JOURNAL OF COMPUTATIONAL PHYSICS 91, 413–423 (1990)

A Sixth-Order Exponentially Fitted Method for the Numerical Solution of the Radial Schrödinger Equation

J. R. CASH

Department of Mathematics, Imperial College, South Kensington, London SW7 2AZ, England

AND

A. D. RAPTIS AND T. E. SIMOS

Department of Mathematics, National Technical University of Athens, Zografou Campus, Athens 157 73, Greece

Received January 4, 1989; revised October 27, 1989

A new sixth-order method of Runge–Kutta type is developed for the numerical integration of the single channel radial Schrödinger equation. The formula derived contains certain free parameters which allows it to be fitted automatically to exponential functions. Extensive numerical testing on the resonance problem and on the bound states problem indicates that this new method is generally more efficient than other previously developed finite difference methods.

INTRODUCTION

In recent years there has been considerable interest in the numerical solution of the one-dimensional Schrödinger equation,

$$y''(r) = [l(l+1)/r^2 + V(r) - k^2]\, y(r), \tag{1.1}$$

where one boundary condition is $y(0) = 0$ with the other boundary condition being specified at $r = \infty$. Equations of this type occur very frequently in theoretical physics, for example [1], and there is a real need to be able to solve them both efficiently and reliably by numerical methods. In (1.1) the function $l(l+1)/r^2 + V(r)$ is the effective potential, which tends to zero with increasing r, and k^2 is a real number denoting the energy. Boundary value methods based on either collocation or finite differences (so-called global methods) are not very popular for the solution of (1.1) due to the fact that the problem is posed on an infinite interval. Initial value approaches, such as shooting, need to take into account the fact that $|y'(r)|$ is very large near $r = 0$. It is therefore inappropriate to use standard library packages for

0021-9991/90 $3.00

the solution of (1.1) and, as a result, many alternative methods have been proposed in an attempt to solve (1.1) efficiently. One of the most popular of these formulae is Numerov's method which is the optimal linear two step method [2]. Although Numerov's method is only of order four it has been found in practice to have a performance which is generally superior to higher order four step methods. It was postulated in [3] that one of the reasons for this is that Numerov's method has a relatively large interval of periodicity whereas the intervals of periodicity of linear k-step methods tend to decrease as k increases. It therefore seems that the investigation of linear multistep methods is not a fruitful way of deriving efficient high order methods for (1.1).

An alternative approach to deriving high order methods for (1.1) was given by Cash and Raptis [4]. In [4] a sixth-order Runge–Kutta type method with a large interval of periodicity was derived. The theoretical results derived in that paper suggested that this new method would be superior to Numerov's method on a wide class of problems of the form (1.1) and this expectation was borne out in practice by numerical testing. A novel aspect of the approach described in [4] was the idea of using a variable step algorithm based on an embedded error estimate. This error estimate was obtained very efficiently by taking the difference of the solutions obtained using Numerov's method and the sixth-order Runge–Kutta-like method and the numerical results showed there is a big advantage to be gained by using variable step, rather than fixed step, formulae for the integration of (1.1).

An alternative approach for developing efficient methods for the solution of (1.1) is to use exponential fitting. This approach is particularly appropriate because for large r the solution of (1.1) is periodic. An early investigation of exponential fitting was by Raptis and Allison [5] who derived exponentially fitted formulae based on Numerov's method. Numerical results presented in [5] indicate that these fitted methods are considerably more efficient than Numerov's method for the solution of (1.1). Since the work of Raptis and Allison, the idea of exponential fitting has been investigated and extended by many authors. Perhaps the most significant work in this general area was that of Ixaru and Rizea [6]. In particular they showed that for the resonance problem defined by (1.1) it is generally more efficient to derive methods which exactly integrate functions of the form:

$$\{1, r, r^2, \ldots, r^s, \exp(\pm\omega r), r\exp(\pm\omega r), r^2\exp(\pm\omega r), \ldots, r^m\exp(\pm\omega r)\} \quad (1.2)$$

than to use classical exponential fitting methods. A powerful low order method of this type was developed by Raptis [7]. However, efficient higher order formulae are much harder to derive. The difficulty is that as the integers m and s increase, more and more free parameters are required in the integration formula to allow the fitting to be carried out. This in turn means that if a linear k-step formula is used the value of k will increase with m and s and, as pointed out earlier, linear k-step methods with $k > 2$ have largely proved to be inefficient for the solution of (1.1). An additional problem with linear k-step methods with $k > 2$ is that step changing tends to be complicated and there are extra difficulties with starting due to the need to derive additional initial conditions.

