

Article

Multi-Wavelets Galerkin Method for Solving the System of Volterra Integral Equations

Hoang Viet Long ^{1,2}, Haifa Bin Jebreen ^{3,*} and Stefania Tomasiello ⁴

¹ Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City 70000, Vietnam; hoangvietlong@tdtu.edu.vn

² Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City 70000, Vietnam

³ Department of Mathematics, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia

⁴ Institute of Computer Science, University of Tartu, 50090 Tartu, Estonia; stefania.tomasiello@ut.ee

* Correspondence: hjebreen@ksu.edu.sa

Received: 1 July 2020; Accepted: 12 August 2020; Published: 15 August 2020



Abstract: In this work, an efficient algorithm is proposed for solving the system of Volterra integral equations based on wavelet Galerkin method. This problem is reduced to a set of algebraic equations using the operational matrix of integration and wavelet transform matrix. For linear type, the computational effort decreases by thresholding. The convergence analysis of the proposed scheme has been investigated and it is shown that its convergence is of order $O(2^{-Jr})$, where J is the refinement level and r is the multiplicity of multi-wavelets. Several numerical tests are provided to illustrate the ability and efficiency of the method.

Keywords: Volterra integral equations; operational matrix of integration; multi-wavelets

1. Introduction

In this paper, we study and construct a novel numerical algorithm for the system of Volterra integral equations of the second kind

$$\mathbf{u}(x) = \mathbf{f}(x) + \int_0^x \mathbf{g}(x, t, \mathbf{u}(t)) dt, \quad x \in \Omega := [0, 1], \quad (1)$$

where $\mathbf{f} : \Omega \rightarrow \mathbb{R}^n$ ($n \in \mathbb{N}$) is a given real-valued continuous function, $\mathbf{u} : \Omega \rightarrow \mathbb{R}^n$ is the unknown function that will be determined and the function $\mathbf{g} : S \rightarrow \mathbb{R}^n$ with $S = \{(x, t) : x, t \in \Omega\}$ is a given linear or nonlinear function of \mathbf{u} which satisfies the following Lipschitz condition with respect to the third variable: for all $x, t \in [0, 1]$ and for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$,

$$|\mathbf{g}(x, t, \mathbf{u}_1(t)) - \mathbf{g}(x, t, \mathbf{u}_2(t))| \leq A|\mathbf{u}_1 - \mathbf{u}_2|. \quad (2)$$

Therefore, the functions \mathbf{f} and \mathbf{g} are considered so that the Equation (1) has a unique solution.

Equation (1) is the general form of second-order Volterra integral equation and appears in scientific applications in chemistry, engineering, mathematics, and physics [1–4]. Numerical and analytical solutions of linear and nonlinear Volterra integral equations have been investigated in many papers. A useful method to solve such equations is the Adomian decomposition method. This method was used to investigate the existence and uniqueness of solutions of this type of equation [5,6]. One of the best paper which utilizes the multi-wavelets for solving integro-differential equations was presented by Saray [7]. In [7], an efficient algorithm was proposed for solving the Volterra integro-differential equation. This method outperforming former approaches. Golbabai et al. [8] developed a general

method based on radial basis function networks to solve the system of Volterra integral equations. The modified homotopy perturbation method for solving this type of equation has been proposed by Aminikhah et al. [5,9]. Kılıçman et al. [10] used Simpson’s 3/8 rule to solve this equation. Aguilar and Brunner used collocation techniques based on spline polynomials [11]. The umbral calculus and the Laplace transform methods were used as solution approaches as well [12].

Wavelets and specially multi-wavelets Galerkin method represent an efficient way to solve a variety of equations, including ordinary differential equations (ODEs), partial differential equations (PDEs), and integral equations [7,13,14]. Due to the discrete-time characterization of wavelet coefficient decay, the sparse form of the coefficients matrices arises. This property is very useful to reduce the computational cost. In this work, we aim to solve the system of the Volterra integral equation using Alpert’s multi-wavelets by exploiting the above-mentioned property. Some results are formally proved and supported by numerical experiments.

The paper is structured as follows. A brief introduction of the Alpert’s multi-wavelets is provided in Section 2. In Section 3, the wavelet Galerkin method is used to approximate the solution of the problem, and the convergence analysis is investigated. Some numerical experiments are performed to illustrate the efficiency and accuracy of the proposed method.

2. Alpert’s Multi-Wavelets and Multiresolution Analysis

Alpert et al. [13,15] introduced a class of multi-wavelets for L^2 , which are indexed by a parameter $r \geq 0$ and built via Lagrange polynomials of degree less than r . These multi-wavelets are piecewise polynomials that are locally supported and orthonormal. The multiresolution analysis (MRA) framework, introduced and developed by Mallat [16] and Meyer [17], is useful to construct these bases.

