Solution of Nonlinear Volterra-Fredholm Integrodifferential Equations via Hybrid of Block-Pulse Functions and Lagrange Interpolating Polynomials

Hamid Reza Marzban and Sayyed Mohammad Hoseini

Department of Mathematical Sciences, Isfahan University of Technology, P.O. Box 8415683111, Isfahan, Iran

Correspondence should be addressed to Hamid Reza Marzban, hmarzban@cc.iut.ac.ir

Received 22 May 2012; Accepted 8 November 2012

1. Introduction

Integral and integrodifferential equations have many applications in various fields of science and engineering such as biological models, industrial mathematics, control theory of financial mathematics, economics, electrostatics, fluid dynamics, heat and mass transfer, oscillation theory, queuing theory, and so forth [1].

It is well known that it is extremely difficult to analytically solve nonlinear integrodifferential equations. Indeed, few of these equations can be solved explicitly. So it is required to devise an efficient approximation scheme for solving these equations. So far, several numerical methods are developed. The solution of the first order integrodifferential equations has been obtained by the numerical integration methods such as Euler-Chebyshev [2] and Runge-Kutta methods [3].
Moreover, a differential transform method for solving integrodifferential equations was introduced in [4]. Shidfar et al. [5] applied the homotopy analysis method for solving the nonlinear Volterra and Fredholm integrodifferential equations. As a concrete example, we can express the mathematical model of cell-to-cell spread of HIV-1 in tissue cultures considered by Mittler et al. [6]. Yalcinbas and Sezer [7] proposed an approximation scheme based on Taylor polynomials for solving the high-order linear Volterra-Fredholm integrodifferential equations of the following form:

\[
\sum_{j=0}^{q} f_j(t)y^{(j)}(t) = f(t) + \lambda_1 \int_{0}^{t} k_1(t,s)y(s)ds + \lambda_2 \int_{0}^{t} k_2(t,s)y(s)ds, \quad 0 \leq t, s \leq t_f. \quad (1.1)
\]

Maleknejad and Mahmoudi [8] developed a numerical method by using Taylor polynomials to solve the following type of nonlinear Volterra-Fredholm integrodifferential equations:

\[
\sum_{j=0}^{q} f_j(t)y^{(j)}(t) = f(t) + \lambda_1 \int_{0}^{t} k_1(t,s)[y(s)]^\gamma ds + \lambda_2 \int_{0}^{t} k_2(t,s)y(s)ds, \quad 0 \leq t, s \leq t_f. \quad (1.2)
\]

Darania and Ivaz [9] suggested an efficient analytical and numerical procedure for solving the most general form of nonlinear Volterra-Fredholm integrodifferential equations

\[
\sum_{j=0}^{q} f_j(t)y^{(j)}(t) = f(t) + \lambda_1 \int_{0}^{t} k_1(t,s)g_1(s,y(s))ds + \lambda_2 \int_{0}^{t} k_2(t,s)g_2(s,y(s))ds, \quad (1.3)
\]

under the mixed conditions

\[
\sum_{j=0}^{q-1} \left( \alpha_{ij}y^{(j)}(0) + \beta_{ij}y^{(j)}(t_f) \right) = \mu_i, \quad i = 0, 1, \ldots, q - 1, \quad (1.4)
\]

where \( \lambda_1 \) and \( \lambda_2 \) are constants and \( 0 \leq t, s \leq t_f \). Moreover, \( f(t), k_1(t,s), k_2(t,s), g_1(s,y(s)), g_2(s,y(s)) \), and \( f_j(t), j = 0, 1, \ldots, q \), are functions that have suitable derivatives on the interval \( 0 \leq t, s \leq t_f \).

These kinds of equations can be found in numerous applications such as electromagnetics, electromagnetic, biomechanics, and elasticity [10–12]. Akyuz and Sezer [13] presented a Taylor polynomial approach for solving high-order linear Fredholm integrodifferential equations in the most general form. Streltsov [14] developed an effective numerical method based on the use of Chebyshev and Legendre polynomials for solving Fredholm integral equations. Babolian et al. [15] suggested an effective direct method to determine the numerical solution of the specific nonlinear Volterra-Fredholm integrodifferential equations. Their approach was based on triangular functions. In [16, 17], the variational iteration method (VIM) was employed for solving integral and integrodifferential equations. In addition, iterative and noniterative methods for the solution of nonlinear Volterra integrodifferential equations were presented and their local convergence was proved. The iterative methods provide a sequence solution and make use of fixed-point theory whereas the noniterative ones result in series solutions and also make use of the fixed-point principles.
In recent years, the meshless methods have gained more attention not only by mathematicians but also in the engineering community. In [19], the meshless moving least square method was employed for solving nonlinear Fredholm integrodifferential equations.

