JOURNAL OF COMPUTATIONAL PHYSICS 80, 17–26 (1989)

Vectorized Partial Wave Calculation Using a Cauchy-Type Propagation Matrix

KENJI ISHIBASHI, YUKI MIURA, HIROSHI TAKADA,
TAKEJI SAKAE, YUZURU MATSUMOTO, AND AKIRA KATASE

Department of Nuclear Engineering, Faculty of Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, Japan

Received April 3, 1987; revised December 17, 1987

The partial wave analysis is an important method of obtaining the collision cross section in atomic and nuclear physics. An algorithm is devised to reduce the computing time by using a vector computer. In solving the Schrödinger equation numerically, this algorithm vectorizes the recurrent calculation in an ingenious way. The method is suited for those problems that are related to relatively a small number of quantum states (reaction channels). A simple example of the calculation is shown for elastic scattering. A speed-up gain of 8 is achieved in comparison to a scalar computer. © 1989 Academic Press, Inc.

1. INTRODUCTION

Partial wave analysis [1] has been used to calculate the collision cross section in atomic and nuclear physics. The computing methods for the analysis were summarized in Ref. [2]. The methods have been further improved by many authors [3–8]. They were applied, for example, to such subjects as simple elastic scattering [9–11] and inelastic and reactive collisions [12]. Although the methods are very useful in a low energy region, the numerical calculation usually takes a long computation time.

Vector computers have emerged recently [13] and come to be widely used [14, 15]. They are capable of performing a calculation in a do-loop much faster than scalar computers. The faster speed, however, is achieved only for the do-loop that contains many parallel calculations of the same kind. When the do-loop has a recurrent logic, the processing speed does not become faster at all. This is a marked feature of the pipeline processing in the vector computer [16].

The vector computer has already been utilized for the partial wave analysis in a molecular–molecular collision [17]. A great number of quantum states (reaction channels) were taken into account in this work. There were a lot of parallel calculations. Matrices with very large sizes were used and their calculations were vectorized. Some recurrent relations remained without being vectorized in solving the expanded Schrödinger equation. However, there were so many parallel calculations as a whole that a sufficient vectorization was achieved. The above method then produced an excellent processing speed inspite of the remaining recurrent calculation.

There are many problems which are related to a small number of parallel calculations. It is useful to develop a vectorization method which is applicable to such problems. The use of this method allows us to make their computation faster. In this case, however, recurrent calculations themselves need to be vectorized. An attempt will be made to perform parallel processing directly in solving the Schrödinger equation.

2. Method of Solving a Single Schrödinger Equation

The radial Schrödinger equation is usually expressed by the simple form [1]

$$d^2u_n/dr^2 + F_n \cdot u_n = 0, \tag{1}$$

where the suffix n is the quantum number of interest, u_n the wave function, r the radial distance, and F_n a function of r which generally contains the potentials and the energy.

In this section, Eq. (1) is represented by omitting the suffix n:

$$d^2u/dr^2 + F \cdot u = 0. \tag{2}$$

For numerical calculation, the radial region is divided into a number of meshes. The points of divisions are numbered as $i = 1$ to $N + 1$ in the outward direction. The function and its derivative with respect to r are designated as u_i and u'_i at the ith point, respectively. Equation (2) is converted into an equation by the use of the Cauchy matrix [7, 18]:

$$\begin{pmatrix} u_{i+1} \\ u'_{i+1} \end{pmatrix} = \mathbf{A}_i \begin{pmatrix} u_i \\ u'_i \end{pmatrix}, \tag{3}$$

where the matrix is given by

$$\mathbf{A}_i = \begin{pmatrix} a_{i11} & a_{i12} \\ a_{i21} & a_{i22} \end{pmatrix}. \tag{4}$$

Other types of propagation matrices have been used in some cases [4, 7]. In the present study, however, we choose the form of Eq. (3) because of its more suitable character for vectorization.

The vector processor performs calculations of a do-loop in an almost parallel manner. The computer attains its ultimate processing speed for the do-loop that has complete parallel calculations. One can see in Eq. (3) that u_i and u'_i depend on those values at the $(i-1)$th point. When it is coded in the usual way, the do-loop contains a typical recurrent logic in the calculation from $i=1$ to N. Such a do-loop calculation does not make the vector computer any faster.