According to MRA, a set of primal scaling functions $\{\phi_{0,0}^0, \dots, \phi_{0,0}^{r-1}\}$ is introduced for primal subspace $V_0^r \in L^2[0, 1]$. By translation and dilation of primal scaling functions $\{\phi^k\}, k = 0, \dots, r - 1$, we determine a space V_j^r ,

$$V_j^r = \text{Span}\{\phi_{j,b}^k := \mathcal{D}_{2^j}\mathcal{T}_b\phi^k, b \in \mathcal{B}_j, k = 0, 1, \dots, r - 1\},$$

where \mathcal{D}_a and \mathcal{T}_b are the dilation and translation operators, respectively such that for a given function h , $\mathcal{D}_a h(x) = a^{\frac{1}{2}}h(ax)$ and $\mathcal{T}_b h(x) = h(x - b)$, also $\mathcal{B}_j := \{0, 1, \dots, 2^j - 1\}$ for $j \in \mathbb{Z}^+ \cup \{0\}$. Therefore, $\phi_{j,b}^k$ is a polynomial of degree less than k which is restricted to $I_{j,b} = [x_{j,b}, x_{j,b+1}]$ where $x_{j,b} := 2^{-j}b$ and $\Omega := [0, 1] = \cup_{b \in \mathcal{B}_j} I_{j,b}$.

For a fixed integer $J \geq 0$, the orthogonal projection $\mathcal{P}_J^r h$ of $h \in L^2[0, 1]$ onto V_J^r is determined by

$$h \approx \mathcal{P}_J^r(h) = \sum_{b \in \mathcal{B}_J} \sum_{k=0}^{r-1} \langle h, \phi_{J,b}^k \rangle \phi_{J,b}^k. \tag{3}$$

The coefficients $h_{j,b}^k = \langle f, \phi_{j,b}^k \rangle$ are determined by $h_{j,b}^k = \int_{I_{j,b}} h(x) \phi_{j,b}^k(x) dx$ [18–20]. To avoid computing integrals, the Gauss–Legendre Quadrature are applied as follows

$$h_{j,b}^k \approx 2^{-J/2} \sqrt{\frac{\omega_k}{2}} h \left(2^{-J} \left(\frac{\tau_k + 1}{2} + b \right) \right), \quad k = 0, \dots, r - 1, b \in \mathcal{B}_J, \tag{4}$$

where ω_k and τ_k are the Gauss–Legendre Quadrature weights and the roots of Legendre polynomial of order r , respectively which are introduced in [21–23]. The projection $\mathcal{P}_J^r h$ converges to h if the function $h \in C^r(\Omega)$ (r times continuously differentiable) [15]. $\mathcal{P}_J^r h$ approximates h with mean error bounded as follows

$$\|\mathcal{P}_J^r(h) - h\| \leq 2^{-Jr} \frac{2}{4^r r!} \sup_{x \in \Omega} |h^{(r)}(x)|. \tag{5}$$

Assume that $\Phi_J^r := [\Phi_{r,J,0}, \dots, \Phi_{r,J,2^J-1}]^T$, where $\Phi_{r,J,b} := [\phi_{J,b}^0, \dots, \phi_{J,b}^{r-1}]$. Φ_J^r refers to the function vector called a multi-scaling function. In fact, Φ_J^r is a function vector which includes the scaling function in the space V_J^r . By this definition, one can rewrite (3) viz, $\mathcal{P}_J^r(h) = H^T \Psi_J^r$ where H is a vector with entries $H_{br+k+1} := h_{J,b}^k$ and has dimension $N := r2^J$.

This construction of multi-wavelets for $L^2(\Omega)$ can be extended to the two-dimensional space including $L^2(\Omega)^2$. Let us introduce the space $V_J^{r,2} := V_J^r \times V_J^r$ which is spanned by orthonormal bases $\{\phi_{J,b}^k, \phi_{J,b'}^{k'} : b, b' \in \mathcal{B}_J, k, k' = 0, 1, \dots, r-1\}$. Therefore, any function $h \in L^2(\Omega)^2$ can be projected onto the $V_J^{r,2}$ by the projection \mathcal{P}_J^r via,

$$h \approx \mathcal{P}_J^r h = \sum_{b \in \mathcal{B}_J} \sum_{k'=0}^{r-1} \sum_{k=0}^{r-1} H_{rb+(k+1),rb'+(k'+1)} \phi_{J,b}^k(x) \phi_{J,b'}^{k'}(y) = \Phi_J^{r,T}(x) H \Phi_J^r(y), \tag{6}$$

where H is an $(N \times N)$ matrix whose elements are obtained by

$$H_{rb+(k+1),rb'+(k'+1)} \approx 2^{-J} \sqrt{\frac{\omega_k}{2}} \sqrt{\frac{\omega_{k'}}{2}} h \left(2^{-J}(\hat{\tau}_k + b), 2^{-J}(\hat{\tau}_{k'} + b') \right), \tag{7}$$

where $\hat{\tau}_k = (\tau_k + 1)/2$. By the following theorem, it is possible to bound the error for such projection, if the function h is sufficiently smooth.