In the present paper, we introduce a new numerical method for solving (1.3) with the mixed conditions (1.4). The method is based on a hybrid of block-pulse functions and Lagrange interpolating polynomials. The associated operational matrices of integration and product together with the Kronecker property of hybrid functions are then used to reduce the solution of (1.3) and (1.4) to the solution of nonlinear algebraic equations whose solution is much more easier than the original one.

The paper is organized as follows. In Section 2, we describe the basic properties of hybrid functions. Section 3 is devoted to the solution of the nonlinear Volterra-Fredholm integrodifferential equations. Finally, in Section 4, various types of integrodifferential equations are given to demonstrate the efficiency and the accuracy of the proposed method.

2. Hybrid Functions

2.1. Properties of Hybrid Functions

In order to define hybrid functions, we first divide the interval $[0, t_f]$ into $N$ equidistant subintervals $([(n - 1)/N] t_f, (n/N) t_f)$, for $n = 1, 2, \ldots, N$. The hybrid of block-pulse functions and Lagrange interpolating polynomials, $b_{nm}(t), n = 1, 2, \ldots, N, m = 0, 1, \ldots, M - 1$, are defined on the interval $[0, t_f]$ as,

\[
b_{nm}(t) = \begin{cases} 
L_m\left(\frac{2N}{t_f} t - 2n + 1\right), & t \in \left(\frac{1}{N} t_f, \frac{n}{N} t_f\right), \\
0, & \text{otherwise}
\end{cases}
\]

(2.1)

where $n$ and $m$ are the orders of block-pulse functions and Lagrange interpolating polynomials, respectively. Here, $L_m(t)$ are defined in [21] as

\[
L_m(t) = \prod_{i=0}^{M-1} \left(\frac{t - \tau_i}{\tau_m - \tau_i}\right)^{m-1}, \quad m = 0, 1, \ldots, M - 1,
\]

(2.2)

in which $\tau_i, i = 0, 1, \ldots, M - 1$ are the zeros of Legendre polynomial of order $M$, with the Kronecker property

\[
L_m(\tau_i) = \delta_{mi} = \begin{cases} 
1, & \text{if } i = m, \\
0, & \text{if } i \neq m
\end{cases}
\]

(2.3)

where $\delta_{mi}$ is the Kronecker delta function. No explicit formulas are known for the points $\tau_i$. However, they can be computed numerically using existing subroutines [21]. Since $b_{nm}(t)$ consists of block-pulse functions and Lagrange interpolating polynomials, which are both complete and orthogonal, the set of hybrid of block-pulse functions and Lagrange interpolating polynomials is a complete orthogonal set in the Hilbert space $L^2[0, t_f]$. A more detailed error analysis concerning this type of hybrid functions was discussed in [22].
2.2. Function Approximation

A function \( f(t) \in \mathcal{L}^2[0,t_f] \) may be expanded as

\[
f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} f_{nm} b_{nm}(t). \tag{2.4}
\]

If the infinite series in (2.4) is truncated, then (2.4) can be written as

\[
f(t) \approx \sum_{n=1}^{N} \sum_{m=0}^{M-1} f_{nm} b_{nm}(t) = F^TB(t), \tag{2.5}
\]

where

\[
F = [f_{10}, \ldots, f_{1(M-1)}, \ldots, f_{N0}, \ldots, f_{N(M-1)}]^T,
\]

\[
B(t) = [b_{10}(t), \ldots, b_{1(M-1)}(t), \ldots, b_{N0}(t), \ldots, b_{N(M-1)}(t)]^T.
\] \tag{2.6}

