

De Vogelaere's Method for the Automatic Solution of Systems of Coupled Homogeneous Second-Order Differential Equations

J. L. MOHAMED

*Department of Statistics and Computational Mathematics,
University of Liverpool, Liverpool L69 3BX, United Kingdom*

Received September 23, 1982; revised September 27, 1983

Systems of coupled second-order ordinary differential equations, arising in quantum mechanical scattering problems, are frequently solved over a fixed mesh by de Vogelaere's method (a fourth-order step-by-step method) and the reactance matrix subsequently calculated. The numerical integration stage of any such problem invariably consumes a large proportion of the computer time required to solve the problem in full. Initial tests with an automatic implementation of de Vogelaere's method to calculate the reactance matrix for electron-hydrogen scattering in the "strong-coupling approximation" appear very promising, with savings of over one-half of the total number of function evaluations required in the numerical integration stage.

1. INTRODUCTION

In atomic and nuclear scattering problems we are interested in solving sets of coupled homogeneous second-order differential equations of the general form

$$\frac{d^2}{dx^2} y_i^\nu(x) = \left[\frac{l_i(l_i + 1)}{x^2} - k_i^2 \right] y_i^\nu(x) + \sum_{j=1}^N V_{ij}(x) y_j^\nu(x), \quad i = 1, \dots, N, \quad (1)$$

subject to the boundary conditions

$$y_i^\nu(x) \xrightarrow{x \rightarrow 0} x^{l_i+1} \quad (a)$$

$$y_i^\nu(x) \xrightarrow{x \rightarrow \infty} \delta_{i\nu} \sin(k_i x - \frac{1}{2} l_i \pi) + R_{i\nu} \cos(k_i x - \frac{1}{2} l_i \pi), \quad k_i^2 > 0, \quad (b1)$$

$$\xrightarrow{x \rightarrow \infty} \delta_{i\nu} \exp(-|k_i| x), \quad k_i^2 < 0, \quad (b2)$$

where the index ν serves to identify the initial state of the system. The potential functions $V_{ij}(x)$ are such that $x^2 V_{ij}(x) \rightarrow 0$ as $x \rightarrow 0$ and $V_{ij}(x) \rightarrow 0$ rapidly as $x \rightarrow \infty$. In the so-called asymptotic region ($x \geq x_a$ say) the potentials V_{ij} are negligible. R , the

symmetric reactance matrix, is the object of most calculations for atomic collision processes and the solution of Eqs. (1) generally involves three stages:

- (i) unless the differential equation solver is self starting, suitable starting values must be generated;
- (ii) the integration of the system of second-order differential equations;
- (iii) the calculation of R .

In what follows we shall assume that stage (ii) is effected by means of the direct fourth-order method of de Vogelaere [1]. By a direct method we mean a method which treats the differential equations in their original form, rather than as a larger first-order system. The question of whether it is more efficient to treat the second-order system directly or to reduce it to a first-order system is open to debate and is *not* the subject of this paper. Rather, we seek to improve on existing implementations of de Vogelaere's method for the direct solution of systems of equations of type (1). De Vogelaere's method has been used extensively (e.g., see [2-5]) to solve such equations.