Theorem 1 ([15]). *Suppose that the function $h : [0, 1]^2 \rightarrow \mathbb{R}^2$ has continuous partial derivatives of order r and mixed partial derivative of order $2r$. Then*

$$\|\mathcal{P}_J^r h - h\| \leq \mathcal{M}_{\max} \frac{2^{1-rJ}}{4^r r!} \left(2 + \frac{2^{1-Jr}}{4^r r!} \right), \tag{8}$$

where

$$\mathcal{M}_{\max} = \max \left\{ \sup_{\xi \in [0,1]} \left| \frac{\partial^r}{\partial x^r} h(\xi, y) \right|, \sup_{\eta \in [0,1]} \left| \frac{\partial^r}{\partial y^r} h(x, \eta) \right|, \sup_{\xi', \eta' \in [0,1]} \left| \frac{\partial^{2r}}{\partial x^r \partial y^r} h(\xi', \eta') \right| \right\}.$$

As the subspaces V_j^r are nested, there exist complementary orthogonal subspaces W_j^r such that

$$V_{j+1}^r = V_j^r \oplus W_j^r, \quad j \in \mathbb{Z} \cup \{0\}, \tag{9}$$

here and in the following \oplus denotes orthogonal sums. There is a family of other bases such that the dilations and translations of these bases span the complementary subspaces W_j^r , namely,

$$W_j^r = \text{Span}\{\psi_{j,b}^k := \mathcal{D}_{2^j} \mathcal{T}_b \psi^k, b \in \mathcal{B}_j, k = 0, 1, \dots, r-1\}.$$

The functions $\psi_{j,b}^k$ are called multi-wavelets. Due to (9), the multi-scale decomposition can be inductively found, $V_j^r = V_0^r \oplus (\oplus_{i=0}^{j-1} W_i^r)$. This decomposition gives rise to the multi-scale projection operator \mathcal{M}_j^r that maps $L^2(\Omega)$ onto V_j^r via

$$h \approx \mathcal{M}_j^r(h) = (\mathcal{P}_0^r + \sum_{i=0}^{j-1} \mathcal{Q}_i^r)(h), \tag{10}$$

where \mathcal{Q}_j^r is the orthonormal projection operator that maps $L^2(\Omega)$ onto W_j^r . In fact, by using multi-scale projection operator, any function $h \in L^2(\Omega)$ can be approximated by multi-wavelets of higher levels $W_j^r, j = 0, 1, \dots, J - 1$ and the multi-scaling functions of the coarse space V_0^r viz,

$$h \approx \mathcal{M}_J^r(h) = \sum_{k=0}^{r-1} h_{0,0}^k \phi_{0,0}^k + \sum_{j=0}^{J-1} \sum_{b \in \mathcal{B}_j} \sum_{k=0}^{r-1} \tilde{h}_{j,b}^k \psi_{j,b}^k, \tag{11}$$

where

$$h_{0,0}^k := \langle h, \phi_{0,0}^k \rangle, \quad \tilde{h}_{j,b}^k := \langle h, \psi_{j,b}^k \rangle. \tag{12}$$

Note that the single-scale coefficients $h_{0,0}^k$ can be determined by (4). However, for evaluating the multi-wavelets coefficients $\tilde{h}_{j,b}^k$ of higher levels, in many cases, they have to be calculated numerically. To avoid such numerical computations, the wavelet transform matrix T_J can be applied, as introduced in [14,24]. This matrix is useful to find the multi-wavelets by using the scaling functions

$$\Psi_J^r = T_J \Phi_J^r, \tag{13}$$

where $\Psi_J^r := [\Phi_{r,0,b}, \Psi_{r,0,b}, \Psi_{r,1,b}, \dots, \Psi_{r,J-1,b}]^T$ and $\Psi_{r,j,b} := [\psi_{j,b}^0, \dots, \psi_{j,b}^{r-1}]$, $b \in \mathcal{B}_j$. Using the vector function Ψ_J^r and (11), we can write

$$h \approx \mathcal{M}_J^r(h) = \tilde{H}_J^T \Psi_J^r, \tag{14}$$

where \tilde{H}_J is an N -dimensional vector with entries $h_{0,0}^k$ and $\tilde{h}_{j,b}^k$ for $b \in \mathcal{B}_j, j = 0, \dots, J - 1$ and $k = 0, \dots, r - 1$. Besides, it is obvious that $\tilde{H}_J = T_J^{-1T} H_J$.

