Internat. J. Math. & Math. Sci.
VOL. 21 NO. 4 (1998) 755-760

EFFICIENT ALGORITHMS FOR SOLVING NONLINEAR VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS WITH TWO POINT BOUNDARY CONDITIONS

by

L.E. GAREY † and R.E. SHAW ††

University of New Brunswick
P.O. Box 5050, Saint John, N.B.
Canada, E2L 4L5

(Received August 19, 1996 and in revised form December 2, 1996)

ABSTRACT. In this paper a collection of efficient algorithms are described for solving an algebraic system with a symmetric Toeplitz coefficient matrix. Systems of this form arise when approximating the solution of boundary value Volterra integro-differential equations with finite difference methods. In the nonlinear case, an iterative procedure is required and is incorporated into the algorithms presented. Numerical examples illustrate the results.

KEY WORDS AND PHRASES. Nonlinear Volterra integro-differential equation, symmetric Toeplitz matrix, algorithm efficiency.

AMS CLASSIFICATION CODE. 65R, 45.

1. INTRODUCTION

The form of the system to be solved in this paper is

$$AY = F(Y) \tag{1.1}$$

where F(Y) is a nonlinear function in an unknown vector Y and A is a symmetric Toeplitz matrix. In [1,4] circulant matrices and symmetric band matrices are discussed in connection with solving a linear system. In [7] there is an application of a fast algorithm to a system of the form (1.1). In [5] this algorithm involves the solution of a second order Volterra integro-differential equation. The background for that work and the work considered here is based on the Sherman-Morrison formula [2,p.113]. Consider two square matrices A and B and two column vectors **u** and **v** related by

$$A = B - \mathbf{u}\mathbf{v}^T$$

If the inverse of B exists and $\mathbf{v}^T\mathbf{B}^{-1}\mathbf{u} \neq 1$, then

$$A^{-1} = B^{-1} + \alpha B^{-1}\mathbf{u}\mathbf{v}^T B^{-1}, \quad \alpha = 1/(1 - \mathbf{v}^T\mathbf{B}^{-1}\mathbf{u})$$

In this paper an application involving the numerical solution of a second order boundary value problem of the Volterra type will be discussed. In particular, the problem model will be nonlinear in the unknown and the computer implementation will employ the Sherman-Morrison formula. Two numerical examples will be given to compare this method with an efficient form of the LU method and a variant of the method described in [5] and [7].

2. THE DISCRETE SOLUTION

Consider a nonlinear integro-differential equation of the form

$$y^{(2)}(x) = f(x,y(x),z(x)), \quad 0 \leq x \leq a \tag{2.1}$$

† Supported by NSERC of Canada

†† Supported by NSERC of Canada

where

$$z(x) = \int_0^x K(x,t,y(t))dt$$

subject to the boundary conditions $y(0)=b_0$ and $y(a)=b_1$. Let $R_1=\{(x,t,y):0\le x\le a,|y|<\infty\}$ and $R_2=\{(x,y,z):0\le x\le a,|y|<\infty,|z|<\infty\}$. For equation (2.1) defined for points in R_1 and R_2, the following conditions are assumed:

I) f and K are uniformly continuous in each variable

ii) for the function f and for all $(x,\bar{y},z)$, $(x,y,\bar{z})$ and (x,y,z) in R_2,

$$|f(x,y,z) - f(x,\bar{y},z)| \le \square_1|y - \bar{y}|;$$

$$|f(x,y,z) - f(x,y,\bar{z})| \le \square_2|z - \bar{z}|;$$

iii) for the function K and for all (x,t,y) and $(x,t,\bar{y})$ in R_1,

$$|K(x,t,y) - K(x,t,\bar{y})| \le \square_3|y - \bar{y}|; \text{ and}$$

iv) the functions f_y, f_z and K_y are continuous and satisfy $f_y(x,y,z)\ge 0$, $f_z(x,y,z)\ge 0$ and $K_y(x,t,y)\le 0$ for all $(x,t,y)\in R_1$ and $(x,y,z)\in R_2$.

A proof of the uniqueness of a solution for problems satisfying the above can be found in Shaw [6].

To develop a discrete solution for (2.1) let $I_N=\{x_i,\ x_i=ih,\ I=0(1)N,\ Nh=a\}$ be a partition of I=[0,a]. A general k-step method of solution is given by

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h^2 \sum_{i=0}^{k} \beta_i f(x_{n+i},y_{n+i},z_{n+i}) \ , \ n=0(1)N-k \tag{2.2}$$

where

$$z_n = h \sum_{j=0}^{n} w_{nj}K(x_n,x_j,y_j) \ , \ n > s \ , \ z_0 = 0$$

The coefficients $\{w_{nj}\}$ denote the weights for the choice of a quadrature rule. In addition, there are special rules required in connection with starting values. These rules have weights w_{mj}, $m\le\max\{k,s\}$ and s is related to the order of the method. Note that when k=2 the two boundary conditions provide the required auxiliary conditions and the (N+1)x(N+1) system can then be solved. In the linear case only one solve is required but in the nonlinear case an iterative scheme must be employed.

