}$$

The eigenfunctions are separable and can be written as:

$$\phi_{\gamma v j}(r_\gamma, \theta_\gamma) = \frac{\chi_{\gamma v j}(r_\gamma)}{r_\gamma} P_j[\cos(\theta_\gamma)] \tag{3.26}$$

The Legendre functions P_j are eigenfunctions of the $\hat{j}^2$ operator:

$$\hat{j}^2 P_j(\cos(\theta_\gamma)) = j(j+1) P_j(\cos(\theta_\gamma)) \tag{3.27}$$

If we substitute this into Eq. (3.25) and multiply from the left with r_γ we get for the radial part of the vibrational functions $\chi_{\gamma v j}(r_\gamma)$:

$$\left[-\frac{\hbar^2}{2m_\gamma}\frac{\partial^2}{\partial r_\gamma^2} + \frac{j(j+1)}{2m_\gamma r_\gamma^2} + V_\gamma(r) - \epsilon_{\gamma v j}\right] \chi_{\gamma v j}(r_\gamma) = 0 \tag{3.28}$$

From now on we will use the composite index n to label the channels ($\gamma v j$).

For a total energy E the translational energy for channel n is:

$$\frac{\hbar^2 k_n^2}{2\mu_\gamma} = E - \epsilon_n \tag{3.29}$$

Spherical Hankel functions are incoming waves that asymptotically solve the Schrödinger equation:

$$\left[\hat{T}_{R_\gamma} + \frac{\hbar^2 j(j+1)}{2\mu_\gamma R_\gamma^2} - \frac{\hbar^2 k_n^2}{2\mu_\gamma}\right] h_j^{(2)}(k_n R_\gamma) = 0 \tag{3.30}$$

To regularize the incoming wave at the origin one can use a cutoff function:

$$\frac{u_n(R_\gamma)}{R_\gamma} = -i k_n h_j^{(2)}(k_n R_\gamma) v_n^{-\frac{1}{2}} f(R_\gamma) \tag{3.31}$$

The phase $-i$ is chosen to get the asymptotic form:

$$u_n(R) \sim e^{-i[k_n R - \pi l/2]} v_n^{-\frac{1}{2}} \tag{3.32}$$

As in equation (3.10) the factor $v_n^{-\frac{1}{2}}$ normalizes the flux. The cutoff function $f(R)$ should go smoothly from $f(0) = 0$ to $f(\infty) = 1$. In section 3.5 we give an alternative way of choosing the incoming wave.

So finally we can define the trial wave function:

$$\begin{aligned} \psi_n &= -\frac{u_0(R_\gamma)}{R_\gamma}\phi_n(r_\gamma,\theta_\gamma) \\ &\quad + \sum_{n'} \frac{u_1(R_{\gamma'})}{R_{\gamma'}}\phi_{n'}(r_{\gamma'},\theta_{\gamma'})\tilde{S}_{n',n} \\ &\quad + \sum_i \chi_i(\vec{x})c_i \end{aligned} \tag{3.33}$$

The first term is an incoming wave in channel n. The second term is a sum of outgoing waves in all open channels (and all arrangements). The last term is a linear combination of L^2 basis functions in the - as yet unspecified - interation region coordinates $\vec{x}$.

One possible choice for $\vec{x}$ is Jacobi coordinates of one of the arrangements, e.g. $\vec{x} = (R_B, r_B, \theta_B)$. Here we deviate somewhat from the original approach of Zhang et al. [13] who used a multi-centered expansion for the L^2 functions. In a multi-centered expansion L^2 basis functions are specified in all three sets of Jacobi coordinates. As a result complicated integrals have to be evaluated for matrix elements involving basis functions specified in coordinates of different arrangements. The reason we abandoned this approach is that we want to use a DVR which is only straightforward if a direct product basis is used in a single coordinate system. Instead of Jacobi coordinates of one of the arrangements we can also use e.g. valence (bond lengths, bond angle), Radau or transition state normal coordinates.