Thresholding

Alpert’s multi-wavelets provide vanishing moments of order $N_\psi^k = k + r - 1$ for $k = 0, 1, \dots, r - 1$, i.e.,

$$\mathcal{N}_p^k = \int_{-\infty}^{\infty} x^p \psi_{0,0}^k(x) dx, \quad 0 \leq p < N_\psi^k, \text{ and } k = 0, 1, \dots, r - 1. \tag{15}$$

Furthermore, Alpert’s multi-wavelets are uniformly bounded concerning to L_∞ and L_1 , i.e.,

$$\|\psi_{j,b}^k\|_{L_\infty(\Omega)} \lesssim 1, \quad \|\psi_{j,b}^k\|_{L_1(\Omega)} \lesssim 1. \tag{16}$$

The vanishing moments and normalization (16) imply that the detail coefficients $\tilde{h}_{j,b}^k$ become small when the underlying function is locally smooth. Therefore it is possible to obtain [25]

$$\tilde{h}_{j,b}^k = |\langle h, \psi_{j,b}^k \rangle| \leq \inf_{P \in \Pi_{N_\psi^k}} |\langle h - P, \psi_{j,b}^k \rangle| \lesssim 2^{-jN_\psi^k} \|h\|_{W^{1,N_\psi^k}(\Omega)}. \tag{17}$$

So the detail coefficients decay at the rate of $2^{-jN_\psi^k}$ and in the regions where the function is smooth, most of the detail coefficients may be discarded when the refinement level J increases. This gives rise to thresholding. The thresholding operator \mathcal{T}_{D_ϵ} is introduced by

$$\mathcal{T}_{D_\epsilon}(\tilde{H}_J) = \mathcal{H}_J, \tag{18}$$

where $D_\epsilon := \{(j, b, k) : |\tilde{h}_{j,b}^k| > \epsilon\}$ and the elements of \mathcal{H}_J are defined by

$$\bar{h}_{j,b}^k := \begin{cases} \tilde{h}_{j,b}^k, & (j, b, k) \in D_\epsilon, \\ 0, & \text{else,} \end{cases} \quad b \in \mathcal{B}_j, j = 0, \dots, J - 1, k = 0, \dots, r - 1. \tag{19}$$

Note that the thresholding affects only the detail coefficients while the coarse scale coefficients remain unchanged.

The approximation error due to the thresholding can be estimated similarly to the classical wavelets. Let \mathcal{A}_{D_ϵ} be the approximation operator $\mathcal{A}_{D_\epsilon} := \mathcal{M}_J^{r-1} \mathcal{T}_{D_\epsilon} \mathcal{M}_J^r$. The approximation error due to the thresholding can be bounded as stated by the following proposition [25].

Proposition 1. (Approximation error). *Let Ω be bounded and $\epsilon_j = \bar{a}^{j-J} \epsilon$ with $\bar{a} > 1$. Then the approximation error concerning to the set of significant details D_ϵ is uniformly bounded concerning to $L^q(\Omega)$, $q \in [1, \infty]$, i.e.,*

$$\|\mathcal{P}_J^r h - \mathcal{P}_{J,D_\epsilon}^r h\|_{L^q(\Omega)} \leq C_{thr} \epsilon, \tag{20}$$

for some constant $C_{thr} > 0$ independent of J, ϵ . Here $\mathcal{P}_J^r h$ and $\mathcal{P}_{J,D_\epsilon}^r h$ are the projections according to (11) corresponding to the coefficients \tilde{H}_J and $\mathcal{A}_{D_\epsilon} \tilde{H}_J$.

