1978

Algorithms for the Solution of Systems of Coupled Second-Order Ordinary Differential Equations

Brendan O'Shea

Technological University Dublin, brendan.oshea@tudublin.ie

Recommended Citation

O'Shea, B. (1978). Algorithms for the solution of systems of coupled second-order ordinary differential equations. *Journal of Computational and Applied Mathematics*, vol. 4, no. 1, 1978. doi:10.1016/0771-050X(78)90014-1

Algorithms for the solution of systems of coupled second-order ordinary differential equations

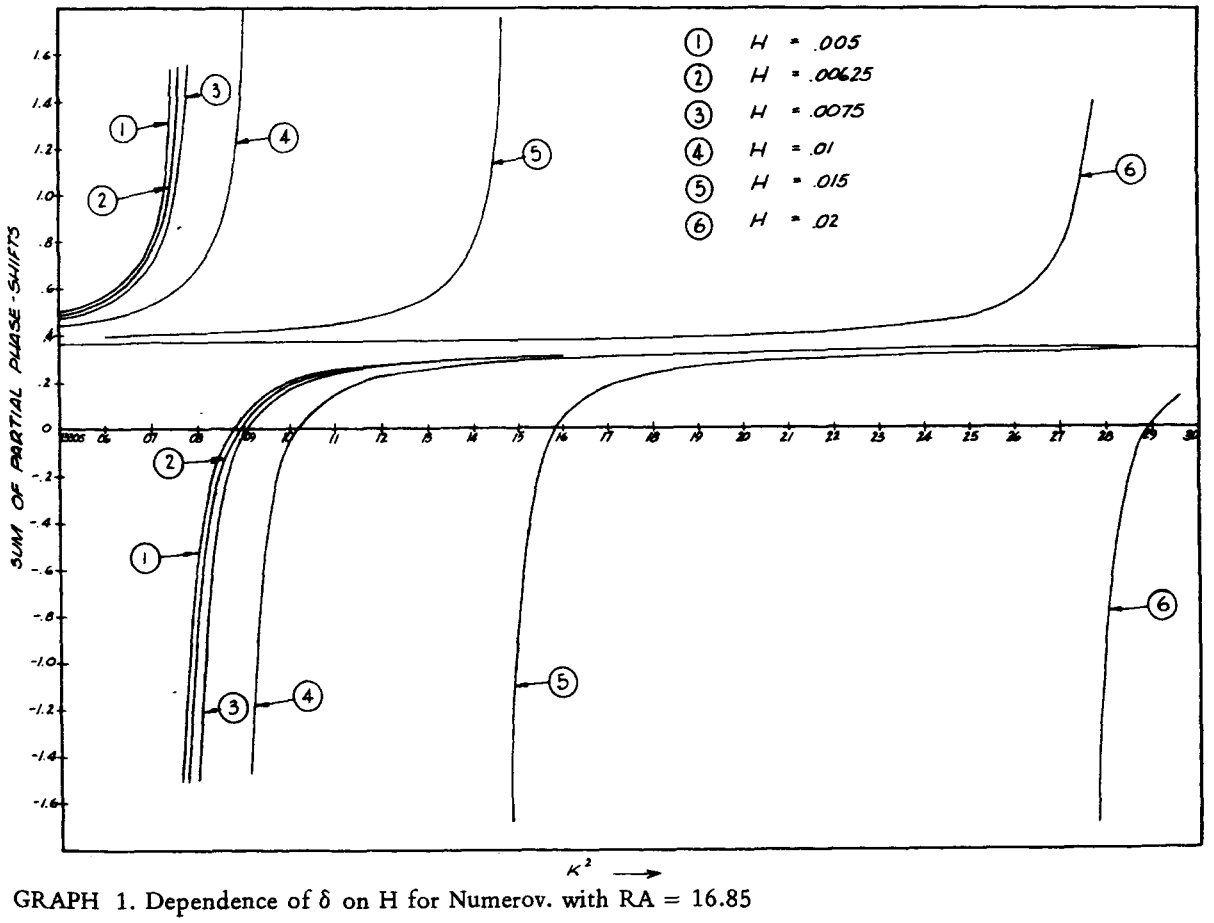
Brendan B. O'Shea (*)

ABSTRACT

Several step-by-step methods for the computer solution systems of coupled second-order ordinary differential equations, are examined from the point of view of efficiency "time-wise" and "storage-wise". Particular reference is made to a system arising in the close-coupling approximation of the Schroedinger equation. The stability of the solution is also considered.