Now that we have defined the trial wave function we can follow the recipe given in the previous section and in [3] to obtain the following equations for the S-matrix:

$$S = \frac{i}{\hbar}(B - C^T B^{\star -1} C) \tag{3.34}$$

$$B = M_{0,0} - M_0^T M^{-1} M_0 \tag{3.35}$$

$$C = M_{1,0} - M_0^\dagger M^{-1} M_0 \tag{3.36}$$

where

$$(M_{0,0})_{n,n'} = < \frac{u_{0n}(R_\gamma)}{R_\gamma}\phi_n|\hat{H} - E|\frac{u_{0n'}(R_{\gamma'})}{R_{\gamma'}}\phi_{n'} > \tag{3.37}$$

$$(M_{1,0})_{n,n'} = < \frac{u_{1n}}{R_\gamma}\frac{\phi_n}{r_\gamma}|\hat{H} - E|\frac{u_{0n'}(R_{\gamma'})}{R_{\gamma'}}\phi_{n'} > \tag{3.38}$$

$$(M_0)_{i,n} = < \chi_i|\hat{H} - E|\frac{u_{0n'}(R_{\gamma'})}{R_{\gamma'}}\phi_{n'} > \tag{3.39}$$

$$M_{i,j} = < \chi_i|\hat{H} - E|\chi_j > \tag{3.40}$$

Because the trial wave function consists of free (or scattering) functions and bound (L^2) functions three types of integrals show up. The $(M_{0,0})$ and $(M_{1,0})$ contain the free-free integrals and the dimension of these square matrices is the number of open channels. We note that

the integrals involving channels of different arrangements can be avoided by using a cutoff function that cuts of the free functions in the region where they would overlap otherwise.

The M_0 matrix elements are integrals involving one bound and one free function. So M_0 is a rectangular matrix, which is "large" (the number of L^2 basis functions) by "small" (the number of open channels).

The M matrix consists of bound-bound integrals and is "large" by "large". Note that this matrix is real and symmetric.

3.3 The DVR applied to the S-matrix Kohn method

The DVR is used for the L^2 functions $|\chi_i(\vec{x})>$. Here we will show the formalism for a general interaction region coordinate system $\vec{x} = (x, y, z)$. This coordinate system does not have to be cartesian, so we assume that is has some Jacobian $J(\vec{x})$ associated with it, and so the scalar product is defined as:

$$< f|g > \equiv \int\int\int J(\vec{x}) f(\vec{x}) g(\vec{x}) d\vec{x} \tag{3.41}$$

where the ranges of integration depend on the type of coordinate. We assume that we have a quadrature for every coordinate:

$$\int f(x)dx = \sum_{i=1}^{n_x} w_i^x f(x_i) \tag{3.42}$$

(and similarly for the y and z coordinate).

So we start with a direct product quadrature with $n_x \times n_y \times n_z$ points which should include the entire interaction region. Then we truncate this grid by throwing out all points where the potential is above a certain cutoff energy ($V_{\rm cut}$) i.e. where we assume the wave function to be (almost) zero [14]. Furthermore we throw out the points for which the Jacobi scattering coordinate in one of the arrangements is larger then a certain $R_\gamma^{\rm max}$ (γ is again the arrangement label). Of course $V_{\rm cut}$ and $R_\gamma^{\rm max}$ are convergence parameters.

