Research Article

Bezier Curves Method for Fourth-Order Integrodifferential Equations

F. Ghomanjani,1 A. V. Kamyad,1 and A. Kilicman2

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

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

Received 3 September 2013; Accepted 22 September 2013

Academic Editor: Abdon Atangana

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.

The Bezier curves method is applied to solve both linear and nonlinear BVPs for fourth-order integrodifferential equations. Also, the presented method is developed for solving BVPs which arise from the problems in calculus of variation. These BVPs result from the Euler-Lagrange equations which are the necessary conditions of the extrema of problems in calculus of variation. Some numerical examples demonstrate the validity and applicability of the technique.

1. Introduction

Recently, there has been much attention devoted to the search for reliable and more efficient solution methods for equations modelling physical phenomena in various fields of engineering (see [1, 2]). One of the methods which has received much concern is the Adomian decomposition method (ADM) (see [3, 4]). The ADM has been employed to solve various scientific models. In [5], Wazwaz’s main objective was to obtain the exact solutions to two fourth-order integrodifferential equations. Hashim [4] determined the accuracy and efficiency of the ADM in solving integrodifferential equations. In the present work, we suggest a technique similar to the one which was used in [6–8] for solving both linear and nonlinear boundary value problems (BVPs) for fourth-order integrodifferential equations.

Now, we consider the following class of two-point BVPs for fourth-order integrodifferential equations

\[ y(t) = \alpha_0, \quad y''(t) = \alpha_1, \]
\[ y(t) = \beta_0, \quad y''(t) = \beta_1, \]

where \( F \) is a real nonlinear continuous function, \( \gamma, \alpha_0, \alpha_1, \beta_0, \) and \( \beta_1 \) are real constants, and \( f, g, \) and \( h \) are given and can be approximated by Taylor polynomials. The conditions for existence and uniqueness of solutions of (1) are given in [9].

The rest of this paper is organized as follows. In Section 2, we review the Bezier curves method. Several illustrative examples are provided in Section 3 for confirming the effectiveness of the presented method. Section 4 contains some conclusions and notations about the future works.

2. The Bezier Curves Method

Consider the problem (1). Divide the interval \([t_0, t_f]\) into a set of grid points such that

\[ t_j = t_0 + jh, \quad j = 0, 1, \ldots, k, \]

where \( h \) is the step size.
where \( h = (t_j - t_0)/k \) and \( k \) is a positive integer. Let \( S_j = [t_{j-1}, t_j] \) for \( j = 1, 2, \ldots, k \). Then, for \( t \in S_j \), the problem (I) can be decomposed to the following problems:

\[
y^{(p)}_j(t) = f(t) + yy_j(t)
\]

\[
+ \int_{t_{j-1}}^{t_j} \left( g(t) y_j(t) + h(t) F(y_j(t)) \right) dt,
\]

\[ t \in [t_{j-1}, t_j], \quad (3) \]

where \( y_j(t) \) is considered in \( t \in S_j \). Let \( y(t) = \sum_{j=1}^{k} \chi_j^1(t) y_j(t) \) where \( \chi_j^1(t) \) is the characteristic function of \( y_j(t) \) for \( t \in [t_{j-1}, t_j] \). It is trivial that \( [t_0, t_f] = \bigcup_{j=1}^{k} S_j \).

Our strategy is using Bezier curves to approximate the solutions \( y_j(t) \) by \( y_j(t) \) where \( y_j(t) \) is given below. Individual Bezier curves that are defined over the subintervals are joined together to form the Bezier spline curves. For \( j = 1, 2, \ldots, k \), define the Bezier polynomials \( v_j(t) \) of degree \( n \) that approximate the action of \( y_j(t) \) over the interval \([t_{j-1}, t_j]\) as follows:

\[
v_j(t) = \sum_{r=0}^{n} a_r^j B_{r,n} \left( t - t_{j-1} \right), \quad (4)
\]