In (2.6), \( f_{nm} \), \( n = 1,2,\ldots,N \), \( m = 0,1,\ldots,M-1 \) are the expansion coefficients of the function \( f(t) \) in the \( n \)th subinterval \([((n-1)/N)t_f, (n/N)t_f)\), and \( b_{nm}(t) \), \( n = 1,2,\ldots,N \), \( m = 0,1,\ldots,M-1 \) are defined in (2.1). In order to evaluate the coefficients \( f_{nm} \), we use Gaussian nodes which are defined in \([-1,1]\) and let \( t_{nm} \), \( n = 1,2,\ldots,N \), \( m = 0,1,\ldots,M-1 \) be the corresponding Gaussian nodes in the \( n \)th subinterval \([((n-1)/N)t_f, (n/N)t_f)\]. The relation between \( \tau_m \) and \( t_{nm} \) is given by

\[
t_{nm} = \frac{t_f}{2N}(\tau_m + 2n - 1). \tag{2.7}
\]

With the aid of (2.3) we have

\[
b_{nm}(t_{ij}) = \begin{cases} 
1, & \text{if } i = n, j = m, \\
0, & \text{otherwise.}
\end{cases} \tag{2.8}
\]

Using (2.8), the coefficients \( f_{nm} \), can be obtained as

\[
f_{nm} = f(t_{nm}). \tag{2.9}
\]

Now, let \( g(t,s) \) be a function of two independent variables defined for \( t \in [0,t_f] \) and \( s \in [0,s_f] \). Then \( g(t,s) \) can be expanded in terms of hybrid functions as follows:

\[
g(t,s) \approx B^T(t)GB(s), \tag{2.10}
\]
where \( G \) is a matrix of order \( NM \times NM \) whose elements can be calculated for \( i, n = 1, 2, \ldots, N \), and \( j, m = 0, 1, \ldots, M - 1 \) as follows:

\[
G_{(n-1)M+m,(i-1)M+j} = g(t_{nm}, s_{ij}),
\]

in which \( s_{ij} \) can be determined by the following relation:

\[
s_{ij} = \frac{s_f}{2N}(\tau_j + 2i - 1).
\]

The integration of the vector \( B(t) \) can also be expanded in terms of hybrid functions as

\[
\int_0^t B(s) \, ds \approx PB(t),
\]

where \( P \) is the \( NM \times NM \) operational matrix of integration and given by, [20],

\[
P = \begin{bmatrix}
E & H & H & \cdots & H \\
0 & E & H & \cdots & H \\
0 & 0 & E & \cdots & H \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & E \\
\end{bmatrix},
\]

in which \( E \) and \( H \) can be obtained in the following manner.

Let

\[
E = (e_{ij}), \quad H = (h_{ij}),
\]

then, for \( i, j = 0, 1, \ldots, M - 1 \), we have

\[
e_{ij} = \frac{t_f}{2N} \int_{-1}^{\tau_j} L_i(t)dt,
\]

\[
h_{ij} = \frac{t_f}{2N} \int_{-1}^{1} L_i(t)dt,
\]

where \( \tau_j, j = 0, 1, \ldots, M - 1 \) are the zeros of Legendre polynomial of order \( M \). It is noted that \( E \) is the associated operational matrix of integration for Lagrange interpolating polynomials on interval \([((n-1)/N)t_f, (n/N) t_f]\).

Integrating (2.5) from 0 to \( t \) and using (2.13) imply

\[
\int_0^t f(s)ds \approx F^T PB(t).
\]
From (2.13), by using the interpolating property of hybrid functions we get

$$
\int_0^{t_{nm}} f(s)ds \approx F^n PB(t_{nm}) = \sum_{i=1}^{n-1} \sum_{j=0}^{M-1} f_{ij} h_{jm} + \sum_{j=0}^{M-1} f_{nj} e_{jm}. \tag{2.18}
$$