After truncation we have a set of N grid points:

$$\vec{x}_q = (x_{k_i}, y_{l_i}, z_{m_i}) \; ; \quad i = 1, \ldots, N \tag{3.43}$$

for which we have:

$$V(\vec{x}_i) \leq V_{\rm cut} \tag{3.44}$$

$$R_\gamma(\vec{x}_i) \leq R_\gamma^{\rm max} \tag{3.45}$$

Associated with each point there is a quadrature weight:

$$w_i = w_{k_i} w_{l_i} w_{m_i} \tag{3.46}$$

and a DVR function:

$$|\chi_i > = \frac{1}{\sqrt{J(x_i)}} \phi_{k_i}^x(x) \phi_{l_i}^y(y) \phi_{m_i}^z(z) \tag{3.47}$$

with the property:

$$\chi_i(\vec{x}_j) = \frac{1}{\sqrt{w_i J(\vec{x}_i)}} \delta_{ij} \tag{3.48}$$

Using this quadrature to evaluate the bound-free elements (Eq. 3.39) we get:

$$(M_0)_{i,n} = < \chi_i|\hat{H} - E|\frac{u_{0n}(R_\gamma)}{R_\gamma}\phi_n > \tag{3.49}$$

$$= \sqrt{w_i J(\vec{x}_i)}\left[(\hat{H} - E)\frac{u_{0n}(R_\gamma)}{R_\gamma}\phi_n\right]_{\vec{x}_i} \tag{3.50}$$

For the free-free elements we have:

$$M_{ij} = < \chi_i|\hat{H} - E|\chi_j > \tag{3.51}$$

$$= \int\int\int J(x)\chi_i(\vec{x})(\hat{H} - E)\chi_j(\vec{x})d\vec{x} \tag{3.52}$$

For multiplicative terms in the Hamiltonian (such as the potential) we get only diagonal elements (because of the quadrature approximation):

$$< \chi_i|V|\chi_j >= V(\vec{x}_i)\delta_{ij} \tag{3.53}$$

and for kinetic energy terms like:

$$T_x = \frac{1}{\sqrt{J}}T'_x\sqrt{J} \tag{3.54}$$

we get:

$$< \chi_i|T_x|\chi_j >= (T'_x)_{k_i,k_j}\delta_{l_i,l_j}\delta_{m_i,m_j} \tag{3.55}$$

where

$$(T'_x)_{k_i,k_j} = \int \phi^x_{k_i}(x)T'_x\phi^x_{k_j}(x)dx \tag{3.56}$$

For kinetic energy terms with a coordinate dependent prefactor we use the quadrature approximation for the prefactor and the analytic formulas for the (second) derivative part:

$$< \chi_i|C(x,y)T_z|chi_j >= \delta_{k_i,k_j}\delta_{l_i,l_j}C(x_{k_i},y_{k_j})(T'_z)_{m_i,m_j} \tag{3.57}$$

and for cross terms we have:

$$< \chi_i|T_xT_y|\chi_j >= (T'_x)_{k_i,k_j}(T'_y)_{l_i,l_j}\delta_{m_i,m_j} \tag{3.58}$$

where the primed and the unprimed operators are related as in Eq. (3.54).

3.4 Implementation of $M\vec{c}$

To solve the linear set of equations

$$M\vec{x} = \vec{b} \tag{3.59}$$

with an iterative method we need to implement the matrix vector multiplication:

$$\vec{d} = M\vec{c} \tag{3.60}$$

In this multiplication we want to exploit the sparsity of M, i.e. we want to avoid multiplications involving zeros of M. We will use as an example a 2-D Hamiltonian to explain our strategy

$$\hat{H} = \hat{T}_x + \hat{T}_y + V(x, y) \tag{3.61}$$

At the end of this section we will show how to handle different types of terms in a Hamiltonian.