where

\[
B_{r,n} \left( t - t_{j-1} \right) = \binom{n}{r} \frac{1}{h^r} (t_j - t)^{n-r} (t - t_{j-1})^r, \quad (5)
\]

is the Bernstein polynomial of degree \( n \) over the interval \([t_{j-1}, t_j]\) and \( a_r^j \) is the control points (see [7]). By substituting (4) in (3), one can define \( R_{1,j}(t) \) for \( t \in [t_{j-1}, t_j] \) as

\[
R_{1,j}(t) = y^{(p)}_j(t)
\]

\[
- \left( f(t) + yy_j(t) \right)
\]

\[
+ \int_{t_{j-1}}^{t_j} \left( g(t) y_j(t) + h(t) F(y_j(t)) \right) dt.
\]

(6)

Let \( v(t) = \sum_{j=1}^{k} \chi_j^1(t)v_j(t) \) where \( \chi_j^1(t) \) is the characteristic function of \( v_j(t) \) for \( t \in [t_{j-1}, t_j] \). Besides the boundary conditions on \( v(t) \), at each node, we need to impose continuity condition on each successive pair of \( v_j(t) \) to guarantee the smoothness. Since the differential equation is of first order, the continuity of \( y \) (or \( v \)) and its first derivative gives

\[
\frac{\partial v}{\partial y_i} - \frac{d}{dt} \left( \frac{\partial G}{\partial y_i} \right) = 0, \quad i = 1, 2, \ldots, N, \quad (14)
\]

where \( v^{(p)}_j(t_j) \) is the \( s \)th derivative \( v_j(t) \) with respect to \( t \) at \( t = t_j \). Thus, the vector of control points \( a_r^j \) \( (r = 0, 1, n - 1, n) \) must satisfy (see [7])

\[
a_r^j (t_j - t_{j-1})^n = a_r^{j+1} (t_{j+1} - t_j)^n,
\]

\[
(a_r^n - a_{n-1}^n) (t_j - t_{j-1})^{n-1} = (a_{r+1}^n - a_r^n) (t_{j+1} - t_j)^{n-1}.
\]

(8)

According to the definition of the \( t_j = t_0 + jh \), we get that \( t_j - t_{j-1} = h \). Therefore

\[
a_r^n = a_0^n, \quad (9)
\]

Let \( \gamma = 0 \). Thus, the vector of control points is \( a_r^n = a_0^n \).

Ghomanjani et al. [7] proved the convergence of this method where \( h \to 0 \).

Now, the residual function can be defined in \( S_j \) as follows:

\[
R_j = \int_{t_{j-1}}^{t_j} \left\| R_{1,j}(t) \right\|^2 dt,
\]

(10)

where \( \| \cdot \| \) is the Euclidean norm. Our aim is solving the following problem over \( S = \bigcup_{j=1}^{k} S_j \):

\[
\min \sum_{j=1}^{k} R_j
\]

s.t.

\[
a_0^n = a_0^{j+1},
\]

(11)

\[
(a_n^n - a_{n-1}^n) = (a_{n+1}^n - a_n^n),
\]

\[
v_1(t_0) = \alpha_0, \quad v_1''(t_0) = \alpha_1,
\]

\[
v_k(t_f) = \beta_0, \quad v_k''(t_f) = \beta_1.
\]

The mathematical programming problem (11) can be solved by many subroutine algorithms. Here, Maple 12 can solve this optimization problem.

Remark 1. Consider the following functional (see [10]):

\[
J \left[ y_1(t), y_2(t), \ldots, y_N(t) \right] = \int_{0}^{1} G(x, y_1(t), \ldots, y_N(t), y'_1(t), \ldots, y'_N(t)) dt,
\]

(12)

to find the extreme value of \( J \), the boundary points of the admissible curves are known as

\[
y_i(0) = \theta_i, \quad i = 1, 2, \ldots, N,
\]

(13)

\[
y_i(1) = \delta_i, \quad i = 1, 2, \ldots, N.
\]

