MATHEMATICS OF COMPUTATION
VOLUME 39, NUMBER 159
JULY 1982, PAGES 165-171

On Certain Extrapolation Methods for the Numerical Solution of Integro-Differential Equations*

By S. H. Chang

Abstract. Asymptotic error expansions have been obtained for certain numerical methods for linear Volterra integro-differential equations. These results permit the application of extrapolation procedures. Computational examples are presented.

1. Introduction. Consider the linear Volterra integro-differential equation

$$y'(x) = a(x) + b(x)y(x) + \int_{x_0}^{x} k(x,s)y(s)\,ds, \tag{1}$$
$$y(x_0) = y_0, \quad x_0 \leqslant x \leqslant L,$$

where $a(x)$, $b(x)$, and $k(x, s)$ are given continuous functions for $x_0 \leqslant x, s \leqslant L$, and y_0 is a given real number. Numerical solutions of more general Volterra integro-differential equations have been investigated by many authors. Methods that use finite difference and quadrature techniques have been studied by, for example, Brunner and Lambert [1], Day [2], Feldstein and Sopka [3], Goldfine [4], Linz [6], Makroglou [8], McKee [9], Mocarsky [10], Wolfe and Phillips [11]. Feldstein and Sopka [3] have also discussed asymptotic error expansion and extrapolation for their Taylor algorithms for integro-differential equations.

It is the purpose of this paper to study the asymptotic expansions for the errors associated with certain simple numerical methods. Such a study will permit the application of extrapolation procedures. As a consequence, high order of accuracy in the numerical solution of (1) can be obtained with only a modest amount of work. This will then be demonstrated by computational examples. Our work is inspired by Linz [7] in which the extrapolation, based on a simple numerical method for linear Volterra integro-differential equations of the first kind, is very effective.

In the subsequent discussion, y_n will denote an approximate value of $y(x_n)$, where $x_n = x_0 + nh$, $n = 1, 2, \ldots, N$, and $h = (L - x_0)/N$. For the known functions $a(x)$, $b(x)$, and $k(x, s)$, a_i, b_i, and $k_{i,j}$ will denote $a(x_0 + ih)$, $b(x_0 + ih)$, and $k(x_0 + ih, x_0 + jh)$.

Received October 2, 1981.
1980 *Mathematics Subject Classification.* Primary 65R20; Secondary 65D30.
*This research was supported in part by the CSU RACA grants program.

0025-5718/82/0000-0065/$02.25

2. The Algorithms and Asymptotic Error Expansions. Integrating (1) from x_{n-1} to x_n, we have

$$(2)\quad y(x_n) = y(x_{n-1}) + \int_{x_{n-1}}^{x_n} [a(t) + b(t)y(t)]\,dt + \int_{x_{n-1}}^{x_n} \int_{x_0}^{t} k(t,s)y(s)\,ds\,dt.$$

Replacing the integrals from x_{n-1} to x_n by the two-step Adams-Moulton rule

$$(3)\qquad \int_{x_{n-1}}^{x_n} \phi(x)\,dx = \frac{h}{12}[5\phi(x_n) + 8\phi(x_{n-1}) - \phi(x_{n-2})] - \frac{h^4}{24}\phi'''(\xi),$$

and replacing the remaining inner integral by the Euler-Maclaurin formula (see Hildebrand [5, p. 202])

$$(4)\qquad \begin{aligned}\int_{x_0}^{x_r} \phi(x)\,dx = h\left[\tfrac{1}{2}\phi(x_0) + \phi(x_1) + \cdots + \phi(x_{r-1}) + \tfrac{1}{2}\phi(x_r)\right]\\ - \frac{h^2}{12}[\phi'(x_r) - \phi'(x_0)] + O(h^4),\end{aligned}$$