Assume that the indices of the (`n`) points in the truncated grid are stored in the one-dimensional arrays `ix(1..n)` and `iy(1..n)`, the potential at the grid points is store in `V(1..n)` and the 1-D kinetic energy operators in `Tx(1..nx,1..nx)` and `Ty(1..ny,1..ny)` (see Eq. 3.54). In addition to this we will need four additional integer arrays `ilx(1..n)`, `irx(1..n)`, `ily(1..n)`, and `iry(1..n)`, which specify for each point (`i`) in the truncated grid how many points there are in to the left [`ilx(i)`], to the right [`irx(i)`], below [`ily(i)`] and above [`ilr(i)`]. With these arrays set properly we can write down the implementation for different terms in the Hamiltonian. We will start with the potential energy term, which is the simplest one:

```
c routine V
      do i=1,n
         d(i)=V(i)*c(i)
      end do
```

For the $\hat{T}_x$ kinetic energy term we make the additional assumption that the points are sorted such that the x-index is running fastest, i.e. all points with the same y-index are numbered consecutively.

```
c routine Tx
      do i=1,n
         sum=0
         k=ix(i)
         do j=-ilx(i),irx(i)
            sum=sum+Tx(k,k+j)*c(i+j)
         end do
         d(i)=d(i)+sum
      end do
```

We can understand this algorithm if we realize that the $\hat{T}_x$ operator only connects points on the same row of the grid. The points along a row can be retrieved as `c(i+j)` because the points are stored consecutively. Operating the $\hat{T}_y$ operator is a little bit more complicated, since we need the values `c(i)` along a column in the grid. This would be no problem for a rectangular grid since the values would be stored at regular intervals in the array [we could even use a 2D array `c(1..nx,1..ny)` instead of the 1D array `c(1..n)`]. We solved this problem by reordering the array `c(1..n)` into a new array `cy(1..n)` which has the y-coordinates running fastest. To reorder the array we need the *ranking* [`irank(1..n)`] of each point:

```
c routine reorder
      do i=1,n
         cy(irank(i))=c(i)
      end do
```

After this reordering the routine for the $\hat{T}_y$ operator is almost identical to the one for $\hat{T}_x$:

```
c routine Ty
      do i=1,n
         sum=0
         k=iy(i)
         l=irank(i)
         do j=-ily(i),iry(i)
            sum=sum+Ty(k,k+j)*cy(l+j)
         end do
         d(i)=d(i)+sum
      end do
```

It is important to note that although the reordering has to be done every iteration, the array `irank(1..n)` only has to be determined only once. Finding this array is a standard sorting problem which can be done with a general purpose sorting algorithm. The only thing a general purpose sorting algoritms needs is a function `f(i,j)` which returns `.true.` if point `i` comes after point `j`. This function only needs the indexing arrays:

```
c f is a logical function
      if (iy(i).eq.iy(j)) then
         f=ix(i).gt.ix(j)
      else
         f=iy(i).gt.iy(j)
      end if
```

We note that in this implementation we set kinetic energy matrix elements to zero that connect points of opposite sides of a region that has been cut out. We think that this is physically correct, since cutting out points corresponds to assuming the wave function to be zero in a certain region. This implies the assumption that the potential is infinite in that region and so it should not hurt to set the couplings through that region to zero. Of course if there is tunneling, we should increase the value of $V_{\rm cut}$ (Eq. 3.44) in order to include points in the grid where the tunneling occurs. Also, indirect coupling, i.e. *around* the region with a high potential is still present in the grid.

Cross terms in the Hamiltonian should be implemented in two steps, i.e.

$$\vec{d} = T_x T_y \vec{c} \tag{3.62}$$

should be implemented as:

$$\vec{q} = T_y \vec{c} \tag{3.63}$$

$$\vec{d} = T_x \vec{q} \tag{3.64}$$

In this way cross terms will only be twice as expensive as simple terms. However, setting couplings across a forbidden region to zero can break the hermiticity of these cross terms. The hermiticity of the operator can be restored by using an explicitly symmetric form, i.e. replacing Eq. (3.62) by:

$$\vec{d} = \frac{1}{2} T_x T_y \vec{c} + \frac{1}{2} T_y T_x \vec{c} \tag{3.65}$$

Of course this makes these cross terms another factor of two more expensive.