1. INTRODUCTION

The close-coupling method, for approximating the solution of the Schroedinger equation for any scattering process, demands that a system of coupled second-order ordinary differential equations be solved numerically. In this paper, J wish to examine this system of equations, as derived by Smith et al. [1], for the scattering of electrons by atomic systems with configuration $2P^q 3P^q$. We have noticed, for some time, a certain amount of instability in these numerical solutions which was made apparent by the dependence of such physical phenomena as the phase shift, resonant positions and widths on mathematical parameters, e.g. step length, being used in the calculations (see graph 1). J will here endeavour to establish the cause of this instability and show how it may be removed.



GRAPH 1. Dependence of δ on H for Numerov. with RA = 16.85

(*) Brendan B. O'Shea, Department of Mathematics, College of Technology, Kevin Street, Dublin 8, Ireland.

Directly related to the problem of solving the relevant system of differential equations is the question of the efficiency of the integrating method "time-wise" and "storage-wise". In other words which integrating method has the smallest computer time (assuming it satisfies certain accuracy conditions) and which method demands the least amount of computer storage in execution ? These are very important when calculations are long, resulting in a big computer time, and storage requirements are large; this is the situation with ATOMNP [2], [13]. I am not aware if any such empirical investigation has previously been undertaken though Lester [3] and Froese [4] have considered the relative efficiency of various integrating algorithms as applied to very much simpler systems. Blatt [5] has dealt, in some length, with the use of the Numerov method in solving the Schoedinger equation but not in a comparative sense. Allison [14] has considered the "time-wise" element of this problem in the case of Numerov and De Vogelaere.

2. SYSTEM OF DIFFERENTIAL EQUATIONS AND METHOD OF SOLUTION

The system of coupled second-order ordinary differential equations, obtained from the Schroedinger equation, that is to be solved has already been detailed by Smith et al [1]. A computer program, called ATOMNP, for its solution and for calculating the scattering and photo-ionization cross-sections has also been published [2]; where relevant, we shall use the notation employed in ATOMNP. Hence we will confine ourselves here to a statement of the form of the system of equations and a very brief summary of the method of solution.
The equations are of the type

$$\frac{d^2F_i^{\nu}}{dr^2} = \sum_{j=1}^{M} A_{ij}^{(r)} F_j^{\nu} + \sum_{K=1}^{NE} a_K Y_{\lambda_K}(P_K F_K : r) P_K(r)$$

$$+ 2CV_i(r) + \sum_{q=1}^{NMU} m_q P_{n_q \ell_q}^{(r)} \qquad (1)$$

Subject to the boundary conditions

$$F_i^{\nu} \quad r \simeq 0 \quad r^{\ell_i + 1}$$

$$F_i^{\nu} \quad r \simeq \infty \quad k_i^{-\frac{1}{2}} [\delta_{i\nu}\sin(\theta_i) + R_{i\nu}\cos(\theta_i)] \quad k_i^2 > 0$$

$$r \simeq \infty \quad e^{-|k_i|r - |\eta_i| \ln(2k_i r)} \quad k_i^2 < 0$$

Where

$$\theta_i = k_i r - \ell_i \pi/2 - \eta_i \ln 2k_i r + \sigma_{\ell_i}$$

$$\eta_i = -(Z-N) / k_i$$

$$\sigma_{\ell_i} = \arg \Gamma(\ell_i + 1 + i\eta_i)$$

The index ν serves to identify the initial state of the system, M denotes the number of different F's i.e. the number of channels, NE is the total number of exchange terms appearing in all the F equations, NMU is the number of Lagrange multipliers involved due to orthogonality conditions, P_K represents the wave-functions of the bound states of the target. For future reference we introduce the notation, NA represents the number of open channels (i.e. those channels for which $k_i^2 > 0$), NB is the number of closed channels (i.e. those for which $k_i^2 < 0$), the parameter NV is zero if all $V_i = 0$, otherwise it is unity. The quantity NTØT = M + NE.
According to Hartree [6] the exchange functions Y_λ satisfy the second order differential equation