In particular Chandra [6] has published a computer program which unfortunately makes substantial demands on the user: the integration mesh is specified *a priori* by the user, based on his knowledge of the expected behaviour of the solution. The accuracy of the calculation therefore depends on the chosen mesh. In addition the choice of steplength strategy is left entirely to the user; the integration mesh is divided into a number of predetermined regions and changes in steplength occur only when stepping from one region to another, and then only by a factor of 2 or $\frac{1}{2}$. No regard is paid to the estimation of the local truncation error of the method and to its subsequent control. Indeed, until recently (see [7]), despite the frequent use of de Vogelaere's method, the author was unaware of any previous error analysis for the method. In the context of the practical solution of ordinary differential equations, the basic idea of varying the steplength so that an estimate of the local truncation error of a numerical method remains less than a specified error tolerance at each step of the integration, is well developed and understood (for example, see Hull et al. [8]). Clearly, the use of such a facility should prove more efficient than computing with a constant steplength. The results of Coleman and Mohamed [7] have been used (see [9]) to provide an automatic implementation of de Vogelaere's method applied to the solution of the single-channel Schrödinger equation; in addition the method has been compared with other automatic versions of popular techniques for the solution of single-channel equations (see [10]). Appealing features of de Vogelaere's method are the low-cost provision of local error estimates and the ease with which necessary changes in steplength may be performed (see Sects. 2.1 and 2.2). The step-by-step integration of (1) consumes a large proportion of the total computing time and it is the aim of this paper to show that with minor modifications, an existing implementation (namely that of Chandra [6]) of the fixed-step de Vogelaere method for the numerical solution of the equations in (1) can be extended to provide a fully automatic method, with the advantage of substantial savings in the number of

integration steps and hence the number of function evaluations required in the numerical integration stage.

In particular, (in Sect. 3), we shall describe a two-channel test problem, previously solved by Chandra. Chandra's program can be used for problems involving open and closed channels; for the sake of completeness we give below the strategy used by Chandra to solve such problems. Our modified version will use the same strategy but adopt an in-built error control on the local error of the method.

If, following Chandra, we assume that there are n_a open channels (corresponding to $k_i^2 > 0$, $i = 1, \dots, n_a$) and $n_b = N - n_a$ closed channels (corresponding to $k_i^2 < 0$, $i = n_a + 1, \dots, N$) stages (i) and (ii) consist of the following: If the integration is started at x_0 , in the vicinity of the origin, use can be made of the initial boundary condition (1a) to provide the necessary $2N$ values for the approximate solution and its first derivative at x_0 . N different choices of the initial boundary conditions should be specified, as the general solution of (1) involves N arbitrary constants and N linearly independent solutions. Thus a total of N (hopefully) linearly independent solution vectors are generated on integrating from x_0 . Chandra chooses $y_{ij}(x_0) = \delta_{ij} x_0^{l_i+1}$ and $(d/dx) y_{ij}(x_0) = \delta_{ij} (l_i + 1) x_0^{l_i}$ for $i, j = 1, \dots, N$, with $y_{ij}(x)$ the solution at x of the i th equation solved subject to the j th boundary condition. Now find x_a such that an asymptotic expansion is valid for $x \geq x_a$. If the outward integration were continued as far as x_a the solution would contain undesirable components of $\exp(|k_i|x)$ ($i = n_a + 1, \dots, N$) in the closed channels, and for large values of x these terms would dominate the required physical solutions. The standard compromise is to integrate inwards N times from x_a to some value x_m , where the offending terms from the inward and outward integrations have not become significant; since $N + n_a$ constants can be specified arbitrarily in the asymptotic region (see Eqs. (1b)), $(N + n_a)$ different choices of the asymptotic boundary conditions will lead to a total of $(N + n_a)$ linearly independent solution vectors in the region $[x_m, x_a]$; Chandra computes the asymptotic boundary conditions at x_a as the solutions of the asymptotic form of (1). The solutions in the inner region $[x_0, x_m]$ and the outer region $[x_m, x_a]$, which Chandra generates in matrices of dimension $N \times N$ and $N \times (N + n_a)$, respectively, must now be matched at the point x_m so that the solutions and their first derivatives are continuous at x_m . Stage (iii) then requires the solution of the matching equations thus enabling the coefficients of the $n_a \times n_a$ matrix R to be extracted. (Further details of this procedure are contained in [6].)

The two-channel test problem to be considered in this paper has both channels open, so that only outward integration need be performed. However, for some reason Chandra has chosen to treat this problem somewhat artificially by performing both inward and outward integration with matching at a pre-assigned point $x_m \neq x_a$. Thus, in order to provide a meaningful comparison of Chandra's program with our automatic program for this example, we treat the problem in a similar fashion, although clearly needless inefficiency will arise in both approaches. It is hoped that future refinements to the automatic program will allow the solution of more complex problems to be considered.