3.5 Distorted waves

The trial wavefunction consists of free waves (the incoming wave and its complex conjugate) and L^2 basis functions (Eq. 3.9). The free waves solve the Schrödinger equation from $R = R^{\max} \ldots \infty$, if:[1]

$$\Delta V(R, r, \theta) = V(R, r, \theta) - V(\infty, r, \theta) = 0 \text{ for } \quad R \geq R^{\max} \tag{3.66}$$

Here we have assumed that the free waves take into account the centrifugal potential, i.e. are equal to Hankel functions for $R \geq R^{\max}$ and also we have assumed that the cutoff function $f(R) = 1$ for $R \geq R^{\max}$. The L^2 basis functions only have to be placed in the region $R < R^{\max}$ so we are interested in minimizing $R^{\max}$.

Instead of an analytical form we can also use a propagation method to generate "distorted" free waves that solve the Schrödinger equation from infinity to a much smaller value of $R^{\max}$. A first approach could be to include the elastic distortion, since this can be expected to be the main long range effect. Secondly, we can take into account the mixing among the open channels. However, if we want to propagate as close in as possible we will have to include closed channels (see below) in the propagation. These closed channels are known to cause numerical problems in simple propagation schemes like the Numerov propagation, but this problem has been solved in the *renormalized* Numerov method [15]. With this method we can easily generate inelastic scattering wave functions. However, we need free waves with asymptotically incoming wave boundary conditions. Our strategy is the following:

Since we do not want a distorted wave of one arrangement to overlap with a distorted wave of a different arrangement, we start by adding a short range repulsive term $[V^{(1)}(R)]$ to the potential. We will call $V^{(1)}(R)$ a *barrier* potential. Then we solve the resulting inelastic scattering problem using the renormalized Numerov method. We repeat this calculation with a slightly different barrier potential and finally we take a linear combination of the two sets of solutions to obtain the desired asymptotic boundary conditions. This approach has several advantages:

- We don't need a cutoff function since the inelastic scattering functions automatically have the proper boundary conditions at $R = 0$.
- $R^{\max}$ is no longer a convergence parameter since it is simply the point after which both barrier potentials are zero.
- $(\hat{H} - E)$ on the free waves can be implemented as a multiplicative operator (see below).

The disadvantage is of course that we have to choose the somewhat arbitrary barrier potential. Before we discuss this problem, we will first summarize the renormalized Numerov method, show how to impose the right boundary conditions and how to evaluate the bound-free and free-free matrix elements.

[1] in this section we drop all arrangement labels

The renormalized Numerov method [15]

We will start with the atom-diatom (3D, J=0) example but most of this discussion is completely general. The inelastic scattering problem can be written as:

$$(\hat{H} - E)\frac{\chi(R,r,\theta)}{Rr} = 0 \tag{3.67}$$

and the Hamiltonian is again given by Eq. (3.20). The close coupling expansion for χ is:

$$\chi_j(R,r,\theta) = \sum_{i=1}^{n_{\rm tot}} \phi_i(r,\theta) U_{ij}(R) \; ; \quad j = 1,2,\ldots,n_{\rm tot} \tag{3.68}$$

where $n_{\rm tot}$ is the total number of channels (open+closed) and i is a composite index for the rotational quantum number j_i and the vibrational quantum number v_i. Substituting Eq. (3.68) into Eq. (3.67), multiplying from the left with $Rr\phi_k(r,\theta)$ and integrating over r and θ gives:

$$\frac{\partial^2}{\partial R^2} U(R) = W(R)U(R) \tag{3.69}$$

where

$$W_{i,j} = \frac{2\mu}{\hbar^2} < \phi_i | V(R,r,\theta) - V(\infty,r,\theta) | \phi_j > + \left[\frac{j_i(j_i+1)}{R^2} - k_i^2\right]\delta_{ij} \tag{3.70}$$

and

$$\frac{\hbar^2 k_i^2}{2\mu} \equiv E - \epsilon_i \tag{3.71}$$

Here ϵ_i is the eigenvalue corresponding to the diatom-eigenfunction ϕ_i. Note that k_i is imaginary for closed channels.