For the remainder of this section we consider a model of the form

$$y^{(2)}(x) = y(x) + g(x,y(x),z(x)) \tag{2.3}$$

with boundary values $y(0)=b_0$ and $y(a)=b_1$. Applying a two-step method to (2.3) gives rise to an (N+1)x(N+1) system. Furthermore, if the method for solving the differential equation is denoted by (ρ,σ) with

$$\rho(z) = \sum_{i=0}^{2} \alpha_i z^i \ , \ \sigma(z) = \sum_{i=0}^{2} \beta_i z^i$$

then a particular method, known as Numerov's method, has $(\alpha_0,\alpha_1,\alpha_2)=(1,-2,1)$ and $(\beta_0,\beta_1,\beta_2)=(1/12,10/12,1/12)$. The quadrature rule Q selected as part of this particular method is the fourth order Newton-Gregory rule. Conditions for the convergence of two part methods $((\rho,\sigma),Q)$ can be found in [6]. To present the method in a way which is more appropriate for the next section, remove the terms

involving y_0 and y_N to the right hand side to give an (N-1)x(N-1) system. This system can be expressed as

$$(J-h^2B)Y = h^2(BG(Y) + S(Y)) + R \tag{2.4}$$

where

$$J=\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \cdots & 0 \\ \alpha_0 & \alpha_1 & \alpha_2 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & \cdots & \alpha_0 & \alpha_1 \end{bmatrix}, B=\begin{bmatrix} \beta_1 & \beta_2 & \cdots & \cdots & 0 \\ \beta_0 & \beta_1 & \beta_2 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & \cdots & \beta_0 & \beta_1 \end{bmatrix}, Y=\begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ y_{N-1} \end{bmatrix}, G(Y)=\begin{bmatrix} g_1 \\ \vdots \\ \vdots \\ g_{N-1} \end{bmatrix}, S(Y)=\begin{bmatrix} \frac{g_0}{12} \\ 0 \\ \vdots \\ 0 \\ \frac{g_N}{12} \end{bmatrix}, R=\begin{bmatrix} (\frac{h^2}{12}-1)b_0 \\ 0 \\ \vdots \\ 0 \\ (\frac{h^2}{12}-1)b_1 \end{bmatrix}$$

Denoting $(J-h^2B)$ by C we have

$$C=\begin{bmatrix} c_1 & c_2 & \cdots & \cdots & 0 \\ c_0 & c_1 & c_2 & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & \cdots & c_0 & c_1 \end{bmatrix}$$

Of particular interest is that C is a diagonally dominant symmetric Toeplitz matrix.

3. ALGORITHM DEVELOPMENT

To solve system (2.4) an iteration scheme must be employed. Using {i} to denote the i^{th} iterate gives

$$(J-h^2B)Y^{i+1} = h^2(BG(Y^i) + S(Y^i)) + R \tag{3.1}$$

There are several ways to solve system (3.1). For example, by efficiently using LU decomposition on the Toeplitz tridiagonal symmetric system the operation count is about 9n, where n is the size of the system. For repeated applications with the same coefficient matrix C and different right hand sides the count is 5n. The form of the LU method used can be described as follows:

1) assign values to elements of C:
   $c_0=1-h^2/12$
   $c_1=-2.0-10h^2/12$
   $c_2=c_0$
2) compute superdiagonal of U
   $u_1=c_1$
   for j=2,N-1
   $u_j=(u_{j-1}c_1-c_0^2)/u_{j-1}$
3) perform forward substitution to solve Lx=b (b represents the right hand side of (3.1)) and the subdiagonal of L is generated from the superdiagonal of U
   $x_1=b_1$
   for j=2,N-1
   $x_j=b_j-\ell x_{j-1}$ where $\ell=c_0/u_{j-1}$
4) perform backward substitution to solve Uy=x
   $y_{N-1}=x_{N-1}/u_{N-1}$
   for j=N-2,1,-1
   $y_j=(x_j-c_0y_{j+1})/u_j$
5) if $\|Y^{i+1}-Y^i\|$ > tolerance, repeat from step 3

This method of efficient LU decomposition will be referred to as Method 1.

