Numerical solution of Volterra integro-differential equation by direct method via block-pulse functions

E. Babolian:Tarbiat Moallem University
Z. Masouri: Islamic Azad University, Science and Research Branch

Abstract

A direct method to determine numerical solutions of linear Volterra integro-differential equations is presented in this paper.. This method is based on block-pulse functions and its operational matrix. By using this approach, the integro-differential equation reduces to a linear lower triangular system of algebraic equations which can be solved easily. Some numerical examples are provided to illustrate accuracy and computational efficiency of the method.

MSC: 45J05; 41A30

1. Introduction

Many problems of theoretical physics and engineering lead to integro-differential equations. An integro-differential equation involves one (or more) unknown functions $x(t)$ and both of its differential and integral. Such a description covers a very broad class of functional relations [1],[2]. To solve these equations, several analytic and numerical approaches have been proposed [1],[2],[3]. In recent years, some different numerical methods to solve integro-differential equations were presented [4],[5].

The direct approach, proposed in this paper, applies block-pulse functions(BPFs) and its operational matrix and transforms a Volterra integro-differential equation to a linear lower triangular system of algebraic equations which can be easily solved. The numerical

KeyWords: Volterra integro-differential equation; Direct method; Numerical solution; Block-pulse functions; Operational matrix.

Received 2 Dec. 2007 Revised 10 Ago. 2008

babolian@tmu.ac.ir- nmasouri@yahoo.com

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

results of the examples illustrate efficiency of this method.

2. Block-pulse functions

Block-pulse functions have been studied by many authors and applied for solving different problems [6],[7],[8].

2.1. Definition

An m-set of block-pulse functions(BPFs) is defined over the interval $[0,T)$as:

$$\phi_i(t)=\begin{cases}1, & \frac{iT}{m}\le t<\frac{(i+1)T}{m},\\ 0, & otherwise,\end{cases} \tag{1}$$

where $i=0,1,...,m-1$ with a positive integer value for m. Also, consider $h=T/m$, and ϕ_i is the ith block-pulse function.

In this paper, it is assumed that $T=1$, so BPFs is defined over $[0,1)$, and $h=1/m$.

There are some properties for BPFs, the most important properties are disjointness, orthogonality, and completeness.

The disjointness property can be clearly obtained from the definition of BPFs:

$$\phi_i(t)\phi_j(t)=\begin{cases}\phi_i(t), & i=j,\\ 0, & i\neq j,\end{cases} \tag{2}$$

where $i,j=0,1,...,m-1$.

The other property is orthogonality. It is clear that:

$$\int_0^1\phi_i(t)\phi_j(t)dt=h\delta_{ij}, \tag{3}$$

where δ_{ij} is the Kroneker delta.

The third property is completeness. For every $f\in L^2([0,1))$ when m approaches to the infinity, Parseval's identity holds:

$$\int_0^1 f^2(t)dt=\sum_{i=0}^{\infty}f_i^2\,\|\phi_i(t)\|^2, \tag{4}$$

where

$$f_i=\frac{1}{h}\int_0^1 f(t)\phi_i(t)dt. \tag{5}$$

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

2.2. Vector forms

Consider the first m terms of BPFs and write them concisely as m-vector:

$$\Phi(t)=[\phi_0(t),\phi_1(t),\ldots,\phi_{m-1}(t)]^T, \qquad t\in[0,1) \tag{6}$$

above representation and disjointness property, follows:

$$\Phi(t)\Phi^T(t)=\begin{pmatrix} \phi_0(t) & 0 & \ldots & 0 \\ 0 & \phi_1(t) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \phi_{m-1}(t) \end{pmatrix}, \tag{7}$$

$$\Phi^T(t)\Phi(t)=1, \tag{8}$$

$$\Phi(t)\Phi^T(t)V=\tilde{V}\,\Phi(t), \tag{9}$$

where V is an m-vector and $\tilde{V}=diag(V)$. Moreover, It can be clearly concluded that for every $m\times m$ matrix B:

$$\Phi^T(t)B\Phi(t)=\hat{B}^T\Phi(t), \tag{10}$$

where $\hat{B}$ is an m-vector with elements equal to the diagonal entries of matrix B.