### 2.3. The Integration of the Cross-Product of Two Hybrid Function Vectors

The following property of the cross-product of two hybrid function vectors will also be used. Integrating the square matrix $B(t)B^T(t)$ from 0 to $t_f$, and using the orthogonality of hybrid functions, we get

$$
W = \int_0^{t_f} B(t)B^T(t)dt, \tag{2.19}
$$

where $W$ is a block diagonal matrix of order $NM \times NM$ given by

$$
W = \begin{bmatrix}
A & 0 & \cdots & 0 \\
0 & A & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & A
\end{bmatrix}, \tag{2.20}
$$

moreover $A$ is also a diagonal matrix of order $M \times M$ given as

$$
A = \text{diag}[a_0, a_1, \ldots, a_{M-1}]. \tag{2.21}
$$

According to the orthogonality of hybrid functions, the values of $a_m$, for $m = 0, 1, \ldots, M - 1$, can be computed as follows:

$$
a_m = \frac{t_f}{2N} \int_{-1}^{1} L_m^2(t)dt. \tag{2.22}
$$

### 2.4. The Operational Matrix of Derivative

Let $d/dt$ denote the derivative operator. By expanding $(d/dt)B(t)$ in terms of $B(t)$ we get

$$
\frac{d}{dt} B(t) = D^* B(t). \tag{2.23}
$$
In (2.23), $D^*$ is called the operational matrix of derivative for hybrid functions which is a block diagonal matrix of order $NM \times NM$ and is given by

$$D^* = \begin{bmatrix}
  D & 0 & \cdots & 0 \\
  0 & D & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & D
\end{bmatrix}, \quad (2.24)$$

where $D$ is also a matrix of order $M \times M$ and the $(i, j)$ th element of this matrix denoted by $[D]_{ij}$ for $i, j = 0, 1, \ldots, M - 1$ can be calculated as follows:

$$[D]_{ij} = \frac{dL_i}{dt}(t_{ij}) = \frac{2N}{t_f} L'_i \left( \frac{2N}{t_f} t_{ij} - 2j + 1 \right) = \frac{2N}{t_f} L'_i(\tau_j). \quad (2.25)$$

Let $f^{(1)}(t)$ denote the first derivative of $f(t)$ with respect to $t$. By expanding $f^{(1)}(t)$ in terms of hybrid functions and using (2.5) and (2.23) we obtain

$$f^{(1)}(t) \approx \sum_{n=1}^{N} \sum_{m=0}^{M-1} f^{(1)}(t_{nm}) b_{nm}(t) = F^T D^* B(t). \quad (2.26)$$

According to (2.8) we get

$$f^{(1)}(t_{nm}) = F^T D^* B(t_{nm}) = \sum_{l=0}^{M-1} f_{nl} [D]_{lm}. \quad (2.27)$$

Let $f^{(j)}(t)$ denote the $j$th derivative of $f(t)$ with respect to $t$. In order to obtain the relation between the expansion coefficients of $f^{(j)}(t)$ and $f(t)$, we use the following fact:

$$\frac{d^j}{dt^j} B(t) = (D^*)^j B(t), \quad (2.28)$$

where $(D^*)^j$ is the $j$th power of the matrix $D^*$ given in (2.23). Equation (2.28) can easily be obtained from (2.23) by differentiating $f(t)$, $j$ times with respect to $t$, $j = 2, 3, \ldots, q$. It should be noted that the matrix $(D^*)^j$ has the following structure:

$$(D^*)^j = \begin{bmatrix}
  D^j & 0 & \cdots & 0 \\
  0 & D^j & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & D^j
\end{bmatrix}. \quad (2.29)$$
By expanding \( f^{(j)}(t) \) in terms of hybrid functions and using (2.5) and (2.28) we get

\[
f^{(j)}(t) = \sum_{n=1}^{N} \sum_{m=0}^{M-1} f^{(j)}(t_{nm}) b_{nm}(t) = F^T (D^*)^j B(t). \tag{2.30}
\]

According to (2.8) it immediately follows that

\[
f^{(j)}(t_{nm}) = F^T (D^*)^j B(t_{nm}) = \sum_{l=0}^{M-1} y_{nl} [D^j]_{lm}. \tag{2.31}
\]

where \([D^j]_{lm}\) denotes the \((l, m)\)th element of the matrix \(D^j\).

### 3. Solution of the Nonlinear Volterra-Fredholm Integrodifferential Equations

Consider the nonlinear Volterra-Fredholm integrodifferential equation given in (1.3). Since the set of hybrid functions is a complete orthogonal set in the Hilbert space \(L^2[0, t_f]\), we can expand any function in this space in terms of hybrid functions.