3. Multi-Wavelets Galerkin Method

In this section, we use the wavelet Galerkin method to solve the system of the Volterra integral Equation (1). To this end, we will apply the interpolation property of scaling functions to reach an efficient algorithm. Assume that the solution $\mathbf{u}(x)$ of Equation (1) can be expanded using multi-scale projection operator \mathcal{M}_J^r based on multi-wavelets as follows

$$\mathbf{u}(x) \approx \mathcal{M}_J^r(\mathbf{u})(x) = (\mathcal{P}_0^r + \sum_{j=0}^{J-1} \mathcal{Q}_j^r)(\mathbf{u})(x) = \tilde{\mathbf{U}}^T \otimes \Psi_J^r(x), \tag{21}$$

where \otimes is the Kronecker product and $\tilde{\mathbf{U}} = (\tilde{U}_1^T, \dots, \tilde{U}_n^T)^T$ is a $(1 \times nr2^J)$ vector whose elements are n unknown sub-vectors \tilde{U}_i with a dimension of $(r2^J \times 1)$ such that

$$u_i(x) := \tilde{U}_i^T \Psi_J^r(x), \quad i = 1, 2, \dots, n.$$

One can imagine two types of equations, linear and nonlinear. For the linear type, the i -th component of the vector function $\mathbf{g}(x, t, \mathbf{u}(t))$ has the form

$$\mathbf{g}_j(x, t, \mathbf{u}(t)) := \sum_{i=1}^n a_{ji}(x, t) u_i(t), \quad j = 1, 2, \dots, n, \tag{22}$$

and it can be approximated by multi-scale operator, i.e.,

$$\begin{aligned} \sum_{i=1}^n a_{ji}(x, t) u_i(t) &\approx \sum_{i=1}^n \mathcal{P}_J^r(a_{ji} u_i)(x, t) = \sum_{i=1}^n \Phi_J^{rT}(x) A_{ji}^T \Phi_J^r(t) \\ &\sum_{i=1}^n \Psi_J^{rT}(x) T_J A_{ji}^T T_J^{-1} \Psi_J^r(t), \quad j = 1, 2, \dots, n, \end{aligned}$$

where A_{ji} ($j, i = 1 : n$) are $(r2^J \times r2^J)$ matrices. Integrating from 0 to x , we get

$$\begin{aligned} \int_0^x \mathbf{g}_j(x, t, \mathbf{u}(t)) dt &= \sum_{i=1}^n \underbrace{\Psi_J^{rT}(x) T_J A_{ji}^T T_J^{-1} I_\psi \Psi_J^r(x)}_{p_j(x)} \\ &= \sum_{i=1}^n P_i^T T_J^{-1} \Psi_J^r(x) = \sum_{i=1}^n \tilde{U}_i^T \mathcal{A}_{ji} T_J^{-1} \Psi_J^r(t), \end{aligned} \tag{23}$$

where \mathcal{A}_{ji} ($j, i = 1 : n$) are $(r2^J \times r2^J)$ matrices and the rest are $(r2^J \times 1)$ vectors. But if the j -th component of vector function $\mathbf{g}(x, t, \mathbf{u}(t))$ is nonlinear, one can consider the following expansion

$$\int_0^x g_j(x, t, \mathbf{u}(t))dt \approx \mathcal{P}_j^r \left(\int_0^x g_j \left(x, t, \mathcal{M}_j^r(\mathbf{u})(t) \right) dt \right) \tag{24}$$

$$\mathbf{G}_j^T \Phi_j^r(x) = G_j^T T_J^{-1} \Psi_J^r(x), \quad j = 1, 2, \dots, n,$$

where G_j is a $(r2^J \times 1)$ vector whose elements are nonlinear algebraic equations. In view of the Equations (23) and (24), and using operational matrix I_ψ of integration for multi-wavelets introduced in [7,14,15], one can write

$$\int_0^x \mathbf{g}(x, t, \mathbf{u}(t))dt \approx \begin{cases} \tilde{\mathbf{U}}^T \Gamma^T \otimes T_J^{-1} \Psi_J^r(x), & \text{linear,} \\ \mathbf{G}^T \otimes T_J^{-1} \Psi_J^r(x), & \text{nonlinear,} \end{cases} \tag{25}$$

with $\Gamma := (\mathcal{A})_{ji}$, ($j, i = 1 : n$) and $\mathbf{G} := (G_1^T, G_2^T, \dots, G_n^T)^T$.

Such an approximation can be considered for the j -th element of \mathbf{f} viz,

$$f_j(x) \approx \mathcal{P}_j^r(f_j)(x) = F_j^T \Phi_j^r(x) = F_j^T T_J^{-1} \Psi_J^r(x),$$

and thus by putting $\mathbf{F} := (F_1^T, F_2^T, \dots, F_n^T)^T$ we have

$$\mathbf{f} \approx \mathbf{F}^T \otimes T_J^{-1} \Psi_J^r(x). \tag{26}$$

Now, we introduce the residual as

$$r_j^r(x) = \tilde{\mathbf{U}}^T \otimes \Psi_J^r(x) - \mathbf{F}^T \otimes T_J^{-1} \Psi_J^r(x) - \begin{cases} \tilde{\mathbf{U}}^T \Gamma^T \otimes T_J^{-1} \Psi_J^r(x), & \text{linear,} \\ \mathbf{G}^T \otimes T_J^{-1} \Psi_J^r(x), & \text{nonlinear.} \end{cases} \tag{27}$$

