Abstract and Applied Analysis
Volume 2013, Article ID 490689, 4 pages
http://dx.doi.org/10.1155/2013/490689

Research Article

Numerical Solution for IVP in Volterra Type Linear
Integrodifferential Equations System

F. Ghomanjani,1 A. Kılıçman,2 and S. Effati1

1 Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran
2 Department of Mathematics, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Correspondence should be addressed to A. Kılıçman; kilicman@yahoo.com

Received 23 May 2013; Accepted 9 July 2013

Academic Editor: Santanu Saha Ray

Copyright © 2013 F. Ghomanjani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A method is proposed to determine the numerical solution of system of linear Volterra integrodifferential equations (IDEs) by using Bezier curves. The Bezier curves are chosen as piecewise polynomials of degree \( n \), and Bezier curves are determined on \([t_0, t_f]\) by \( n + 1 \) control points. The efficiency and applicability of the presented method are illustrated by some numerical examples.

1. Introduction

Integrodifferential equations (IDEs) have been found to describe various kinds of phenomena, such as glass forming process, dropwise condensation, nanohydrodynamics, and wind ripple in the desert (see [1, 2]).

There are several numerical and analytical methods for solving IDEs. Some different methods are presented to solve integral and IDEs in [3, 4]. Maleknejad et al. [5] used rationalized Haar functions method to solve the linear IDEs system. Linear IDEs system has been solved by using Galerkin methods with the hybrid Legendre and block-Pulse functions on interval \([0, 1]\) (see [6]). Yusufoğlu [7] presented an application of He’s homotopy perturbation (HPM) method to solve the IDEs system. He’s variational iteration method has been used for solving two systems of Volterra integrodifferential equations (see [8]). Arikoglu and Ozkol [9] presented differential transform method (DTM) for integrodifferential and integral equation systems. He’s homotopy perturbation (HPM) method was proposed for system of integrodifferential equations (see [10]). A numerical method based on interpolation of unknown functions at distinct interpolation points has been introduced for solving linear IDEs system with initial values (see [11]). Recently, Biazar introduced a new modification of homotopy perturbation method (NHPM) to obtain the solution of linear IDEs system (see [12]). Taylor expansion method has been used for solving IDEs (see [13, 14]). Rashidinia and Tahmasebi developed and modified Taylor series method (TSM) introduced in [15] to solve the system of linear Volterra IDEs.

In the present work, we suggest a technique similar to the one which was used in [16] for solving the system of linear Volterra IDEs in the following form:

\[
\sum_{i=1}^{n} \sum_{j=0}^{\alpha_m} p_{mij}(t) y_i^{(j)}(t) + \sum_{i=1}^{n} \int_{t_0}^{t} k_{mi}(t, x) \sum_{j=0}^{\alpha_m} y_i^{(j)}(x) \, dx = f_m(t), \quad m = 1, 2, \ldots, n, \quad t_0 \leq t \leq t_f,
\]

with the initial conditions

\[
y_i^{(0)}(t_0) = c_{i0}, \quad y_i^{(1)}(t_0) = c_{i1}, \ldots, y_i^{(\alpha_m-1)}(t_0) = c_{i(\alpha_m-1)},
\]

where \( y_i^{(j)}(t) \) stands for \( j \)-th order derivative of \( y_i(t) \), \( f_m(t) \), \( k_{mi}(t, x) \), and \( p_{mij}(t) \) are known functions \( (m, i = 1, 2, \ldots, n; j = 0, 1, \ldots, \alpha_m) \), and \( t_0, t_f \), and \( c_{ij} \) \( (i = 1, 2, \ldots, n; j = 0, 1, \ldots, \alpha_m - 1) \) are appropriate constants.

The current paper is organized as follows. In Section 2, function approximation will be introduced. Numerical examples will be stated in Section 3. Finally, Section 4 will give a conclusion briefly.
2. Function Approximation

Our strategy is to use Bezier curves to approximate the solutions \( y_i(t) \), for \( 1 \leq i \leq n \), which are given below. Define the Bezier polynomials of degree \( N \) that approximate, respectively, the actions of \( y_i(t) \) over the interval \([t_0, t_f]\) as follows:

\[
y_i(t) = \sum_{r=0}^{N} a_i^r B_{r,N}(t-t_0),
\]

where \( h = t_f - t_0 \) and \( a_i \) is the control point of Bezier curve, and

\[
B_{r,N}(t-t_0) = \binom{N}{r} \frac{(t-t_0)^r}{h^r},
\]

is the Bernstein polynomial of degree \( N \) over the interval \([t_0, t_f]\) (see [17]). By substituting (3) in (2), \( R_m(t) \) can be defined for \( t \in [t_0, t_f] \) as

\[
R_m(t) = \sum_{i=1}^{n} \sum_{j=0}^{\alpha_m} p_{mij}(t) y_i^{(j)}(t)
+ \sum_{i=1}^{n} \int_{t_0}^{t} \left( k_m(t,x) \sum_{j=0}^{\alpha_m} y_i^{(j)}(x) \right) dx - f_m(t),
\]

where (2) is satisfied. The convergence was proved in the approximation with Bezier curves when the degree of the approximate solution, \( N \), tends to infinity (see [18]).