we obtain from (2) that

$$(5)\qquad \begin{aligned} y(x_n) = y(x_{n-1}) + \frac{h}{12}\big[5(a(x_n) + b(x_n)y(x_n))\\ +8(a(x_{n-1}) + b(x_{n-1})y(x_{n-1}))\\ - (a(x_{n-2}) + b(x_{n-2})y(x_{n-2}))\big]\\ + \frac{h^2}{12}\Bigg\{5\left[\frac{1}{2}k(x_n, x_0)y(x_0) + \sum_{i=1}^{n-1} k(x_n, x_i)y(x_i) + \frac{1}{2}k(x_n, x_n)y(x_n)\right]\\ +8\Bigg[\frac{1}{2}k(x_{n-1}, x_0)y(x_0) + \sum_{i=1}^{n-2} k(x_{n-1}, x_i)y(x_i)\\ + \frac{1}{2}k(x_{n-1}, x_{n-1})y(x_{n-1})\Bigg]\\ -\Bigg[\frac{1}{2}k(x_{n-2}, x_0)y(x_0) + \sum_{i=1}^{n-3} k(x_{n-2}, x_i)y(x_i)\\ + \frac{1}{2}k(x_{n-2}, x_{n-2})y(x_{n-2})\Bigg]\Bigg\}\\ + Q_n,\end{aligned}$$

where

$$\begin{aligned} Q_n = -\frac{h^3}{144}\Bigg\{5\left[\frac{\partial}{\partial s}(k(x_n, s)y(s))\right]_{x_0}^{x_n} + 8\left[\frac{\partial}{\partial s}(k(x_{n-1}, s)y(s))\right]_{x_0}^{x_{n-1}}\\ -\left[\frac{\partial}{\partial s}(k(x_{n-2}, s)y(s))\right]_{x_0}^{x_{n-2}}\Bigg\} + O(h^4),\end{aligned}$$

or, using (3) again,

$$Q_n = -\frac{h^2}{12}\int_{x_{n-1}}^{x_n} f(x)\,dx + O(h^4), \tag{6}$$

where

$$f(x) = \left[\frac{\partial}{\partial s}(k(x,s)y(s))\right]_{x_0}^{x} \tag{7}$$

if the function $f(x)$ is C^3 in $[x_0, L]$.

From (5) we have the following algorithm.

ALGORITHM A.

Get starting values: y_0, y_1.

Compute y_n, for $n = 2, 3, \ldots, N$, according to

$$\begin{aligned} y_n = y_{n-1} &+ \frac{h}{12}[5(a_n + b_n y_n) + 8(a_{n-1} + b_{n-1}y_{n-1}) - (a_{n-2} + b_{n-2}y_{n-2})] \\ &+ \frac{h^2}{12}\Bigg[5\left(\frac{1}{2}k_{n,0}y_0 + \sum_{i=1}^{n-1} k_{n,i}y_i + \frac{1}{2}k_{n,n}y_n\right) \\ &\quad + 8\left(\frac{1}{2}k_{n-1,0}y_0 + \sum_{i=1}^{n-2} k_{n-1,i}y_i + \frac{1}{2}k_{n-1,n-1}y_{n-1}\right) \\ &\quad - \left(\frac{1}{2}k_{n-2,0}y_0 + \sum_{i=1}^{n-3} k_{n-2,i}y_i + \frac{1}{2}k_{n-2,n-2}y_{n-2}\right)\Bigg]. \end{aligned} \tag{8}$$

Now, let $e(x)$ be the solution of

$$e'(x) = b(x)e(x) + \int_{x_0}^{x} k(x,s)e(s)\,ds - \frac{1}{12}f(x), \quad e(x_0) = 0, \tag{9}$$

where $f(x)$ is given by (7). Using the same approach as before and with appropriate assumption on smoothness, we have

$$\begin{aligned} e(x_n) = e(x_{n-1}) &+ \frac{h}{12}[5b(x_n)e(x_n) + 8b(x_{n-1})e(x_{n-1}) - b(x_{n-2})e(x_{n-2})] \\ &+ \frac{h^2}{12}\Bigg\{5\left[\sum_{i=1}^{n-1} k(x_n, x_i)e(x_i) + \frac{1}{2}k(x_n, x_n)e(x_n)\right] \\ &\quad + 8\left[\sum_{i=1}^{n-2} k(x_{n-1}, x_i)e(x_i) + \frac{1}{2}k(x_{n-1}, x_{n-1})e(x_{n-1})\right] \\ &\quad - \left[\sum_{i=1}^{n-3} k(x_{n-2}, x_i)e(x_i) + \frac{1}{2}k(x_{n-2}, x_{n-2})e(x_{n-2})\right]\Bigg\} \\ &- \frac{1}{12}\int_{x_{n-1}}^{x_n} f(x)\,dx + O(h^3). \end{aligned} \tag{10}$$