To apply the Galerkin method, it is necessary that $\langle r_j^r, \Psi_J^r \rangle = 0$. Thus we have

$$\tilde{\mathbf{U}}^T - \mathbf{F}^T \otimes T_J^{-1} - \begin{cases} \tilde{\mathbf{U}}^T \Gamma^T \otimes T_J^{-1} \\ \mathbf{G}^T \otimes T_J^{-1} \end{cases} = 0, \quad \begin{matrix} \text{linear,} \\ \text{nonlinear.} \end{matrix} \tag{28}$$

By solving this system of linear and nonlinear equations using restarted generalized minimal residual method (GMRES) and Newton methods, respectively, we obtain the approximate solution of the Equation (1). Note that because we use the Galerkin method with orthogonal bases, such a system will have a unique solution [26].

Convergence Analysis

Theorem 2. Suppose that $\mathbf{e}_j = \mathbf{u} - \mathcal{M}_j^r(\mathbf{u})$ where \mathbf{u} and $\mathcal{M}_j^r(\mathbf{u})$ are the exact and approximate solutions of nonlinear system (1), respectively. Let X be an open set in \mathbb{R} and let $g : \Omega \times X \rightarrow \mathbb{R}$ be a function such that $g(x, t, \mathbf{u}(x)) \in C^r(\Omega)$ for any $\mathbf{u} \in X$ and the condition (2) is satisfied. Furthermore, presume that $\mathbf{f} \in C^r(\Omega)$. Furthermore, assume that the residual $\mathbf{e} := \tilde{r} - (r_j^r)$ where the residual \tilde{r} is specified as

$$\tilde{r}(x) = u(x) - f(x) - \int_0^x g(x, t, u(t))dt, \tag{29}$$

and r_j^r is introduced in (27).

If $\mathbf{u} \in C^r(\Omega)$, and the method used to solve system (28) is convergent then one has

$$\|\mathbf{e}\|_2^2 \leq \frac{2^{1-Jr}}{4^r r!} \left(|1 + \sqrt{nr2^J} \kappa| \sup_{x \in \Omega} |\mathbf{u}(x)| + \sup_{x \in \Omega} |\mathbf{f}(x)| \right),$$

where κ is a positive constant. Consequently, $\mathbf{e} \rightarrow 0$ when $J \rightarrow \infty$.

Proof. Using (27), (29) and the hypotheses of the theorem, we can write

$$\mathbf{e}(x) = \mathbf{e}_J(x) - \left(\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x) \right) - \left(\int_0^x \mathbf{g}(x, t, \mathbf{u}(t)) dt - \mathcal{M}_J^r \left(\int_0^x \mathbf{g}(x, t, \mathcal{M}_J^r(\mathbf{u}))(t) dt \right) \right), \quad (30)$$

where $\mathbf{e} := \tilde{r} - (r_J^r)$. Since the function \mathbf{g} satisfies the Lipschitz condition (2), Equation (30) can be reduced to

$$\mathbf{e}(x) = \mathbf{e}_J(x) - \left(\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x) \right) - A \int_0^x \mathbf{e}_J(t) dt.$$

Now, suppose that

$$\mathbf{e} \approx \mathcal{E} \otimes \Psi_J^r, \quad \mathbf{e}_J \approx \mathcal{E}_J \otimes \Psi_J^r,$$

where \mathcal{E} and \mathcal{E}_J are the $(1 \times nr2^J)$ vectors and thus, one can write

$$\mathcal{E} \otimes \Psi_J^r = \mathcal{E}_J \otimes \Psi_J^r - \left(\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x) \right) - A \mathcal{E}_J \otimes I_\Psi \Psi_J^r,$$