An approach which overcomes all of these problems is to combine the ideas of [4, 6, 7] and to use exponential fitting with Runge–Kutta-like methods. Such formulae allow several free parameters for fitting without requiring that the step numbers be increased above 2. The purpose of this paper is to investigate the possibility of deriving Runge–Kutta methods fitted to (1.2) and in particular to derive and test a method with $m=1$ and $s=3$. Following the work of Ixaru and Rizea [6] we would expect this new method to be superior to methods based on classical exponential fitting for a large class of problems of the form (1.1). Numerical results presented in Section 3 show that this expectation is borne out in practice with the new formula being considerably more accurate than previously developed finite difference formulae.

2. Derivation of the Methods

The one-dimensional radial Schrödinger equation can be written in the form:

$$y''(r)=f(r)\,y(r), \qquad r\in[0,\infty), \tag{2.1}$$

where $f(r)=W(r)-E$, $W(r)=l(l+1)/r^2+V(r)$ is an effective potential with $W(r)\to 0$ as $r\to\infty$ and E is a real number denoting the energy. The Schrödinger equation is normally posed in boundary value form with one boundary condition being $y(0)=0$ and the other boundary condition being specified at large r. Since both $V(r)$ and $l(l+1)/r^2$ tend to zero for large values of r it follows that the function f can be approximated by a constant for $r>r_c$ say and so the solution is nearly periodic for $r>r_c$. It is this periodicity of the solution that exponential fitting attempts to exploit.

The numerical integration method which we will derive in this paper is of the form

$$y_{n+1}+ay_n+y_{n-1}=h^2[\beta_0(y''_{n+1}+y''_{n-1})+\beta_1(y''_{n+1/2}+y''_{n-1/2})+\gamma y''_n], \tag{2.2}$$

where, for example, $y''_{n+1}\equiv f(r_{n+1})\,y_{n+1}$ with $r_{n+1}=r_n+h$.

We require that the method (2.2) should be exact for any linear combination of the functions:

$$\{1, r, r^2, r^3, \exp(\pm\omega r), r\exp(\pm\omega r)\} \tag{2.3}$$

(that is, of the form (1.2) with $s=3$, $m=1$). Generally speaking it is the fact that the formula exactly integrates all polynomials of degree $\leqslant 3$ which is important for $r\leqslant r_c$ whereas it is the fact that it integrates exactly $r^i\exp(\pm\omega r)$, $i=0, 1$, that is important for $r>r_c$. Perhaps the simplest way of deriving a formula which integrates the functions (2.3) exactly is to derive one which integrates

$$[1, r, \exp(\pm\omega_0 r), \exp(\pm\omega_1 r), \exp(\pm\omega_2 r)] \tag{2.4}$$

exactly and then put $\omega_0 = 0$, $\omega_1 = \omega_2 = \omega$ [13]. Putting $a = -2$ so that the functions 1 and r are integrated exactly and demanding that (2.2) integrates (2.4) exactly, we obtain the system of equations for β_0, β_1, and γ:

$$2w_j^2\beta_0\cosh(w_j) + 2w_j^2\beta_1\cosh(w_j/2) + w_j^2\gamma = 2(\cosh(w_j) - 1), \qquad j = 0, 1, 2, \tag{2.5}$$

where $w_j = \omega_j h$.