The differential equation is solved numerically on the grid:

$$R_k = R_0 + k\Delta; \quad k = 1,\ldots,n_{\rm step} \tag{3.72}$$

We define:

$$U_k \equiv U(R_k) \tag{3.73}$$

The Numerov progator is identical to the corrector formula of Milne's sixth order formula (see Abramowitz & Stegun [11]):

$$U_k = 2U_{k-1} - U_{k-2} + \frac{\Delta}{12}(W_k U_k + 10 W_{k-1}U_{k-1} + W_{k-2}U_{k-2}) \tag{3.74}$$

which can be rewritten as:

$$(1 - \frac{\Delta^2}{12}W_k)U_k - 2(I + \frac{5\Delta^2}{12}W_{k-1})U_{k-1} + (I - \frac{\Delta^2}{12}W_{k-2})U_{k-2} = 0 \tag{3.75}$$

This three term recursion relation for U_k can be turned into a two term recursion for the ratio of two adjacent U matrices. Defining:

$$Q_k = U_{k-1}U_k^{-1} \tag{3.76}$$

and

$$F_k = I - \frac{\Delta^2}{12} W_k \tag{3.77}$$

we get:

$$Q_k = [12I - 10F_{k-1} - F_{k-2}Q_{k-1}]^{-1} F_k \tag{3.78}$$

With $U_0 = 0$ and $U_1 \neq 0$ we get $Q_1 = 0$. With the recursion relation 3.78 we can generate $Q_2, Q_3, \ldots, Q_{\text{nstep}}$. Using Eq. (3.76) we can generate the coefficient matrices U_k from ($k =$ nstep) back to $k = 1$. To start this process we need U_{nstep}, which we can extract from Q_{nstep}:

Asymptotically $U(R)$ can be written as:

$$U_{ij}(R) \sim v_i^{-\frac{1}{2}} [\cos(k_i R - \frac{\pi j_i}{2}) K_{ij} + \sin(k_i R - \frac{\pi j_i}{2}) \delta_{ij}] \tag{3.79}$$

for $i = 1, \ldots, n_{\text{open}}$, $j = 1, \ldots, n_{\text{tot}}$ and

$$U_{ij}(R) \sim |v_i|^{-\frac{1}{2}} (e^{-|k_i|R} K_{ij} + e^{|k_i|R} \delta_{ij}) \tag{3.80}$$

for $i = n_{\text{open}} + 1, \ldots, n_{\text{tot}}$, $j = 1, \ldots, n_{\text{tot}}$.

In matrix notation this can be written as:

$$U(R) = C(R)K + S(R) \tag{3.81}$$

where $C(R)$ and $S(R)$ are diagonal matrices:

$$C_{i,i}(R) = |v_i|^{-\frac{1}{2}} \cos(k_i R - \frac{\pi j_i}{2}) \tag{3.82}$$

$$S_{i,i}(R) = |v_i|^{-\frac{1}{2}} \sin(k_i R - \frac{\pi j_i}{2}) \tag{3.83}$$

So at $R = R_k$ we have:

$$U_k = C_k + S_k \tag{3.84}$$

substituting this into Eq. (3.76) gives:

$$Q_k[C_k K + S_k] = C_{k-1} K + S_{k-1} \tag{3.85}$$

from which we get the following expression for the K matrix:

$$K = -[Q_k C_k - C_{k-1}]^{-1} [Q_k S_k - S_{k-1}] \text{ for large } k \tag{3.86}$$