$$\frac{d^2(rY_\lambda)}{dr^2} = \frac{(\lambda + 1)}{r^2} (rY_\lambda) - (2\lambda + 1) \frac{P(r) F(r)}{r} \qquad (2)$$

subject to the boundary conditions

$$rY_\lambda \quad r \simeq 0 \quad r^{\lambda+1}$$

$$rY_\lambda \quad r \simeq \infty \quad r^{-\lambda}$$

It is necessary to adopt a method of inward and outward integration, with subsequent matching to obtain a final continuous solution, to enable a solution of (1) to be obtained when some of the channels are virtual: otherwise components of dominant parasitic solutions of the type $e^{|K|r}$ would be included as a result of round-off and truncation errors. The matching process involves function values at two points or the function and derivative values at one point. We adopt the former alternative. Briefly, we calculate NIN (= M + NE + NV + NMU) linearly independent solutions of the system of equations comprised of (1) and (2) for the inner region $0 < r < R1$ and store the solutions at two match points R1 and R5 (< R1). Asymptotically, NØUT (= M + NA + NE + NV + NMU) independent solutions are calculated where initial values are provided by the expansion method of Burke and Schey [7], and integration proceeds backwards until function values are obtained, and stored, at R1 and R5. A further NA + NV + NMU equations are obtained by various stratagems [1] to provide the required total of 2(M + NE) + NA + NV + NMU equations for the same number of unknown parameters. This enables us to calculate the correct linear combination of the linearly independent solutions which provides us with F_i^{ν}.

3. INTEGRATING ALGORITHMS

It is convenient to make use of matrix notation and combine (1) and (2) as

$$\frac{d^2Z}{dr^2} = BZ + G \qquad (3)$$

where

$$Z = \begin{vmatrix} F_i \\ rY_K \end{vmatrix} \quad 1 \leq i \leq M, \; 1 \leq K \leq NE$$

is a column vector which has NTØT elements, as is also G, and B is a square matrix with NTØT x NTØT elements. We may look upon (3) as a special case of the more general system

$$\frac{d^2Z}{dr^2} = f(r, Z^*) \qquad (4)$$

where f is a column vector which has NTØT elements and Z* denotes the transpose of Z.
We may replace the second-order equation (3), or (4), by an equivalent system of first order equations by defining

$$\frac{dZ}{dr} = u$$

$$\frac{du}{dr} = BZ + G$$

Hence

$$\frac{dW}{dr} = \Phi \qquad (5)$$

where

$$W = \begin{pmatrix} Z \\ U \end{pmatrix}, \quad \Phi = \begin{pmatrix} U \\ BZ + G \end{pmatrix}$$

are column vectors each with 2*NTØT elements.
The method of solution [1], [2] demands that NIN independent solutions of Z be generated in the inner region and NØUT independent solutions in the outer region. Hence in the actual numerics, Z is treated as a NTØT x NIN matrix in the inner region but as a NTØT x NØUT matrix in the outer region. However, it is not necessary to explicitely distinguish between the two regions in the general description of algorithms suitable for generating solutions of (3).
Initially J considered many integrating algorithms for generating solutions but finally rejected them all as being unsuitable, but the Numerov method, Runge-Kutta type methods (including Nystrom's algorithms) and the De Vogelaere hybrid method. As there is some disparity, at times, in the naming of these algorithms J feel it is best to state explicitly the forms of the various algorithms J have used.
The subscript n denotes that the function value is being approximated at the point r_n; H denotes the step-length of the integrating technique.