In Section 2 we describe the method of de Vogelaere to solve a single equation,

with an obvious extension to systems of coupled equations. A suitable method for generating the required starting values is presented together with a means of estimating (at no extra cost) the leading term in the local truncation error at each step of the calculation. An automatic program using de Vogelaere's method to solve the single channel Schrödinger equation has previously been presented (see [9]) and the strategy described in this paper, for choosing a steplength based on the magnitude of the estimated local error, is a coupled-channel version of that adopted in [9]. Section 3 describes the two-channel test problem, previously solved by Chandra, and we comment on the performance of a modified automatic version (CHANDRAM) of Chandra's program on this problem. (CHANDRAM is a production code which is not currently available for general use.)

2. NUMERICAL INTEGRATION

2.1. De Vogelaere's Method for a Single Equation

This is a fourth-order step-by-step method designed to solve the second-order differential equation with first derivative missing:

$$\frac{d^2}{dx^2} y(x) = f(x, y(x))$$

subject to the initial conditions

$$y(x_0) = y_0, \quad z(x_0) = \left. \frac{dy}{dx} \right|_{x=x_0} = z_0.$$

For a fixed steplength h the general step of the algorithm, leading from x_{2n} to $x_{2n+2} = x_{2n} + 2h$, is as follows.

Given y_{2n} , z_{2n} , f_{2n} and f_{2n-1}

- (i) $y_{2n+1} = y_{2n} + hz_{2n} + \frac{h^2}{6} (4f_{2n} - f_{2n-1})$
- (ii) $f_{2n+1} = f(x_{2n+1}, y_{2n+1})$
- (iii) $y_{2n+2} = y_{2n} + 2hz_{2n} + \frac{h^2}{3} (4f_{2n+1} + 2f_{2n})$
- (iv) $f_{2n+2} = f(x_{2n+2}, y_{2n+2})$
- (v) $z_{2n+2} = z_{2n} + \frac{h}{3} (f_{2n} + 4f_{2n+1} + f_{2n+2})$

where

$$y_n \simeq y(x_n), \quad z_n \simeq \left. \frac{dy}{dx} \right|_{x=x_n}, \quad f_n \equiv f(x_n, y_n).$$

[The generalisation of the algorithm to the system

$$\frac{d^2}{dx^2} y_i(x) = f_i(x, y_1(x), y_2(x), \dots, y_N(x)), \quad i = 1, 2, \dots, N,$$

is a direct extension of the method for a single equation.]

De Vogelaere's method requires two function evaluations per step and makes use of an intermediate point at the midpoint of the interval $[x_{2n}, x_{2n+2}]$; the solution is predicted at this point to third order in (i) and the predicted value is then used in the corrector equation (iii) to obtain the solution correct to fourth order at the end of the interval. Equation (v) is just Simpson's rule and provides a fourth-order approximation to the derivative of the solution at the end of the interval.

Cyclic use of (i)–(v) enables the solution to be obtained over the range of integration and it only remains to start the method. The values of y_0 and z_0 above are insufficient to start the calculation and de Vogelaere [1] suggested two separate methods for providing the additional starting value.

Method 1.

$$y_{-1} = y_0 - h z_0 + \frac{h^2}{2} f_0 + \mathcal{O}(h^3)$$

$$f_{-1} = f(x_0 - h, y_{-1})$$

Now use (i)–(v) with $n = 0, 1, \dots$

Method 2.

$$\tilde{y}_1 = y_0 + h z_0 + \frac{h^2}{2} f_0 + \mathcal{O}(h^3)$$

$$\tilde{f}_1 = f(x_1, \tilde{y}_1)$$

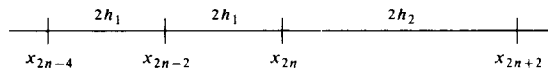
$$\tilde{\tilde{y}}_1 = y_0 + h z_0 + \frac{h^2}{6} (2f_0 + \tilde{f}_1) + \mathcal{O}(h^4)$$