Solving for β_0, β_1, and β_2 and then setting $\omega_0 = 0$, $\omega_1 = \omega_2 = \omega$, we obtain

$$\begin{aligned}
\beta_0 &= (-8w(2 - 2\cosh(w) + \cosh(3w/2) - \cosh(w/2)) - 2w^2(4\sinh(w) - \sinh(3w/2) \\
&\quad -5\sinh(w/2)) + 2w^4\sinh(w/2))/D \\
\beta_1 &= (8w(3 - 4\cosh(w) + \cosh(2w)) - 4w^4\sinh(w))/D \qquad (2.6) \\
\gamma &= (-16w(1 - 2\cosh(w) + \cosh(2w) - \cosh(3w/2) + \cosh(w/2)) + 4w^2(4\sinh(w) \\
&\quad -\sinh(3w/2) - 5\sinh(w/2)) + 2w^4(\sinh(3w/2) + 3\sinh(w)))/D,
\end{aligned}$$

where

$$D = -2w^4(4\sinh(w) - \sinh(3w/2) - 5\sinh(w/2))$$

with $w = \omega h$.

In order to use formula (2.2) in practice we need to find computable approximations to the terms $y_{n+1/2}$, $y_{n-1/2}$, where we still require the method (2.2) to be exact for any linear combination of the functions (2.3). Following the approach of [4] we look for approximations of the form

$$\begin{aligned}
y_{n+1/2} + y_{n-1/2} &= A_1[y_{n+1} + y_{n-1}] + A_2 y_n + h^2\{A_3[y''_{n+1} + y''_{n-1}] + A_4 y''_n\} \\
y_{n-1/2} &= B_1[y_{n+1} + y_{n-2}] + B_2[y_n + y_{n-1}] + h^2\{B_3[y''_{n+1} + y''_{n-2}] \\
&\quad + B_4[y''_n + y''_{n-1}]\},
\end{aligned}$$

where the constants A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4 are to be determined. In order for these approximations to be exact for (2.3) it can be shown that we need

$$\begin{aligned}
A_1 &= (-8w(2 - 2\cosh(w) + \cosh(3w/2) - \cosh(w/2)) + 2w^2R + w^4\sinh(w))/X \\
A_2 &= (-16w(1 - 2\cosh(w) + \cosh(2w) - \cosh(3w/2) + \cosh(w/2)) \\
&\quad -4w^2R + 6w^4\sinh(w))/X \\
A_3 &= (2w(3 + \cosh(w) - 4\cosh(w/2)) - w^2(\sinh(w) - 2\sinh(w/2)) - 2R)/X \\
A_4 &= (-2w(1 + 6\cosh(w) + \cosh(2w) - 4\cosh(3w/2) - 4\cosh(w/2)) - 2w^2(3\sinh(w) \\
&\quad -\sinh(3w/2) - 3\sinh(w/2)) + 4R)/X
\end{aligned}$$

with

$$
\begin{aligned}
R &= 4\sinh(w) - \sinh(3w/2) - 5\sinh(w/2)\\
X &= -8w(3 - 4\cosh(w) + \cosh(2w)) + 4w^4\sinh(w)\\
B_1 &= (-16w(1 - \cosh(2w) + 2\cosh(3w/2) - 2\cosh(w/2)) + 8w^2T - w^4S)/Y\\
B_2 &= (-16w(1 - \cosh(w) - \cosh(2w) + \cosh(3w) - 2\cosh(3w/2) + 2\cosh(w/2))\\
&\quad -8w^2T + 9w^4S)/Y \qquad (2.7)\\
B_3 &= (2w(9 + 8\cosh(w) - \cosh(2w) - 16\cosh(w/2)) + w^2(2\sinh(w) + \sinh(2w)\\
&\quad -8\sinh(w/2)) - 8T)/Y\\
B_4 &= (2w(1 - 9\cosh(w) - 9\cosh(2w) + \cosh(3w) + 16\cosh(3w/2)) - 3w^2(6\sinh(w)\\
&\quad +3\sinh(2w) - 8\sinh(3w/2)) + 8T)/Y
\end{aligned}
$$

and

$$
\begin{aligned}
S &= 2\sinh(w) + \sinh(2w)\\
T &= 2\sinh(w) + \sinh(2w) - 3\sinh(3w/2) + \sinh(w/2)\\
Y &= -32w(2 - \cosh(w) - 2\cosh(2w) + \cosh(3w)) + 16w^4S.
\end{aligned}
$$

