Research Article
On the Multiwavelets Galerkin Solution of the Volterra–Fredholm Integral Equations by an Efficient Algorithm

H. Bin Jebreen

Department of Mathematics, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia

Correspondence should be addressed to H. Bin Jebreen; hjebreen@ksu.edu.sa

Received 18 July 2020; Revised 19 October 2020; Accepted 24 October 2020; Published 11 November 2020

Academic Editor: Shiv Kaushik

Copyright © 2020 H. Bin Jebreen. 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.

We develop the multiwavelet Galerkin method to solve the Volterra–Fredholm integral equations. To this end, we represent the Volterra and Fredholm operators in multiwavelet bases. Then, we reduce the problem to a linear or nonlinear system of algebraic equations. The interesting results arise in the linear type where thresholding is employed to decrease the nonzero entries of the coefficient matrix, and thus, this leads to reduction in computational efforts. The convergence analysis is investigated, and numerical experiments guarantee it. To show the applicability of the method, we compare it with other methods and it can be shown that our results are better than others.

1. Introduction
A mathematical model of the spatiotemporal development of an epidemic yields the following Volterra–Fredholm integral equation (VFIE):

\[ u(x) = f(x) + \int_0^x k_1(x, s, u(s))ds + \int_0^1 k_2(x, s, u(s))ds, \quad x \in \Omega = [0, 1], \]  

(1)

where the given functions \( f: \Omega \to \mathbb{R} \) and \( k_1: S \times \mathbb{R} \to \mathbb{R} \) are assumed to be continuous functions. Furthermore, we consider \( k_2 = p(x, s)h(u(s)) \) to be integrable, where \( h(x) \) is a nonlinear function and \( p(x, s): S \to \mathbb{R} \) is a continuous function. Besides, the given functions are selected so that equation (1) has a unique solution.

The parabolic boundary value problems lead to these types of integral equations and widely arise from various physical and biological models. The VFIE also appears in the literature in mixed form as

\[ u(x) = f(x) + \int_0^x \int_0^1 k(t, s)u(s)ds dt, \]  

(2)

where \( f \) and \( k \) are analytic functions. Many authors studied the mixed form of VFIE numerically. Among these, we can mention collocation method [1], projection method [2], spline collocation method [3], wavelet collocation method [4], Adomian decomposition method [5], and so on [6–8]. Among these studies, we focus on a paper that uses the multiwavelet Galerkin method to solve linear mixed VFIE as mentioned in [9]. In [9], the wavelet transform matrix and the operational matrix of integration are utilized to reduce the problem of linear mixed VFIE to a sparse linear system of algebraic equations. By searching among the sources, we can find a small number of papers in the field of numerical and analytical solutions to problem (1). In [10], the Lagrange collocation method is employed to solve this problem. Wang and Wang [11] applied the Taylor collocation method to find the numerical solution of the equation. Also, the convergence analysis is investigated for the proposed method. The Taylor collocation method was applied by Karamete and Sezer [12] to solve this equation as well as the high-order linear Fredholm–Volterra integrodifferential equations [13]. The Bell polynomials have been employed for solving this equation [14].

The motivation of our work is to develop the multiwavelet Galerkin method to solve (1). We split the problem into two configurations, linear and nonlinear. After using
the multiwavelet Galerkin method, both types reduce to the system of the linear and nonlinear algebraic equations, respectively. The interesting results arise in the linear type where thresholding is employed to decrease the nonzero entries of the coefficient matrix. This gives the sparse system. This property is very useful to reduce the computational cost. We use Alpert’s multiwavelet bases constructed in [15] following [16], and these bases have been used to solved PDEs, ODEs, and integral equations [9, 17–19]. These bases allow us to have high vanishing moments, compact support, and properties such as orthogonality and interpolation [15]. These characteristics of Alpert’s multiwavelets lead to a sparse representation of differential and integral operators [16, 20].

This paper is organized as follows: In Section 2, we briefly introduce Alpert’s multiwavelet bases. In Section 3, the Multiwavelets Galerkin method is used to solve this probble, and the conditions for convergence of the proposed method are discussed. Section 5 contains some numerical results to confirm the validity and efficiency of the proposed method, and Section 6 contains a few concluding remarks.

