Solutions of System of Volterra Integro Differential Equation of Second Kind by Using Piecewise Constant Functions

Mostafa Safavi*
Department of Mathematics, Payame Noor University, Islamic Republic of Iran

Abstract
In this paper we may use piece wise constant functions for the special type of system of second kind integro differential equation of the first order. The main problem is reduced to linear system of algebraic equations. Some numerical examples are dedicated for showing efficiency and validity of the method.

Keywords: Second kind integro differential equation; Block pulse functions; Collocation points

Introduction
Many different basis functions are used for approximating the solution of integro differential equations like Haar wavelets, lagrange functions, Taylor polynomials, Chebyshev polynomials, sine-cosine wavelets, Tau method, Adomian decomposition method, hybrid Legendre and Block Pulse functions and so on [1-8]. In this study, we use Block Pulse functions (BPfs) for solving system of Volterra integro differential equation of the form.

\[ y_i(x) + \lambda \int_0^x k_i(t, \tau) y_i' \, d\tau = g_i(x) \quad i = 1, 2, \ldots, m \]
\[ y_m(x) + \lambda \int_0^x k_m(t, \tau) y_m' \, d\tau = g_m(x) \]

where, \( k_i, g_i \in [0,1] \) for \( i = 1, \ldots, m \) are known functions and \( y_i(x) \) is the unknown function and \( y_i(0) = y_i, y_j(0) = y_j, \ldots, y_m(0) = y_m \cdot \) we set 
\[ F_i(x) = \int_0^x k_i(t, x) y_i' \, dt \quad for \quad i = 1, 2, \ldots, m \]

so (1) becomes

\[ y_i(x) + \lambda F_i(x) = g_i(x) \]
\[ y_j(x) + \lambda F_j(x) = g_j(x) \]
\[ \vdots \]
\[ y_m(x) + \lambda F_m(x) = g_m(x) \]

by collocating (3) at the points \( x_1, x_2, \ldots, x_k \) we get

\[ y_i(x_1) + \lambda F_i(x_1) = g_i(x_1) \]
\[ y_j(x_1) + \lambda F_j(x_1) = g_j(x_1) \]
\[ \vdots \]
\[ y_m(x_1) + \lambda F_m(x_1) = g_m(x_1) \]
\[ y_i(x_2) + \lambda F_i(x_2) = g_i(x_2) \]
\[ y_j(x_2) + \lambda F_j(x_2) = g_j(x_2) \]
\[ \vdots \]
\[ y_m(x_2) + \lambda F_m(x_2) = g_m(x_2) \]
\[ \vdots \]
\[ y_i(x_k) + \lambda F_i(x_k) = g_i(x_k) \]
\[ y_j(x_k) + \lambda F_j(x_k) = g_j(x_k) \]
\[ \vdots \]
\[ y_m(x_k) + \lambda F_m(x_k) = g_m(x_k) \]

At first, we define a \( k \)-set of BPfs for every row of (4) as:

\[ B_i(t) = \begin{cases} 1 & \frac{i-1}{k} \leq t \leq \frac{i}{k} \quad \text{for all} \quad i = 1, 2, \ldots, k \\ 0 & \text{elsewhere} \end{cases} \]

The functions \( B_i(t) \) are disjoint and orthogonal. That is,

\[ B_i(t)B_j(t) = \begin{cases} 0 & i \neq j \\ B_i(t) & i = j \end{cases} \]

\[ \langle B_i(t)B_j(t) \rangle = \begin{cases} 0 & i \neq j \\ \frac{1}{k} & i = j \end{cases} \]

Since the BPfs is not continuous so the derivatives don't exist at these points of discontinuity therefore we can't apply the BPfs in a direct manner to solve differential equations. Therefore we may expand \( y_i'(x), y_j'(x), \ldots, y_m'(x) \) into the BPfs series and \( y_i(x), y_j(x), \ldots, y_m(x) \) will be obtain through integration:

\[ y_i'(x) = \sum_{i=1}^{m} y_i B_i(x) \]
\[ y_j'(x) = \sum_{i=1}^{m} y_j B_i(x) \]
\[ \vdots \]
\[ y_m'(x) = \sum_{i=1}^{m} y_m B_i(x) \]

where,

\[ y_{i,1} = k(y_i(x), B_i(x)) = \int_0^1 y_i(x)B_i(x) \, dx \]
\[ y_{i,2} = k(y_i(x), B_i(x)) = \int_0^1 y_i(x)B_i(x) \, dx \]
\[ \vdots \]
\[ y_{i,m} = k(y_i(x), B_i(x)) = \int_0^1 y_i(x)B_i(x) \, dx \]

*Corresponding author: Mostafa Safavi, Department of Mathematics, Payame Noor University, Islamic Republic of Iran, E-mail: mostafa.safavi@gmail.com

Received December 02, 2012; Accepted January 28, 2013; Published February 06, 2013


Copyright: © 2013 Safavi M. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
In practice, only the first $k$-term of (7) are considered, that is,

$$y'_i(x) = \sum_{j=1}^{k} y_{ij} B_j(x)$$

$$y''_i(x) = \sum_{j=1}^{k} y_{ij} B_j(x)$$

$$\vdots$$

$$y'''_i(x) = \sum_{j=1}^{k} y_{ij} B_j(x)$$

with matrix form

$$y'_i(x) = B'(x)Y$$

$$y''_i(x) = B'(x)Y$$

$$\vdots$$

$$y''''_i(x) = B'(x)Y$$

with $Y = \{y_1, y_2, \ldots, y_k\}'$, $Y = \{y_1, y_2, \ldots, y_k\}'$, $\ldots$, and $B(x) = \{B_1(x), B_2(x), \ldots, B_k(x)\}'$.