This specifies completely our integration formula (2.2). However, the above formulae are subject to heavy cancellations for small values of $w = \omega h$. In this case it is much more convenient to use the series expansions for the coefficients of the formula. These are laborious to derive and in what follows we merely list our results.

```
β0=1/60-w2/3780+13w4/3628800-41w6/958003200+19457w8/41845579776000

β1=4/15+w2/945+17w4/907200+w6/29937600-3077w8/10461394944000

γ=13/30-w2/630-w4/22400+w6/53222400-2383w8/6974263296000


A1=3/32-43w2/32256+151w4/32514048+24539w6/450644705280-686657w8/
   1073681494179840

A2=29/16+43w2/16128-151w4/16257024-24539w6/225322352640+686657w8/
   536840747089920

A3=-1/384+43w2/387072-6173w4/1950842880+442543w6/5407736463360-
   296084471w8/141725957231738880

A4=31/192-215w2/193536+1643w4/975421440-589777w6/2703868231680+
   341403833w8/70862978615869440
```

```
B1=37/128-699w2/14336+31823w4/4816896-55748531w6/66762178560+
   177383276499w8/1749703175700480

B2=27/128+699w2/14336-31823w4/4816896+55748531w6/66762178560-

   177383276499w8/1749703175700480

B3=-9/512+233w2/57344-188473w4/289013760+72603677w6/801146142720-
   245121791839w8/20996438108405760

B4=-171/512+2563w2/57344-1720907w4/289013760+10843249w6/14566293504-
   1883477525549w8/20996438108405760                                  (2.8)
```

where $w2 = w^2$, $w4 = w^4$, $w6 = w^6$, $w8 = w^8$.

Having derived our numerical method we wish to demonstrate its efficiency on some problems of practical interest. This we do in the next section.

3. Numerical Results

In this section we present some numerical results to illustrate the performance of our method. We consider the numerical integration of the Schrödinger equation:

$$y''(r) = (V(r) - E)\, y(r) \tag{3.1}$$

in the well-known case where the potential $V(r)$ is the Woods–Saxon potential

$$W(r) = V(r) = u_0/(1+t) - u_0 t/[a(1+t)^2] \tag{3.2}$$

with $t = \exp((r - R_0)/a)$, $u_0 = -50$, $a = 0.6$, and $R_0 = 7.0$. In order to solve this problem numerically we need to approximate the true (infinite) range of integration $[0, \infty)$ by a finite range. For the purpose of our numerical computation we take the domain of integration as $0 \leqslant r \leqslant 15$. We consider Eq. (3.1) in a rather large domain of energies, i.e., $E_{\min} = -50$, $E_{\max} = 1010$. The problems we consider are (a) the so-called bound states problem and (b) the so-called resonance problem.

We first consider:

(a) *The Bound States Problem*

For negative energies we solve the so-called bound states problem, i.e., with the boundary conditions $y(0) = 0$ and $y(r) \approx \exp(-\sqrt{-E}r)$ for large values of r. In order to solve this problem numerically we use a strategy which has been proposed by Cooley [8] and subsequently has been improved by Blatt [9]. Basically this involves integrating forward from the point $r = 0$, backward from the point $r_b = 15$

and matching up the solution at some internal point in the range of integration. As initial conditions for the backward integration we take [6]

$$y(r_b) = \exp(-\sqrt{-E}\, r_b) \qquad \text{and} \qquad y(r_b - h) = \exp[-\sqrt{-E}\,(r_b - h)],$$

where h is the steplength of integration of the numerical method.

In order to use method (2.2) we need to compute one extra "initial" condition at $r_b - 2h$ for the backwards integration and this is obtained using the method developed in Raptis and Allison [5]. Similarly, in the forward direction we take $y(0) = 0$, $y(h) = h$, and we compute a numerical solution at $2h$ using the method of [5].

TABLE I

Absolute Errors, in 10^{-9} Units, of the Negative Eigenvalues Calculated by the Seven Algorithms I, II, III, IV, V, VI, VII

