Computer Assisted Mechanics and Engineering Sciences, 5: 3–8, 1998.

Uniform-graded mesh block method for second kind Volterra integral equations

Gamal M. Attia
Physical Science Department, Faculty of Engineering
Mansoura University, Mansoura, Egypt

(Received September 30, 1996)

In this paper a block method is developed to use on uniform-graded mesh for the solution of Volterra integral equations of the second kind. This method permits the use of a variable step size when solving Volterra integral equations. Means of reducing the error. Extensive results are presented.

1. Introduction

There are many different methods for solving the second kind Volterra integral equation:

$$y(t) = g(t) + \int_0^1 K(t, s, y(s))\,\mathrm{d}s\,. \tag{1}$$

For instance, a selection can be found in Delves and Mohamed [4] or Baker [2]. Jones [5] used variable step size method and developed a family of predictor–corrector methods for solving this equation.

Attia and Nersessian [1] solved Volterra integral equation with singular kernel by product Simpson's on graded mesh, but their technique require a change of the quadrature formulas to use it.

The purpose of this paper is to develop block method to use on uniform-graded mesh for solving Volterra integral equation without any change of the quadrature formula.

2. Preliminaries

The Volterra integral equation of the second kind is assumed to satisfy the condition for a unique solution (see, for example Tricomi [9] or Smithies [8]). This equation can be written in the form

$$y(t) = g(t) + \int_0^1 K(t, s, y(s))\,\mathrm{d}s\,, \qquad t \in [0, T]\,. \tag{2}$$

The interval $[0, T]$ is divided into $N = 2M$ subintervals. The nodes are chosen to satisfy with

$0 = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N \le T$

$$\begin{aligned} t_{2k} &= \left(\frac{2k}{N}\right)^{\beta} = \left(\frac{k}{M}\right)^{\beta}, && k = 0, 1, \ldots, M,\\ t_{2k+1} &= \frac{1}{2}[t_{2k} + t_{2k+2}]\,, && k = 0, 1, \ldots, M-1,\\ h_k &= t_{k+1} - t_k\,, && k = 0, 1, \ldots, 2M-1,\\ s_k &= t_k\,, && k = 0, 1, \ldots, 2M. \end{aligned} \tag{3}$$

3. Block method

The concept of a block method seems to have been described by Young [10], a similar technique for use with differential equations was given by Miline [7]. A block method is essentially an extrapolation procedure and has the advantage of being self starting. As we shall see it produces a block of values at a time. One drawback of the method however is that it requires the kernel to be evaluated at points for which it may not be defined. We shall follow Linz [6] description of the method. For simplicity we confine ourselves to the simplest nontrivial case in which a block of two values is produced at each stage. The generalization will be obvious.

The method depends on the use of two three-point quadrature formulas. The first is Simpson's rule, and the second is given by [3]

$$\int_0^h \phi(x)\,\mathrm{d}x = \frac{h}{12}[5\phi_0 + 8\phi_1 - \phi_2] + \frac{h^4}{24}\phi^{(3)}(\zeta)\,. \tag{4}$$

Suppose now that $y_0, y_1, \ldots, y_{2k}$ have been found (k may be zero), then since.

$$y_{2k+1} = g_{2k+1} + \int_0^{t_{2k}} K(t_{2k+1}, s, y(s))\,\mathrm{d}s + \int_{t_{2k}}^{t_{2k+1}} K(t_{2k+1}, s, y(s))\,\mathrm{d}s\,, \tag{5}$$

$$y_{2k+2} = g_{2k+2} + \int_0^{t_{2k}} K(t_{2k+2}, s, y(s))\,\mathrm{d}s + \int_{t_{2k}}^{t_{2k+2}} K(t_{2k+2}, s, y(s))\,\mathrm{d}s\,. \tag{6}$$

We can use Simpson's rule and (4) to obtain the approximations

$$\begin{aligned} y_{2k+1} = g_{2k+1} &+ \sum_{j=0}^{k-1} \frac{h_{2j}}{3}[K(t_{2k+1}, t_{2j}, y_{2j}) + 4K(t_{2k+1}, t_{2j+1}, y_{2j+1}) + K(t_{2k+1}, t_{2j+2}, y_{2j+2})] \\ &+ \frac{h_{2k}}{12}[5K(t_{2k+1}, t_{2k}, y_{2k}) + 8K(t_{2k+1}, t_{2k+1}, y_{2k+1}) - K(t_{2k+1}, t_{2k+2}, y_{2k+2})]\,, \end{aligned} \tag{7}$$

$$y_{2k+2} = g_{2k+2} + \sum_{j=0}^{k} \frac{h_{2j}}{3}[K(t_{2k+2}, t_{2j}, y_{2j}) + 4K(t_{2k+2}, t_{2j+1}, y_{2j+1}) + K(t_{2k+2}, t_{2j+2}, y_{2j+2})]\,. \tag{8}$$

Thus we have a pair of nonlinear or linear equations (dependent on the kernel of the integral equation to be nonlinear or linear) to solve for y_{2k+1} and y_{2k+2}. The linear equations are solved in recurrence by the direct substitution's method to find y_k and $k = 0, 1, 2, \ldots, N$.