To extremize the necessary condition, \( J[y_1(t), y_2(t), \ldots, y_N(t)] \) is that it should satisfy the Euler-Lagrange equations

\[
\frac{\partial G}{\partial y_i} - \frac{d}{dt} \left( \frac{\partial G}{\partial y'_i} \right) = 0, \quad i = 1, 2, \ldots, N,
\]

(14)
with boundary conditions given in (13). The system of BVPs (14) does not always have a solution, and if the solution exists, it cannot be unique. Note that in many variational problems, the existence of a solution is obvious from the geometrical or physical meaning of the problem, and if the solution of the Euler equation satisfies the boundary conditions, it is unique. Also this unique extremal will be the solution of the given variational problem. Thus, another approach for solving the variational problem (12) is finding the solution of the system of ordinary differential equations (ODEs) (14) which satisfies the boundary conditions in (13) which were called systems of BVPs. The simplest form of the variational problem (12) is

$$J[y(t)] = \int_{0}^{1} G\left(t, y(t), y'(t)\right) dt,$$

with the given boundary conditions

$$y(0) = \theta, \quad y(1) = \delta.$$  \hspace{1cm} (15)

For the extremum of the functional (15), the necessary condition is to satisfy the following second-order differential equation:

$$\frac{\partial G}{\partial y} - \frac{d}{dt} \left( \frac{\partial G}{\partial y'} \right) = 0, \quad i = 1, 2, \ldots, N, \hspace{1cm} (17)$$

with boundary conditions given in (16).

One can emphasize that our method is to solve BVPs such as (14) and (17).

### 3. Numerical Results and Discussion

To demonstrate the accuracy of the decomposition method, we consider some examples with known exact solutions.

**Example 1.** First, we consider the linear fourth-order integrodifferential equation as in (1) with $f(t) = t(1 + e^t) + 3e^t$, $\gamma = 1$, $g(t) = -1$, and $h(t) = 0$; that is,

$$y^{(iv)}(t) = t \left(1 + e^t\right) + 3e^t + y(t)$$

$$- \int_{0}^{t} y(t) dt, \quad 0 < t < 1,$$

$$y(0) = 1, \quad y''(0) = 2,$$

$$y(1) = 1 + e, \quad y''(1) = 3e,$$

where the exact solution is

$$y(t) = 1 + te^t, \hspace{1cm} (19)$$

and the approximated solution

$$y(t) = 0.9935754347t^3 + 0.7033912400t^2 + 1.975345738t - 0.6105447271t^2 + 1.56255820t^3, \quad t \in \left[\frac{2}{3}, 1\right], \hspace{1cm} (20)$$

where the maximum absolute errors in [4] and presented method are, respectively, 0.008 and 0.0009. The graphs of approximated and exact solution are shown in Figure 1.

**Example 2.** Now, consider the nonlinear fourth-order BVP (1) with $f(t) = 1$, $\gamma = 0$, $g(t) = 0$, and $h(t) = e^{-t}$, and $F(t) = y^2(t)$,

$$y^{(iv)}(t) = 1 + \int_{0}^{t} e^{-t} y^2(t) dt, \quad 0 < t < 1,$$

$$y(0) = 1, \quad y''(0) = 1,$$

$$y(1) = e, \quad y''(1) = e,$$

where the exact solution is

$$y(t) = e^t, \hspace{1cm} (22)$$

with $n = 3$ and $k = 3$, and one can find the following approximated solution

$$y(t) = 0.9999999997 + 1.004858451t$$

$$+ 0.9327855932t^2 + 0.7033912400t^3, \quad t \in \left[0, \frac{1}{3}\right],$$

$$+ 0.935111555 + 1.063258052t$$

$$+ 0.757867804t^2 + 0.8785900700t^3, \quad t \in \left[\frac{1}{3}, \frac{2}{3}\right],$$

$$+ 0.7908249981 + 1.975345738t$$

$$- 0.6105447271t^2 + 1.56255820t^3, \quad t \in \left[\frac{2}{3}, 1\right], \hspace{1cm} (21)$$