2. Alpert’s Multiwavelet Bases

Let $J \in \mathbb{Z}^+ \cup \{0\}$. We consider the uniform finite discretizations $\Omega = [0, 1] = \bigcup_{b \in \mathcal{B}} X_{J,b}$ where the subintervals $X_{J,b} = [x_b, x_{b+1}]$ are determined by the point $x_b = (b/2^J)$ with $\mathcal{B} = \{0, 1, \ldots, 2^J - 1\}$. For $k \in \mathcal{B} = \{0, 1, \ldots, 2^J - 1\}$, we introduce the subspace $V_J$ as a piecewise polynomial bases of degree less than multiplicity parameter $r$ that is spanned by

$$V_J = \text{span}\{\psi_{J,b}^k = \mathcal{D}_b \mathcal{T}_b \phi^k, b \in \mathcal{B}, k \in \mathcal{R}\} \subset L^2(\Omega), \quad r \geq 0,$$

where $\mathcal{D}$ and $\mathcal{T}$ are the dilation and translation operators, respectively, and $\{\psi^k\}_{k \in \mathcal{R}}$ are the primal interpolating scaling bases introduced by Alpert et al. [16]. Given nodes $\{t_k\}_{k \in \mathcal{R}}$ which are the roots of Legendre polynomial of degree $r$, the interpolating scaling bases are defined as

$$\phi^k(t) = \begin{cases} \frac{2}{\omega_k} L_k(2t - 1), & t \in [0, 1], \\ 0, & \text{otherwise}, \end{cases}$$

where $\{L_k(t)\}_{k \in \mathcal{R}}$ are the Lagrange interpolating polynomials at the point $\{t_k\}_{k \in \mathcal{R}}$ and $\{\omega_k\}_{k \in \mathcal{R}}$ are the Gauss–Legendre quadrature weights [16, 18]. These bases form an orthonormal bases on $\Omega$ with respect to the $L^2$-inner product. Due to the definition of the space $V_J$, the spaces $\{V_J\}_{j \in \mathbb{Z}^+ \cup \{0\}}$ have dimension $N_j = 2^jr$ and obviously are nested:

$$V_0 \subset V_1 \subset \cdots \subset V_J \subset \cdots \subset L^2(\Omega).$$

Hence, we consider the complement subspace $W_J$ of $V_J$ in $V_{J+1}$ such that

$$V_{J+1} = V_J \oplus W_J, \quad W_J \perp V_J,$$

where $\oplus$ denotes orthogonal sums. According to (6), the space $V_J$ may be inductively decomposed to

$$V_J = V_0 \oplus \bigoplus_{j=0}^{J-1} W_j.$$

The complement subspace $W_J$ has dimension $2^J r$ and is spanned by multiwavelet bases $\{\psi^k\}_{k \in \mathcal{R}}$, as

$$W_J = \text{span}\{\psi_{J,b}^k = \mathcal{D}_b \mathcal{T}_b \phi^k, \quad b \in \mathcal{B}, k \in \mathcal{R}\}.$$

Because Alpert’s multiwavelets are completely introduced in [16], we avoid this and refer the readers to [15, 16, 19].

Every function $p \in L^2(\Omega)$ can be represented in the form

$$p \approx p_J = \sum_{b \in \mathcal{B}, k \in \mathcal{R}} c_{J,b}^k \psi_{J,b}^k,$$

where $\mathcal{P}_J$ is the orthogonal projection that maps $L^2(\Omega)$ onto the subspace $V_J$. To find the coefficients $c_{J,b}^k$ that are determined by $\langle p, \psi_{J,b}^k \rangle = \int_{X_{J,b}} f(x) \psi_{J,b}^k(x) \, dx$, we shall compute the integrals. We apply the $r$-point Gauss–Legendre quadrature by a suitable choice of the weights $\omega_k$ and nodes $t_k$ for $k \in \mathcal{R}$ to avoid these integrals [16, 19], via

$$p_{J,b}^k \approx 2^{-\frac{J}{2}} \sqrt{\frac{2}{\pi}} \left( 2^{-\frac{1}{2}} \left( 1 + \frac{1}{2} \right) \right) \sup_{x \in [0, 1]} |p^{(r)}(x)|.$$

Convergence analysis of the projection $P_J p$ is investigated for the $r$-times continuously on differentiable function $p \in C^r(\Omega)$:

$$\|P_J p - p\|_2 \leq 2^{-\frac{J}{2}} \sqrt{\frac{2}{\pi}} \left( 2^{-\frac{1}{2}} \left( 1 + \frac{1}{2} \right) \right) \sup_{x \in [0, 1]} |p^{(r)}(x)|.$$

For the full proof of this approximation and further details, we refer the readers to [15]. Thus, we can conclude that $P_J p$ converges to $p$ with rate of convergence $O(2^{-Jr})$.