Working with C, note that $|c_1|>|c_0|+|c_2|$, $c_0=c_2$ and the division by $-c_0$ gives a matrix which is tridiagonal symmetric and diagonally dominant. The nonzero elements in each row are given by (-1, λ, -1) with $\lambda=2+(10h^2/12)$. Denoting this matrix by D gives

$$DY^{i+1} = -\frac{1}{c_0}(h^2(BG(Y^i) + S(Y^i)) + R) \tag{3.2}$$

Let $\mathbf{u}=(\mu-\lambda)\mathbf{e}_1$ and $\mathbf{v}=\mathbf{e}_1$ where $\mathbf{e}_1=(1,0,...,0)^T$. Perturb D such that $\hat{D}$ is given by

$$\hat{D} = \begin{bmatrix} \mu & -1 & \dots & \dots & 0 \\ -1 & \lambda & -1 & \dots & 0 \\ \vdots & & & & \\ 0 & \dots & -1 & \lambda & -1 \\ 0 & \dots & \dots & -1 & \lambda \end{bmatrix}$$

Then it is easily seen that $D=\hat{D}-\mathbf{u}\mathbf{v}^T$. Method 2 uses an LU decomposition of $\hat{D}$ from [4] as follows:

$$L = \begin{bmatrix} 1 & & & \\ -\mu^{-1} & 1 & & \\ & \ddots & & \\ & & -\mu^{-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \mu & -1 & & \\ & \mu & -1 & \\ & & \ddots & \\ & & & \mu \end{bmatrix}$$

For ease of notation, let $K^i=h^2(BG(Y^i)+S(Y^i))+R$ represent the i^{th} iterate of the right hand side of equation (3.2). Let $a_j=(\mu^{-j}-\mu^{j-2m-2})d$ where $d=\mu^{-1}(1-\mu^{-2(m+1)})^{-1}$, m=N-1. The steps in the method are given by

1) factor $\hat{D}$ into its LU decomposition and determine the a_j, j=1(1)N-1
2) solve the system $Z^i = K^i$ where $Z^i = (z_j^i)$
3) evaluate $Y^{i+1} = (y_j^{i1})$ where $y_j^{i+1} = z_j^i - a_j y_{j-1}^i$, j=1(1)N-1
4) if $\|Y^{i+1} - \bar{Y}\| >$ tolerance, repeat from step 2

There are just a few calculations for step 1 and approximately 4n calculations for step 2. The remaining steps require about 5n operations for a total of 9n. For repeated applications with the same $\hat{D}$ the operation count is about 6n.

As a third approach, Method 3 utilizes the method given in [2,p.113]. To obtain an approximate solution the steps are as follows:

1) factor $\hat{D}$ into its LU decomposition
2) solve $\hat{D}W = u$
3) solve $\hat{D}Z = Y^i$
4) calculate the constant $\beta=v^TZ/(1-v^TW)$ and set $Y^i = Z+\beta W$
5) if $\|Y^{i+1} - \bar{Y}\| >$ tolerance, repeat from step 3

As before, step 1 essentially requires just a few operations and step 2 requires 4n operations. Since $v=(1,0,...,0)^T$, step 4 requires 2n operations. The remaining steps give a total of 9n operations that reduces to 6n with repeated iterations.

To close this section, we will look at the convergence of the three methods. For the nonlinear problems, each method is iterative in nature. For the direct LU method, the application of the method is equivalent to solving

$$Y^{i+1} = A^{-1}(h^2(BG(Y^i) + S(Y^i)) + R).$$

Subtracting the same equation with the exact solution of the discrete system being used gives us

$$E^{i+1} = h^2A^{-1}B(G(Y^i) - G(Y) + S(Y^i) - S(Y))$$

from which

$$|E^{i+1}| \le h^2 L |A^{-1}| |B| |E^i| \tag{3.3}$$

where L is a Lipschitz constant and convergence takes place provided $h^2L|A^{-1}||B|<1$. Note that A is a continuous function of h which tends to J as $h \to 0$. The matrix -A is monotone [3,p.360] and hence is invertible. The inverse of a monotone matrix has nonnegative elements. Since -J and -A are monotone and $(-A)-(-J) \ge 0$ we see that $(-J^{-1})-(-A^{-1}) \ge 0$ [3,Theorem 7.5, p.362]. It follows, therefore, that $0 \le (-A^{-1}) \le (-J^{-1})$. From [3] it is also known that $|J| \le N^2/8$ and therefore $|A^{-1}|$ is bounded.