4. Error analysis

Equation (1) can be written in the symbolic form

$$(I - \hat{K})y(t) = g(t) \tag{9}$$

where the operator is defined as

$$(\hat{K}y)(t) = \int_0^t K(t, s, y(s))\,\mathrm{d}s\,, \qquad t \in [0, T]\,. \tag{10}$$

If $\tilde{y}(t)$ is the approximate value of $y(t)$, then Eq. (9) is replaced by the approximation

$$(I - \hat{K}_n)\tilde{y}(t) = g(t)\,. \tag{11}$$

The operator $(\hat{K}_n y)(t)$ $(n = 2M)$ is defined as

$$(\hat{K}_n y)(t) = \sum_{j=0}^{M-1} \frac{h_{2j}}{3} \left[K(t, t_{2j}, y_{2j}) + 4K(t, t_{2j+1}, y_{2j+1}) + K(t, t_{2j+2}, y_{2j+2})\right]. \tag{12}$$

Subtract (11) from (9) then we have

$$y(t) - \tilde{y}(t) = (\hat{K} - \hat{K}_n)y(t) + \hat{K}_n(y(t) - \tilde{y}(t)) \tag{13}$$

but $y(t) - \tilde{y}(t)$ is equal to the value of error $E(t)$, then,

$$(I - K_n)E(t) \approx -\sum_{j=0}^{M-1} \frac{h_j^5}{90} \frac{\partial^4 K(t, s, y(s))}{\partial s^4}. \tag{14}$$

5. THE ALGORITHM OF SOLUTION

- Uses the procedures for:
  - Solution of two linear equations;
  - Solution of two nonlinear equations.
- Uses the Functions for:

  $K(s,t) :=$;

  $g(t) :=$;

  $y(t) :=$.
- Input β, number of subintervals $(N) = 2^p$, $p = 1, 2, \ldots, 10$.
- Compute the places of the nodes

$$t_{2k} = \left(\frac{2k}{N}\right)^{\beta}, \qquad k = 0, 1, \ldots, \frac{N}{2},$$

$$t_{2k+1} = \frac{t_{2k} + t_{2k+2}}{2}, \qquad k = 0, 1, \ldots, \frac{N}{2} - 1,$$

$$h_k = t_{k+1} - t_k, \qquad k = 0, 1, \ldots, N-1,$$

$$s_k = t_k, \qquad k = 0, 1, \ldots, 2M.$$

- Evaluate $y(t)$, $t \in [0, 1]$.
- Find the root mean square error
- Compute the value of max. error and its position.

6. TEST EXAMPLES

An algorithm using this method applied as described in Section 5. This algorithm was tested using a number of different test equations. These results are presented here in a condensed form, $x \in [0, 1]$.

