Sample-and-Hold Functions for Solving Linear Volterra-Integral-Algebraic Equations

Osama H. Mohammed(a), Safaa H. Rasool(b)
AL-Nahrain University, College of science, Department of Mathematics and Computer Applications, Baghdad, Iraq.
Corresponding Author: Osama H. Mohammed

Abstract: In this paper, Sample-and-Hold functions are implemented to give approximate solutions for linear volterra integral-algebraic equations. The proposed method will transform the problem to a linear lower triangular system of algebraic equations using the operation matrix associated with the Sample-and-Hold functions. Convergence result and tested examples are given in order to check the validity and efficiency of the proposed method.


Date of Submission: 01-07-2017 Date of acceptance: 28-08-2017

I. Introduction

Couple system of integral algebraic equations (IAEs) including of the first and second kind volterra integral algebraic equations. These systems are also called as singular system of integral equations, naturally appear in mathematical model processes, e.g., the kernel identification subject in heat conduction and viscoelasticity [1], development of a chemical reaction within a small cell [2], the two dimensional biharmonic equation in a semi-infinite strip [3], dynamic procedure in chemical reactors [4] and Kirchhoff’s laws. (For more applications see [5,6] and references therein.)

In this paper, Sample-and-Hold functions will be used to solve linear volterra integral equations (LVIEs) with variable coefficients as given below:

\[ A(t)Y(t) = G(t) + \int_{0}^{t} K(t, r)Y(r) \, dr, \quad 0 \leq t \leq 1, \]  

Where \( A(t) = [a_{ij}(t)], \quad i, j = 1, 2, \ldots, n \),
\[ Y(t) = [Y_{1}(t), Y_{2}(t), \ldots, Y_{n}(t)]^{T}, \]
\[ G(t) = [G_{1}(t), G_{2}(t), \ldots, G_{n}(t)]^{T}, \]
\[ K(t, r) = [k_{ij}(t, r)], \quad i, j = 1, 2, \ldots, n \]

Since \( A(t), G(t), K(t, r) \) are known as functions and \( Y(t) \) is unknown. If \( \text{det} A(t) = 0 \), this system is denoted as Volterra Integral-Algebraic Equations (IAEs). Under the condition \( \text{det} A(t) = 0 \), the system can have a number of solutions or no solution at all. Sufficient conditions for the existence of unique continuous solution have been introduced in [7].

A little numerical methods are found, to solve these systems such as polynomial spline collocation method and its convergence results [8], Legendre collocation method [9], Jacobi collocation method including the matrix-vector multiplication representation [10], Multistep methods based on Adams quadratures rules and extrapolation formulas [11], Piecewise constant orthogonal functions such as Walsh functions [12,13], block-pulse functions [14], Haar functions [15,16].

This paper is ordered as following: in section 2, summary of Sample-and-Hold functions and their properties have been described. In section 3, the proposed method for solving volterra integral-algebraic equations have been presented. In section 4, we display illustrative examples, finally a conclusion have been drawn in section 5.

II. Generalized Sample-and-Hold functions (SHFs)

For any \( Y(t) \in L^2[0, A), Y(t) \) can be represented by a Sample-and-Hold functions. Consider the interval \([0, A) \) and by considering:

\[ y_{i}(t) \approx y(ih), \quad i = 1, \ldots, m \]
Where \( h = \frac{A}{m} \) \( y(ih) \) is the generosity of the function \( y(t) \) at time \( t \) and \( y(ih) \) is the sample of function \( y(t) \) at the point \( t = ih \).

The \( m \)-set of SHFs \( S_m(t) \), consisting of \( m \) element functions, and is defined as following [17]:

\[
S_i(t) = \begin{cases} 1, & \text{for } (i-1)h \leq t < ih \\ 0, & \text{elsewhere} \end{cases}
\]

Where \( i = 1, \ldots, m \).

### 2.1 Properties:

The Sample-and-Hold functions are similar to Block Pulse functions in aspects. Further it's easy to satisfy the following properties and by same way of [14].