Taking L_2 -norm from both sides and using the triangle inequality yields

$$\begin{aligned} \|\mathcal{E} \otimes \Psi_J^r\|_2^2 &\leq \|\mathcal{E}_J \otimes \Psi_J^r\|_2^2 + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 + A \|\mathcal{E}_J \otimes I_\Psi \Psi_J^r\|_2^2 \\ &= \|\mathcal{E}_J\|_2^2 \|\Psi_J^r\|_2^2 + A \|\mathcal{E}_J\|_2^2 \|I_\Psi \Psi_J^r\|_2^2 + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 \\ &\leq \|\mathcal{E}_J\|_2^2 \|\Psi_J^r\|_2^2 + A \|\mathcal{E}_J\|_2^2 \|I_\Psi\|_2^2 \|\Psi_J^r\|_2^2 + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 \end{aligned}$$

where the second row comes from theorem 8 in [27]. Since Alpert multi-wavelets are orthonormal, one can write

$$\begin{aligned} \|\mathcal{E}\|_2^2 &\leq \|\mathcal{E}_J\|_2^2 + A \|\mathcal{E}_J\|_2^2 \|I_\Psi\|_2^2 + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 \\ &= \|\mathcal{E}_J\|_2^2 \left(\|I_{nr2^J}\|_2^2 + A \|I_\Psi\|_2^2 \right) + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2. \end{aligned}$$

According to the previous section, when Ψ_J^r has high vanishing moments and the function h is smooth, $\langle h, \Psi_J^r \rangle$ decays fast in $J \rightarrow \infty$. By means of vanishing moments of Alpert’s multi-wavelets and the matrix norms inequalities, we get

$$\|\mathcal{E}\|_2^2 \leq \|\mathcal{E}_J\|_2^2 |1 + \sqrt{nr2^J} \kappa| + \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2,$$

where $\kappa = A \|I_\Psi\|_\infty$. Now Equation (5) leads to the desired result

$$\|\mathcal{E}\|_2^2 \leq \frac{2^{1-Jr}}{4^r r!} \left(|1 + \sqrt{nr2^J} \kappa| \sup_{x \in \Omega} |\mathbf{u}(x)| + \sup_{x \in \Omega} |\mathbf{f}(x)| \right).$$

□

Theorem 3. Let the condition of Theorem 2 be valid. Suppose that \mathbf{u}_J is the approximate solution obtained from solving (28) using restarted GMRES or Newton methods. If these methods solve (28) with proper accuracy, the error can be estimated from

$$\|\mathbf{u} - \mathbf{u}_J\|_2^2 \leq (1 - A \|I_\Psi\|_2^2)^{-1} \frac{2^{1-Jr}}{4^r r!} \sup_{x \in \Omega} |\mathbf{f}(x)| + \eta, \quad (31)$$

where η is a small positive number that desire to zero.

Proof. Taking $\mathcal{M}_J^r(\mathbf{u})$ as the approximate solution of (1) and \mathbf{u}_J as the approximate solution obtained from solving (28) using restarted GMRES or Newton methods, the convergence can be concluded from

$$\|\mathbf{u} - \mathbf{u}_J\| \leq \|\mathbf{u} - \mathcal{M}_J^r(\mathbf{u})\| + \|\mathcal{M}_J^r(\mathbf{u}) - \mathbf{u}_J\|. \tag{32}$$

The approximate solution of (1) satisfies

$$\mathcal{M}_J^r(\mathbf{u})(x) = \mathcal{M}_J^r(\mathbf{f})(x) + \mathcal{M}_J^r\left(\int_0^x \mathbf{g}(x, t, \mathcal{M}_J^r(\mathbf{u})(t) dt\right). \tag{33}$$

Subtracting (33) from (1), and using the Lipschits condition (2), one can write

$$\mathbf{e}_J \leq \mathbf{f} - \mathcal{M}_J^r(\mathbf{f}) + A \int_0^x \mathbf{e}_J dt. \tag{34}$$

Let us consider $\mathbf{e}_J \approx \mathcal{E}_J \otimes \Psi_J^r$ where \mathcal{E}_J is the $(1 \times nr2^J)$ vector and thus, we have

$$\mathcal{E}_J \otimes \Psi_J^r = \left(\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\right) + A\mathcal{E}_J \otimes I_\psi \Psi_J^r. \tag{35}$$

Taking L_2 -norm from both sides of (35) and using Theorem 8 in [27] yields

$$\begin{aligned} \|\mathcal{E}_J \otimes \Psi_J^r\|_2^2 &\leq \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 + A\|\mathcal{E}_J \otimes I_\psi \Psi_J^r\|_2^2 \\ &\leq \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 + A\|\mathcal{E}_J\|_2^2\|I_\psi \Psi_J^r\|_2^2 \end{aligned}$$

Since Alpert multi-wavelets are orthonormal, one can write

$$\|\mathcal{E}_J\|_2^2 \leq \|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2 + A\|\mathcal{E}_J\|_2^2\|I_\psi\|_2^2. \tag{36}$$

Therefore one can bound the error of $\|\mathbf{u} - \mathcal{M}_J^r(\mathbf{u})\|$ via

$$\|\mathcal{E}_J\|_2^2 \leq (1 - A\|I_\psi\|_2^2)^{-1}\|\mathbf{f}(x) - \mathcal{M}_J^r(\mathbf{f})(x)\|_2^2. \tag{37}$$