Assume that the vector function $\Phi_J = [\Phi_{J,b}, \Phi_{J,b+1}, \ldots, \Phi_{J,2^J-2J-1}]^T$ with $\Phi_{J,b} = [\psi_{J,b}^0, \psi_{J,b}^1, \ldots, \psi_{J,b}^{2^J-1}]$ includes the scaling functions and is called multiscaling function. Approximation (9) may be rewritten using the vector $P$ that includes entries $P_{b,b+1} = p_{J,b}^k$ as follows:

$$\Phi_J p = P^T \Phi_J,$$

where $P$ is an $N$-dimensional vector. The building blocks of these bases construction can be applied to approximate a higher-dimensional function. To this end, one can introduce the two-dimensional subspace $V_J^{2^J} = V_J \times V_J \subset L^2(\Omega)^2$ that is spanned by

$$\left\{\phi_{J,b,b'}^k \psi_{J,b}^k \psi_{J,b'}^k, \quad b, b' \in \mathcal{B}, k, k' \in \mathcal{R}\right\}.$$

Thus, by this assumption, to derive an approximation of the function $p \in L^2(\Omega)^2$ by the projection operator $\mathcal{P}_J$, we have

$$p(x, y) \approx \Phi_J^T (x) P \Phi_J^T (y),$$
where the components of the square matrix $P$ of order $N$ are obtained by
\[ P_{r,b}^k, (k+1)r, b'_{(k+1)} \approx 2^{-l} \frac{|a_k|}{2} \sqrt{\frac{\mu_k}{2}} p \left( 2^{-l} \left( \tau_k + b, 2^{-l} \left( \tau_k' + b' \right) \right) \right), \]

where $\tau_k = (\tau_k + 1)/2$. Consider the $2r$-th partial derivatives of $p$: $\Omega^2 \rightarrow \mathbb{R}$ to be continuous. Utilizing this assumption, the error of this approximation can be bounded as follows:
\[ \| P_P - P \| \leq M_{\max} 2^{-l} \left( 2 + 2^{-1} 4^{r} \right), \]

where $M_{\max}$ is a constant.

Given orthogonal projection operator $Q_j^t = P_{r+1}^t - P_j^t$ that maps $L^2(\Omega)$ onto $W_j^t$, the multiscale projection operator $M_j^t$ can be represented as
\[ M_j^t = Q_0^0 + \sum_{j=0}^{l-1} Q_j^t, \]

and consequently, any function $p \in L^2(\Omega)$ can be approximated as a linear combination of multiwavelets and single-scale interpolating scaling functions as
\[ p \approx M_j^t(p) = \sum_{k=0}^{r-1} \sum_{j=0}^{l-1} \sum_{k=0}^{r-1} \sum_{j=0}^{l-1} \sum_{k=0}^{r-1} \sum_{j=0}^{l-1} p_{j,b}^k \Psi_{j,b}^k, \]

where
\[ p_{j,b}^k \equiv \langle p, \phi_{j,b}^k \rangle, \]
\[ \bar{p}_{j,b}^k \equiv \langle p, \psi_{j,b}^k \rangle. \]

Note that we can compute the coefficients $p_{j,b}^k$ by using (10). But in many cases, multiwavelet coefficients from zero up to higher-level $J - 1$ must be evaluated numerically. To avoid this, we use multiwavelet transform matrix $T_j$, introduced in [18, 19]. This matrix connects multiwavelet bases and multiscaling functions, via
\[ \Psi_j^t = T_j \Phi_j^t, \]

where $\Psi_j^t = [\Phi_j^0, \Phi_j^1, \Psi_{r,0,0}, \Psi_{r,1,0}, \ldots, \Psi_{r,1,b}, \ldots]$ is a vector with the same dimension $\Phi_j^t$ (here $\Psi_{r,j,b}^t = [\psi_{r,j,b}^0, \ldots, \psi_{r,j,b}^r]$). This representation helps to rewrite equation (18) as to form
\[ p \approx M_j^t(p) = T_j^T \Psi_j^t, \]

where we have the $N$-dimensional vector $\bar{P}_j$ whose entries are $p_{j,b}^k$ and $\bar{P}_j$ is and is given by employing the multiwavelet transform matrix $T_j$ as $\bar{P}_j = T_j \bar{P}_j$.

The multiwavelet coefficients (details) become small when the underlying function is smooth (locally) with increasing refinement levels. If the multiwavelet bases have $N_\psi$ vanishing moments [19, 21], then details decay at the rate of $2^{-tN_\psi}$ [17]. Because vanishing moments of Alpert’s multiwavelets are equal to $r$, consequently, one can obtain $p_{j,b}^k \approx O(2^{-t})$. This allows us to truncate the full wavelet transforms while preserving most of the necessary data. Thus, we can set to zero all details that satisfy a certain constraint $\epsilon$ using thresholding operator $\mathcal{C}_\epsilon$:
\[ \mathcal{C}_\epsilon(\bar{P}_j) = \bar{P}_j, \]

and the elements of $\bar{P}_j$ are determined by
\[ \bar{p}_{j,b}^k = \begin{cases} \bar{p}_{j,b}^k & (j,b,k) \in D_\epsilon, \\ 0 & \text{else}, \end{cases} \]

where $D_\epsilon = \{ (j,b,k) : |p_{j,b}^k| > \epsilon \}$. Now, we can bound the approximation error after thresholding [17] via
\[ \| P_P - P_P^{r,D} \|_{L^2(\Omega)} \leq C_{\text{thr}} \epsilon, \]

3. Multiwavelet Galerkin Method

In order to obtain multiwavelet Galerkin solution of (1), assume that solution can be approximated as an expansion of the Alpert’s multiwavelets, i.e.,
\[ u(x) \approx P_j^t(u)(x) = U_j^T \Psi_j^t(x), \]

where the $N$ dimension vector $U$ of unknowns must be specified. This solution is selected such that it satisfies (1) approximately. Also, it is obtained from the solution of the minimization problem
\[ \| u - P_j^t(u) \| = \min_{z \in L^2(\Omega)} \| u - z \|. \]

Since $L^2(\Omega)$ is an inner product space with finite dimension, it can be shown that this minimization problem has a unique solution [22].