2.3. BPFs expansion

The expansion of a function $f(t)$ over $[0,1)$, with respect to $\phi_i(t),\ i=0,1,\ldots,m-1$ may be compactly written as:

$$f(t)\cong\sum_{i=0}^{m-1} f_i\phi_i(t)=F^T\Phi(t)=\Phi^T(t)F, \tag{11}$$

where $F=[f_0,f_1,\ldots,f_{m-1}]^T$ and f_i s are defined by Eq.(5).

Now, assume $k(t,s)$ is a function of two variables in $L^2([0,1)\times[0,1))$. It can be similarly expanded with respect to BPFs as

$$k(t,s)\cong\Phi^T(t)K\Psi(s), \tag{12}$$

where $\Phi(t)$ and $\Psi(s)$ are m_1 and m_2 dimensional BPF vectors respectively, and K is the $m_1\times m_2$ block-pulse coefficient matrix with
$k_{ij},\ i=0,1,\ldots,m_1-1,\ j=0,1,\ldots,m_2-1$ as follows:

$$k_{ij}=m_1m_2\int_0^1\int_0^1 k(t,s)\phi_i(t)\psi_j(s)\,dtds. \tag{13}$$

For convenience, we put $m_1=m_2$.

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

2.4. Operational matrix

Computing $\int_0^t \phi_i(\tau)d\tau$ follows:

$$\int_0^t \phi_i(\tau)d\tau = \begin{cases} 0, & t < ih, \\ t - ih, & ih \le t < (i+1)h, \\ h, & (i+1)h \le t < 1. \end{cases} \tag{14}$$

Note that $t - ih$, equals to $h/2$, at mid-point of $[ih,(i+1)h]$. So, we can approximate $t - ih$, for $ih \le t < (i+1)h$, by $h/2$.

Now, expressing $\int_0^t \phi_i(\tau)d\tau$, in terms of the BPFs follows:

$$\int_0^t \phi_i(\tau)d\tau \cong \left[0,\ldots,0,\frac{h}{2},h,\ldots,h\right]\Phi(t) \tag{15}$$

in which $h/2$, is i th component.

Therefore

$$\int_0^t \Phi(\tau)d\tau \cong P\Phi(t), \tag{16}$$

where $P_{m\times m}$ is called operational matrix of integration and can be represented:

$$P = \frac{h}{2}\begin{pmatrix} 1 & 2 & 2 & \ldots & 2 \\ 0 & 1 & 2 & \ldots & 2 \\ 0 & 0 & 1 & \ldots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}. \tag{17}$$

So, the integral of every function $f(t)$ can be approximated as follows:

$$\int_0^t f(\tau)d\tau \cong \int_0^t F^T\Phi(\tau)d\tau \cong F^T P\Phi(t). \tag{18}$$

3. Direct method

In this section, a direct method is proposed to solve Volterra integro-differential equations. For convenience, we consider the following linear first order Volterra integro-differential equation of the form:

$$\begin{cases} x'(t) + q(t)x(t) = f(t) + \lambda\int_0^t k(t,s)x(s)ds, \\ x(0) = x_0 \end{cases} \tag{19}$$

In the above equation, $f(t)$, $q(t)$ and $k(t,s)$ are known functions but $x(t)$ is not.

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

Moreover, $k(t,s)\in L^2([0,1)\times[0,1))$ and $f(t),q(t)\in L^2([0,1))$. Note that the appearance in Eq.(19) of initial condition equation. This is necessary to ensure the existence of a unique solution.