(i) \( S_m(t)S_m(t) = \text{diag}(S_m(t)) \), where \( S_m(t) = [ S_1(t)S_2(t) \ldots S_m(t) ]^T \).

(ii) \( S_m(t)^T S_m(t) = 1 \).

(iii) \( S_m(t)U = US_m(t) \), \( U = \text{diag}(U) \), where \( U \) be an \( m \)-vector. \( \cdots \) (2)

(iv) \( S_m(t)^T S_m(t) = \hat{J} S_m(t) \) for every \( m \times m \) matrix \( J \) and \( \hat{J} \) is an \( m \)-vector with entries equal to the diagonal elements of \( J \). \( \cdots \) (3)

### 2.2 Operational Matrix of the Integration for SHFs

The integral \( \int_0^t S_m(\tau) d\tau \) is expanded in terms of SHFs and by arranging the coefficients in matrix form, we have

\[
\int_0^t S_m(\tau) d\tau = P_m S_m(t).
\]

Where \( P \) is given by [18]:

\[
P_m = \frac{h}{2} \begin{pmatrix} 1 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{m \times m}
\]

### 2.3 Function approximation

A square integrable time function \( y(t) \) of Lebesgue measure may also be extended into an \( m \)-term Sample-and-Hold functions series in \( t \in [0, A] \) as

\[
y(t) = \sum_{i=1}^{m} y_i S_i(t) = Y_m^T S_m(t) = S_m(t)^T Y_m ...
\]

Where \( Y_m \triangleq [y_1 \ y_2 \ldots \ y_m]^T \).

Where \( y_i = y(ih) \) the \( i^\text{th} \) sample of the function \( y(t) \). Actually, \( y_i \)'s are the sample of \( y(t) \) with the sampling period \( h \).

### 2.4 Convergence Analysis

Assume \( y \in L^2[0, A] \). Let \( S_m \) is called the set of all linear combination of Sample-and-Hold Functions \( S_i \), where \( i = 1, 2, \ldots, m \). We defined partial sum of SHFs of \( y \) as \( Z_m = \sum_{i=1}^m y_i S_i(t) \), where \( S_i \) is the \( i^\text{th} \) Sample-and-Hold functions.

By using orthogonality property,

\[
\|Z_m\|^2 = \left( \sum_{i=1}^m y_i S_i(t), \sum_{i=1}^m y_i S_i(t) \right) = \sum_{i=1}^m |y_i|^2 \|S_i\|^2
\]

Also, \( y - Z_m, S_i = \{y, S_i\} - \{z_m, S_i\} = y_i - y_i = 0 \), where \( i = 1, 2, \ldots, m \). Since \( \{y - Z_m, S_i\} = 0 \), we get \( y - Z_m = 0 \) then \( \|y - Z_m\| = 0 \), since \( \{y, S_i\} = \int_0^1 y S_i(t) dt \), where \( i = 1, 2, \ldots, m \).

Pythagorean theorem state,

\[
\|y\|^2 = \|y - Z_m\|^2 + \|Z_m\|^2
\]

From this we have, \( \|Z_m\|^2 = \sum_{i=1}^m |y_i|^2 \|S_i\|^2 \leq \|y\|^2 \).

The infinite series \( \sum_{i=1}^\infty |y_i|^2 \|S_i\|^2 \) has non-negative terms and it's partial sum are bounded above by \( \|y\|^2 \). It's convergence and satisfies Bessel's inequality \( \sum_{i=1}^\infty |y_i|^2 \|S_i\|^2 \leq \|y\|^2 \).

Since \( S_m \subseteq S_{m+1} \), the norm \( \|y - Z_m\| \) are decreasing and hence \( \lim_{m \to \infty} \|y - Z_m\| = 0 \), this garanties that the numerical solution converges to exact solution.

Further, from equation (5), we have \( \|y\|^2 = \lim_{m \to \infty} \|y - Z_m\|^2 = \sum_{i=1}^\infty |y_i|^2 \|S_i\|^2 \).