Let

\[
y(t) = \sum_{n=1}^{N} \sum_{m=0}^{M-1} y_{nm} b_{nm}(t) = Y^T B(t), \tag{3.1}
\]

where

\[
y_{nm} = y(t_{nm}). \tag{3.2}
\]

Using (2.30), for \(j = 1, 2, \ldots, q\), we have

\[
y^{(j)}(t) = \sum_{n=1}^{N} \sum_{m=0}^{M-1} y^{(j)}(t_{nm}) b_{nm}(t). \tag{3.3}
\]

where

\[
y^{(j)}(t_{nm}) = \sum_{l=0}^{M-1} y_{nl} [D^j]_{lm}. \tag{3.4}
\]

Moreover, by expanding \(k_2(t, s)\) in terms of hybrid functions we get

\[
k_2(t, s) = B^T(t) K_2 B(s), \tag{3.5}
\]
where $K_2$ is a matrix of order $NM \times NM$ and can be calculated similar to matrix $G$ in (2.10). From (2.5) and (2.9) we obtain

\[
g_2(t, y(t)) = \sum_{n=1}^{N} \sum_{m=0}^{M-1} g_2(t_{nm}, y(t_{nm})) b_{nm}(t) = \sum_{n=1}^{N} \sum_{m=0}^{M-1} g_2(t_{nm}, y_{nm}) b_{nm}(t) = B^T(t)G_2,
\]

where $G_2$ is a vector of order $NM \times 1$. From (2.19), (3.5) and (3.6), we have

\[
\int_0^1 k_2(t, s) g_2(s, y(s)) ds = \int_0^1 B^T(t)K_2B(s)B^T(s)G_2 ds = B^T(t)V,
\]

where

\[
V = K_2WG_2.
\]

It should be noted that $V$ is a vector of order $NM \times 1$. Substituting (3.1)–(3.7) in (1.3) yields

\[
\sum_{j=0}^{q} f_j(t)y^{(j)}(t) = f(t) + \lambda_1 \int_0^t k_1(t, s) g_1(s, y(s)) ds + \lambda_2 B^T(t)V.
\]

We now collocate (1.3) at $NM$ points. For a suitable collocation points, we use Gaussian nodes $t_{nm}, n = 1, 2, \ldots, N$, $m = 0, 1, \ldots, M - 1$, which are introduced in (2.7). Therefore we have

\[
\sum_{j=0}^{q} f_j(t_{nm})y^{(j)}(t_{nm}) = f(t_{nm}) + \lambda_1 \int_0^{t_{nm}} k_1(t_{nm}, s) g_1(s, y(s)) ds + \lambda_2 B^T(t_{nm})V.
\]

Also, using (2.8) we get

\[
B^T(t_{nm})V = V_{(n-1)M+m},
\]
where \( V_{(n-1)M+m} \) denotes the \((n - 1)M + m)\text{th} element of vector \( V \). Consequently, with the aid of (2.18) and (3.10) we obtain

\[
\sum_{j=0}^{q} f_j(t_{nm})y^{(j)}(t_{nm}) = f(t_{nm}) + \lambda_2 V_{(n-1)M+m} \\
+ \lambda_1 \left( \sum_{i=1}^{n-1M-1} \sum_{j=0}^{M-1} k_1(t_{nm}, s_{ij}) g_1(s_{ij}, y_{ij}) h_{jm} \\
+ \sum_{j=0}^{M-1} k_1(t_{nm}, s_{nj}) g_1(s_{nj}, y_{nj}) e_{jm} \right),
\]

(3.12)

where \( y^{(j)}(t_{nm}) \) for \( n = 1, 2, \ldots, N, \ m = 0, 1, \ldots, M - 1 \) can be calculated from (3.4). Equation (3.12) gives \( NM \) nonlinear equations. Finally, we approximate the mixed conditions given in (1.4). From (1.4), (3.1), (3.3), and the definition of hybrid functions we obtain

\[
\sum_{j=0}^{q-1} \left( \alpha_{ij} \sum_{m=0}^{M-1} y^{(j)}(t_{1m}) b_{1m}(0) + \beta_{ij} \sum_{m=0}^{M-1} y^{(j)}(t_{Nm}) b_{Nm}(t_f) \right) = \mu_i, \quad i = 0, 1, \ldots, q - 1.
\]

(3.13)

In order that the approximate solution obtained by the present method be continuous, we impose the continuity condition at the points \((n/N)t_f, \ n = 1, 2, \ldots, N - 1\). Equations (3.12)-(3.13) together with the continuity conditions give \( N(M+1)+q-1 \) nonlinear equations that can be solved by using the well-known Tau method [23] for the unknown coefficients \( y_{nm}, \ n = 1, 2, \ldots, N, \ m = 0, 1, \ldots, M - 1 \).