Let us rewrite (1) in the operator form
\[ (I - \mathcal{F})u = g, \]

where $g(x) = f(x) + V(u)(x)$ with $V(u)(x) = \int_0^{k_1} k_1 (x, s) u(s) ds$. Furthermore, the operator $\mathcal{F}(u) = \int_0^{k_1} P(x, s) h(u(s)) ds$ is assumed to be compact on $L^2(\Omega)$ to $L^2(\Omega)$ and $k_1$ is a given continuous function. Due to these assumptions, $V(u)(x)$ is a continuous function and, consequently, $g$ is also continuous function. Due to (11), the function $g(x)$ can be approximated at a rate of at least $2^{-b}$:
\[ g(x) \approx P_j^t(g)(x) = F_j^T \Phi_j^t(x) + K_j^T \Phi_j^t(x) \]
\[ = F_j^T \Phi_j^t(x) + K_j^T \Phi_j^t(x) \]
\[ = F_j^T \Psi_j^t(x) + \mathcal{C}_\epsilon(\Phi_j^t(x)), \]

where $\mathcal{F}_1$ and $\mathcal{C}_1$ are $N \times N$ matrices and the rest are $N$-dimensional vectors. Now, we introduce the residual in the approximation of (1):
where \( u_j = \mathcal{M}_j'(u) \) and
\[
\mathcal{M}_j'(\mathcal{F}_j'(u_j)) = \mathcal{M}_j' \left( \int_0^1 \mathcal{M}_j'(k_2(x,s,u_j(s))) ds \right)
= \mathcal{M}_j' \left( \int_0^1 \mathcal{M}_j'(p(x,s)h(u_j(s))) ds \right)
= \int_0^1 H^T \Psi_j'(s)\Psi_j'(s) P\Psi_j'(x) ds = H^T \Psi_j'(x)
= \{ \Psi_2^j(x), \text{ linear,} \}
\]
\[
\{ \Psi_2^j(x), \text{ nonlinear.} \}
\]

Symbolically,
\[
r_j = \mathcal{M}_j'((1 - \mathcal{F}_j')u_j - g).
\]

To find \( u_j \), it requires that the approximate solution \( u_j \) satisfies
\[
\langle r_j, [\Psi_j'] \rangle = 0, \quad i = 1, \ldots, N.
\]

This is a multivariate Galerkin’s method and yields a linear or nonlinear system that must be solved to obtain the approximate solution. Note that \( \mathcal{M}_j'(z) = 0 \) if and only if \( \langle z, [\Psi_j'] \rangle = 0 \). Thus, we can rewrite (31) as
\[
\mathcal{M}_j'(r_j) = 0,
\]
or equivalently,
\[
\mathcal{M}_j'(1 - \mathcal{F}_j')u_j = \mathcal{M}_j'(g).
\]

Note that \( \mathcal{M}_j'(u_j) = u_j \) whenever \( u_j \in V_j \). Due to this, equation (34) can be rewritten as
\[
(1 - \mathcal{M}_j'\mathcal{F}_j')(u_j) = \mathcal{M}_j'(g).
\]

According to (35), we obtain the system of linear or nonlinear equations. Due to the higher vanishing moments \( r \) and increasing refinement level \( J \), for the linear type of this equation, we can discard coefficients by high thresholding introduced in the previous section. We can reduce the computational efforts using proper methods for this type of system such as the GMRES method. The GMRES method is introduced by Saad and Schultz [23] for solving sparse and large linear systems. The GMRES generates an approximate solution whose residual norm is minimum by using a Krylov subspace. In this paper, we use restarted GMRES Algorithm 2 [24]. To use this method, we must first define Arnoldi’s algorithm. Arnoldi’s procedure is an algorithm for building an orthogonal basis of the Krylov subspace \( k_m \). The \( N \)-th Krylov subspace is defined as follows:
\[
k_m(A, w) = \text{span}\{ w_1, \Lambda w_1, \ldots, \Lambda^{m-1} w_1 \}.
\]

Here, we assume that system (35) of the linear type to be of form \( AU = D \) and \( W_m \) is an \( N \times m \) matrix with column vectors \( w_1, \ldots, w_m \). Also, \( \mathcal{H}_m \) is a \( (m+1 \times m) \) Hessenberg matrix whose nonzero entries \( h_{i,j} \) are defined by Algorithm 1.