Numerov [5], [8] for $\frac{d^2Z}{dr^2} = BZ + G$

$$(1 - \frac{H^2}{12} B_{n+1}) Z_{n+1} = 2^* (1 - \frac{H^2}{12} B_n) Z_n - (1 - \frac{H^2}{12} B_{n-1}) Z_{n-1} + H^2 [B_n Z_n + \frac{1}{12} (G_{n+1} + 10G_n + G_{n-1})] \qquad (6)$$

Fourth order Runge-Kutta [9] for $\frac{dZ^2}{dr^2} = f(r, Z^*)$ (7)

where $\frac{dZ}{dr} = U$

$$Z_{n+1} = Z_n + HU_n + \frac{H^2}{6}(k_1 + k_2 + k_3)$$

$$U_{n+1} = U_n + \frac{H}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where $k_1 = f(r_n, Z_n^*)$

$$k_2 = f(r_n + \frac{H}{2}, Z_n^* + \frac{H}{2} U_n^* + \frac{H^2}{6} k_1^*)$$

$$k_3 = k_2$$

$$k_4 = f(r_n + H, Z_n^* + HU_n^* + \frac{H^2}{2} k_2^*)$$

Classical fourth-order Runge-Kutta [10] for $\frac{dW}{dr} = \Phi(r, W^*)$

$$W_{n+1} = W_n + \frac{H}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad (8)$$

where $k_1 = \Phi(r_n, W_n^*)$

$$k_2 = \Phi(r_n + \frac{H}{2}, W_n^* + \frac{k_1^*}{2})$$

$$k_3 = \Phi(r_n + \frac{H}{2}, W_n^* + \frac{k_2^*}{2})$$

$$k_4 = \Phi(r_n + H, W_n^* + k_3^*)$$

Classical fourth-order Runge-Kutta combined with a Richardson-type truncation estimate correction.
When integrating over the interval (r_n, r_{n+1}) let
(i) $W_{n+1}^{(H)}$ denote the calculated value for W_{n+1} when one application of (8) with a step-length of H is involved.
(ii) $W_{n+1}^{(H/2)}$ denote the calculated value for W_{n+1} when two applications of (8) with step-length of H/2 is involved.
Then it can be shown that a corrected value for W_{n+1} is given by

$$W_{n+1} = W_{n+1}^{(H/2)} + \frac{W_{n+1}^{(H/2)} - W_{n+1}^{(H)}}{2^4 - 1} \qquad (9)$$

Nystrom [9] *fourth-order algorithm* for $\frac{dZ^2}{dr^2} = f(r, Z^*)$

$$Z_{n+1} = Z_n + HU_n + \frac{H^2}{6}(k_1 + 2k_2)$$

$$U_{n+1} = U_n + \frac{H}{6}(k_1 + 4k_2 + k_3) \qquad (10)$$

where

$$k_1 = f(r_n, Z_n^*)$$

$$k_2 = f(r_n + \frac{H}{2}, Z_n^* + \frac{H}{2} U_n^* + \frac{H^2}{8} k_1^*)$$

$$k_3 = f(r_n + H, Z_n^* + H U_n^* + \frac{H^2}{2} k_2^*)$$

De Vogelaere [11] *algorithm* for $\frac{d^2Z}{dr^2} = f(r, Z^*)$

$$Z_{n+\frac{1}{2}} = Z_n + \frac{H}{2} U_n + \frac{H^2}{24}(4 f_n - f_{n-\frac{1}{2}})$$

$$Z_{n+1} = Z_n + HU_n + \frac{H^2}{6}(2f_{n+\frac{1}{2}} + f_n)$$

$$U_{n+1} = U_n + \frac{H}{6}(f_{n+1} + 4 f_{n+\frac{1}{2}} + f_n) \qquad (11)$$

4. COMPUTING PRELIMINARIES

In my investigation I concentrated on the problem of an electron colliding with an oxygen ion, in particular e^-0^+ ($L = 0$, $S = 1$, π = odd) and confined myself to the energy region k^2 = (.13305 - .13330) rydbergs as this incorporates a resonant energy. The width of this resonance is very narrow, Γ = .0058 Å [12] and hence the resonant energy is very sensitive to inaccuracies in the solution of (1) which makes it a very illustrative case for our purposes. Energies near threshold were not considered as the expansion method of Burke and Schey results in severe instability in the asymptotic region at these threshold energies.