Let $\rho_n = y(x_n) - y_n - h^2 e(x_n)$, $n = 0, 1, \ldots, N$. We see from (5), (8), and (10) that ρ_n satisfies the following equation.

$$\begin{aligned}\rho_n = \rho_{n-1} &+ \frac{h}{12}(5b_n\rho_n + 8b_{n-1}\rho_{n-1} - b_{n-2}\rho_{n-2}) \\ &+ \frac{h^2}{12}\left[5\left(\sum_{i=1}^{n-1} k_{n,i}\rho_i + \frac{1}{2}k_{n,n}\rho_n\right) + 8\left(\sum_{i=1}^{n-2} k_{n-1,i}\rho_i + \frac{1}{2}k_{n-1,n-1}\rho_{n-1}\right)\right. \\ &\left. - \left(\sum_{i=1}^{n-3} k_{n-2,i}\rho_i + \frac{1}{2}k_{n-2,n-2}\rho_{n-2}\right)\right] \\ &+ O(h^4), \qquad n \geqslant 2.\end{aligned} \tag{11}$$

By assumption on initial conditions, $\rho_0 = 0$. Suppose we choose the starting value y_1 so that

$$y(x_1) - y_1 = O(h^4).$$

Furthermore,

$$\begin{aligned}e(x_1) &= e(x_0) + e'(x_0)h + \tfrac{1}{2}e''(x_0)h^2 + \ldots \\ &= \tfrac{1}{2}e''(x_0)h^2 + \ldots,\end{aligned} \tag{12}$$

since $e(x_0) = e'(x_0) = 0$ by (9). Then we have

$$\rho_1 = y(x_1) - y_1 - h^2 e(x_1) = O(h^4). \tag{13}$$

Then, it is clear that (11) implies $\rho_n = O(h^4)$, for $n \geqslant 2$.

We have thus proved the following theorem.

THEOREM 1. *Assume that $a(x)$ and $b(x)$ are C^3, and $k(x, s)$ is $C^{3,4}$, for $x_0 \leqslant x$, $s \leqslant L$. Then the approximations y_n, $n \geqslant 2$, computed from Algorithm* A *with $O(h^4)$ starting values, satisfy the relation*

$$y(x_n) = y_n + h^2 e(x_n) + O(h^4).$$

Now, the extrapolation procedure can be used. Let $Y(x, h)$ denote the approximate solution at x with step-size h. Then, by Theorem 1, we have

$$y(x) = Y(x, h) + h^2 e(x) + O(h^4).$$

We then obtain immediately that

$$y(x) = \frac{1}{3}\left(4Y\left(x, \frac{h}{2}\right) - Y(x, h)\right) + O(h^4).$$

Thus a better approximate value at x is obtained with fourth order accuracy.

Now, instead of (3), let us use the three-step Adams-Moulton rule

$$\begin{aligned}\int_{x_{n-1}}^{x_n} \phi(x)\,dx = \frac{h}{24}&[9\phi(x_n) + 19\phi(x_{n-1}) - 5\phi(x_{n-2}) + \phi(x_{n-3})] \\ &- \frac{19h^5}{720}\phi^{(4)}(\xi),\end{aligned} \tag{14}$$

together with the Euler-Maclaurin formula (4), in Eq. (2). This leads to the following algorithm.

ALGORITHM B.

Get starting values: y_0, y_1, y_2.

Compute y_n, for $n = 3, 4, \ldots, N$, according to

$$
\begin{aligned}
y_n = y_{n-1} + \frac{h}{24}[9(a_n + b_n y_n) + 19(a_{n-1} + b_{n-1}y_{n-1}) \\
-5(a_{n-2} + b_{n-2}y_{n-2}) + a_{n-3} + b_{n-3}y_{n-3}] \\
+ \frac{h^2}{24}\left[9\left(\frac{1}{2}k_{n,0}y_0 + \sum_{i=1}^{n-1} k_{n,i}y_i + \frac{1}{2}k_{n,n}y_n\right)\right. \\
+19\left(\frac{1}{2}k_{n-1,0}y_0 + \sum_{i=1}^{n-2} k_{n-1,i}y_i + \frac{1}{2}k_{n-1,n-1}y_{n-1}\right) \\
-5\left(\frac{1}{2}k_{n-2,0}y_0 + \sum_{i=1}^{n-3} k_{n-2,i}y_i + \frac{1}{2}k_{n-2,n-2}y_{n-2}\right) \\
\left.+\left(\frac{1}{2}k_{n-3,0}y_0 + \sum_{i=1}^{n-4} k_{n-3,i}y_i + \frac{1}{2}k_{n-3,n-3}y_{n-3}\right)\right].
\end{aligned}
\tag{15}
$$