1. Compute \( r_0 = D - \Lambda U \), \( \beta = \| r_0 \|_2 \) and \( w_1 = r_0 / \beta \)
2. Generate the Arnoldi basis and the matrix \( \mathcal{H}_m \) using the Arnoldi algorithm starting with \( w_1 \).
3. Compute \( y_m \) the minimizer of \( \| \beta e_1 - H_m y \|_2 \) and \( U_m = U_0 + W_m y_m \).
4. If satisfied then stop, else set \( U_0 = U_m \) and go to 1.

To investigate the convergence analysis, one can prove that \( \| F_j' - \mathcal{M}_j'F_j' \| = O(2^{-J/2}) \). Thus, \( \| F_j' - \mathcal{M}_j'F_j' \| \rightarrow 0 \) when \( J \rightarrow \infty \) because \( F_j \) is an compact operator. Now, we can raise the convergence theorem.

**Theorem 1**. Let \( F \) be a compact operator and \( I - F \) be injective. Assume that the sequence \( \mathcal{F}_j' : L^2(\Omega) \rightarrow L^2(\Omega) \) is collectively compact and pointwise convergent to \( F \).

Then, \( (I - \mathcal{F}_j')^{-1} \) exists and is uniformly bounded. Also, the solution of (27) and (31) satisfy the error estimate
\[
\| u - u_j \| \leq \frac{\| (I - \mathcal{F}_j')^{-1} \|}{1 - \epsilon_j} \| (I - \mathcal{F}_j')^{-1} \| \| u - u_j \| + \| (\mathcal{M}_j'F_j' - \mathcal{M}_j'F_j') u_j \|.
\]

Proof. Because the sequence \( \mathcal{F}_j \) converges pointwise to \( F \) in \( L^2(\Omega) \) and is collectively compact, we conclude that
\[
\| (\mathcal{F}_j - F) \mathcal{F}_j \| \rightarrow 0, \quad \text{as} \quad J \rightarrow \infty.
\]

For all sufficiently large \( J \), we have
\[
\| (I - F)^{-1}(\mathcal{F}_j - F) \mathcal{F}_j \| < 1,
\]
and as a consequence of this, \( I - \mathcal{F}_j' \) is reversible. Note that the inverse operator \( (I - F) \) exists due to the Riesz theorem. The investigation is based on the approximation of \( I - \mathcal{F}_j' \) by \( I - \mathcal{M}_j'\mathcal{F}_j' \).
\[
I - \mathcal{M}_j'\mathcal{F}_j' = (I - \mathcal{F}_j') + (\mathcal{F}_j' - \mathcal{M}_j'\mathcal{F}_j')
= (I - \mathcal{F}_j') \left( I + (I - \mathcal{F}_j')^{-1}(\mathcal{F}_j' - \mathcal{M}_j'\mathcal{F}_j') \right).
\]

To prove the existence of \( (I - \mathcal{M}_j'\mathcal{F}_j')^{-1} \), assume that
\[
\epsilon_j = \sup_{J \geq 1} \| \mathcal{F}_j' - \mathcal{M}_j'\mathcal{F}_j' \| < \frac{1}{\| (I - \mathcal{F}_j')^{-1} \|}.
\]

Thus, \( (I + (I - \mathcal{F}_j')^{-1}(\mathcal{F}_j' - \mathcal{M}_j'\mathcal{F}_j'))^{-1} \) exists and is uniformly bounded due to geometric series theorem, i.e.,
\[
\| (I + (I - \mathcal{F}_j')^{-1}(\mathcal{F}_j' - \mathcal{M}_j'\mathcal{F}_j'))^{-1} \| \leq \frac{1}{1 - \epsilon_j} \| (I - \mathcal{F}_j')^{-1} \|.
\]

Using (42), \( (I - \mathcal{M}_j'\mathcal{F}_j') \) exists. Taking norm and using (42), we obtain
\[ \left\| (1 - \mathcal{M}_J F)^{-1} \right\| \leq \frac{\left\| (1 - F)^{-1} \right\|}{1 - \epsilon_j \left\| (1 - F)^{-1} \right\|}. \quad (43) \]

Applying the operator \( \mathcal{M}_J \) to both sides of (27) and then rearranging, we can obtain
\[ (1 - \mathcal{M}_J F)u = \mathcal{M}_J g + (u - \mathcal{M}_J u) + (\mathcal{M}_J F - \mathcal{M}_J)u. \quad (44) \]

Subtracting (35) from (44), we obtain
\[ (1 - \mathcal{M}_J F)(u - u_J) = (u - \mathcal{M}_J u) + (\mathcal{M}_J F - \mathcal{M}_J)u. \quad (45) \]