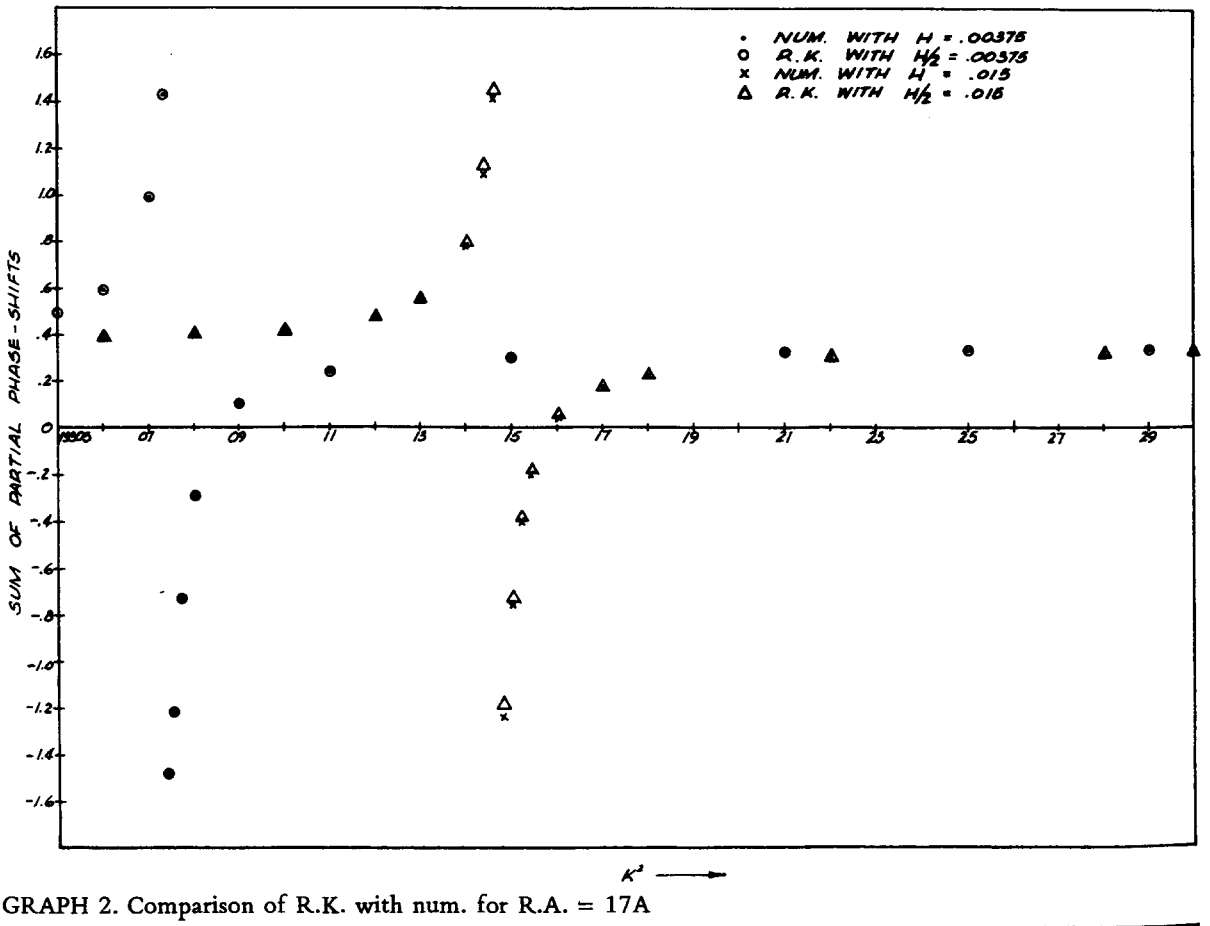
As some of the integrating algorithms demand some auxiliary technique for calculating function values at two initial points (one is usually provided by the initial boundary conditions) it was thought best, for the sake of consistency, to use the classical fourth order Runge-Kutta algorithm for this calculation in all cases (this is the technique employed in ATOMNP [2]. All calculations were computed on the IBM 360/65 with double precision. Hence round-off error is negligible and may be neglected.