Equation (3) is solved in a special way in this study. The values of u and u' at the last point $i=N+1$ are given by

$$\begin{pmatrix} u_{N+1} \\ u'_{N+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} u_1 \\ u'_1 \end{pmatrix}. \tag{5}$$

The matrix $\mathbf{A}$ is a product of individual matrices $\mathbf{A}_i$ at all positions:

$$\mathbf{A} = \prod_{i=1}^{N} \mathbf{A}_i. \tag{6}$$

When the radial positions are grouped evenly into m sections, each section has $p=N/m$ points. A matrix $\mathbf{B}_j$ in the section j is defined as a product of $\mathbf{A}_i$ in that section:

$$\mathbf{B}_j = \prod_{g=1}^{p} \mathbf{A}_{(j-1)p+g}. \tag{7}$$

The matrix $\mathbf{A}$ is thus expressed by

$$\mathbf{A} = \prod_{j=1}^{m} \mathbf{B}_j. \tag{8}$$

In Eqs. (7) and (8), the multiplication in Eq. (6) is rearranged in a special manner. In the evaluation of Eq. (7), the calculation for $\mathbf{B}_j$ is independent of that for $\mathbf{B}_{j'}$, when $j' \neq j$. Therefore, the calculation is suited for the parallel processing by the vector computer. The do-loop for evaluating the value of Eq. (7) can be completely vectorized with respect to the do-variable j. The do-loop with the index j may enclose an inner do-loop with the do-variable g in a natural sense. One can see that the calculation results are the same when the inner do-loop is exchanged by the outer. For convenience of vectorization, therefore, the do-variables are exchanged in this study. The inner do-loop with the index j has a parallel calculation alone, while the outer do-loop becomes recurrent with regard to g. Since the principal calculation is already vectorized in the inner do-loop, the vector computer can evaluate all $\mathbf{B}_j$ with a very high speed.

As indicated in Eq. (6), the final matrix is expressed by a linear product of matrices at individual positions. This linearity enables us to vectorize the calculation procedure. In contrast, there are other propagation methods for solving wave equations. Both R-matrix propagation [4–6] and log-derivative [7, 8] methods have the matrices that produce nonlinear relations in their propagation calculations. Since the calculation procedure inevitably becomes recurrent for these methods, they are not suited for direct vectorization.

3. Coupled Equations

In a complex collision problem, coupled equations [2] are usually used, differing from Eq. (1). They have the form

$$d^2\mathbf{u}/dr^2 + \mathbf{F}\mathbf{u} = 0, \tag{9}$$

where $\mathbf{u}$ stands for a vector expressing wave functions. The wave functions of different quantum states (reaction channels) are contained in $\mathbf{u}$ as elements of the vector. The operator matrix $\mathbf{F}$ couples the individual wave functions in $\mathbf{u}$, and is converted into a diagonal matrix $\mathbf{D}$ by introducing a unitary matrix $\mathbf{T}$:

$$\mathbf{D} = \mathbf{T}^{-1}\mathbf{F}\mathbf{T}. \tag{10}$$

The vector $\mathbf{u}$ is transformed into

$$\mathbf{v} = \mathbf{T}^{-1}\mathbf{u}. \tag{11}$$

Neglecting higher order terms [19], we have an equation without coupling:

$$d^2\mathbf{v}/dr^2 + \mathbf{D}\mathbf{v} = 0. \tag{12}$$

A global vector and a global matrix are defined to extend the relationship in Eq. (3). In a region between r_i and r_{i+1}, an equation is obtained which is similar to Eq. (3):

$$\begin{pmatrix} \mathbf{v}_{i+1} \\ \mathbf{v}'_{i+1} \end{pmatrix} = \mathbf{G}_i \begin{pmatrix} \mathbf{v}_i \\ \mathbf{v}'_i \end{pmatrix}. \tag{13}$$

The matrix $\mathbf{G}_i$ has a size $2N \times 2N$. This matrix is derived from the diagonal matrix $\mathbf{D}_i$ using a constant or linear reference potential [18, 19]. Then, the original wave functions are expressed by

$$\begin{pmatrix} \mathbf{u}_{i+1} \\ \mathbf{u}'_{i+1} \end{pmatrix} = \mathbf{A}_i \begin{pmatrix} \mathbf{u}_i \\ \mathbf{u}'_i \end{pmatrix}, \tag{14}$$

where

$$\mathbf{A}_i = \begin{pmatrix} \mathbf{T}_i & 0 \\ 0 & \mathbf{T}_i \end{pmatrix} \mathbf{G}_i \begin{pmatrix} \mathbf{T}_i^{-1} & 0 \\ 0 & \mathbf{T}_i^{-1} \end{pmatrix}. \tag{15}$$