Also the integration of BPfs is expandable into BPfs series:

$$\int_0^x B(x) = PB(x)$$

the $k$-square matrix $P$ is called the operational matrix of integration of the transform and is defined as follows:

$$P_{kk} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 2 & \cdots & 2 \\ 0 & 1 & 2 & \cdots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Integrating (8) over the interval $[0, x]$ and using (9) yields

$$y_1(x) = y_1 + \int_0^x B'(x) dx \ Y$$

$$y_2(x) = y_2 + \int_0^x B'(x) dx \ Y$$

$$\vdots$$

$$y_k(x) = y_k + \int_0^x B'(x) dx \ Y$$

Similarly, $k(x, t) \in L^2([0, t])$, $\forall i, 1, 2, \ldots, m$ may be approximated as:

$$k_1(x, t) = B'(x)K_1B(x)$$

$$k_2(x, t) = B'(x)K_2B(x)$$

$$\vdots$$

$$k_k(x, t) = B'(x)K_kB(x)$$

where

$$K_1 = \{K_{11}, K_{12}, \ldots, K_{1k}\}$$

$$K_2 = \{K_{21}, K_{22}, \ldots, K_{2k}\}$$

$$K_k = \{K_{k1}, K_{k2}, \ldots, K_{kk}\}$$

and

$$K_{ij} = k^2(B_i(x), \langle k_j(x, t), B_j(x) \rangle) \quad \text{for} \quad s = 1, 2, \ldots, m$$

Now using (5) leads to

$$K_{ij} = \begin{bmatrix} B_i(x) & 0 \\ \vdots & \vdots \\ 0 & B_j(x) \end{bmatrix}$$

$$= \text{diag}[B_i(x), B_2(x), \ldots, B_k(x)]$$

Now we choose

$$x_i = \frac{i}{k}, i = 1, 2, \ldots, k$$

so for evaluating $F_i(x)$, $\forall i = 1, 2, \ldots, m$ by substituting the matrix form of functions $k_1(x, t), k_2(x, t), \ldots, k_m(x, t)$ and $y_1(x), y_2(x), \ldots, y_m(x)$ and using the fact that $B(x)$ where $\phi_i$ is the $i$-th column of the identity matrix of order $k$, we may proceed as follows:

$$F_i(x) = B(x)K_1 + \int_0^x B(x) B'(t) dt \ Y$$

$$= \epsilon K_1 \sum_{j=1}^{m} \text{diag}[B(x), B_2(x), \ldots, B_k(x)] \ dt \ Y$$

$$= \frac{1}{k} \epsilon K_1 \ Y$$

where $\epsilon$ is the $k \times k$ diagonal matrix defined as follows:

$$\epsilon = \frac{1}{k} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Now by substituting (11) and (14) into (4) we obtain:

$$y_1(x) = y_1 + \frac{\lambda}{k} \epsilon K_1 \ Y$$

$$y_2(x) = y_2 + \frac{\lambda}{k} \epsilon K_2 \ Y$$

$$\vdots$$

$$y_m(x) = y_m + \frac{\lambda}{k} \epsilon K_m \ Y$$

$$A_i Y_i = g_i(x_i)$$

$$A_i Y_2 = g_2(x_i)$$

$$\vdots$$

$$A_i Y_m = g_m(x_i)$$

$$A_i^2 Y_i = g_i(x_i) - y_1$$

$$A_i^2 Y_2 = g_2(x_i) - y_2$$

$$\vdots$$

$$A_i^2 Y_m = g_m(x_i) - y_m$$

where,
\[
\begin{align*}
A_1' &= e_P' + \frac{\lambda}{k} e_{K_1} d' \\
A_2' &= e_P' + \frac{\lambda}{k} e_{K_2} d' \\
&\vdots \\
A_m' &= e_P' + \frac{\lambda}{k} e_{K_m} d'
\end{align*}
(17)
\]

Solving the linear system of algebraic equations (16) gives column vectors \(Y_1, Y_2, \ldots, Y_m\) and we can approximate \(y_1(x), y_2(x), \ldots, y_m(x)\) by (11) at every points \(x_i \in [1, 0]\).

**Illustrative Example**

Consider system of second kind volterra integro differential equation with initial conditions

\[
\begin{align*}
y_1(x) + \int_0^1 \cos(x-t)y_1(t) dt &= x^2 - 2\cos(x) + 2, \\
y_2(x) + \lambda \int_0^1 xy_1(t) dt &= x + \frac{1}{2} x^3,
\end{align*}
\]

the exact solution of system is

\[
\begin{align*}
y_1(x) &= x^2 \\
y_2(x) &= x.
\end{align*}
\]

Table 1 show the computed error for the example with \(k=16\) and table 2 show the computed error for the example with \(k=32\).

**Conclusions**

In this paper, Block Pulse functions were used to solve special type of system of the second kind volterra integro differential equation which convert the main problem to solve linear system of algebraic equations. For showing validity and efficiency, the method is applied for test problem with two different values of the parameter \(k\) which states approximations may be more accurate using larger \(k\). The benefit of the method is simplicity for execution and using sparse matrices which make the method cheap as computational costs.

**References**