Taking norm and employing (43),
\[ \left\| u - u_J \right\| \leq \frac{\left\| (I - F)^{-1} \right\|}{1 - \epsilon_j \left\| (1 - F)^{-1} \right\|} \left( \left\| u - \mathcal{M}_J u \right\| + \left\| \mathcal{M}_J F - \mathcal{M}_J \right\| \right). \quad (46) \]

This is equivalent to (37). It is straightforward to show that \( \left\| \mathcal{M}_J F - \mathcal{M}_J \mathcal{F} \right\| \to 0 \) as \( J \to \infty \). Assume that \( \{u_j\} \) be a sequence of continuous functions so that \( u_j \to u \) as \( J \to \infty \). Since the orthonormal projection \( \mathcal{M}_J \) satisfies \( \left\| \mathcal{M}_J \right\| = 1 \), we can obtain
\[ \left\| u - \mathcal{M}_J u \right\| \leq \left\| u - u_j \right\| + \left\| u_j - \mathcal{M}_J u_j \right\| + \left\| \mathcal{M}_J (u - u_j) \right\| \leq 2 \left\| u - u_j \right\| + \left\| u_j - \mathcal{M}_J u_j \right\|. \quad (47) \]

Thus, for each real number \( \epsilon > 0 \), there exists a number \( J_0 \) such that, for every number \( J \geq J_0 \), one can write \( \left\| u - u_j \right\| \leq (\epsilon/4) \). This then implies that
\[ \left\| u - \mathcal{M}_J u \right\| \leq \frac{\epsilon}{2} + \left\| u_j - \mathcal{M}_J u_j \right\|. \quad (48) \]

This implies that \( \left\| u - \mathcal{M}_J u \right\| \leq \epsilon \), for sufficiently large value of \( J \) and because \( \epsilon \) is arbitrary, consequently, \( \mathcal{M}_J u \to u \) as \( J \to \infty \).

4. Numerical Experiments

To verify the accuracy and efficiency of the proposed method, we consider a series of numerical examples. In the linear type of equation (1), we aim to generate a sparse matrix to reduce the computational costs. We illustrate the rate of sparsity \( S_e \) which is defined by [21,25]
\[ S_e = \frac{N_e - N_x}{N_0} \times 100\%, \quad (49) \]

where \( \epsilon \) is the threshold (small positive number) and \( N_e \) and \( N_0 \) are the number of elements remaining after thresholding and the total number of elements, respectively.

Example 1. Consider the linear VFIE given in [11] as
\[ u(x) = e^{-x} - e^{x}(x - 1) + \int_{0}^{x} e^{x}u(s)ds - \int_{0}^{1} e^{x}u(s)ds. \quad (50) \]

The exact solution is \( e^{-x} \) [11].

The effects of the multiplicity parameter \( r \), the refinement level \( J \), and thresholding with different thresholds are reported in Table 1 and Figure 1. The results confirm the theoretical claims and demonstrate the effectiveness of the method. Note that the \( L^2 \) error decreases as parameters \( r \) and \( J \) increase due to the rate of convergence \( O(2^{-n}) \).

We compare the error of the proposed method, the Lagrange collocation method [10], Taylor collocation method [11], and Taylor polynomial method [26] in Table 2. Due to Table 2, our method is flexible than other methods, and without changing the multiplicity parameter \( r \), we can improve the results. Figure 2 illustrate the effect of thresholding with different threshold parameters \( e \) on the coefficient matrix. It can be seen that the number of matrix elements decreases when the threshold parameter increases.

Example 2. Let us consider the following VFIE:
\[ u(x) = \frac{e^{2x} - 1}{2} \cos(x) + e^{x} + \frac{1 - e^{2}}{2} \sin(x) \]
\[ + \int_{0}^{x} e^{r} \cos(x)u(s)ds - \int_{0}^{1} e^{r} \sin(x)u(s)ds. \quad (51) \]

In Table 3 and Figure 3, we show the effect of the parameters \( r \), \( J \), and \( e \) on sparsity and \( L^2 \)-error. It is obvious that increasing the parameters \( r \) and \( J \) reduces the error. A comparison of the proposed method with other methods such as Taylor method [26] and the Lagrange collocation method [10] is reported in Table 4. In Figure 4, we illustrate the effect of thresholding on the coefficient matrix by taking different thresholds \( e \) when \( r = 8 \) and \( J = 2 \).

Example 3. Let us consider nonlinear VFIE (1) with
\[ u(x) = \sin(x) + \left(1 - e^{-\sin(1)}\right)x^{3} - \frac{x^{2} - \sin^{2}(x)}{4} \]
\[ - \int_{0}^{x} (x - s)u^{2}(s)ds - \int_{0}^{1} x^{3} \cos(s)u^{(3)}(s)ds. \quad (52) \]

The exact solution of this equation is \( u(x) = \sin(x) \).