DOI: 10.9790/5728-1304033037 www.iosrjournals.org 31 | Page
III. The Approach

Let us consider the system of linear volterra integral-algebraic equations with variable coefficients (1), which can be written as:

\[ \sum_{q=1}^{n} a_{p,q}(t)y_q(t) = g_p(t) + \sum_{q=1}^{n} \int_{0}^{t} k_{p,q}(r, t)y_q(r)dr, \quad 0 \leq t \leq 1, \ldots, (6) \]

Where \( p = 1, 2, \ldots, n \) and \( r = \frac{t}{\lambda} \).

Writing the coefficient \( a_{p,q}, y_q, g_p \) and \( k_{p,q} \) in terms of Sample-and-Hold functions yields:

\[ a_{p,q}(t) \approx A_{p,q}^s \sum_{m} S_m(t)A_{p,q}, \]

\[ y_q(t) \approx Y_q^s S_m(t) = S_m^T(t)Y_q, \]

\[ g_p(t) \approx G_p^s S_m(t) = S_m^T(t)G_p, \]

\[ k_{p,q}(r, t) \approx \kappa_{p,q}^s T_m(r)S_m^T(r)Y_q \]

Where \( Y_q = [y_q^0, y_q^1 \ldots y_q^n]^T \), \( K_{p,q} = \left[ K_{p,q}^{l,j} \right] \) and \( A_{p,q} = \left[ a_{p,q}^{l,j} \right] \) with \( i, j = 1, \ldots, m \). The \( p \)th equations, from the system (6) can be extended in \( m \)-terms of Sample-and-Hold functions expansion as follows,

\[ \sum_{q=1}^{n} S_m^T(t)A_{p,q} Y_q^s S_m(t) \approx G_p^s S_m(t) + \sum_{q=1}^{n} \int_{0}^{t} S_m^T(t)K_{p,q} S_m(r) S_m^T(r)Y_q dr \]

\[ = G_p^s S_m(t) + S_m^T(t) \sum_{q=1}^{n} \int_{0}^{t} K_{p,q} S_m(r) S_m^T(r)Y_q dr \]

According to equation (2), we have:

\[ S_m^T(t) \sum_{q=1}^{m} A_{p,q} Y_q^s S_m(t) = G_p^s S_m(t) + S_m^T(t) \sum_{q=1}^{m} K_{p,q} \bar{Y}_{m+1}(r) dr \]

\[ = G_p^s S_m(t) + S_m^T(t) \sum_{q=1}^{m} K_{p,q} \bar{Y}_q^{m,1} S_m(t) \]

By using equation (3) we obtain:

\[ \bar{A}_p^{T} S_m(t) \approx G_p^s S_m(t) + \bar{Y}_p^{T} S_m(t) \]

Hence

\[ \bar{A}_p^{T} \approx G_p + \bar{Y}_p \]

Or

\[ \bar{A}_p \approx G_p + \bar{Y}_p \ldots (7) \]

Where \( \bar{Y}_p \) is an \( m \)-vector with element equal to the diagonal entries of the \( m \times m \) matrix \( \sum_{q=1}^{m} K_{p,q} Y_q P \) and \( \bar{A}_p \) is an \( m \)-vector with element equal to the diagonal entries of the \( m \times m \) matrix \( \sum_{q=1}^{m} A_{p,q} Y_q^T \), since \( \bar{Y}_q = \text{diag}(Y_q) \).

Now we can calculate \( \bar{Y}_p \) as

\[
\bar{Y}_p = \begin{bmatrix}
\frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(1,1)} Y_q^1 \\
\frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(2,1)} Y_q^1 + \frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(2,2)} Y_q^2 \\
\frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(3,1)} Y_q^1 + \frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(3,2)} Y_q^2 + \frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(3,3)} Y_q^3 \\
\vdots \\
\frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(m,1)} Y_q^1 + \frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(m,2)} Y_q^2 + \cdots + \frac{h}{2} \sum_{q=1}^{n} K_{p,q}^{(m,m)} Y_q^n
\end{bmatrix}
\]

Which can be written as:

DOI: 10.9790/5728-1304033037 www.iosrjournals.org 32 | Page
-held Functions for Solving Linear Volterra Integral-Algebraic Equations

\[
\begin{bmatrix}
\frac{1}{2} \sum_{q=1}^{n} K_{p,q}^{(1,1)} & 0 & 0 & \ldots & 0 \\
\frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(2,1)} & \frac{1}{2 h} \sum_{q=1}^{n} K_{p,q}^{(2,2)} & 0 & \ldots & 0 \\
\frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(3,1)} & \frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(3,2)} & \frac{1}{2 h} \sum_{q=1}^{n} K_{p,q}^{(3,3)} & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(m,1)} & \frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(m,2)} & \frac{1}{h} \sum_{q=1}^{n} K_{p,q}^{(m,3)} & \ldots & \frac{1}{2 h} \sum_{q=1}^{n} K_{p,q}^{(m,m)}
\end{bmatrix}
\]

Hence

\[
\begin{bmatrix}
\frac{1}{2m} K_{p,q}^{(1,1)} & 0 & 0 & \ldots & 0 \\
\frac{1}{m} K_{p,q}^{(2,1)} & \frac{1}{2m} K_{p,q}^{(2,2)} & 0 & \ldots & 0 \\
\frac{1}{m} K_{p,q}^{(3,1)} & \frac{1}{m} K_{p,q}^{(3,2)} & \frac{1}{2m} K_{p,q}^{(3,3)} & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{m} K_{p,q}^{(m,1)} & \frac{1}{m} K_{p,q}^{(m,2)} & \frac{1}{m} K_{p,q}^{(m,3)} & \ldots & \frac{1}{2m} K_{p,q}^{(m,m)}
\end{bmatrix}
\]

And

\[
\begin{bmatrix}
\sum_{q=1}^{n} a_{p,q}^{(1)} y_{q1} \\
\sum_{q=1}^{n} a_{p,q}^{(2)} y_{q2} \\
\vdots \\
\sum_{q=1}^{n} a_{p,q}^{(m)} y_{qm}
\end{bmatrix}
\]

Substituting equation (7) into equation (1), also replace \( \approx \) with equality, we obtains,

\[
\begin{bmatrix}
A^{(1)} & Y^{(1)} \\
A^{(2)} & Y^{(2)} \\
\vdots & \vdots \\
A^{(m)} & Y^{(m)}
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2m} K^{(1,1)} & 0 & 0 & \ldots & 0 \\
\frac{1}{m} K^{(2,1)} & \frac{1}{2m} K^{(2,2)} & 0 & \ldots & 0 \\
\frac{1}{m} K^{(3,1)} & \frac{1}{m} K^{(3,2)} & \frac{1}{2m} K^{(3,3)} & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{m} K^{(m,1)} & \frac{1}{m} K^{(m,2)} & \frac{1}{m} K^{(m,3)} & \ldots & \frac{1}{2m} K^{(m,m)}
\end{bmatrix}
\]

This can be written as:

\[
A^{(i)} Y^{(i)} = G^{(i)} + \frac{1}{2m} K^{(i,i)} Y^{(i)} + \frac{1}{m} \sum_{j=1}^{i-1} K^{(i,j)} Y^{(j)}, \quad i = 1, 2, \ldots, m , \quad \ldots (8)
\]

Where \( Y^{(i)} = [y_{1i} \ldots y_{ni}]^T \), \( A^{(i)} = [a_{p,q}^{(i)}] \) and \( K^{(i,i)} = [k_{p,q}^{(i,i)}] \) for \( p,q = 1, 2, \ldots, m \). By simplifying Eq. (8), we get:

\[
Y^{(i)} = [A^{(i)} - \frac{1}{2m} K^{(i,i)}]^{-1} \begin{bmatrix} G^{(i)} + \frac{1}{m} \sum_{j=1}^{i-1} K^{(i,j)} Y^{(j)} \end{bmatrix}, i = 1, 2, \ldots, m , \quad \ldots (9)
\]

If \( A^{(i)} = \frac{1}{2m} K^{(i,i)} \) transform to be nonsingular and hence equation (9) given Sample-and-Hold coefficients recursively. Employing this coefficients with \( Y(t) = [Y^{(0)} \ Y^{(1)} \ldots Y^{(m-1)}]S_m(t) \), now the numerical solution can be facilitate calculated.