Now we want to construct functions with incoming wave boundary conditions:

$$U_{ij}^{\pm}(R) \sim |v_i|^{-\frac{1}{2}} e^{\pm i(k_i R - \frac{\pi j_i}{2})} \delta_{ij} \tag{3.87}$$

or in matrix notation:

$$U^{\pm} = C \pm iS \tag{3.88}$$

First we define two inelastic scattering problems ($p = 1, 2$):

$$(\hat{H}^{(p)} - E) \frac{\chi^{(p)}}{Rr} = 0 \tag{3.89}$$

where the two hamiltonians are:

$$\hat{H}^{(p)} = \hat{H} + V^{(p)}(R) \tag{3.90}$$

for the wave functions we have:

$$\chi_j^{(p)} = \sum_{i=1}^{n_{\rm tot}} \phi_i(r,\theta) U_{ij}^{(p)}(R) \tag{3.91}$$

We can write the asymptotic form of both solutions in one matrix equation:

$$\left[\tilde{U}^{(1)}(R)\tilde{U}^{(2)}(R)\right] \sim \left[\tilde{C}(R)\tilde{S}(R)\right] \begin{bmatrix} \tilde{K}^{(1)} & \tilde{K}^{(2)} \\ \tilde{I} & \tilde{I} \end{bmatrix} \tag{3.92}$$

Here the tilde indicates the upper left $n_{\rm open} \times n_{\rm open}$ block of the matrix. From this equation we can derive what linear combination of the columns of $\tilde{U}^{(p)}$ to take to get asymptotically cosines and sines:

$$\left[\tilde{C}(R)\tilde{S}(R)\right] \sim \left[\tilde{U}^{(1)}(R)\tilde{U}^{(2)}\right] E \tag{3.93}$$

where the $2n_{\rm open} \times 2n_{\rm open}$ matrix E is given by:

$$E = \begin{bmatrix} \tilde{K}^{(1)} & \tilde{K}^{(2)} \\ \tilde{I} & \tilde{I} \end{bmatrix} \tag{3.94}$$

Finally we only have to combine the sines and the cosines to get asymptotically incoming $(-)$ or outgoing $(+)$ waves:

$$\bar{U}^{\pm}(R) = \left[\bar{U}^{(1)}(R)\bar{U}^{(2)}(R)\right] E \begin{bmatrix} \tilde{I} \\ \pm i\tilde{I} \end{bmatrix} \tag{3.95}$$

$$\equiv \bar{U}^{(1)}(R)E^{(1),\pm} + \bar{U}^{(2)}(R)E^{(2),\pm} \tag{3.96}$$

The bar indicates the first $n_{\rm open}$ columns (of length $n_{\rm tot}$). So the incoming and outgoing waves are defined by $(j = 1, \ldots, n_{\rm open})$:

$$\chi_j^{\pm} = \sum_{i=1}^{n_{\rm tot}} \phi_i(r,\theta) U_{ij}^{\pm}(R) \tag{3.97}$$

$$= \sum_{p=1}^{2} \sum_{k=1}^{n_{\rm open}} \chi_k^{(p)} E_{kj}^{(p),\pm} \tag{3.98}$$

Bound-free matrix elements

It is easy to operate $\hat{H} - E$ on the incoming waves if we observe from Eq. (3.89) that:

$$(\hat{H} - E)\chi_j^{(p)} = -V^{(p)}\chi_j^{(p)} \tag{3.99}$$

So using Eq. (3.98) we get:

$$(\hat{H} - E)\frac{\chi_j^-}{Rr} = -\frac{1}{Rr} \sum_{p=1}^{2} \sum_{k=1}^{n_{\rm open}} V^{(p)}(R)\chi_k^{(p)}(R,r,\theta) E_{kj}^{(p),-} \tag{3.100}$$

Free-free matrix elements

Using the same trick as for the bound-free elements we get:

$$< \frac{\chi_i^{\pm}}{Rr}|\hat{H} - E|\frac{\chi_j^{-}}{Rr} >= \sum_{q=1}^{2}\sum_{k=1}^{n_{\rm tot}} \int\int\int \chi_i^{\pm} V^{(q)} \chi_k^{(q)} E_{kj}^{(q),-} dR\ dr\ d\cos\theta \tag{3.101}$$