In Figure 5, we illustrate the effect of parameter \( r \) and \( J \) on the \( L^2 \)-error and the error of approximation is plotted in Figure 6 taking \( r = 7 \) and \( J = 2 \).

Example 4. Consider the nonlinear VFIE as
\[ u(x) = f(x) + \int_{0}^{x} \cos(x + s)u^{2}(s)ds + \int_{0}^{1} \sin(s - x)u^{2}(s)ds, \quad (53) \]

with
\[ f(x) = \frac{(-2 \cos(2x) - \sin(2x))e^{2x}}{5} + \frac{2e^{2} \sin(-1 + x)}{5} \]
\[ + \frac{e^{2} \cos(-1 + x) + \cos(x)}{5} + \frac{e^{x} - \sin(x)}{5}. \quad (54) \]
(1) Choose a vector $w_1$, such that $\|w_1\|_2 = 1$
(2) For $j = 1, 2, \ldots, m$ do
(3) Compute $h_{ij} = (Aw_i, w_i)$ for $i = 1, 2, \ldots, j$
(4) $w_j = (Aw_j, w_i)$ for $i = 1, 2, \ldots, j$
(5) $h_{j+1,j} = \|w_j\|_2$
(6) If $h_{j+1,j} = 0$ then stop
(7) $w_{j+1} = (w_j/h_{j+1,j})$
(8) End do

ALGORITHM 1: Arnoldi’s algorithm.

Figure 1: Effects of the refinement level $J$ and multiplicity parameter $r$ for Example 1.

(1) Compute $r_0 = D - \Lambda U_0$, $\beta = \|r_0\|_2$ and $w_1 = r_0/\beta$
(2) Generate the Arnoldi basis and the matrix $H_m$ using the Arnoldi algorithm starting with $w_1$
(3) Compute $y_m$ the minimizer of $\|\beta e_1 - H_my\|_2$ and $U_m = U_0 + W_my_m$
(4) If satisfied then stop, else set $U_0 = U_m$ and go to 1

ALGORITHM 2: Restarted (GMRES).

Figure 2: The effect of thresholding with threshold $\epsilon = 10^{-5}$ (left), $\epsilon = 10^{-3}$ (middle), and $\epsilon = 10^{-1}$ (right) on the coefficient matrix for Example 1.
Table 1: Effects of parameters $r$, $J$, and $\varepsilon$ on sparsity and $L_2$-error for Example 1.

| $r$ | $J$ | Without thresholding | $\varepsilon = 10^{-5}$ | $\varepsilon = 10^{-3}$ | $\varepsilon = 10^{-1}$ |
|-----|-----|----------------------|-------------------------|-------------------------|-------------------------|
| 5   | 2   | 6.39e-9              | 24.8                    | 54                      | 91                      |
|     | 3   | 2.00e-10             | 59.8                    | 78.8                    | 96.5                    |
| 7   | 2   | 5.97e-13             | 43                      | 65.9                    | 97.5                    |
|     | 3   | 4.71e-15             | 72.4                    | 85.4                    | 93.8                    |

Table 2: Comparison of the error for Example 1.

| $r$ | Presented method | Taylor collocation | Taylor polynomial method | Lagrange collocation |
|-----|------------------|--------------------|--------------------------|----------------------|
| 2   | 1.55e-3          | 7.87e-2            | 3.41e-2                  | 7.87e-2              |
| 5   | 6.39e-9          | 6.23e-5            | 3.68e-4                  | 6.23e-5              |
| 8   | 4.63e-15         | 1.89e-8            | 1.24e-5                  | 1.77e-7              |
| 9   | 4.01e-16         | 2.35e-8            | 3.46e-7                  | 7.21e-6              |

Figure 3: Effects of the refinement level $J$ and multiplicity parameter $r$ for Example 2.

Figure 4: The effect of thresholding with threshold $\varepsilon = 10^{-5}$ (left), $\varepsilon = 10^{-4}$ (middle), and $\varepsilon = 10^{-3}$ (right) on the coefficient matrix for Example 2.
Table 3: Effects of parameters $r$, $J$, and $\varepsilon$ on sparsity and $L_2$-error for Example 2.

| $r$ | $J$ | Without thresholding | $\varepsilon = 10^{-6}$ | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-2}$ |
|-----|-----|----------------------|----------------------|----------------------|----------------------|
|     |     | $S_s$ | $L_2$-error | $S_s$ | $L_2$-error | $S_s$ | $L_2$-error | $S_s$ | $L_2$-error |
| 5   | 2   | 0       | 1.73e-8   | 22.2   | 1.00e-6   | 51.5   | 1.74e-4   | 79.2   | 5.15e-3   |
|     | 3   | 0       | 5.43e-10  | 58.1   | 1.18e-6   | 76.2   | 1.74e-4   | 91.9   | 5.15e-3   |
| 6   | 2   | 0       | 1.81e-10  | 32.5   | 5.24e-10  | 59.5   | 7.81e-5   | 82.1   | 1.37e-2   |
|     | 3   | 0       | 2.84e-12  | 65.3   | 1.02e-6   | 81.7   | 7.80e-5   | 93.7   | 1.37e-2   |
| 7   | 2   | 0       | 1.62e-12  | 39.4   | 6.78e-12  | 65.1   | 9.95e-5   | 84.8   | 1.84e-2   |
|     | 3   | 0       | 1.27e-14  | 71.3   | 2.74e-7   | 84.7   | 9.95e-5   | 95.1   | 1.84e-2   |
The exact solution is $u(x) = e^x$.