Now use (ii)–(v) with $n = 0$ and (i)–(v) with $n = 1, 2, \dots$

We note here that Method 1 had been used exclusively, until very recently (see [9]), in all applications of de Vogelaere's method to solve the radial Schrödinger equation. The methods are equivalent in terms of asymptotic error estimates, both giving third-order approximations for $y(x_1)$. The leading terms in the error expansions for each method are of fourth order and are very similar in form; however, if we consider the equation $y''(x) = f(x, y(x))$ in the form $y''(x) = g(x)y(x)$, then it transpires that the fifth-order contributions to the error expansions for Methods 1 and 2 involve a factor of $g(x_{-1})$ and $g(x_1)$, respectively. For the radial Schrödinger equation with nonzero angular momentum, $|g(x)|$ is generally a rapidly decreasing

function of x near the origin. Thus $g(x_{-1})$ may be substantially greater than $g(x_1)$ and the contribution to the error from the fifth-order term may exceed that from the fourth-order term. A numerical example, given in Coleman and Mohamed [9], clearly demonstrates this effect by showing that, for sufficiently accurate starting values y_0 , z_0 (and f_0), the ratio of $g(x_{-1})$ to $g(x_1)$ is close to the factor by which the error in the approximation to $y(x_1)$ obtained by Method 1 exceeds that obtained by Method 2. We conclude that when solving the Schrödinger equation, Method 2 of starting is superior to Method 1. Chandra's program uses Method 1, the less accurate method.

An appealing feature of de Vogelaere's method is that it allows an arbitrary change of steplength without the need for further function evaluations. Suppose a steplength of $2h_1$ is used as far as x_{2n} . If now we change the steplength to $2h_2 = 2(ch_1)$



a sufficiently accurate (first-order) estimate for f at $x_{2n} - h_2$ is given by

$$\bar{f}_{2n-1} = f_{2n} + c(f_{2n-1} - f_{2n}).$$

2.2. Truncation Error Estimates

The leading term in the local truncation error in the step from x_{2n} to x_{2n+2} is $(2h^5/45)f_{2n}'''$ and an automatic routine based on de Vogelaere's method will require an estimate of this error at each step of the integration. For the case where a constant steplength of $2h$ is used over two consecutive intervals $[x_{2n-2}, x_{2n}]$ and $[x_{2n}, x_{2n+2}]$ de Vogelaere (see [1]) calculates fourth-order approximations y_{2n-1}^* , y_{2n+1}^* for $y(x_{2n-1})$, $y(x_{2n+1})$ in terms of previously calculated information:

$$y_{2n-1}^* = y_{2n} - hz_{2n} + \frac{h^2}{24} (7f_{2n} + 6f_{2n-1} - f_{2n-2})$$

$$y_{2n+1}^* = y_{2n+2} - hz_{2n+2} + \frac{h^2}{24} (7f_{2n+2} + 6f_{2n+1} - f_{2n})$$

Then a suitable linear combination of the quantities $(y_{2n+1}^* - y_{2n+1})$ and $(y_{2n-1}^* - y_{2n-1})$ may be used to estimate e , the local truncation error per unit step. We choose to control the estimated error per unit step rather than the error per step; the former quantity varies approximately as h^4 , reflecting the behaviour of the global error of the method (see [7]), whereas the latter is proportional to h^5 , reflecting the behaviour of the local truncation error. In addition when a differential equation is integrated over a fixed interval, a decrease in h results in an increased number of steps, and the adopted error criterion takes this effect into account. We have

$$e = \frac{4}{45h} [(y_{2n+1}^* - y_{2n+1}) - (y_{2n-1}^* - y_{2n-1})].$$

This expression must be modified when the steplength is changed, so as to cater for all possible sequences of steplengths (see [7]). There are three separate cases:

(a) *Immediately following a step change.*

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & 2h_1 & & 2h_1 & & 2h_2 & \\
 | & | & | & | & | & | & \\
 x_{2n-4} & x_{2n-2} & x_{2n} & & x_{2n+2} & &
 \end{array} & h_2 = ch_1 \\
 e = \frac{8c[(y_{2n+1}^* - y_{2n+1}) - \alpha(y_{2n-1}^* - y_{2n-1})]}{5h_1(12 + 7c - c^2)}, & \alpha = \frac{1}{3}c^3(2 + c).
 \end{array}$$

(b) *The second step after a step change.*

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & 2h_1 & & 2h_1 & & 2h_2 & & 2h_2 & \\
 | & | & | & | & | & | & | & | & \\
 x_{2n-4} & x_{2n-2} & x_{2n} & & x_{2n+2} & & x_{2n+4} & &
 \end{array} & h_2 = ch_1 \\
 e = \frac{4c[\beta(y_{2n+3}^* - y_{2n+3}) - (y_{2n+1}^* - y_{2n+1})]}{5h_1(1 + 5c + 3c^2)}, & \beta = \frac{\alpha}{c^4}
 \end{array}$$

(c) *Two step changes in succession.*

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & 2h_1 & & h_1 & & 2h_2 & & 2h_3 & \\
 | & | & | & | & | & | & | & | & \\
 x_{2n-4} & x_{2n-2} & x_{2n} & & x_{2n+2} & & x_{2n+4} & &
 \end{array} & h_2 = ch_1 \\
 & & & & & & & & h_3 = c_1 h_2 \\
 e = \frac{24c^2c_1^2[\beta(y_{2n+3}^* - y_{2n+3}) - \alpha_1(y_{2n+1}^* - y_{2n+1})]}{5h_3\gamma}, & \alpha_1 = \frac{1}{3}c_1^3(2 + c_1) \\
 \gamma = c^2(12 + 7c_1 - c_1^2) + c(20 + 12c_1 - 2c_1^2) + 2c_1 + 4.
 \end{array}$$

2.3. Steplength Strategy for Coupled-Channel Routine

The user is required to specify as input a parameter ϵ which is the largest allowed local error per unit step. Since the magnitude of the solution may not be known in advance, the modification (CHANDRAM) to Chandra's program (CHANDRA) calculates at each step

$$\tau_{ij} = \epsilon \cdot \max\{1, |y_{ij}|\}, \quad \begin{array}{l} i = 1, \dots, N \\ j = 1, \dots, M \end{array} \quad (2)$$

where $M = N$, $N + n_a$ in the outward, inward integrations, respectively, and y_{ij} is the calculated solution of the i th equation for the j th boundary condition at the end of the step from x_{2n-2} to x_{2n} . This provides an absolute or relative error criterion according as the calculated solution is less than or greater than unity and the step is accepted if the relation

$$|e_{ij}| \leq \tau_{ij} \quad (3)$$

is satisfied for all i and j .

Since the local error per unit step e_{ij} (corresponding to y_{ij}) varies approximately as

h^4 , a steplength of ah gives a corresponding value of approximately $\alpha^4 e_{ij}$. Thus if (3) is not satisfied for a particular i, j pair the step is rejected and the solution restarted from x_{2n-2} with a decreased steplength $\alpha^* h$; for each i, j pair which violates (3) the quantity

$$\alpha_{ij} = \left(\frac{0.5 \tau_{ij}}{|e_{ij}|} \right)^{1/4}$$

is formed, with α^* denoting the minimum of all such quantities.