Fully expanding the χ's gives and integrating over r and θ gives:

$$< \frac{\chi_i^{\pm}}{Rr}|\hat{H} - E|\frac{\chi_j^{-}}{Rr} >= -\sum_{p=1}^{2}\sum_{q=1}^{2}\sum_{k=1}^{n_{\rm open}}\sum_{l=1}^{n_{\rm open}} E_{ki}^{(p),\pm} E_{lj}^{(q),-} X_{kl}^{p,q} \tag{3.102}$$

where

$$X_{kl}^{pq} = \int_0^{\infty} V^(q)(R) \sum_{j=1}^{n_{\rm tot}} U_{jk}^{(p)}(R) U_{jl}^{(q)}(R) dR \tag{3.103}$$

Choice of the barrier potentials

The barrier potentials $V^{(p)}(R)$ need to satisfy somewhat contradictory conditions. First of all they have to be strongly repulsive for small R in order to prevent overlap of the incoming waves of different arrangements. Furthermore the barriers need to be as short range as possible, since the size of the interaction region grid is determined by the range of the barriers. These two conditions could be met with a hard wall potential. However, if the barrier switches on too sharply, the incoming waves will change abruptly in a small range, and we will need a high density of grid points in order to get acurate bound-free integrals. In other words, the barriers have to smooth as well.

If we have found one good barrier potential we can generate the second one by simply multiplying the first with a factor close to unity (e.g. 1.1). If the two barrier potentials are to similar the two sets of inelastic scattering waves will be almost linearly dependent and we can expect numerical problems. We had good results with a barrier of the following form:

$$V^{(1)}(R) = \begin{cases} a(R - R^{\rm max})^4 & \text{for } R < R^{\rm max} \\ 0 & \text{otherwise} \end{cases} \tag{3.104}$$

Bibliography

[1] See D. E. Manolopoulous, J. Chem. Phys. **85**, 6425 (1986) *and references therein.*

[2] (a) A. Kuppermann and P. G. Hipes, J. Chem. Phys. **84**,5962 (1986) ; (b) C. A. Parker, R. T. Pack, B. J. Archer, and R. B. Walker, Chem. Phys. Lett. **137**, 564 (1987).

[3] J. Z. H. Zhang, S.-I Chu, W. H. Miller, J. Chem. Phys. **1988** 6233 (1988).

[4] Z. Bacic and J. C. Light, J. Chem. Phys. **85**, 4594 (1986).

[5] A. S. Dickinson and P. R. Certain, J. Chem. Phys. **49**, 4209 (1968).

[6] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice-Hall, Englewood Cliffs, NH, 1980), Chap. 13.

[7] J. Stoer and R. Bulirsch, *Introduction to numerical analysis* (Springer, New York, 1980).

[8] D. T. Colbert and W. H. Miller, J. Chem. Phys. **96**, 1982 (1992).

[9] J. Cullum and R. A. Willoughby, J. Comput. Phys. **44**, 329 (1981).

[10] C. C. Paige and M. A. Saunders, Siam J. Numer. Anal. **12**, 617 (1975).

[11] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (National Bureau of Standards, Washington, D. C. 1968).

[12] M. L. Goldberger, K. M. Watson, *Collision Theory* (John Wiley, New York, 1964).

[13] J. Z. H. Zhang and W. H. Miller, J. Chem. Phys. **91** 1528 (1989).

[14] Z. Bačić and J. C. Light, J. Chem. Phys. **85**, 4594 (1986).

[15] (a) B. R. Johnson, National Resource for Computation in Chemistry, University of California, Berkeley, NRCC, Proceedings No. 5, 86 (1979) ; (b) L. M. Hubbard, S-hua Shi, and W. H. Miller, J. Chem. Phys. **78**, 2381 (1983).