A matrix $\mathbf{A}$ is defined as a products of $\mathbf{A}_i$:

$$\mathbf{A} = \prod_{i=1}^{N} \mathbf{A}_i. \tag{16}$$

The wave function at the last point is thus written by

$$\begin{pmatrix} \mathbf{u}_{N+1} \\ \mathbf{u}'_{N+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}'_1 \end{pmatrix}. \tag{17}$$

Equations (16) and (17) have the same form as Eqs. (5) and (6). The rearrangement method used in Eq. (7) is applicable to Eq. (16). Hence, vector processing is useful in evaluating Eq. (16). The collision cross section is obtained from the results at a radial distance sufficiently far from the origin. The cross section is obtained, for instance, from the reactance matrix [1] that is constructed by the use of these values.

4. Simple Example of the Vector Calculation

As an example of calculation, the method in Section 2 is applied to the elastic scattering of electrons by a simple spherical potential [9, 10]. In this case, the quantum number n in Eq. (1) stands for the angular momentum l. Then, n is replaced by l in the following. The function F becomes $F(r) = k^2 - U(r) - l(l+1)/r^2$, where k is the wave number of the incident electron, and $U(r)$ the potential. The increment h of the radial point r_i is taken to be constant along the whole length. The boundary condition appears at the origin ($r=0$) as $u_l = 0$. The Magnum approximation [19] is made to first order in h. When a matrix $\mathbf{M}$ is defined as

$$\mathbf{M}(r) = \begin{pmatrix} 0 & 1 \\ F(r) & 0 \end{pmatrix}, \tag{18}$$

the matrix $\mathbf{A}_i$ in Eq. (4) becomes [19]

$$\begin{aligned} \mathbf{A}_i &= \exp(h\mathbf{M}(r_i + h/2)) \\ &= \begin{pmatrix} \cos(\lambda_i h) & \lambda_i^{-1} \sin(\lambda_i h) \\ \lambda_i \sin(\lambda_i h) & \cos(\lambda_i h) \end{pmatrix}, \end{aligned} \tag{19}$$

where

$$\lambda_i^2 = -F(r_i + h/2). \tag{20}$$

When $F > 0$, the value of λ_i becomes complex so that the trigonometric functions in Eq. (19) are equivalently written by hyperbolic ones. It may be possible to use a linear reference potential for creating the matrix elements [18]. In the present work, nevertheless, Eqs. (18) and (19) are chosen for their simplicity. The value of the radial wave function u_l is computed from the origin in the outward direction according to Eqs. (7) and (8). The phase shift η_l is determined by comparing the calculated values of u_l with its asymptotic form [1]

$$u_l \propto k^{-1} \sin(k \cdot r - l\pi/2 + \eta_l) \tag{21}$$

at a distance sufficiently far from the origin.

5. Calculation Procedure

Figure 1 shows the procedure for the calculation described in Section 4. The partial wave has an angular momentum l. A radial position r_s is first taken as a starting point. Calculations are divided into three stages A, B, and C. In stage A, a certain radial distance d_c is added to r_s. The calculation goes from $r_1 = r_s$ to $r_{N+1} = r_s + d_c$. The method is explained in Fig. 2. The matrices $\mathbf{B}_j$ are evaluated according to Eq. (7). They are calculated in the double do-loop with the do-variables j and g as shown in Fig. 2. The inner do-loop carries out the multiplication of individual matrices $\mathbf{A}_i$ by the do-variable j. Since this calculation is completely independent of the other with regard to the do-variable j, there is no recursion in the inner do-loop. It is therefore possible to vectorize the inner do-loop by the do-variable j.

Stage B calculates the matrix $\mathbf{A}$ as the product of the matrices $\mathbf{B}_j$ according to Eq. (8). Although this calculation is basically recurrent, it is evaluated again by vector processing. The matrices $\mathbf{B}_j$ are classified into q groups, so that the number of sections in each group becomes $s = m/q$. A matrix $\mathbf{C}_t$ in group t is expressed as product of $\mathbf{B}_j$:

$$\mathbf{C}_t = \prod_{d=1}^{s} \mathbf{B}_{(t-1)s+d}. \tag{22}$$

Equation (8) is thus written by

$$\mathbf{A} = \prod_{t=1}^{q} \mathbf{C}_t. \tag{23}$$

The do-loop for evaluating Eq. (22) is vectorized with respect to t for the same reason as for Eq. (7). The procedure is shown in Fig. 3. After the vector calculation, the matrix $\mathbf{A}$ is obtained by scalar processing by Eq. (23). The functions u_{N+1} and u'_{N+1} at the last point r_{N+1} are easily obtained from Eq. (5).