If in addition to (3) the relation

$$C^4 |e_{ij}| < 0.5 \tau_{ij}$$

is satisfied for three consecutive steps (where $C > 1$), an increase in steplength, by the factor C , is performed. Here C is set to be 2 and the value 0.5 is a tuning factor. A more obvious strategy would be to choose C such that

$$C^4 \max \left| \frac{e_{ij}}{\tau_{ij}} \right| = 0.5$$

is satisfied; however, experience has shown that the adoption of such a strategy for the numerical solution of a single differential equation often results in a change of steplength being performed at almost every step. The restrictions on C are introduced to avoid frequent increases by small amounts and, much worse, increases followed by immediate decreases. (The particular choice of $C = 2$ also agrees with the factor used by Chandra for allowed increases in the steplength.)

The initial steplength for the inner region of integration must be supplied by the user and this value changes according to the strategy above. If on approaching x_m (the end point of the inner region) the current steplength h_1 would cause us to step beyond the matching point, a decrease in steplength from h_1 to h_2 is forced; the initial steplength for the outer region is then taken to be $-h_1$.

3. TEST RUN

The modified automatic version (CHANDRAM) of Chandra's FORTRAN 1V program (CHANDRA) was run for a range of values of ϵ on an IBM 370/168 computer (using double precision for all real variables) to calculate the reactance matrix for electron-hydrogen scattering in the "strong-coupling approximation," where only the 1s and 2s atomic states are included in the eigenfunction expansion, with exchange neglected. This is a two-channel problem, corresponding to Chandra's test run No. 1 (see [6]), with potentials $V_{ij}(x)$ given explicitly by

$$V_{11}(x) = -2 \left(1 + \frac{1}{x} \right) \exp(-2x)$$

$$V_{12}(x) = \frac{4\sqrt{2}}{27} (2 + 3x) \exp(-1.5x) = V_{21}(x)$$

$$V_{22}(x) = -2 \left(\frac{1}{x} + \frac{3}{4} + \frac{1}{4}x + \frac{1}{8}x^2 \right) \exp(-x)$$

CHANDRA uses the following user-supplied data:

$$N = 2 = n_a; \quad l_1 = 0 = l_2; \quad k_1^2 = 1.0 \text{ a.u.}; \quad k_2^2 = 0.25 \text{ a.u.}$$

The integration mesh is divided into 4 regions:

$[x_0, x_1]$	$[x_1, x_2]$	$[x_2, x_3]$	$[x_3, x_4]$
region 1	region 2	region 3	region 4

where

$$x_0 = 0.0304572$$

$$x_1 = x_m = 1.52286$$

$$x_2 = 3.959436$$

$$x_3 = 8.832588$$

$$x_4 = x_a = 13.70574.$$

Outward integration is performed over $[x_0, x_1]$ for $N = 2$ different choices of the initial boundary conditions and with a constant steplength of $h = 0.0304572$. Inward integration is performed over $[x_3, x_4]$, $[x_2, x_3]$, $[x_1, x_2]$ for $(N + n_a) = 4$ different choices of the asymptotic boundary conditions, with constant steplengths of $8h$, $4h$, $2h$, respectively. The values of the potential functions V_{ij} at the mesh points must be supplied by the user in advance of the calculation; since the number of points in the integration mesh is fixed so is the number of function evaluations which are required.

CHANDRAM uses the same starting points x_0 and x_4 and the same boundary conditions as in CHANDRA (see Sect. 1) with an initial steplength of $h = 0.0304572$ in the inner region. h is varied automatically during the outward and inward integrations and the parameter ε takes the following values: 10^{-n} , $n = 3, 4, 5, 6, 8$. The values of the potentials V_{ij} at the mesh points are determined automatically as the calculation proceeds, from the analytic expressions.

The elements of the 2×2 symmetric reactance matrix R_C , obtained from CHANDRA, were compared with those of the reactance matrix $R_{CM, \varepsilon}$ obtained from CHANDRAM, for the range of values of ε . By inspection the elements of R_C are best reproduced by those of $R_{CM, 10^{-4}}$. These matrices are shown below, together with $R_{CM, 10^{-8}}$.