Method:		I	II	III	IV	V	VI	VII
The eigenvalue	h	Absolute error in 10^{-9} units (Real time of Computation in CPU's)						
-49.457788728	1/8	236	49	5	0	15	0	0
		(0.02)	(0.02)	(0.02)	(0.03)	(0.07)	(0.05)	(0.05)
	1/16	13	2	0	0	1	0	0
		(0.042)	(0.045)	(0.045)	(0.062)	(0.138)	(0.110)	(0.111)
	1/32	1	0	0	0	0	0	0
		(0.080)	(0.088)	(0.088)	(0.130)	(0.278)	(0.221)	(0.230)
-41.232607772	1/8	494883	579	173	53	5874	15	0
		(0.023)	(0.021)	(0.02)	(0.025)	(0.065)	(0.047)	(0.052)
	1/16	30798	61	12	7	330	0	0
		(0.042)	(0.045)	(0.045)	(0.063)	(0.130)	(0.110)	(0.108)
	1/32	1918	5	3	0	22	0	0
		(0.088)	(0.088)	(0.088)	(0.130)	(0.275)	(0.220)	(0.230)
-26.873448915	1/8	29910934	15082	1722	213	85567	85	12
		(0.022)	(0.022)	(0.02)	(0.022)	(0.063)	(0.05)	(0.05)
	1/16	549634	1297	127	15	4660	4	0
		(0.043)	(0.045)	(0.045)	(0.06)	(0.128)	(0.108)	(0.111)
	1/32	34234	183	15	1	576	0	0
		(0.088)	(0.088)	(0.088)	(0.129)	(0.278)	(0.219)	(0.230)
-8.676081670	1/8	47212345	268957	15323	1846	606785	123	45
		(0.025)	(0.025)	(0.02)	(0.025)	(0.065)	(0.05)	(0.05)
	1/16	2895215	1877	1543	213	31526	21	1
		(0.044)	(0.045)	(0.045)	(0.062)	(0.130)	(0.110)	(0.112)
	1/32	178233	393	113	21	6768	1	0
		(0.090)	(0.090)	(0.090)	(0.130)	(0.252)	(0.21)	(0.215)

Note. Real time of computation for any negative eigenvalue for the seven algorithms in CPUs. Potential (3.2). Bound states problem.

In Table I we list the results obtained for seven numerical methods:

Method I. Numerov's method
Method II. Derived by Raptis and Allison [5]
Method III. Derived by Ixaru and Rizea [6]
Method IV. Derived by Raptis [7]
Method V. The algorithm CV_2 derived by Ixaru and Berceanu [10]
Method VI. Derived by Raptis and Cash [3]
Method VII. Proposed in this paper.

The true solutions to the Woods–Saxon bound states problem were obtained corrected to nine decimal places using the analytic solution and the numerical results obtained for the seven methods were compared to this true solution. Table I shows the absolute errors of the eigenenergies in 10^{-9} units and also shows the real time of computation for different choices of constant stepsize (which are displayed in column 2). The empty areas indicate that the corresponding absolute errors are larger than 1.

The performance of the different methods is dependent on the choice of the fitting parameter ω. For the purpose of obtaining our numerical results it is appropriate to choose ω in the way suggested by Ixaru and Rizea [6]. That is, we choose:

$$\omega = \begin{cases} (-50-E)^{1/2} & \text{for } r \in [0, 6.5] \\ (-E)^{1/2} & \text{for } r \in (6.5, 15]. \end{cases}$$

For a discussion of the reasons for choosing the values 50 and 6.5 and the extent to which the results obtained depend on these values see [6, p. 25].

(b) *The Resonance Problem*

For positive energies one has the so-called resonance problem. This involves finding all values of E in the range $1 \leqslant E \leqslant 1000$ for which the phase shift $\delta(E)$ is equal to $\pi/2$. We follow the same strategy as in case (a). Since the solution for large r is $y(r) \approx A\cos(\sqrt{E}r)$ we take as our initial conditions for the backward integration [6]:

$$y(r_b) = \cos(\sqrt{E}\, r_b) \qquad \text{and} \qquad y(r_b - h) = \cos[\sqrt{E}\,(r_b - h)].$$

In order to use method (2.2) we need to compute one extra "initial" condition at $r_b - 2h$ for the backwards integration and this is obtained using the method developed in Raptis [7]. Similarly in the forward direction we take $y(0) = 0$, $y(h) = h$, and we compute a numerical solution at $2h$ using the method of [7].

In Table II we list the results obtained for the same methods as in the case (a).