4. **Illustrative Examples**

In this section, four examples are given to demonstrate the efficiency, the accuracy, and the applicability of the proposed method.

**Example 4.1.** Consider the nonlinear Volterra-Fredholm integrodifferential equation

\[
t^2 y''(t) + 2y'(t) = 2 - \frac{5}{6} t + \frac{1}{2} t e^{-t^2} + \int_0^t t s e^{-y^2(s)} ds + \int_0^1 t y^2(s) ds, \quad y(0) = 0, \quad y'(0) = 1
\]

(4.1)

whose exact solution is given in [9] as \( y(t) = t \). We applied the present method to solve this problem.

Let us define

\[
E = \left( \int_0^{t_f} (y_e(t) - y(t))^2 dt \right)^{1/2},
\]

(4.2)
Table 1: The computational results of $L^2$-norm error for Example 4.1.

<table>
<thead>
<tr>
<th>Present method</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 2, M = 8$</td>
<td>$9.28e-12$</td>
</tr>
<tr>
<td>$N = 2, M = 9$</td>
<td>$6.32e-12$</td>
</tr>
<tr>
<td>$N = 2, M = 10$</td>
<td>$1.85e-13$</td>
</tr>
<tr>
<td>$N = 4, M = 8$</td>
<td>$1.11e-13$</td>
</tr>
<tr>
<td>$N = 4, M = 9$</td>
<td>$6.26e-15$</td>
</tr>
<tr>
<td>$N = 4, M = 10$</td>
<td>$9.29e-17$</td>
</tr>
</tbody>
</table>

Table 2: The computational results of $L^2$-norm error for Example 4.2.

<table>
<thead>
<tr>
<th>Present method</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 2, M = 8$</td>
<td>$8.83e-12$</td>
</tr>
<tr>
<td>$N = 2, M = 9$</td>
<td>$3.37e-12$</td>
</tr>
<tr>
<td>$N = 2, M = 10$</td>
<td>$2.15e-13$</td>
</tr>
<tr>
<td>$N = 4, M = 8$</td>
<td>$7.00e-14$</td>
</tr>
<tr>
<td>$N = 4, M = 9$</td>
<td>$4.63e-15$</td>
</tr>
<tr>
<td>$N = 4, M = 10$</td>
<td>$9.70e-17$</td>
</tr>
</tbody>
</table>

where $y(t)$ and $y_e(t)$ denote the approximate solution obtained by the present method and the exact solution, respectively. In Table 1, the computational results of the $L^2$-norm error denoted by $E$ between the approximate solution and the exact solution for different values of $N$ and $M$ are given. The simulation results reported in Table 1, indicate that there is an excellent agreement between the approximate and exact solution.

Example 4.2. As the second example, consider the nonlinear Volterra integrodifferential equation

$$y'(t) = 2t - \frac{1}{2} \sin(t^4) + \int_0^t t^2 s \cos(t^2 y(s)) ds,$$

$$y(0) = 0,$$  (4.3)

whose exact solution is given in [19] as $y(t) = t^2$. We applied the present method to solve this problem. In Table 2, the computational results of the $L^2$-norm error between the approximate solution and the exact solution for different values of $N$ and $M$ are summarized. The simulation results obtained by the present method are superior to those reported in [19].

