The Log-Derivative and Renormalized Numerov Algorithms

I. Introduction

Two algorithms for solving the coupled channel differential equations which arise in atomic and molecular scattering theory will be presented. They are the log-derivative method$^{1,2}$ and the renormalized Numerov method$^{2,3}$ Both these algorithms share the following desirable properties: They are simple and easy to implement, no special difficulties are encountered with closed channels, the step size can be easily changed and no linear dependence or overflow difficulties arise when propagating the solution though classically forbidden regions.

The log-derivative method will be discussed first, then the renormalized Numerov method and finally a model calculation using both these algorithms will be discussed and features of the two methods compared.

II. Log-Derivative Method

The "coupled-channel Schrödinger equation" is most conveniently written in the following matrix differential equation form:

$$\left[ I \frac{d^2}{dx^2} + Q(x) \right] \psi(x) = 0$$  \hspace{1cm} (1)

where

$$Q(x) = (2 \mu / t)^2 [Ei - V(x)].$$  \hspace{1cm} (2)
Here, I is the unit matrix, $\mu$ is the reduced mass, $V(x)$ is the symmetric potential matrix which has the centrifugal potential and the diagonal threshold energy terms included in it and $E$ is the total energy. The wave function $\Psi(x)$ is a square-matrix function of $x$.

The log-derivative matrix is defined to be

$$\gamma(x) = \Psi'(x) \Psi^{-1}(x),$$

where the prime means differentiation with respect to $x$. Differentiating Eq. (3) and using Eq. (1) to eliminate the second derivative term, we obtain the matrix Ricatti equation

$$\gamma'(x) + Q(x) + \gamma^2(x) = 0,$$  

This equation cannot be integrated by the usual numerical techniques for solving first order differential equations because $\gamma(x)$ diverges for certain values of $x$. This is illustrated by the solution to the simple one channel problem in which $Q$ is a constant:

$$\gamma(x) = Q^{1/2} \text{etn} (Q^{1/2} x).$$

This function is infinite at the points $x = n \pi Q^{-1/2}$ and the usual numerical algorithms for solving a first order differential equation cannot propagate the solution across these points. Our algorithm has no difficulty propagating the solution across these singular points.
The algorithm is as follows:

\[ y_n = (I + h y_{n-1})^{-1} y_{n-1} - h^{-1} U_n \]  \hspace{1cm} (6)

where \( h \) is the spacing between the \( N + 1 \) grid points \( x_0, x_1, \ldots, x_N \) and where

\[ U_n = \begin{cases} 
  \frac{h^2}{3} Q(x_n), & n = 0, N \\
  2 \left( \frac{h^2}{3} \right) Q(x_n), & n = 2, 4, \ldots, N-2 \\
  8I + 8 \left[ I - \left( \frac{h^2}{6} \right) Q(x_n) \right]^{-1}, & n = 1, 3, \ldots, N-1 
\end{cases} \]  \hspace{1cm} (7)

Eq. (6) is a two term recurrence relation that can be iteratively solved once the term \( y_0 \) is specified. The initial term is related to the initial value of the log-derivative function by the relation

\[ y_0 = y(x_0) - h^{-1} U_0. \]  \hspace{1cm} (8)

The calculated value of \( y_n \) is equal to \( y(x_n) \) only at the final integration point \( n = N \). This is no problem however, since only this value is needed to calculate the S-matrix.

In actual practice it is somewhat more convenient and efficient to solve for the quantity

\[ z_n = I + h y_n. \]  \hspace{1cm} (9)
Substituting this into Eq. (6) it is easy to show that

\[ z_n = (2I - u_n) - z_{n-1}^{-1} \quad (10) \]

The initial term is calculated from the relation

\[ z_0 = (I - u_0) + h\nu(x_0) \quad (11) \]

For most scattering problems, this leads to \( z_0^{-1} = 0 \). The matrix \( y(x_N) \) is obtained from \( z_N \) in a final calculation

\[ y(x_N) = h^{-1} (z_N - I) \quad (12) \]

The reaction matrix \( K \) is defined by the asymptotic behavior of the wave function. In the region \( x \geq x_N \), in which all but the centrifugal part of the potential has become negligible, the wave function is

\[ \Psi(x) = \begin{cases} J(x) + N(x)K, & x \geq x_N \end{cases} \quad (13) \]

The matrices \( J(x) \) and \( N(x) \) are diagonal. The matrix elements of the open channels are made up of Riccati-Bessel functions.
\[ [J(x)]_{ij} = \delta_{ij} k_j^{-\frac{1}{2}} \hat{j}_j(k_j x) \]  