Now, the residual function is defined over the interval \([t_0, t_f]\) as follows:

\[
R(t) = \int_{t_0}^{t_f} \left( \sum_{m=1}^{n} \left\| R_m(t) \right\|^2 \right) dt,
\]

where \( \| \cdot \| \) is the Euclidean norm. Our aim is to solve the following problem over the interval \([t_0, t_f]\):

\[
\text{min } R(t) \quad \text{s.t. } y_i^{(0)}(t_0) = c_{i0}, \quad y_i^{(1)}(t_0) = c_{i1}, \ldots, y_i^{(\alpha_m-1)}(t_0) = c_{i(\alpha_m-1)}.
\]

The mathematical programming problem (7) can be solved by many subroutine algorithms, and we used Maple 12 to solve this optimization problem.

Remark 1. In Chapter 1 of [19], it was proved that \( N \) satisfies

\[
N > \frac{S}{\delta^2 e},
\]

where \( S = \| y_i(t) \| \), and because of this reason that \( y_i(t) \) is uniformly continuous on \([t_0, t_f]\), we have \( s, t \in [t_0, t_f] \) that \( |t - s| < \delta \) and \(-\epsilon/2 < y_i(t) - y_i(s) < \epsilon/2\), for more details see [19].

3. Applications and Numerical Results

Consider the following examples which can be solved by using the presented method.

Example 1. Consider a system of third-order linear Volterra IDEs on the interval \([0, 1]\) (see [4]):

\[
y''_1(t) + t^2 y'_1(t) - y''_2(t)
+ \int_{t_0}^{t} ((t-x) y_1(x) + y_2(x)) dx = g_1(t),
\]

\[
4t^3 y'_1(t) + 6t^2 y_1(t) + y'''_2(t)
+ \int_{t_0}^{t} (y_1(x) + (t+x) y_2(x)) dx = g_2(t),
\]

\[\]

where (9) can be solved by using the presented method.
Table 1: Computed errors for Example 1.

<table>
<thead>
<tr>
<th>t</th>
<th>Absolute error for $y_1(t)$</th>
<th>Absolute error for $y_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000000</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>0.2</td>
<td>$1.4801 \times 10^{-10}$</td>
<td>$2.2475 \times 10^{-11}$</td>
</tr>
<tr>
<td>0.4</td>
<td>$0.162735585 \times 10^{-5}$</td>
<td>$3.12780502 \times 10^{-7}$</td>
</tr>
<tr>
<td>0.6</td>
<td>$0.251133963 \times 10^{-5}$</td>
<td>$0.1536077787 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.8</td>
<td>$1.8337 \times 10^{-10}$</td>
<td>$0.8864238659 \times 10^{-5}$</td>
</tr>
<tr>
<td>1.0</td>
<td>$4.5905 \times 10^{-10}$</td>
<td>$7.897 \times 10^{-12}$</td>
</tr>
</tbody>
</table>

Table 2: Computed errors for Example 2.

<table>
<thead>
<tr>
<th>t</th>
<th>Absolute error for $y_1(t)$</th>
<th>Absolute error for $y_2(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000000</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>0.2</td>
<td>$3.840 \times 10^{-11}$</td>
<td>$1.5360 \times 10^{-11}$</td>
</tr>
<tr>
<td>0.4</td>
<td>$0.5791064832 \times 10^{-3}$</td>
<td>$0.1560417484 \times 10^{-3}$</td>
</tr>
<tr>
<td>0.6</td>
<td>$0.17373195072 \times 10^{-2}$</td>
<td>$0.1560417484 \times 10^{-3}$</td>
</tr>
<tr>
<td>0.8</td>
<td>$0.69492781056 \times 10^{-2}$</td>
<td>$1.5360 \times 10^{-11}$</td>
</tr>
<tr>
<td>1.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

With $N = 5$, the computed errors are shown in Table 2 which show the high accuracy of the proposed method.

4. Conclusions

In this paper, Bernstein's approximation is used to approximate the solution of linear Volterra IDEs. In this method, we approximate our unknown function with Bernstein's approximation. The present results show that Bernstein's approximation method for solving linear Volterra IDEs is very effective and simple, and the answers are trusty, and their accuracy is high, and we can execute this method in a computer simply. The numerical examples support this claim.

Acknowledgment

The authors are very grateful to the referees for their valuable suggestions and comments that improved the paper.

References


Submit your manuscripts at http://www.hindawi.com