Stage A: Evaluation of $\mathbf{B}_j$ in the region $r = r_s$ to $r_s + d_c$.

↓

Stage B: Evaluation of the matrices $\mathbf{C}_t$ and $\mathbf{A}$ and the functions u_{N+1} and u'_{N+1}.

↓

Stage C: Evaluation of phase shift and check of its convergence.

Fig. 1. Outline of the phase shift calculation.

```
(1) Initial value of $\mathbf{B}_j$ (vector processing)
  Do $j = 1, m$
      $\mathbf{B}_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
      Continue
(2) Matrix product for $\mathbf{B}_j$ (vector processing)
  Do $g = 1, p$
      Do $j = 1, m$
      $i = (j-1)p + g$
      $\mathbf{B}_j = \mathbf{B}_j \begin{pmatrix} \cos(\lambda_i h) & \lambda_i^{-1} \sin(\lambda_i h) \\ \lambda_i \sin(\lambda_i h) & \cos(\lambda_i h) \end{pmatrix}$
      Continue
    Continue
```

FIG. 2. Detailed procedure in stage A.

```
(1) Initial value of $\mathbf{C}_t$ (vector processing)
  Do $t = 1, q$
    $\mathbf{C}_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
    Continue
(2) Matrix product for $\mathbf{C}_t$ (vector processing)
    Do $d = 1, s$
      Do $t = 1, q$
      $\mathbf{C}_t = \mathbf{C}_t \mathbf{B}_{(t-1)s+d}$
      Continue
    Continue
(3) Evaluation of matrix $\mathbf{A}$ (scalar processing)
    $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
    Do $t = 1, q$
    $\mathbf{A} = \mathbf{A}\mathbf{C}_t$
    Continue
(4) Evaluation of $u_{N+1}$ and $u'_{N+1}$ (scalar processing)
    $\begin{pmatrix} u_{N+1} \\ u'_{N+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} u_1 \\ u'_1 \end{pmatrix}$
```

FIG 3. Detailed procedure in stage B.

In stage C, the phase shift η_l is obtained from the relation

$$k \cdot r_{N+1} - l\pi/2 + \eta_l = \arctan(k \cdot u_{N+1}/u'_{N+1}). \tag{24}$$

The procedure is returned to stage A to continue the calculation. The last position r_{N+1} is regarded as an initial position r_s at stage A. If the phase shift is converged, the calculation is completed.

6. Computation Time

A program was coded in FORTRAN according to the scenario in the preceding section. This program was run in both vector and scalar computers at the computer center of Kyushu University. The computers were Fujitsu FACOM VP-200 as the vector, and M-780 as the scalar. The processing speed in the maximum performance was 540 MFLOPS for the vector computer. The computation speed was an order of magnitude lower in the scalar computer than in the vector computer. The do-loops in stage A (Eq. (7)) and a part of stage B (Eq. (22)) were vectorized in the vector computer. Since the amount of calculations were less in Eq. (22) than in Eq. (7), the vectorization for Eq. (22) was less effective for the entire computation time.

The calculation was performed on the elastic scattering of an electron by a neon atom. This problem is similar to our earlier work [10]. The spherically symmetric potential was simply approximated by a simple screened potential [20]. The incident energy of the electron was taken in an intermediate region of 200 to 1000 eV, i.e., $k = 2.7$ to 6.1 in atomic units (a.u.). The parameters were as follows: $h = 0.005$(a.u.), $d_c = 100$(a.u.), $N = 2 \times 10^4$, $m = 1000$, $p = 20$, $q = 50$, and $s = 20$. The fraction of the vectorization was as high as 99.9% in this code. The computation results are shown in Table I for the angular momentums $l = 3$ and 10. The vector computer carried out the calculation about 8 times as fast as the scalar computer did for the same program. This was a great improvement in the computating time.