The true solutions to the Woods–Saxon resonance problem were obtained corrected to six decimal places using the analytic solution and the numerical results obtained for the seven methods were compared to this true solution. Table II shows the absolute errors of the eigenenergies in 10^{-6} units and also shows the real time of computation for different choices of constant stepsize (which are displayed in column 2). The empty areas indicate that the corresponding absolute errors are larger than 1. The choice of ω is the same as in case (a).

TABLE II

Absolute Errors, in 10^{-6} Units, of the Resonances Calculated by the Seven Algorithms I, II, III, IV, V, VI, VII

Method:		I	II	III	IV	V	VI	VII
The resonance	h	Absolute error in 10^{-6} units (Real time of Computation in CPU's)						
53.588872	1/16	228323	4828	811	313	323	11	1
		(0.09)	(0.082)	(0.070)	(0.070)	(0.111)	(0.102)	(0.120)
	1/32	14059	324	61	28	22	5	0
		(0.191)	(0.148)	(0.152)	(0.152)	(0.207)	(0.200)	(0.235)
	1/64	870	85	4	1	8	0	0
		(0.371)	(0.297)	(0.289)	(0.301)	(0.405)	(0.405)	(0.474)
163.215341	1/16	------	67950	6298	633	1921	117	5
			(0.078)	(0.082)	(0.082)	(0.109)	(0.100)	(0.125)
	1/32	476488	4516	461	55	123	12	1
		(0.191)	(0.148)	(0.152)	(0.152)	(0.203)	(0.199)	(0.250)
	1/64	29538	288	32	3	45	2	0
		(0.359)	(0.289)	(0.300)	(0.301)	(0.405)	(0.408)	(0.485)
341.495874	1/16	------	494722	28421	1284	7215	2542	37
			(0.082)	(0.078)	(0.078)	(0.112)	(0.098)	(0.135)
	1/32	------	32166	1628	115	423	85	2
			(0.152)	(0.148)	(0.152)	(0.207)	(0.199)	(0.235)
	1/64	435752	2081	112	8	89	5	1
		(0.362)	(0.289)	(0.289)	(0.301)	(0.410)	(0.405)	(0.474)
989.701916	1/16	------	566260	297804	2228	58695	------	681
			(0.082)	(0.082)	(0.082)	(0.111)		(0.135)
	1/32	------	40128	14608	304	3651	2428	55
			(0.152)	(0.148)	(0.152)	(0.210)	(0.205)	(0.250)
	1/64	------	3115	854	21	251	115	7
			(0.289)	(0.293)	(0.301)	(0.411)	(0.410)	(0.485)

Note. Real time of computation for any resonance for the seven algorithms in CPUs. Potential (3.2).

In Table III some results obtained with another potential in Eq. (3.1) are shown. This potential is

$$V(r) = V_W(r) + C/r + l(l+1)/r^2, \tag{3.3}$$

where V_W is the Woods–Saxon potential (3.2). For the purpose of our experiments we use the same parameters as in [6], i.e., $C = 20$, $l = 2$.

Since $V(r)$ is singular at the origin, we use the special strategy of [6]. We start the forward integration from a point $\varepsilon > 0$ and the initial values $y(\varepsilon)$ and $y(\varepsilon + h)$ for the integration scheme are obtained by a perturbative method (see ref. [11]). As in [6] we use the value $\varepsilon = \frac{1}{4}$ for our numerical experiments.

TABLE III

Absolute Errors, in 10^{-6} Units, of the Positive Eigenvalues Calculated by the Seven Algorithms I, II, III, IV, V, VI, VII