Approximating functions $x(t), x'(t), f(t), q(t)$, and $k(t,s)$ with respect to BPFs by Eqs.(11) and (12) gives:

$$\begin{aligned} x(t) &\cong X^T\Phi(t)=\Phi^T(t)X \\ x'(t) &\cong X'^T\Phi(t)=\Phi^T(t)X' \\ f(t) &\cong F^T\Phi(t)=\Phi^T(t)F \\ q(t) &\cong Q^T\Phi(t)=\Phi^T(t)Q \\ k(t,s) &\cong \Phi^T(t)K\Phi(s) \end{aligned} \tag{20}$$

such that the m-vectors X, X', F, Q, and $m\times m$ matrix K are BPF coefficients of $x(t), x'(t), f(t), q(t)$, and $k(t,s)$, respectively. Note that in Eqs.(20), X and X' are unknown vectors. With substituting Eqs.(20) into Eq.(19) follows:

$$\begin{aligned} \Phi^T(t)X'+Q^T\Phi(t)\Phi^T(t)X &\cong \Phi^T(t)F+\lambda\int_0^t \Phi^T(t)K\Phi(s)\Phi^T(s)X\,ds \\ &\cong \Phi^T(t)F+\lambda\Phi^T(t)K\int_0^t \Phi(s)\Phi^T(s)X\,ds. \end{aligned} \tag{21}$$

Using operational matrix and Eq.(9) gives

$$\Phi^T(t)X'+\Phi^T(t)\tilde{Q}X\cong\Phi^T(t)F+\Phi^T(t)\lambda K\tilde{X}P\Phi(t). \tag{22}$$

Now, $\lambda K\tilde{X}P$ is an $m\times m$ matrix. From Eq.(10) follows:

$$\Phi^T(t)\lambda K\tilde{X}P\Phi(t)=\hat{X}^T\Phi(t), \tag{23}$$

where $\hat{X}$ is an m-vector with components equal to the diagonal entries of matrix $\lambda K\tilde{X}P$. So, combining Eqs.(22) and (23) gives

$$\Phi^T(t)X'+\Phi^T(t)\tilde{Q}X\cong\Phi^T(t)F+\Phi^T(t)\hat{X}, \tag{24}$$

or

$$X'+\tilde{Q}X\cong F+\hat{X}. \tag{25}$$

Now, X' must be computed in terms of X. Note that

$$\begin{aligned} x(t)-x(0) &= \int_0^t x'(\tau)d\tau \\ &\cong \int_0^t X'^T\Phi(\tau)d\tau \\ &\cong X'^T P\Phi(t). \end{aligned} \tag{26}$$

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

Therefore

$$x(t) \cong X'^T P \Phi(t) + X_0^T \Phi(t), \tag{27}$$

where X_0 is the m-vector of the form $X_0 = [x_0, x_0, \ldots, x_0]^T$, consequently, using Eq.(20) we have

$$X' \cong (P^T)^{-1}(X - X_0). \tag{28}$$

Substituting Eq.(28) into Eq.(25) and replacing $=$ with $\cong$, we obtain

$$(I + P^T \tilde{Q})X - P^T \hat{X} = P^T F + X_0. \tag{29}$$

Eq.(29) is a linear lower triangular system of m algebraic equations with m unknowns, components of X, which can be easily solved by forward substitution. So, an approximate solution $x(t) \cong X^T \Phi(t)$, is obtained for Eq.(19). Note that this approach does not use any projection method such as collocation, Galerkin, etc.

4. Numerical examples

Two examples are presented in this section to illustrate computational efficiency of the approach proposed in this paper. In these examples, the approximate solutions are briefly compared with exact solutions only at nine specific points. But, it must be noted that at mid-point of every subinterval $[ih, (i+1)h]$, for $i = 0, 1, \ldots, m-1$, the approximate solution is more accurate. Moreover, this accuracy will increase as m increases. This can be clearly followed from definition of operational matrix P.

All computations were performed using Matlab 7 on a **P**ersonal **C**omputer.

Example 1. Consider the following integro-differential equation [2]:

$$x'(t) = -1 + \frac{1}{2}t^2 - t e^t - \int_0^t s\, x(s)\, ds, \tag{30}$$

with the initial condition $x(0) = 0$, and the exact solution $x(t) = 1 - e^t$. See Table 1 for numerical results.

Example 2. Consider the following integro-differential equation:

$$x'(t) = -t e^t - e^{-t} + \int_0^t e^{t+s}\, x(s)\, ds, \tag{31}$$

with the initial condition $x(0) = 1$, and the exact solution $x(t) = e^{-t}$. The numerical results are shown in Table 2.