DOI: 10.9790/5728-1304033037  www.iosrjournals.org 33 | Page
IV. Illustrative Examples:

In this section, we represent three illustrative examples in order to utilize the efficiency and accuracy of the proposed method.

Example (1):

Assume the system of linear volterra integral equations with variable coefficients.

\[
\begin{bmatrix}
1 \\
-2t
\end{bmatrix}
\begin{bmatrix}
f_1(t) \\
f_2(t)
\end{bmatrix} =
\begin{bmatrix}
sin(t) + t\cos(t) \\
cos(t) - 2t\sin(t)
\end{bmatrix}
+ 
\int_0^t
\begin{bmatrix}
t^2\cos(r) \\
\sin(t)\cos(r)
\end{bmatrix}
\begin{bmatrix}
-\sin(r) \\
-\sin(t)\sin(r)
\end{bmatrix}
\begin{bmatrix}
f_1(r) \\
f_2(r)
\end{bmatrix} dr \quad (10)
\]

Where the exact solution is found in [19] as \( f_1(t) = \sin t \). \( f_2(t) = \cos t \).

Following tables (1 - 3) represent a comparison between the approximate solution of equation (10) using the proposed method and Block Pulse functions (BPFs) [14] with the exact solution.

Table (1): Comparison between the approximate solution of equation (10) using the proposed method and Block Pulse method with the exact solution when \( m=32 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) m=32</th>
<th>( f_1 ) (BPFs) m=32</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>3.952E-001</td>
<td>3.894E-001</td>
<td>3.894E-001</td>
<td>5.749E-003</td>
<td>8.78E-003</td>
</tr>
<tr>
<td>0.7</td>
<td>6.584E-001</td>
<td>6.472E-001</td>
<td>6.442E-001</td>
<td>1.423E-002</td>
<td>2.02E-003</td>
</tr>
<tr>
<td>0.9</td>
<td>7.872E-001</td>
<td>7.649E-001</td>
<td>7.833E-001</td>
<td>3.87E-003</td>
<td>1.839E-002</td>
</tr>
</tbody>
</table>

Table (2): Comparison between the approximate solution of equation (10) using the proposed method and Block Pulse method with the exact solution when \( m=128 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) m=128</th>
<th>( f_1 ) (BPFs) m=128</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>3.952E-001</td>
<td>3.913E-001</td>
<td>3.894E-001</td>
<td>5.749E-003</td>
<td>1.9E-003</td>
</tr>
<tr>
<td>0.7</td>
<td>6.446E-001</td>
<td>6.393E-001</td>
<td>6.442E-001</td>
<td>2.387E-003</td>
<td>4.87E-003</td>
</tr>
<tr>
<td>0.9</td>
<td>7.872E-001</td>
<td>7.716E-001</td>
<td>7.833E-001</td>
<td>3.87E-003</td>
<td>1.174E-002</td>
</tr>
</tbody>
</table>

Table (3): Comparison between the approximate solution of equation (10) using the proposed method and Block Pulse method with the exact solution when \( m=256 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) m=256</th>
<th>( f_1 ) (BPFs) m=256</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>3.916E-001</td>
<td>3.905E-001</td>
<td>3.894E-001</td>
<td>2.158E-003</td>
<td>1.092E-004</td>
</tr>
<tr>
<td>0.7</td>
<td>6.442E-001</td>
<td>6.404E-001</td>
<td>6.442E-001</td>
<td>2.387E-003</td>
<td>3.439E-003</td>
</tr>
<tr>
<td>0.9</td>
<td>7.848E-001</td>
<td>7.705E-001</td>
<td>7.833E-001</td>
<td>1.455E-003</td>
<td>1.283E-002</td>
</tr>
</tbody>
</table>