5. RESULTS

ATOMNP uses the Numerov algorithm for the numerical integration; when Runge-Kutta (7) replaces the Numerov algorithm in ATOMNP the H-dependence of the phase-shift, $\delta(k^2)$, is uneffected (see graph 2). Moreover, one very interesting correlation between the solutions obtained by the two methods becomes apparent; the phase-shifts, $\delta(K^2)$, obtained from ATOMNP using the Numerov algorithm and a step-length of H are almost identical with those obtained when Runge-Kutta (7) replaces Numerov in ATOMNP but with a step-length of 2*H. A similar correlation arose between Numerov and both De Vogelaere and Runge-Kutta (8). This is probably due to the fact that the function values are approximated at all the same values of r for each method, notwithstanding the difference in step-lengths.

Reference to graph (1) shows that the "phase-shift versus energy" graphs converge to a limit as H decreases which is reasonable as it seems logical that the smaller H should provide the more accurate answer as round-off error is insignificant due to double-precision being used in all calculations. Use of the Richardson-type error estimate and its inclusion in the calculations (9) does not improve the rate of convergence of the $\delta(k^2)$-graphs which implies that the inaccuracies shown in graph 1 are not due to the gradual accumulative effect of the truncation errors over the range of integration.

It has been known for some time that the innermost region must be treated with extra care and a small