Again, let $e(x)$ be the solution of (9) and $\rho_n = y(x_n) - y_n - h^2 e(x_n)$, $n = 0, 1, \ldots, N$. This time we find that ρ_n satisfies an equation similar to (11) but for $n \geqslant 3$ and with an $O(h^5)$ error term. Using an argument similar to that leading to (13), we obtain easily that $\rho_1 = \rho_2 = O(h^4)$. Then this leads again to the asymptotic error expansion

$$y(x_n) = y_n + h^2 e(x_n) + O(h^4).$$

Now, from (9) we see that

$$e''(x_0) = -\frac{1}{12}f'(x_0).$$

Suppose that

$$f'(x_0) = O(h). \tag{16}$$

Then from (12) we will have $e(x_0 + h) = O(h^3)$. This in turn will lead to $\rho_1 = \rho_2 = O(h^5)$ if we choose $O(h^5)$ starting values. Then the equation on ρ_n implies that $\rho_n = O(h^5)$, and thus the asymptotic error expansion

$$y(x_n) = y_n + h^2 e(x_n) + O(h^5).$$

By differentiating (7) we have

$$f'(x_0) = k_{ss}(x_0, x_0)y(x_0) + 2k_s(x_0, x_0)y'(x_0) + k(x_0, x_0)y''(x_0). \tag{17}$$

One sufficient condition for (16) to hold is seen to be

$$k(x_0, x_0) = k_s(x_0, x_0) = k_{ss}(x_0, x_0) = 0. \tag{18}$$

THEOREM 2. *Assume that $a(x)$ and $b(x)$ are C^4, and $k(x, s)$ is $C^{4,5}$, for $x_0 \leqslant x$, $s \leqslant L$. Then the approximations y_n, $n \geqslant 3$, computed from Algorithm* B *with $O(h^4)$ starting values, satisfy the relation*

$$y(x_n) = y_n + h^2 e(x_n) + O(h^4).$$

If, furthermore, (16) *is satisfied, then the values* y_n, $n \geqslant 3$, *computed from Algorithm* B *with* $O(h^5)$ *starting values, satisfy the relation*

$$y(x_n) = y_n + h^2 e(x_n) + O(h^5).$$

3. Computational Examples.

Example 1.

$$y'(x) = 1 - \int_0^x y(s)\, ds, \qquad y(0) = 0, \qquad 0 \leqslant x \leqslant 1.$$

The exact solution is $y(x) = \sin x$.

Example 2.

$$y'(x) = 1 + \sin x - y(x) + \int_0^x \sin(x-s) y(s)\, ds,$$
$$y(0) = 0, \qquad 0 \leqslant x \leqslant 1.$$

The exact solution is $y(x) = x$.

For Example 1, we see that $f'(x_0) = 0$ by (17). Both Algorithms A and B, with appropriate starting values, are used in computing the approximate solution. We list in Tables 1, 2, and 3 some of the resulting errors, before and after extrapolation. By error we mean

$$\text{error} = |\, \text{exact value} - \text{approximate value}\, | .$$

For Example 2, the approximate solution is computed using only Algorithm A. The resulting errors are listed in Tables 4 and 5. The effect of extrapolation is apparent from these tables.

The programs are written in FORTRAN in double precision for the IBM 370/158 computer at the Cleveland State University.