DOI: 10.9790/5728-1304033037  www.iosrjournals.org 34 | Page
Example (2) :
Assume the system of linear volterra integral-algebraic equations with variable coefficients.

\[
\begin{pmatrix}
0 & 0 \\
t & -2t
\end{pmatrix}
\begin{pmatrix}
f_1(t) \\
f_2(t)
\end{pmatrix}
= \begin{pmatrix}
\frac{-t^2}{3} \\
-5\frac{t^3}{3} + \frac{7t^4}{6}
\end{pmatrix}
\int_0^t \left( \begin{pmatrix}
3r \\
2r(t + r) + 2
\end{pmatrix}
\begin{pmatrix}
f_1(r) \\
f_2(r)
\end{pmatrix}
\right) dr
\]

...(11)

Where the exact solution is found in [14] as \( f_1(t) = 1 + t \) and \( f_2(t) = -t \).

Following tables (4 - 6) represent a comparison between the approximate solution of equation (11) using the proposed method and Block Pulse functions (BPFs) [14] with the exact solution.

### Table (4)
Comparison between the approximate solution of equation (11) using the proposed method and Block Pulse method with the exact solution when \( m=32 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) ( m=32 )</th>
<th>( f_1 ) (BPFs) ( m=32 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1.376E+000</td>
<td>9.324E-001</td>
<td>1.4E+000</td>
<td>2.374E-002</td>
<td>4.676E-001</td>
</tr>
<tr>
<td>0.7</td>
<td>1.684E+000</td>
<td>9.096E-001</td>
<td>1.7E+000</td>
<td>1.645E-002</td>
<td>7.991E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>1.896E+000</td>
<td>8.368E-001</td>
<td>1.9E+000</td>
<td>3.133E-002</td>
<td>1.063E-001</td>
</tr>
</tbody>
</table>

### Table (5)
Comparison between the approximate solution of equation (11) using the proposed method and Block Pulse method with the exact solution when \( m=128 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) ( m=128 )</th>
<th>( f_1 ) (BPFs) ( m=128 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1.399E+000</td>
<td>9.343E-001</td>
<td>1.4E+000</td>
<td>1.253E-003</td>
<td>4.427E-001</td>
</tr>
<tr>
<td>0.7</td>
<td>1.694E+000</td>
<td>1.232E+000</td>
<td>1.7E+000</td>
<td>5.652E-003</td>
<td>4.681E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>1.897E+000</td>
<td>1.518E+000</td>
<td>1.9E+000</td>
<td>3.188E+003</td>
<td>3.816E-001</td>
</tr>
</tbody>
</table>

### Table (6)
Comparison between the approximate solution of equation (11) using the proposed method and Block Pulse method with the exact solution when \( m=256 \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f_1 ) (SHFs) ( m=256 )</th>
<th>( f_1 ) (BPFs) ( m=256 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1.399E+000</td>
<td>8.721E-001</td>
<td>1.4E+000</td>
<td>1.423E-003</td>
<td>5.279E-001</td>
</tr>
<tr>
<td>0.7</td>
<td>1.699E+000</td>
<td>8.222E-001</td>
<td>1.7E+000</td>
<td>1.276E-003</td>
<td>8.778E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>1.899E+000</td>
<td>7.791E-001</td>
<td>1.9E+000</td>
<td>2.388E-003</td>
<td>1.128E+000</td>
</tr>
</tbody>
</table>

Example (3) :
Assume the system of linear volterra integral-algebraic equations with variable coefficients.

\[
\begin{pmatrix}
0 & 0 \\
t & -2t
\end{pmatrix}
\begin{pmatrix}
f_1(t) \\
f_2(t)
\end{pmatrix}
= \begin{pmatrix}
\frac{-t^2}{3} \\
-5\frac{t^3}{3} + \frac{7t^4}{6}
\end{pmatrix}
\int_0^t \left( \begin{pmatrix}
3r \\
2r(t + r) + 2
\end{pmatrix}
\begin{pmatrix}
f_1(r) \\
f_2(r)
\end{pmatrix}
\right) dr
\]

...(12)

With \( g_1(t) = 1 - (1 + t + t^3) \sin t - \frac{1}{5} (3 + \cos 3t) (\sin^2 \frac{3t}{2}) \)

DOI: 10.9790/5728-1304033037 www.iosjournals.org 35 | Page
where the exact solution is found in [9] as \( f_1(t) = \cos t \cdot f_2(t) = \sin 3t \).