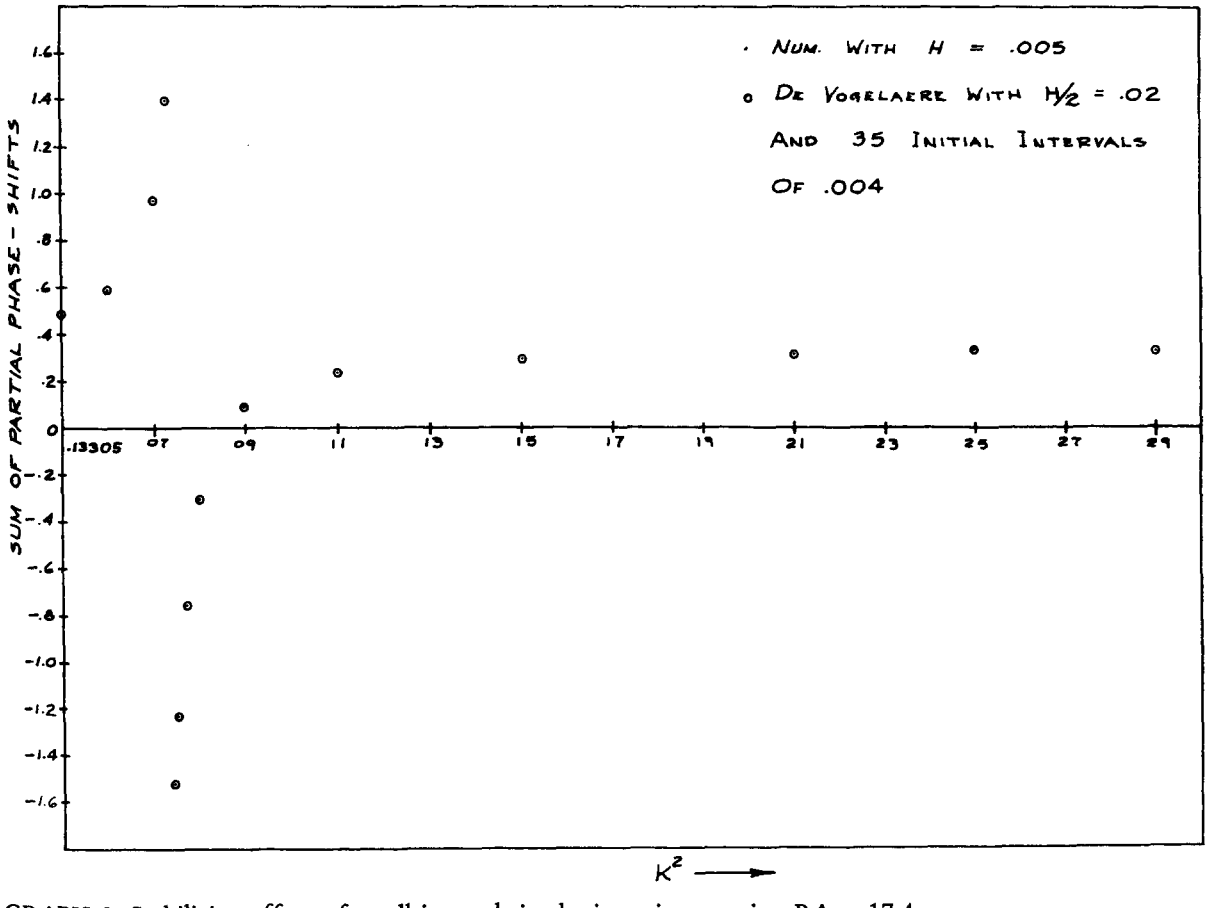


GRAPH 2. Comparison of R.K. with num. for R.A. = 17A

step-length employed there; a step-length of .005 atomic units was considered adequately small and has been the step-length used in ATOMNP. My investigation shows that even .005 a.u. is too large in the immediate neighbourhood of the origin. The optimum step-length seems to be.002 a.u. when ATOMNP uses the Numerov method (.004 suffices for Runge-Kutta and De Vogelaere algorithms) and this step-length should be retained for a distance of .015 a.u. appoximately. One can then revert to one's usual and larger step-length. This inner-inner region appears to be the key to the problem as displayed in graph 3 because the use of this extra-small step-length in the inner-inner region immediately nullifies the strong dependence of the phase-shift δ in H. In addition to stabilizing the $\delta(k^2)$-graph, the use of the extra-small step-length in the inner-inner region provides us with an extra bonus; we may use a larger than usual step-length for all regions outside the inner-inner region and still obtain a solution as accurate as with a very much smaller step-length but with the usual step-length of .005 a.u. for all the inner-region. This point will be illustrated in more detail in teh next section.

6. COMPUTER TIME

The computer times of a set of data for the e^-0^+ ($L = 0$, $S = 1$, π = odd) problem was examined when the various algorithms being investigated were used in turn with ATOMNP. Every effort was made to program these algorithms in the most efficient manner possible. The value of RA was the same for all the algorithms, where RA denotes the distance from the origin beyond which the contribution of the exchange terms is considered to be insignificant. Table 1 shows that the De Vogelaere method is very much superior to all of the others in the efficient use of computer time. This of course is partly due to the fact we can use a step-length of 2*H with De Vogelaere and get the same accuracy in calculating the phase-shift δ as we would using Numerov with a step-length of H. However, if we did use De Vogelaere with a step-length of H, we would still find that De Vogelaere was the most efficient "time-wise". This is somewhat at variance with the findings of Allison [14] but programming efficiency could have some bearing on this.



GRAPH 3. Stabilizing effect of small intervals in the inner-inner region RA = 17,4.

Table 1. Relative computer times of various algorithms for a definite set data.

METHOD	Computer time
Numerov (6) with a step-length of H	9 min 01 s
Runge-Kutta (7) with step-length of 2*H	9 min 49 s
Runge-Kutta (8) with step-length of 2*H	8 min 01 s
Nystrom (10) with a step-length of 2*H	7 min 23 s
Nystrom (10) with a Richardson-type error estimate (9) and step-length of 2*H	10 min 29 s
De Vogelaere (11) with a step-length of 2*H	4 min 14 s

A further very substantial saving of computer time is achieved if we use a small step-length of .004 a.u. in the inner-inner region, in conjunction with the De Vogelaere algorithm, which allows us to use a larger than usual step-length of .02 a.u. instead of the usual .005 a.u. as used hitherto with ATOMNP. We see from graph 3 that the resultant phase-shifts are as accurate as those calculated from ATOMNP using the Numerov algorithm and a step-size of .005, while Table 2 shows the tremendous saving of computer time that is achieved.