Method:		I	II	III	IV	V	VI	VII
The eigenvalue	h	Absolute error in 10^{-6} units (Real time of Computation in CPU's)						
61.482588	1/16	253692	24140	3244	1265	3588	275	11
		(0.092)	(0.090)	(0.090)	(0.090)	(0.153)	(0.100)	(0.130)
	1/32	15621	1472	235	108	244	115	5
		(0.194)	(0.168)	(0.172)	(0.160)	(0.305)	(0.200)	(0.290)
	1/64	967	386	14	4	86	72	0
		(0.390)	(0.328)	(0.332)	(0.332)	(0.620)	(0.398)	(0.480)
173.075711	1/16	------	295434	41986	5200	41332	2925	115
			(0.090)	(0.090)	(0.090)	(0.160)	(0.102)	(0.120)
	1/32	618815	18816	2974	327	7754	300	35
		(0.194)	(0.168)	(0.160)	(0.160)	(0.322)	(0.205)	(0.280)
	1/64	38361	1200	193	18	856	48	0
		(0.391)	(0.332)	(0.320)	(0.321)	(0.673)	(0.400)	(0.480)
352.682070	1/16	------	------	203007	9171	180375	63550	856
				(0.082)	(0.082)	(0.160)	(0.102)	(0.135)
	1/32	------	123715	12523	885	10575	2125	117
			(0.160)	(0.172)	(0.172)	(0.335)	(0.200)	(0.280)
	1/64	573357	7707	889	64	1756	125	7
		(0.390)	(0.320)	(0.328)	(0.328)	(0.660)	(0.405)	(0.480)
1002.768393	1/16	------	------	------	545667	------	------	80927
					(0.090)			(0.130)
	1/32	------	------	162311	3377	182550	60700	813
				(0.172)	(0.160)	(0.210)	(0.205)	(0.280)
	1/64	------	11125	10675	263	82754	2875	17
			(0.320)	(0.293)	(0.320)	(0.411)	(0.405)	(0.495)

Note. Real time of computation for any positive eigenvalue for the seven algorithms in CPUs. Potential (3.3).

For the purpose of obtaining our numerical results it is appropriate to choose ω in the way suggested by Ixaru and Rizea [6]. That is we choose

$$\omega = \begin{cases} [V(a_1) + V(\varepsilon)]/2 & \text{for } r \in [\varepsilon, a_1] \\ V(a_1)/2 & \text{for } r \in (a_1, a_2] \\ V(a_3) & \text{for } r \in (a_2, a_3] \\ V(15) & \text{for } r \in (a_3, 15], \end{cases}$$

where a_1 is taken so that $V(a_1) = V(\varepsilon)/2$, a_2 is approximately the first node of $V(x)$, and $a_3 = 6.25$ is the point where $V(x)$ is approximately half of its minimum negative value.

The asymptotic solution is in our case the Coulomb function G. The values of G at r_b and $r_b - h$, needed for backward integration, have been obtained by the subroutine RCWFN (see ref. [12]).

All computations were carried out on the Micro-Vax II of the Department of Mathematics of the National Technical University of Athens, using double precision arithmetic (16 significant digits accuracy). It can be seen from the results presented in Tables I–III that the new formula is considerably more accurate than the other numerical methods that we have considered. In particular, the results obtained for the method presented in this paper are considerably better than those obtained for the exponentially fitted method VI (although we have been unfair to VI in that it is a variable step method and we have carried out our comparisons using a fixed step) and this bears out the theory developed in [6].

REFERENCES

1. L. D. LANDAU AND F. M. LIFSHITZ, *Quantum Mechanics* (Pergamon, New York, 1965), p. 54.
2. A. C. ALLISON, *J. Comput. Phys.* **6**, 378 (1970).
3. A. D. RAPTIS AND J. R. CASH, *Comput. Phys. Commun.* **44**, 95 (1987).
4. J. R. CASH AND A. D. RAPTIS, *Comput Phys. Commun.* **33**, 299 (1984).
5. A. RAPTIS AND A. C. ALLISON, *Comput. Phys. Commun.* **14**, 1 (1978).
6. L. GR. IXARU AND M. RIZEA, *Comput. Phys. Commun.* **19**, 23 (1980).
7. A. D. RAPTIS, *Computing* **28**, 373 (1982).
8. J. W. COOLEY, *Math. Comput.* **15**, 363 (1961).
9. J. M. BLATT, *J. Comput. Phys.* **1**, 382 (1967).
10. L. GR. IXARU AND S. BERCEANU, *Comput. Phys. Commun.* **44**, 11 (1987).
11. L. GR. IXARU AND M. MICU, in *Topics in Theoretical Physics* (Central Institute of Physics, Bucharest, 1978, p. 259.
12. A. R. BARNETT, D. H. FENG, J. W. STEED, AND L. J. B. GOLDFARB, *Comput. Phys. Commun.* **8**, 377 (1974).
13. A. D. RAPTIS, Exponential multistep methods for ordinary differential equations, *Bull. Greek Math. Soc.* **25**, 113 (1984).