Example 4.3. Let us consider the nonlinear Volterra integrodifferential equation

$$y^{(4)}(t) = 1 + \int_0^t e^{-s} y^2(s) ds, \quad t \in [0, 1],$$

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

$$y(1) = e, \quad y'(1) = e,$$  (4.4)
whose exact solution is given in [24] as $y(t) = e^t$. We solved this problem by using hybrid functions. In Table 3, the computational results of $y(t)$ obtained by the present method with $N = 2$ and $M = 10$ together with the exact solution and the solution determined by the Tau method [24] are summarized. The computational results obtained by the present method are superior to that reported in [24]. Moreover in Table 4, the $L^2$-norm error between the approximate solution obtained by the present method and the exact solution for different values of $N$ and $M$ is given. It should be pointed out that only a very small numbers of hybrid basis functions are needed to obtain a quite satisfactory approximation to the exact solution.

**Example 4.4.** As the last example, consider the nonlinear Volterra-Fredholm integrodifferential equation

$$y'''(t) - ty''(t) = \frac{1}{3}t^3 - 2t + 1 - \int_0^t s \, y(s) \, ds + \int_0^1 \frac{2te^s}{y(s) - s} \, ds,$$

$$y(0) = 1, \quad y'(0) = 1, \quad y(1) = 1 + e, \quad (4.5)$$

which has the exact solution $y(t) = t + e^t$. To measure the accuracy of proposed approach, the $L^2$-norm error between the approximate solution obtained by the present method for different values of $N$ and $M$ and the exact solution is given in Table 5. The computational results reveal that the method is convergent.

### Table 3: The computational results of $y(t)$ for Example 4.3.

<table>
<thead>
<tr>
<th>$t$</th>
<th>Tau method $n = 10$</th>
<th>Present method $N = 2, M = 10$</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>0.2</td>
<td>1.2214027513</td>
<td>1.2214027582</td>
<td>1.2214027582</td>
</tr>
<tr>
<td>0.4</td>
<td>1.4918246782</td>
<td>1.4918246976</td>
<td>1.4918246976</td>
</tr>
<tr>
<td>0.6</td>
<td>1.8221187743</td>
<td>1.8221188004</td>
<td>1.8221188004</td>
</tr>
<tr>
<td>0.8</td>
<td>2.2255409119</td>
<td>2.2255409285</td>
<td>2.2255409285</td>
</tr>
<tr>
<td>1.0</td>
<td>2.7182818285</td>
<td>2.7182818284</td>
<td>2.7182818284</td>
</tr>
</tbody>
</table>

### Table 4: The computational results of $L^2$-norm error for Example 4.3.

<table>
<thead>
<tr>
<th>Present method</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 2, M = 8$</td>
<td>$4.64 \times 10^{-8}$</td>
</tr>
<tr>
<td>$N = 2, M = 9$</td>
<td>$2.94 \times 10^{-10}$</td>
</tr>
<tr>
<td>$N = 2, M = 10$</td>
<td>$3.68 \times 10^{-11}$</td>
</tr>
<tr>
<td>$N = 4, M = 8$</td>
<td>$3.01 \times 10^{-9}$</td>
</tr>
<tr>
<td>$N = 4, M = 9$</td>
<td>$4.52 \times 10^{-12}$</td>
</tr>
<tr>
<td>$N = 4, M = 10$</td>
<td>$5.96 \times 10^{-13}$</td>
</tr>
</tbody>
</table>

### Table 5: The computational results of $L^2$-norm error for Example 4.4.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>$4.64 \times 10^{-8}$</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>$2.94 \times 10^{-10}$</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>$3.68 \times 10^{-11}$</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>$3.01 \times 10^{-9}$</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>$4.52 \times 10^{-12}$</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>$5.96 \times 10^{-13}$</td>
</tr>
</tbody>
</table>

5. **Conclusion**

In the present work, a hybrid approximation method was developed for solving the high-order nonlinear Volterra-Fredholm integrodifferential equations. The nice properties of
hybrid functions together with the associated operational matrices of integration, derivative, and cross-product are used to reduce the solution of problem to the solution of nonlinear algebraic equations whose solution is much more easier than the original one. The matrices $P$, $C$, and $D^*$ are sparse, hence making the method computationally attractive without sacrificing the accuracy of the solution. Moreover, It was shown that small values for $N$ and $M$ are needed to achieve high accuracy and a satisfactory convergence. The numerical results support this claim. The method is general, easy to implement and yields the desired accuracy in a few terms of hybrid basis functions. The simulation results indicate the convergence and the effectiveness of the proposed approach.

**Acknowledgments**

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

**References**