Table 5 is reported to show the efficiency and accuracy of the proposed method. We observe when the refinement level $J$ and multiplicity parameter $r$ increase, the $L^2$-errors decrease. Figure 7 shows the error of proposed method on taking $r = 6$ and $J = 2$.

5. Conclusion

We have employed the multiwavelet Galerkin method to solve the Volterra–Fredholm integral equations. To this end, the Volterra and Fredholm operators are represented in multiwavelet bases. Applying this method leads to a linear or nonlinear system of algebraic equations. In the linear type, we obtain a new sparse system using thresholding due to the decay in the wavelet coefficients. The convergence analysis is investigated, and one can show that the rate of convergence is $O(2^{-3r})$. The numerical examples illustrate the efficiency and accuracy of the method.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time due to legal or ethical reasons.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This project was supported by researchers supporting project number RSP-2020/210, King Saud University, Riyadh, Saudi Arabia.

References

[1] B. G. Pachpatte, "On mixed volterra–fredholm type integral equations," Indian Journal of Pure and Applied Mathematics, vol. 17, pp. 488–496, 1986.
[2] L. Hacia, “On approximate solution for integral equations of mixed type,” ZAMM-Journal of Applied Mathematics and Mechanics, vol. 76, pp. 415–416, 1996.

[3] H. Brunner, “On the numerical solution of nonlinear Volterra-Fredholm integral equations by collocation methods,” SIAM Journal on Numerical Analysis, vol. 27, no. 4, pp. 987–1000, 1990.

[4] E. Banafatem, R. A. Razzaghi, and S. Yousefi, “Two-dimensional Legendre wavelets method for the mixed volterra-fredholm integral equations,” Journal of Vibration and Control, vol. 13, no. 11, pp. 1667–1675, 2007.

[5] A. Karamete and M. Sezer, “A Taylor collocation method and Legendre wavelets Galerkin method,” Journal of Computational and Applied Mathematics, vol. 58, pp. 415–416, 1996.

[6] M. Ghasemi, M. Fardi, and R. Khoshsiar Ghaziani, “Solution of system of the mixed Volterra-Fredholm integral equations by an analytical method,” Mathematical and Computer Modelling, vol. 58, no. 7-8, pp. 1522–1530, 2013.

[7] A. Yildirim, “Homotopy perturbation method for the mixed Volterra-Fredholm integral equations,” Chaos, Solitons & Fractals, vol. 42, no. 5, pp. 2760–2764, 2009.

[8] Y. Saad and M. H. Schultz, “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems,” SIAM Journal on Scientific and Statistical Computing, vol. 7, no. 3, pp. 856–869, 1986.

[9] Y. Saad, “Iterative Methods for Sparse Linear Systems,” SIAM, New Delhi, Delhi, 2003.

[10] K. Wang and Q. Wang, “Lagrange collocation method for solving Volterra-Fredholm integral equations,” Applied Mathematics and Computation, vol. 219, no. 21, pp. 10434–10440, 2013.

[11] K. Wang and Q. Wang, “Taylor collocation method and convergence analysis for the Volterra-Fredholm integral equations,” Journal of Computational and Applied Mathematics, vol. 260, pp. 294–300, 2014.

[12] M. Gülsu and M. Sezer, “Taylor collocation method for solution of systems of high-order linear Fredholm-Volterra integro-differential equations,” International Journal of Computer Mathematics, vol. 83, no. 4, pp. 429–448, 2006.

[13] F. Mirzaee, “Numerical solution of nonlinear fredholm-volterra integral equa-tions via Bell polynomials,” Computational Methods for Differential Equations, vol. 5, no. 2, pp. 88–102, 2017.

[14] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin, “Wavelet-like bases for the fast solution of second-kind integral equations,” SIAM Journal on Scientific Computing, vol. 14, no. 1, pp. 159–184, 1993.

[15] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, “Adaptive solution of partial differential equations in multiwavelet bases,” Journal of Computational Physics, vol. 182, no. 1, pp. 149–190, 2002.

[16] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, “Adaptive Resolution Discontinuous Galerkin Schemes for Conservation Laws,” Mathematics of Computation, vol. 83, pp. 113–151, 2014.