TABLE 1

Example 1, *Algorithm* A

x	$h = 0.1$	$h = 0.05$	$h = 0.025$
0.4	1.13×10^{-5}	2.55×10^{-6}	5.95×10^{-7}
0.6	3.50×10^{-5}	8.06×10^{-6}	1.92×10^{-6}
0.8	7.75×10^{-5}	1.81×10^{-5}	4.35×10^{-6}
1.0	1.42×10^{-4}	3.35×10^{-5}	8.11×10^{-6}

TABLE 2

Example 1, *Algorithm* B

x	$h = 0.1$	$h = 0.05$	$h = 0.025$
0.4	8.47×10^{-6}	2.21×10^{-6}	5.49×10^{-7}
0.6	2.92×10^{-5}	7.28×10^{-6}	1.81×10^{-6}
0.8	6.73×10^{-5}	1.67×10^{-5}	4.17×10^{-6}
1.0	1.26×10^{-4}	3.15×10^{-5}	7.85×10^{-6}

TABLE 3
Example 1, *after extrapolation*

	Algorithm A		*Algorithm* B	
x	$h = 0.1$	$h = 0.05$	$h = 0.1$	$h = 0.05$
0.4	3.54×10^{-7}	5.72×10^{-8}	1.25×10^{-7}	4.10×10^{-9}
0.6	9.28×10^{-7}	1.33×10^{-7}	4.46×10^{-9}	1.13×10^{-8}
0.8	1.70×10^{-6}	2.32×10^{-7}	1.18×10^{-7}	1.70×10^{-8}
1.0	2.60×10^{-6}	3.46×10^{-7}	2.06×10^{-7}	2.07×10^{-8}

TABLE 4
Example 2, *Algorithm* A

x	$h = 0.1$	$h = 0.05$	$h = 0.025$
0.4	1.14×10^{-4}	2.89×10^{-5}	7.27×10^{-6}
0.6	2.42×10^{-4}	6.11×10^{-5}	1.53×10^{-5}
0.8	4.05×10^{-4}	1.02×10^{-4}	2.54×10^{-5}
1.0	5.93×10^{-4}	1.49×10^{-4}	3.72×10^{-5}

TABLE 5
Example 2, *Algorithm* A, *after extrapolation*

x	$h = 0.1$	$h = 0.05$
0.4	7.42×10^{-7}	4.51×10^{-8}
0.6	6.16×10^{-7}	3.76×10^{-8}
0.8	5.28×10^{-7}	3.25×10^{-8}
1.0	4.78×10^{-7}	2.97×10^{-8}

Department of Mathematics
The Cleveland State University
Cleveland, Ohio 44115

1. H. BRUNNER & J. D. LAMBERT, "Stability of numerical methods for Volterra integro-differential equations," *Computing*, v. 12, 1974, pp. 75–89.

2. J. T. DAY, "Note on the numerical solution of integro-differential equations," *Comput. J.*, v. 9, 1967, pp. 394–395.

3. A. FELDSTEIN & J. R. SOPKA, "Numerical methods for nonlinear Volterra integro-differential equations," *SIAM J. Numer. Anal.*, v. 11, 1974, pp. 826–846.

4. A. GOLDFINE, "Taylor series methods for the solution of Volterra integral and integro-differential equations," *Math. Comp.*, v. 31, 1977, pp. 691–707.

5. F. B. HILDEBRAND, *Introduction to Numerical Analysis*, 2nd ed., McGraw-Hill, New York, 1974.

6. P. LINZ, "Linear multistep methods for Volterra integro-differential equations," *J. Assoc. Comput. Mach.*, v. 16, 1969, pp. 295–301.

7. P. LINZ, "A simple approximation method for solving Volterra integro-differential equations of the first kind," *J. Inst. Math. Appl.*, v. 14, 1974, pp. 211–215.

8. A. MAKROGLOU, "Convergence of a block-by-block method for nonlinear Volterra integro-differential equations," *Math. Comp.*, v. 35, 1980, pp. 783–796.

9. S. MCKEE, "Cyclic multistep methods for solving Volterra integro-differential equations," *SIAM J. Numer. Anal.*, v. 16, 1979, pp. 106–114.

10. W. L. MOCARSKY, "Convergence of step-by-step methods for non-linear integro-differential equations," *J. Inst. Math. Appl.*, v. 8, 1971, pp. 235–239.

11. M. A. WOLFE & G. M. PHILLIPS, "Some methods for the solution of non-singular Volterra integro-differential equations," *Comput. J.*, v. 11, 1968/69, pp. 334–336.