No.	Equation	Exact Solution $y(t)$
(1)	$y(t) = \frac{1}{1+t^2}\int_0^t \frac{s}{\sqrt{1+t^2}}\, y(s)\, \mathrm{d}s$	$y(t) = \frac{1}{\sqrt{(1+t^2)^3}}$
(2)	$y(t) = e^t + \int_0^t 2\cos(t-s)\, y(s)\, \mathrm{d}s$	$y(t) = (t+1)^2 e^t$
(3)	$y(t) = 2t + 3 - \int_0^t [3 + 2(t-s)]\, y(s)\, \mathrm{d}s$	$y(t) = 4e^{-2t} - e{-}t$
(4)	$y(t) = \sinh t - \int_0^t \cosh(t-s)\, y(s)\, \mathrm{d}s$	$y(t) = \frac{2}{\sqrt{5}} e^{-t/2} \sinh \frac{\sqrt{5}}{2} t$
(5)	$y(t) = \sin t + \int_0^t e^{t-s}\, y(s)\, \mathrm{d}s$	$y(t) = \frac{1}{5}[e^{3t} - \cos t + 2\sin t]$
(6)	$y(t) = t + 1 - \cos t - \int_0^t \cos(t-s)\, y(s)\, \mathrm{d}s$	$y(t) = t$
(7)	$y(t) = e^{t(t+2)} + \int_0^t 2e^{t-s}\, y(s)\, \mathrm{d}s$	$y(t) = (1+2t)e^{t(t+2)}$
(8)	$y(t) = 1 + t^2 + \int_0^t \frac{1+t^2}{1+s^2}\, y(s)\, \mathrm{d}s$	$y(t) = (1+t^2)e^t$
(9)	$y(t) = 1 + \int_0^t y(s)\, \mathrm{d}s$	$y(t) = e^t$
(10)	$y(t) = (t-1) + (1+t^2)e^{-t^2} + \int_0^t t^2 e^{-st}\, y(s)\, \mathrm{d}s$	$y(t) = t$
(11)	$y(t) = \frac{t^2}{2} e^{-t} + \int_0^t \frac{1}{2}(t-s)^2 e^{s-t}\, y(s)\, \mathrm{d}s$	$y(t) = \frac{1}{3}\left[1 - e^{-1.5t}\left(\cos\frac{\sqrt{3}}{2}t + \sqrt{3}\sin\frac{\sqrt{3}}{2}t\right)\right]$
(12)	$y(t) = t - \int_0^t (t-s)\, y(s)\, \mathrm{d}s$	$y(t) = \sin t$
(13)	$y(t) = t + \int_0^t \sin(t-s)\, y(s)\, \mathrm{d}s$	$y(t) = t$
(14)	$y(t) = t - \int_0^t \sinh(t-s)\, y(s)\, \mathrm{d}s$	$y(t) = \frac{t - t^3}{6}$
(15)	$y(t) = \cos t + \int_0^t y(s)\, \mathrm{d}s$	$y(t) = \frac{e^t + \sin t + \cos t}{2}$
(16)	$y(t) = \sin t + \int_0^t (t-s)\, y(s)\, \mathrm{d}s$	$y(t) = \frac{\sin t + \sinh t}{2}$
(17)	$y(t) = 1 + t(\cos t^2 - 1) + \int_0^t t^2 \sin(ts)\, y(s)\, \mathrm{d}s$	$y(t) = 1$
(18)	$y(t) = \frac{t(1-e^t)}{2} + \int_0^t t^2 e^{st}\, y(s)\, \mathrm{d}s$	$y(t) = t$

7. Results

The results of solving these equations using the algorithm described in this paper are summarized in Table 1. These results were obtained by using P.C. and ($N = 16$).Table (2) shows the relation between N and the maximum error (E_M) when $\beta = 1.2$ for example (1). The results show that the error decreases as N increases.

Table 1

Example	Uniform-Graded Mesh				Uniform Mesh ($\beta = 1$)			
	β	E.r.m.s.	X_{EM}	E_M	E_M	X_{EM}	E.r.m.s.	E_R
1	1.20	1.099×10^{-6}	0.1360	2.42×10^{-6}	5.45×10^{-6}	0.0625	1.6970×10^{-6}	0.64
2	1.15	1.027×10^{-5}	1.0000	2.33×10^{-5}	2.30×10^{-5}	1.0000	1.1900×10^{-5}	0.86
3	1.15	2.520×10^{-5}	0.1473	4.62×10^{-5}	9.27×10^{-5}	0.0625	3.3410×10^{-5}	0.75
4	1.05	2.130×10^{-6}	0.1730	1.237×10^{-6}	2.88×10^{-6}	0.0625	1.2950×10^{-6}	0.95
5	1.00	2.853×10^{-5}	0.9375	7.54×10^{-5}	7.54×10^{-5}	0.9375	2.853×10^{-5}	1
6	1.00	1.069×10^{-6}	0.0625	1.83×10^{-6}	1.83×10^{-6}	0.0625	1.069×10^{-6}	1
7	0.95	5.644×10^{-4}	0.9404	1.56×10^{-3}	1.70×10^{-3}	0.9375	5.836×10^{-4}	0.97
8	0.95	1.717×10^{-6}	0.9404	4.13×10^{-6}	4.58×10^{-6}	0.9375	1.775×10^{-6}	0.97
9	0.95	1.098×10^{-6}	0.9404	2.19×10^{-6}	2.44×10^{-6}	0.9375	1.109×10^{-6}	0.99
10	0.90	1.920×10^{-7}	0.9434	5.23×10^{-7}	6.18×10^{-7}	0.9375	2.084×10^{-7}	0.92
11	0.90	2.386×10^{-7}	0.5955	2.95×10^{-7}	5.44×10^{-7}	0.9375	2.985×10^{-7}	0.80
12	0.85	3.901×10^{-7}	1.0000	5.23×10^{-7}	1.14×10^{-6}	0.9375	5.545×10^{-7}	0.70
13	0.85	3.504×10^{-7}	0.9464	7.06×10^{-7}	1.21×10^{-6}	0.9375	5.082×10^{-7}	0.77
14	0.80	3.504×10^{-7}	0.0947	6.37×10^{-7}	1.17×10^{-6}	0.9375	4.993×10^{-7}	0.70
15	0.75	2.044×10^{-7}	0.1051	5.29×10^{-7}	1.03×10^{-6}	0.9375	3.981×10^{-7}	0.52
16	0.70	3.234×10^{-8}	0.9554	7.81×10^{-8}	2.34×10^{-6}	0.9375	7.304×10^{-8}	0.44
17	0.70	4.715×10^{-8}	0.9554	1.30×10^{-7}	3.41×10^{-7}	0.9375	9.999×10^{-8}	0.47
18	0.65	1.977×10^{-6}	1.0000	6.23×10^{-6}	3.11×10^{-5}	0.9375	7.774×10^{-6}	0.25