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

Table 1: Numerical results for example 1

t	Exact solution	Approximate solution, m=64	Approximate solution, m=128
0.1	-0.105171	-0.106933	-0.102592
0.2	-0.221403	-0.215727	-0.220458
0.3	-0.349859	-0.356241	-0.350924
0.4	-0.491825	-0.489539	-0.495336
0.5	-0.648721	-0.661699	-0.655186
0.6	-0.822119	-0.825017	-0.817865
0.7	-1.013753	-1.004387	-1.012193
0.8	-1.225541	-1.236052	-1.227294
0.9	-1.459603	-1.455819	-1.465389

Table 2: Numerical results for example 2

t	Exact solution	Approximate solution, m=32	Approximate solution, m=64
0.1	0.904837	0.896495	0.903450
0.2	0.818731	0.816261	0.822599
0.3	0.740818	0.743208	0.737371
0.4	0.670320	0.676693	0.671383
0.5	0.606531	0.597173	0.601822
0.6	0.548812	0.543729	0.547965
0.7	0.496585	0.495070	0.498927
0.8	0.449329	0.450767	0.447235
0.9	0.406570	0.410432	0.407214

5. Conclusion

The direct method based on BPFs and its operational matrix to solve Volterra integro-differential equation arising in many physical and engineering problems is presented. This approach transforms a Volterra integro-differential equation to a linear lower triangular system of algebraic equations. Its applicability and accuracy is checked on two examples. In these examples, the approximate solution is briefly compared with exact solution only at nine specific points. But, it must be noted that at mid-point of every subinterval $[ih,(i+1)h]$, for $i=0,1,\ldots,m-1$, the approximate solution is more accurate and this accuracy will increase as m increases. On the other hand, some points farther to mid-points may get worse as m increases. Of course, these oscillations are negligible. This can be clearly followed from definition of operational matrix P. As illustrated in Eqs.(14) and (15), the diagonal elements $h/2$ of operational matrix P are

[Downloaded from isharat.khu.ac.ir on 2026-10-29]

approximate values of $t - ih$. Note that, at mid-point of every subinterval $[ih,(i+1)h]$ the diagonal elements of operational matrix are exactly $h/2$. In general, It follows from numerical results that the accuracy of the obtained solutions are reasonable.

The advantages of this method are low cost of setting up the equations without applying any projection method such as Galerkin and Collocation methods. Also, the linear system (29) is a lower triangular system which can be easily solved by forward substitution with $O(m^2)$ operations. Therefore the count of operations is very low.

Finally, this method can be easily extended and applied to Volterra integro-differential equation of any order and systems of integro-differential equations with suitable initial conditions.

References

1. L. M. Delves and J. L. Mohamed, Computational Methods for Integral Equations, Cambridge University Press, Cambridge(1985).

2. A. M. Wazwaz, A First Course in Integral Equations, World Scientific, Singapor, (1997).

3. E. Babolian and L. M. Delves, A fast Galerkin scheme for linear integro-differential equation , IMA J. Numer. Anal. 1 (1981) 193-213.

4. K. Maleknejad and M. Hadizadeh, Numerical study of nonlinear Volterra integro-differential equations by Adomian's method, J. Sci. I.R. Iran 9(1) (1998).

5. K. Maleknejad and Y. Mahmoudi, Numerical solution of integro-differential equation by using hybrid Taylor and Block-Pulse functions, Far East J. Math. Sci. (FJMS) 9(2) (2003) 203-213.

6. E. Babolian and Z. Masouri, Direct method to solve Volterra integral equation of the first kind using operational matrix with block-pulse functions, J. Comput. Appl. Math.,in press, doi: 10.1016/j.cam.(2007).07.029.

7. E. Babolian and A. Salimi Shamloo, Numerical solution of Volterra integral and integro-differential equations of convolution type by using operational matrices of piecewise constant orthogonal functions, J. Comput. Appl. Math., 214 (2008) 495-508.

8. K. Maleknejad, M. Shahrezaee and H. Khatami, Numerical solution of integral equations system of the second kind by Block-Pulse functions, Appl. Math. Comput. 166 (2005) 15-24.

[Downloaded from isharat.khu.ac.ir on 2026-10-29]