(14)

\[ [N(x)]_{ij} = \delta_{ij} k_j^{-\frac{1}{2}} \hat{n}_j(k_j x) \]  

(15)

and the matrix elements for the closed channels are made up of modified spherical Bessel functions of the first and third kinds.

\[ [J(x)]_{ij} = \delta_{ij} (k_i x)^{\frac{1}{2}} I_{j_{\frac{1}{2}}}(k_j x) \]  

(16)

\[ [N(x)]_{ij} = \delta_{ij} (k_i x)^{\frac{1}{2}} K_{j_{\frac{1}{2}}}(k_j x) \]  

(17)

where \( k_j \) is the channel wave number. Differentiate Eq. (13) with respect to \( x \), then multiply from the right by the inverse of this equation, set \( x = x_N \), and solve the resulting equation for \( K \) in terms of \( y(x_N) \).

\[ K = -[y(x_N) N(x_N) - N'(x_N)]^{-1} \times [y(x_N) J(x_N) - J'(x_N)]. \]  

(18)

The matrix \( K \) is an augmented reaction matrix containing elements connecting closed as well as open channels, e.g., \( K \) can be written in the form

\[
K = \begin{pmatrix}
K_{oo} & K_{oc} \\
K_{co} & K_{cc}
\end{pmatrix}
\]  

(19)
where $K_{oo}$, $K_{oc}$, $K_{co}$, and $K_{cc}$ are open-open, open-closed, closed-open, and closed-closed submatrices of $K$. The S-matrix is given in terms of the open-open submatrix, $K_{oo}$, by the familiar formula

$$S = (I + iK_{oo})^{-1} (I - iK_{oo}).$$ (20)

Each of the matrices in Eq. (18) can be partitioned into open-open, open-closed, closed-open and closed-closed submatrices similar to the partitioning of $K$ in Eq. (19). Written in partitioned form, Eq. (18) is

$$
\begin{pmatrix}
K_{oo} & K_{oc} \\
K_{co} & K_{cc}
\end{pmatrix}
= - \begin{pmatrix}
y_{oo}N_o - N'_o & y_{oc}N_c \\
y_{co}N_o & y_{cc}N_c - N'_c
\end{pmatrix}^{-1}
\times \begin{pmatrix}
y_{oo}J_o - J'_o & y_{oc}J_c \\
y_{co}J_o & y_{cc}J_c - J'_c
\end{pmatrix}
$$

Since the S-matrix depends only $K_{oo}$, it is clear that the calculation can be simplified somewhat by only computing the left hand column of partitions of the K-matrix. The equation then becomes

$$
\begin{pmatrix}
K_{oo} \\
K_{co}
\end{pmatrix}
= - \begin{pmatrix}
y_{oo}N_o - N'_o & y_{oc}N_c \\
y_{co}N_o & y_{cc}N_c - N'_c
\end{pmatrix}^{-1} \begin{pmatrix}
y_{oo}J_o - J'_o \\
y_{co}J_o
\end{pmatrix}
$$ (21)
The matrix $C$, which is the closed channel part of $J(X_N)$ is not used in this equation. Thus, the closed channel functions defined by Eq. (16) are not needed to calculate $K_{oo}$. However, the closed channel elements of $N(X_N)$ defined by Eq. (17) are still required.

The closed channel functions defined by Eq. (17) decrease exponentially with increasing $x$. This is a possible source of numerical difficulty. The problem is easily eliminated by redefining the closed channel elements of both $N(x)$ and $N'(x)$ by multiplying these functions by increasing exponential functions which just cancel the exponential decrease. That is, we make the following simple replacement

$$\begin{align}
[N(x)]_{ii} & \rightarrow [N(x)]_{ii} \exp(k_i x) \\
[N'(x)]_{ii} & \rightarrow [N'(x)]_{ii} \exp(k_i x)
\end{align}$$  

(22a)

(22b)

It should be noted that after this replacement is made, $N'(x)$ is no longer the first derivative of $N(x)$. It is easily verified that replacing $N(x)$ and $N'(x)$ by the expressions given in Eq's. (22a) and (22b) will leave $K_{oo}$ unchanged. These modified closed channel functions can be easily calculated from recurrence relations.