Table 2. Computer times for the same set of data and RA = 11.4 a.u.

METHOD	Computer time
Numerov with H = .005 a.u.	14 min 11 s
De Vogelaere with 35 initial step-lengths of .004 a.u. and the H = .02 a.u.	3 min 41 s

7. STORAGE REQUIREMENTS

We have listed in Table 3 the number of arrays required by the various algorithms in integrating the system of equations (1) over a range with a variable step-length (the variations of step-length is confined to halving as we integrate inwards from the asymptotic region). These arrays may be quite large, in ATOMNP we consider arrays of 800 elements, each element being in double-precision when the code runs on IBM 360/65.

Table 3. Storage requirements of various algorithms

METHOD	storage requirements in arrays
Numerov (6)	10
Runge-Kutta (7)	7
Runge-Kutta (8)	7
Runge-Kutta (8) and a Richardson-type estimate (9)	9
Nystrom (10)	6
Nystrom (10) and a Richardson-type error estimate (9)	8
De Vogelaere (11)	5

De Vogelaere is again the optimum method, this time it has the minimum storage requirements. One may note that 5 storage-arrays are required, as the system of equations (1) is coupled, rather than the usual 4 arrays when the equations are uncoupled. However, there is still a saving of 50 % in storage requirements over the Numerov method.

8. SUMMARY

In the numerical integration of the system of equations (1), one should use a very small step-length in the immediate neighbourhood of the origin. A suitable step-length is .004 a.u. when De Vogelaere or a fourth-order Runge-Kutta type algorithm is being used in the integration or .002 a.u. when Numerov is being used; this step-length should be retained for a distance of approximately 1.5×10^{-2} a.u. A much larger step-length may then be used for the remainder of the integration. The most efficient integrating algorithm is that of De Vogelaere both from the point of view of storage requirements and computer time considerations.

BIBLIOGRAPHY

1 (a) SMITH, K.; HENRY, R. J. W. and BURKE, P. G. : Phys. Rev., *147*, No 1. 21, (1966).

1 (b) SMITH K.; CONNEELY, M. J. and MORGAN, L. A. : Phys. Rev., *177*, 196 (1969)

2. CONNEELY, M. J.; LIPSKY, L.; SMITH, K.; BURKE, P. G.; HENRY, R. J. W. : Computer Phys, Comm., 1, No 5, 306 (1970).

3. LESTER, W. A.; J. of Comp. Phys., *3*, 322 (1968).

4. FROESE, C. : Can J. of Phys., *42*, 1895, (1963).

5. BLATT, J. M. : J. of Comp. Phys., 7, 382, (1967).

6. HARTREE, D. R.: *The calculation of atomic structures,* John Wiley & Sons, Inc., New York, (1957).

7. BURKE, P. G. and SCHEY, H. M. : Phys. Rev., *126,* 163, (1962).

8. COWELL, P. H. and CROMMELIN, A. C. D. : Appendix to Greenwich Observs. for 1909, 84, (1910).

9. NYSTROM, E. J., Acta Soc. Sci. Fenn., *50,* NR 13, 1, (1925).

10. KUTTA, W. : Z. Math. Phys., *46,* 435, (1901).

11. DE VOGELAERE, R. : J. Res. Nat. Bur. Stan., *54,* 119, (1955).

12. HENRY, R. J. W. : Planet Space Sci., *16,* 1503 (1968).

13. SMITH, K. : *Physics of electronic and atomic collisions,* VII ICPEAC, 433 (1971) - North-Holland (1972).

14. ALLISON, A. C. : J. of Comp. Phys. *6,* 378 (1970).