The graphs of the exact and computed solution of the problem for Example 1.
with \( n = 3 \) and \( k = 3 \), and one can find the following approximated solution:

\[
y(t) = \begin{cases} 
0.9999999997 + 1.001035251t \\
+0.4854296266t^2 + 0.2130969500t^3, & t \in [0, \frac{1}{3}], \\
0.983270381 + 1.016091895t \\
+0.4402597035t^2 + 0.2582668700t^3, & t \in \left[\frac{1}{3}, \frac{2}{3}\right], \\
0.955636536 + 1.208199192t \\
+0.1520987099t^2 + 0.4023473900t^3, & t \in \left[\frac{2}{3}, 1\right], 
\end{cases}
\]

where the maximum absolute errors in \([4]\) and presented method are, respectively, 0.001 and 0.00018. The graphs of approximated and exact solution are shown in Figure 2.

**Example 3.** Consider the problem of finding the minimum of the integral

\[
\min \int_0^1 \left( (y'(t))^2 + ty'(t) + (y(t))^2 \right) dt,
\]

with the boundary conditions

\[
y(0) = 0, \quad y(1) = \frac{1}{4},
\]

where the exact solution is

\[
y(t) = \frac{e^{-t} - 1}{4(e^2 - 1)} \left( e - 2e^2 - 2e^2 + e^{t+1} \right).
\]

According to (17), the associated Euler-Lagrange equation is as follows:

\[
y(t) - y''(t) - \frac{1}{2} = 0,
\]

with \( n = 3 \) and \( k = 10 \), and the computational results are shown in Table 1, where the computed values are compared with the values obtained from the analytical solution and Legendre polynomials in [11].

**Table 1: Legendre polynomials, presented method, exact, and absolute error of \( y(t) \) for Example 3.**

<table>
<thead>
<tr>
<th>( t )</th>
<th>Legendre polynomials</th>
<th>Presented method</th>
<th>Exact</th>
<th>Absolute error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00000080</td>
<td>0.000000000000</td>
<td>0.000000000000</td>
<td>0.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0419474</td>
<td>0.04195072854</td>
<td>0.04195072872</td>
<td>1.773458002 \times 10^{-10}</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0793162</td>
<td>0.07931714637</td>
<td>0.07931714637</td>
<td>2.0180492 \times 10^{-12}</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1124756</td>
<td>0.1124732287</td>
<td>0.1124732286</td>
<td>6.880859813 \times 10^{-11}</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1417531</td>
<td>0.1417508127</td>
<td>0.1417508127</td>
<td>1.025976216 \times 10^{-11}</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1674426</td>
<td>0.167429185</td>
<td>0.1417508127</td>
<td>1.12222848 \times 10^{-11}</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1898041</td>
<td>0.1898066812</td>
<td>0.1898066813</td>
<td>6.433997580 \times 10^{-11}</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2090640</td>
<td>0.2090659248</td>
<td>0.2090659248</td>
<td>2.253659231 \times 10^{-11}</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2254149</td>
<td>0.2254134028</td>
<td>0.2254134028</td>
<td>2.503659231 \times 10^{-11}</td>
</tr>
<tr>
<td>0.9</td>
<td>0.2390158</td>
<td>0.2390127258</td>
<td>0.2390127256</td>
<td>2.308402409 \times 10^{-10}</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2499920</td>
<td>0.2500000000</td>
<td>0.2500000000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Figure 2:** Graphs of the exact and computed solution of the problem for Example 2.

4. Conclusions

In this sequel, the Bezier curves method was employed to solve linear and nonlinear BVPs for fourth-order integrodifferential equations. The presented algorithm produced results which are of reasonable accuracy. Numerical examples show that the proposed method is efficient and very easy to use.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

The authors are very grateful to the referees for their valuable suggestions and comments that improved the paper, and they...
would like to express their sincere gratitude to Doctor Y. Damchi. The third author gratefully acknowledges the partial support from University Putra Malaysia.

References