We have seen from Eq. (21) that the closed channel elements of $J(x)$ are not required to calculate $K_{oo}$ but the closed channel elements of $N(x)$ are, in general, required. The need for these functions can also be eliminated, but only if the value of $x_N$ (see Eq. (18)) is sufficiently large. It can be shown that the elements of $y_{oo}(x)$ and $y_{oo}(x)$ must eventually approach zero exponentially as $x$ increases. Thus, the open and closed channel parts of Eq. (21) decouple and only the open channel elements of $N(x)$ are required to calculate $K_{oo}$. 


III. Renormalized Numerov Method

The matrix Numerov algorithm is an efficient method that can be used to obtain numerical solutions of Eq. (1). The basic formula is the three term recurrence relation

\[ [I - T_{n+1}] \Psi_{n+1} - [2I + 10T_n] \Psi_n + [I - T_{n-1}] \Psi_{n-1} = 0 \]  

(23)

where

\[ \Psi_n \equiv \Psi(x_n) \]  

(24)

and

\[ T_n = - \left( \frac{h^2}{12} \right) Q(x_n) \]  

(25)

Here \( h \) is the spacing between the \( N + 1 \) equally spaced grid points \( x_0, x_1, \ldots, x_N \) and the square matrix \( Q(x) \) is defined by Eq. (2). Equation (23) is derived by an obvious generalization of the derivation of the ordinary Numerov algorithm to matrix quantities.

The renormalized Numerov algorithm is derived from Eq. (23) by making two transformations. First define the matrix

\[ F_n = [I - T_n] \Psi_n \]  

(26)
and substitute into Eq. (23). This gives

\[ F_{n+1} - U_n F_n + F_{n-1} = 0. \]  \hspace{1cm} (27)

where

\[ U_n = (I - T_n)^{-1}(2I + 10T_n). \] \hspace{1cm} (28)

Next, define the ratio matrix

\[ R_n = F_{n+1} F_n^{-1}. \] \hspace{1cm} (29)

Substitute this into Eq. (27) to obtain the two term recurrence relation

\[ R_n = U_n - R_{n-1}^{-1}. \] \hspace{1cm} (30)

This is the basic equation of the renormalized Numerov method. It can be solved once the value of the initial term \( R_0 \) is specified. In scattering problems, the usual case is to assume the initial values of the wavefunction are \( \Psi(x_0) = 0 \) and \( \Psi(x_1) \neq 0 \). The corresponding value of the initial inverse ratio matrix is \( R_0^{-1} = 0 \). (For exceptions to this rule see Appendix D in Ref. 3.)
The matrix $U_n'$ defined by Eq. (28), is symmetric. It follows from this and the symmetry of $R_0^{-1}$ and also from Eq. (30) that the matrix $R_n$ is also symmetric. For computational convenience, Eq. (28) can be reformulated as a symmetric matrix inversion problem. Define

$$W_n = I - T_n',$$

(31)

then

$$U_n = 12W_n^{-1} - 10I.$$

(32)

Thus, at each grid point we must invert two symmetric matrices.

Equation (30) can be solved iteratively to obtain $R_N'$. The value of $R_{N-1}^{-1}$ is also readily available and can be saved at the last integration point. Using these two quantities the log-derivative matrix can easily be calculated by means of the formula

$$y(x_n') = h^{-1} (A_{n+1} R_n - A_{n-1} R_{n-1}^{-1})(I - T_n)$$

(33)

where $T_n$ is defined by Eq. (25) and

$$A_n = (0.5 I - T_n)(I - T_n)^{-1}$$

(34)

The $S$-matrix can then be calculated from this log-derivative matrix by the techniques outlined in section II.
Another method for calculating the S-matrix, which avoids calculating the log-derivative matrix, is also possible. Multiply Eq. (13) by \((I - T_n)\) to obtain

\[
F_n = j(x_n) + n(x_n) K
\]  

(35)

where we have defined

\[
j(x_n) = (I - T_n) J(x_n)
\]  

(36)

and

\[
n(x_n) = (I - T_n) N(x_n)
\]  

(37)

Evaluate Eq. (35) at \(x_N\) and \(x_{N+1}\), calculate the ratio matrix \(R_N = \frac{F_{N+1}}{F_N} \left(\frac{F_N}{F_{N+1}}\right)^{-1}\), then solve the resulting equation for \(K\) in terms of \(R_N\)

\[
K = -\left[R_N n(x_N) - n(x_{N+1})\right]^{-1} \left[R_N j(x_N) - j(x_{N+1})\right]
\]  

(38)

This equation is similar to Eq. (18) and can be partitioned and solved in exactly the same way. If \(x_N\) is large enough, the open and closed channel parts of this equation decouple; if not, they are coupled and the closed channel elements of \(n(x_N)\) will be required. In order to avoid any possible numerical difficulty, the closed channel elements of \(N(x)\) in Eq. (37) should be the modified functions defined in Eq. (22a).
The K-matrix and S-matrix computed by the renormalized Numerov algorithm will by symmetric only to within the truncation error of the calculation. In fact, one can obtain an estimate of the magnitude of the truncation error from the error in symmetry. This is in contrast to the log derivative method where there is no relationship between truncation error and the symmetry of the S-matrix.