Here: E.r.m.s. is the root mean square error, E_M is a max. error, X_{EM} is the position of max. error and $E_R = \dfrac{\text{E.r.m.s. of uniform graded mesh}}{\text{E.r.m.s. of uniform mesh}}$.

Table 2

N	2	4	8	16	32
E_M	2.2×10^{-3}	4.65×10^{-4}	2.70×10^{-5}	2.42×10^{-6}	1.56×10^{-7}
N	64	128	256	512	1024
E_M	9.75×10^{-9}	6.08×10^{-10}	3.73×10^{-11}	7.28×10^{-12}	7.28×10^{-12}

8. The effect of graded mesh

In Fig. 1, the relation between $-\log(\text{E.r.m.s.})$ and β is plotted at $N = 16$. We notice that:

- The Max. value of $-\log(\text{E.r.m.s.})$ it means the Min. value of E.r.m.s.
- For example (1) the Min. E.r.m.s. occur at $\beta = 1.2$,
- Example (6) has Min. E.r.m.s. at $\beta = 1$ (Uniform Mesh) and
- In example (17) the Min. E.r.m.s. at $\beta = 0.7$.

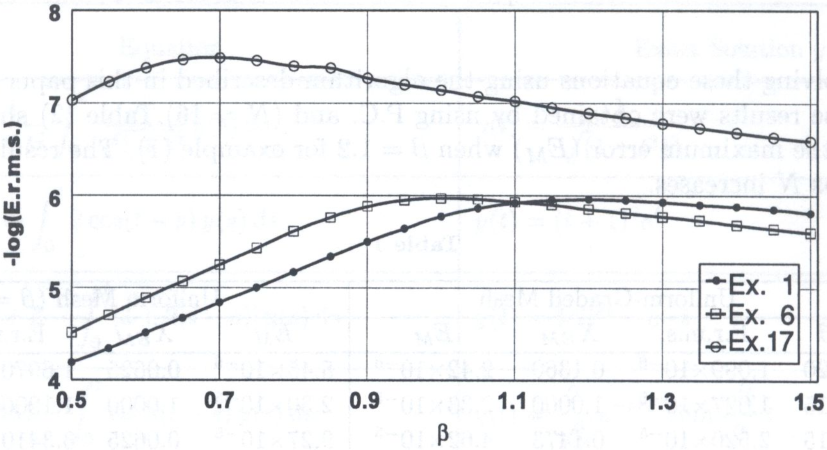

Fig. 1. The relation between E.r.m.s. and β at $N = 16$

REFERENCES

[1] G.M. Attia, A.B. Nersessian. Numerical solution of Volterra integral equations with singular kernels. *Second Int. Conf. on Engineering Mathematics and Physics (ICEMP-1994)*, Vol. **3**: 249–258. Cairo University, Faculty of Engineering, 1994.

[2] C.T.H. Baker. *The Numerical Treatment of Integral Equations.* Oxford University Press, 1977.

[3] P.J. Davis, R. Rabinovitz. *Methods of Numerical Integration.* Academic Press, 1984.

[4] L.M. Delves, J.L. Mohamed. *Computational Methods for Integral Equations.* Cambridge University Press, first published 1985.

[5] H.M. Jones, S. Mckee. Variable step size predictor–corrector scheme for second kind Volterra integral equations. *Mathematics of Computation*, **44**(170), 391-404, April 1985.

[6] P. Linz. The numerical solution of Volterra integral equations by finite difference methods. *M.R.C. Tech Rep.*, 825, 1968

[7] Miline. *Numerical solution of differential equations.* Wiley, 1953.

[8] F. Smithies. *Integral Equations. Cambridge Tracts in Mathematics and Mathematical Physics*, No. 49. Cambridge University Press, W. E. New York, 1958.

[9] F.G. Tricomi. *Integral Equations.* Interscience, New York, 1957.

[10] A. Young. The application of product integration to the numerical solution of integral equations. *Proc. R. Soc. A.*, **224**: 561–573, 1954.