The other two methods are somewhat similar so only the last one will be analyzed. There are two parts to the iteration process. In the first part, an approximation is obtained. By adding an appropriate correction, the next iterate Y^{i+1} is given by

$$Y^{i+1} = \hat{D}^{-1} K^i + \alpha \hat{D}^{-1} v^T \hat{D}^{-1} K^i u. \tag{3.4}$$

Again, exact values for the solution of the discrete problem are substituted and the result is subtracted from (3.4) to give

$$E^{i+1} = \hat{D}^{-1} h^2 B[G(Y^i)-G(Y)+S(Y^i)-S(Y)] + \alpha \hat{D}^{-1} v^T \hat{D}^{-1} h^2 B(G(Y^i)-G(Y)+S(Y^i)-S(Y))u. \tag{3.5}$$

In particular, $v^T \hat{D}^{-1} h^2 B(G(Y^i)-G(Y)+S(Y^i)-S(Y))$ is just a constant because of the definition of v. Hence, equation (3.5) can be written as

$$E^{i+1} = (\hat{D}^{-1} + \alpha \hat{D}^{-1} u v^T \hat{D}^{-1}) h^2 B(G(Y^i)-G(Y)+S(Y^i)-S(Y))$$

$$= h^2 A^{-1} B(G(Y^i)-G(Y)+S(Y^i)-S(Y)).$$

By taking norms and assuming F satisfies a Lipschitz condition we have the same result as (3.3).

4. NUMERICAL RESULTS

Two Volterra integro-differential equations of the form (2.1) are used to illustrate the methods described. Method 1 is the efficient form of the LU method, Method 2 is a variant of the fast algorithm given in [4] and [6] and Method 3 is the implementation of the Sherman-Morrison formula [see, 2]. The iteration count for all methods is given along with the average CPU time in seconds. All programs were run on a SUN SPARC 20 in double precision arithmetic.

Example 1: $y'' = y - x^2 - \frac{x^5}{4} + 2 - x((x-1)e^x + 1) + \int_0^x xty\, dt$, $y(0) = 1$, $y(1) = 1+e$

exact solution: $y = x^2 + e^x$

TABLE I

			Method 1	Method 2	Method 3
step size (h)	N	# iterates	avg CPU time	avg CPU time	avg CPU time
0.01	100	5	0.222884	0.220293	0.219623
0.005	200	6	0.545157	0.542729	0.542624
0.0025	400	6	1.317954	1.309260	1.308940
0.001	1000	7	5.034045	5.008745	5.016750

Example 2: $y'' = e^x + 2 + \frac{11}{30}x^6 - e^{x^2} + \int_0^x (x+t)y^2\,dt, \quad y(0) = 0, \quad y(1) = 1$

exact solution: $y = x^2$

TABLE II

			Method 1	Method 2	Method 3
step size (h)	N	# iterates	avg CPU time	avg CPU time	avg CPU time
0.01	100	6	0.266512	0.277212	0.276059
0.005	200	7	0.670504	0.701899	0.697980
0.0025	400	7	1.722915	1.800295	1.805805
0.001	1000	8	6.779800	7.335638	7.372522

Another approach to solving system (3.1) is given in the paper by Yan and Chung [7]. Their modification reduces the operation count for step 3 from 5n to 4n+2t where t represents the number of operations in the correction term approximation ($t \le n$). The actual value of t depends on the degree of the diagonal dominance of the coefficient matrix C. To describe their algorithm, one can replace the approximation to the correction term in step 3 of Method 2 by

$$Y^{i+1} = Z^i - \mu^{-1} z_1 \mathbf{p} .$$

Let $b=\mu^{-1}$. The vector $\mathbf{p}$ is given by $\mathbf{p}=(b,b^2,\ldots b^t,0,\ldots,0)^T$ where the number of components in $\mathbf{p}$ is dependent on $|\lambda|$ and a tolerance τ. In particular, τ is chosen such that

$$t(\tau) \ge \frac{\log(|\lambda| - 2) + \log(\tau)}{\log(|b|)} - 1$$

For the matrix C as given in (3.1), as the step size h used in the k-step method and quadrature rule in (2.2) decreases $|\lambda|$ approaches 2 and b approaches 1. This gives a value of $t > N$ and the method of Yan and Chung [7], as pointed out in their paper, will not work.

REFERENCES

1. CHEN, M., On the solution of circulant linear systems, *SIAM J. Numer. Anal.* **24** (668-683), 987.

2. HAGER, W., *Applied Numerical Linear Algebra*, Prentice Hall, 1988.

3. HENRICI, P., *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley, 1962.

4. ROJO, O., A new method for solving symmetric circulant tridiagonal systems of linear equations, *Computers Math. Applic.* **20** (1990), 61-67.

5. SHAW, R.E., GAREY, L.E. and GLADWIN, C.J., A fast algorithm for solving second order Volterra integro-differential equations with two-point boundary conditions, *Congressus Numerantium* **106** (1995), 193-203.

6. SHAW, R.E., Numerical Solutions for Two-Point Boundary Value Problems, Ph.D. Thesis, University of New Brunswick, 1994.

7. YAN, W.M. and CHUNG, K.L., A fast algorithm for solving special tridiagonal systems, *Computing* **52** (1994), 203-211.