IV. Example Calculations

In this section several of the characteristics of the log-derivative and renormalized Numerov algorithms will be elucidated and compared by applying them to a model problem.

The model is the atom-collinear harmonic oscillator system described by Secrest and Johnson. The Schröedinger equation for this problem is

\[- \frac{1}{2m} \left( \frac{\partial^2}{\partial x^2} \right) - \frac{1}{2} \left( \frac{\partial^2}{\partial y^2} \right) + \frac{1}{2} y^2 + V(x-y) - \frac{1}{2} E \]\n
where the interaction potential is

\[V(x-y) = A \exp[-\alpha(x-y)]\]

This problem was recently solved very accurately by Stechel, Walker and Light for the particular set of parameters: A = 41000, \( \alpha = 0.3 \), \( m = 2/3 \), and \( E = 6.0, 8.0 \). They used a six channel expansion of the wave function and an integration range from \( x = 0 - 100 \). We have chosen this same set of parameters for our model problem. However, we only solved the \( E = 6.0 \) case.
Accurate, converged values of the transition probabilities are given in Table I. These values were calculated using the renormalized Numerov algorithm with 2000 points and a grid spacing $h = 0.05$. Since the transition probability matrix, computed by the renormalized Numerov method, is symmetric only to within the truncation error, we have symmetrized the results by averaging $P_{mn}$ and $P_{nm}$. It is these averaged probabilities that are given in Table I. The inaccuracy of any of these numbers is no greater than two digits in the last place shown.

In Figs. 1 and 2 we show the relative truncation error of the calculated transition probabilities as a function of the grid spacing. There are several features to observe: The error curves are almost linear (on a log-log scale) with a slope very close to 4. This is consistent with the fact that both algorithms are fourth-order methods. Next we note that for a given grid spacing, the renormalized Numerov method is more accurate than the log-derivative method. Alternatively, in order to obtain the same relative error, we must use a smaller grid spacing with the log-derivative method. The worst case is the 0-1 transition, where the ratio of log derivative to renormalized Numerov grid spacings must be about 0.63 in order to obtain equal relative errors. The best case is the 1-2 transition where this ratio is about 0.88.

On the other hand, the average CPU time per grid point is less for the log derivative method than for the renormalized Numerov method by an approximately constant ratio of about 0.76. This ratio is easy to understand. It is approximately the ratio of the number of matrix inversions. Two inversions per grid point are required for the renormalized Numerov method whereas, on the average, only 1.5 inversions per grid point are required for the log derivative method. The only other procedure that might
require much computer time at each grid point is the calculation of the potential matrix. In the present model problem this is almost negligible. However, if it were not, it would tend to make the time ratio per grid point less favorable to the log derivative method.

The fact that less time per grid point is required, approximately compensates for the increased number of points required by the log derivative method. Based on the figures given, the 1-2 transition could be calculated more efficiently using the log derivative method whereas the 0-1 transition could be calculated faster with the renormalized Numerov method.

The average CPU time per grid point as a function of the number of channels is plotted in Fig. 3. These curves can be approximately extrapolated for N larger than 20 channels by the formulas

\[ T_{RN} = 0.208 \times 10^{-5} N^{2.8} \]

and

\[ T_{LD} = 0.158 \times 10^{-5} N^{2.8} \]

By using the information in Figs. 1, 2 and 3 and given the integration range, the time required to calculate solutions of various accuracies can be determined.
REFERENCES

ATOM - COLLINEAR HARMONIC OSCILLATOR

RENDEREDIZED NUMEROU

Figure 1.
CPU TIME VS. NUMBER OF CHANNELS (CDC 7600)

CPU SECONDS PER GRID POINT

10^{-2}

10^{-3}

10^{-4}

10^{-5}

NUMBER OF CHANNELS

1

10

100

UPPER CURVE = REN NUM. LOWER CURVE = LOG DERIV.

Figure 3.
<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>P_{nm}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.97788564</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.97699265</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.999096929</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.2210932*10^{-1}</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.898031*10^{-3}</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0.503948*10^{-5}</td>
</tr>
</tbody>
</table>