TABLE I

Computation Time by the Scalar and the Vector Computers

Electron energy (eV)	l	Radial region (a.u.)	Phase shift (rad)	Computation time: Scalar (ms)	Computation time: Vector (ms)	Vector/scalar ratio
200	3	0–300	0.367	721	85	8.5
200	10	0–800	0.0229	1890	227	8.3
500	3	0–300	0.543	714	94	7.6
500	10	0–600	0.0437	1417	170	8.3
1000	3	0–300	0.613	720	85	8.5
1000	10	0–600	0.0764	1420	170	10.1

7. Characteristics of the Present Method

The simple example was described in the preceding section. This method enabled us to vectorize the calculation for a single Schrödinger equation. The method applies to the collision that is apparently related to a small number of parallel calculations. When the incident energy becomes higher and is in the intermediate region, the convergence of the phase shifts gets slower [10]. The calculation needs to go a larger distance before the phase shifts converge, so that it takes a longer time to complete the calculation. In such a case the present method is very useful for reducing computation time.

Many collision problems are described by coupled equations. The present method is successfully applied to such problems when they are related to a relatively small number of quantum states (reaction channels). However, when a great number of quantum states is involved, there are apparently a lot of parallel calculations. Then, sufficient vectorization may be achieved by methods other than the present one. In fact, vector computers have been used [17] by a method based on R-matrix propagation [5].

8. Conclusion

The phase shift calculation was vectorized for each quantum state. The use of the Cauchy matrix in the difference equation enabled us to solve the wave equation efficiently by using a vector computer. The speed-up gain of 8 in the computation time was obtained in comparison with that of the scalar computer. The present method is considered to be specially suitable for the partial wave calculation that relates to a relatively small number of reaction channels.

Acknowledgment

The authors wish to express their gratitude to Mr. K. Yamada of Kyushu University for his cooperation.

References

1. N. F. Mott and H. S. W. Massey, *Theory of Atomic Collision*, 3rd ed. (Oxford Univ. Press, London, 1965).
2. B. Alder *et al.* (Ed.), *Methods in Computational Physics*, Vol. 10 (Academic Press, New York, 1971).
3. Gh. Adam, L. Gr. Ixaru, and A. Corciovei, *J. Comput. Phys.* **22**, 1 (1976).
4. J. C. Light and R. B. Walker, *J. Chem. Phys.* **65**, 4272 (1976).
5. E. B. Stechel, R. B. Walker, and J. C. Light, *J. Chem. Phys.* **69**, 3518 (1978).
6. B. C. Garrett, M. J. Redmon, D. G. Truhlar, and C. F. Melius. *J. Chem. Phys.* **74**, 412 (1981).
7. F. Mrugala and D. Secrest, *J. Chem. Phys.* **78**, 5954 (1983).
8. M. H. Alexander and D. E. Manolopoulos, *J. Chem. Phys.* **86**, 2044 (1987).

9. P. S. Ganas and A. E. S. Green, *J. Chem. Phys.* **76**, 1819 (1982).
10. A. Katase, K. Ishibashi, Y. Matsumoto, T. Sakae, S. Maezono, E. Murakami, K. Watanabe, and H. Maki, *J. Phys. B* **19**, 2715 (1986).
11. T. Sawada, P. S. Ganas, and A. E. S. Green, *Phys. Rev. A* **9**, 1130 (1974).
12. Briefly reviewed in Ref. [17].
13. R. M. Russell, *Commun. ACM* **21**, 63 (1978).
14. R. W. Numrich (Ed.), *Supercomputer Applications* (Plenum, New York, 1985).
15. M. Dupuis (Ed.), *Supercomputer Simulations in Chemistry,* Lecture Notes in Chemistry, Vol. 44 (Springer-Verlag, Berlin, 1986).
16. K. Hwang, *Computer Arithmatic-Principles, Architecture and Design* (Wiley, New York, 1979).
17. D. W. Schwenke and D. G. Truhlar, in *Supercomputer Applications*, edited by R. W. Numrich (Plenum, New York, 1985), p. 215.
18. R. G. Gordon, in *Methods in Computational Physics*, Vol. 10, edited by B. Alder *et al.* (Academic Press, New York, 1971), p. 81.
19. J. C. Light, in *Methods in Computational Physics*, Vol. 10, edited by B. Alder *et al.* (Academic Press, New York, 1971), p. 111.
20. A. E. S. Green, D. L. Sellin, and A. S. Zachor, *Phys. Rev.* **184**, 1 (1969).