Following tables (7 - 9) represent a comparison between the approximate solution of equation (12) using the proposed method and Block Pulse functions (BPFs) [14] with the exact solution.

### Table (7)
Comparison between the approximate solution of equation (12) using the proposed method and Block Pulse method with the exact solution when \( m=32 \).

<table>
<thead>
<tr>
<th></th>
<th>( f_1 ) (SHFs) ( m=32 )</th>
<th>( f_2 ) (BPFs) ( m=32 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>9.232E-001</td>
<td>8.765E-001</td>
<td>9.211E-001</td>
<td>1.135E-002</td>
<td>4.455E-002</td>
</tr>
<tr>
<td>0.7</td>
<td>7.700E-001</td>
<td>5.080E-001</td>
<td>7.648E-001</td>
<td>5.166E-003</td>
<td>2.568E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>6.13E-001</td>
<td>1.13E-001</td>
<td>6.21E-001</td>
<td>1.972E-002</td>
<td>5.079E-001</td>
</tr>
</tbody>
</table>

### Table (8)
Comparison between the approximate solution of equation (12) using the proposed method and Block Pulse method with the exact solution when \( m=128 \).

<table>
<thead>
<tr>
<th></th>
<th>( f_1 ) (SHFs) ( m=128 )</th>
<th>( f_2 ) (BPFs) ( m=128 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>9.225E-001</td>
<td>8.676E-001</td>
<td>9.211E-001</td>
<td>1.454E-003</td>
<td>5.342E-002</td>
</tr>
<tr>
<td>0.7</td>
<td>7.674E-001</td>
<td>5.146E-001</td>
<td>7.648E-001</td>
<td>2.584E-003</td>
<td>2.502E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>6.23E-001</td>
<td>8.39E-002</td>
<td>6.21E-001</td>
<td>1.689E-003</td>
<td>5.385E-001</td>
</tr>
</tbody>
</table>

### Table (9)
Comparison between the approximate solution of equation (12) using the proposed method and Block Pulse method with the exact solution when \( m=256 \).

<table>
<thead>
<tr>
<th></th>
<th>( f_1 ) (SHFs) ( m=256 )</th>
<th>( f_2 ) (BPFs) ( m=256 )</th>
<th>Exact ( f_1 )</th>
<th>Absolute Error ( f_1 ) (SHFs)</th>
<th>Absolute Error ( f_1 ) (BPFs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>9.221E-001</td>
<td>8.691E-001</td>
<td>9.211E-001</td>
<td>1.024E-003</td>
<td>5.193E-002</td>
</tr>
<tr>
<td>0.7</td>
<td>7.651E-001</td>
<td>5.114E-001</td>
<td>7.648E-001</td>
<td>2.355E-004</td>
<td>2.335E-001</td>
</tr>
<tr>
<td>0.9</td>
<td>6.23E-001</td>
<td>8.833E-002</td>
<td>6.21E-001</td>
<td>1.449E-003</td>
<td>5.333E-001</td>
</tr>
</tbody>
</table>

### V. Conclusion

In this manuscript, we have used Sample-and-hold functions method for the numerical solutions of linear volterra integral-algebraic equations. The method is calculable on SHFs and operational matrix of integration. The method trade with system of linear volterra integral equations with variable coefficients and volterra integral-algebraic equations with same ease. The systems of algebraic equations are transform to linear lower triangular systems. Numerical results, which assure optical results, demonstrate the effectiveness and applicability of the method. Further, the main advantage is that the attending method, which is facile and direct, needs less computational effort than other numerical methods.

DOI: 10.9790/5728-1304033037 www.iosrjournals.org 36 | Page
References