TABLE I
Ratio of $h_{\max,i}$ to $h_{C,i}$ for Given Values of ε and i

region i	ε				
	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-8}
1	8	8	4	4	0.70
2	8	4	4	2	0.35
3	4	2	2	1	0.18
4	2	1	1	0.5	0.09

$$R_C = \begin{pmatrix} 1.138853 & 3.854753 (-1) \\ 3.854697 (-1) & -3.256686 (-1) \end{pmatrix}$$

$$R_{CM,10^{-4}} = \begin{pmatrix} 1.138839 & 3.854565 (-1) \\ 3.854490 (-1) & -3.256349 (-1) \end{pmatrix}$$

$$R_{CM,10^{-8}} = \begin{pmatrix} 1.138996 & 3.855010 (-1) \\ 3.855010 (-1) & -3.256568 (-1) \end{pmatrix}$$

The value $\varepsilon = 10^{-8}$ is extremely stringent but we may regard the computed elements of the reactance matrix to be "exact" since CHANDRAM uses steplengths which are considerably smaller than those used by CHANDRA.

To illustrate the magnitude of the steplength which CHANDRAM may use while still satisfying the appropriate error criterion, we show in Table I the ratio of the maximum steplength $h_{\max,i}$ used by CHANDRAM in region i for a given ε to the constant steplength $h_{C,i}$ used by CHANDRA in the same region. In addition the number of function evaluations n required by each program is shown in Table II.

Table I shows that greater magnifications of the steplength used by CHANDRA are allowed in CHANDRAM for the larger values of ε , as we would expect. This is also reflected in the function evaluation counts of Table II. (Note that the number of

TABLE II
Numbers of Function Evaluations Required by CHANDRA and CHANDRAM

Program	ε	n
CHANDRA	-	2004
CHANDRAM	10^{-3}	592
	10^{-4}	944
	10^{-5}	1024
	10^{-6}	1880
	10^{-8}	5500

function evaluations required by CHANDRAM could be further reduced by the adoption of a steplength control policy based on the estimated error per step, rather than on the error per unit step.)

4. CONCLUSION

The accuracy of the reactance matrix produced by CHANDRAM when $\varepsilon = 10^{-4}$ is about the same as that produced by CHANDRA, and it should be noted that for this value of ε , CHANDRAM requires less than half the number of function evaluations of CHANDRA. In practice, solutions are not normally required to high numerical accuracy, since the accuracy of the solution of the physical problem is determined to a large extent by the number of channels which are included. The results above, although limited to one test problem, do suggest that the incorporation of an automatic steplength changing facility in the solution of the radial Schrödinger equations, will bring substantial savings in the number of function evaluations required in the numerical integration stage. Since a large proportion of the computer time to solve the problem is devoted to this stage the introduction of such a facility would appear to be highly desirable; with the resulting increased efficiency in solving the differential equations, we would hope to be able to solve much larger systems of equations than has hitherto been possible.

ACKNOWLEDGMENT

This research was supported by a Research Grant from the Science Research Council.

REFERENCES

1. R. DE VOGELAERE, *J. Res. Nat. Bur. Standards Sect. 54* (1955), 119.
2. W. A. LESTER, JR., *J. Comput. Phys.* **3** (1968), 322.
3. L. VERLET, *Phys. Rev.* **159** (1967), 98.
4. S. BASAVIAH AND R. F. BROOM, *IEEE Trans. Magnetics* **MAG 11** (1975), 759.
5. J.-M. LAUNAY, *J. Phys. B* **9** (1976), 1823.
6. N. CHANDRA, *Comput. Phys. Commun.* **5** (1973), 417.
7. J. P. COLEMAN AND J. MOHAMED, *Math. Comp.* **32** (1978), 751.
8. T. E. HULL, W. H. ENRIGHT, B. M. FELLEN, AND A. E. SEDGWICK, *SIAM J. Numer. Anal.* **9** (1972), 603.
9. J. P. COLEMAN AND J. MOHAMED, *Comput. Phys. Commun.* **17** (1979), 283.
10. J. MOHAMED, Ph.D. Thesis, University of Durham, 1979.