According to the theorem hypotheses, the methods used to solve the obtained system are convergent. So $\eta := \|\mathcal{M}_J^r(\mathbf{u}) - \mathbf{u}_J\|$ will be very small. Inequality (31) is obtained using (5) and (32), i.e.,

$$\|\mathbf{u} - \mathbf{u}_J\|_2^2 \leq (1 - A\|I_\psi\|_2^2)^{-1} \frac{2^{1-Jr}}{4^r r!} \sup_{x \in \Omega} |\mathbf{f}(x)| + \eta.$$

□

4. Numerical Examples

In this section, some numerical experiments are considered to illustrate the convergence and efficiency of the proposed method. To this end, we report the L^2 errors of the solution which is defined by

$$\xi_u := \|u - \mathcal{M}_J^r(u)\|_2 = \left(\int_\Omega |u(x) - \mathcal{M}_J^r(u)(x)|^2 dx\right)^{1/2},$$

where u and $\mathcal{M}_J^r(u)$ are the exact and approximate solution of systems (1), respectively. In order to get the sparse coefficients matrix in the linear type, all the entries of this matrix that are less than a small positive number ε are set to zero. Finally, one can find the rate of sparsity S_ε which is defined by [28]

$$S_\varepsilon = \frac{N_0 - N_\varepsilon}{N_0} \times 100\%.$$

where N_0 is the total number of elements and N_ϵ the number of elements remaining after thresholding.

Example 1. Let us run the proposed method on the following linear Volterra integral equation [5,29]

$$u_1(x) = - \int_0^x e^{-(s-x)} u_1(s) ds - \int_0^x \cos(s-x) u_2(s) ds + \cosh(x) + x \sin(x),$$

$$u_2(x) = - \int_0^x e^{s+x} u_1(s) ds - \int_0^x x \cos(s) u_2(s) ds + 2 \sin(x) + x(\sin^2(x) + e^x).$$

The exact solution is $\mathbf{u}(x) = (e^{-x}, 2 \sin(x))$. The effect of thresholding on L_2 -errors and sparsity percentage is reported in Table 1 for different values of r, J and ϵ . To illustrate the effect of the refinement level J and the multiplicity parameter r on L^2 error, Figure 1 is plotted. Figure 2 shows sparse matrix when $r = 5, J = 3$ and $\epsilon = 10^{-4}, 10^{-2}$.

Table 1. Effects of parameters r, J and ϵ on sparsity and L^2 -error for Example 1.

		Without Thresholding		$\epsilon = 10^{-5}$		$\epsilon = 10^{-3}$	
r	J	S_ϵ	L^2 -Error	S_ϵ	L^2 -Error	S_ϵ	L^2 -Error
3	2	6.25	$\zeta_{u_1} = 3.24 \times 10^{-5}$ $\zeta_{u_2} = 8.41 \times 10^{-5}$	24.48	$\zeta_{u_1} = 3.30 \times 10^{-5}$ $\zeta_{u_2} = 8.41 \times 10^{-5}$	49.83	$\zeta_{u_1} = 2.42 \times 10^{-4}$ $\zeta_{u_2} = 3.04 \times 10^{-4}$
	3	17.19	$\zeta_{u_1} = 4.05 \times 10^{-6}$ $\zeta_{u_2} = 1.05 \times 10^{-5}$	53.30	$\zeta_{u_1} = 6.02 \times 10^{-6}$ $\zeta_{u_2} = 1.26 \times 10^{-5}$	74.13	$\zeta_{u_1} = 2.41 \times 10^{-4}$ $\zeta_{u_2} = 5.15 \times 10^{-4}$
5	2	6.25	$\zeta_{u_1} = 6.38 \times 10^{-9}$ $\zeta_{u_2} = 1.66 \times 10^{-8}$	45.69	$\zeta_{u_1} = 4.62 \times 10^{-6}$ $\zeta_{u_2} = 1.28 \times 10^{-6}$	70.69	$\zeta_{u_1} = 1.53 \times 10^{-4}$ $\zeta_{u_2} = 6.51 \times 10^{-4}$
	3	17.19	$\zeta_{u_1} = 2.00 \times 10^{-10}$ $\zeta_{u_2} = 5.19 \times 10^{-10}$	71.42	$\zeta_{u_1} = 1.24 \times 10^{-5}$ $\zeta_{u_2} = 8.54 \times 10^{-6}$	86.75	$\zeta_{u_1} = 1.53 \times 10^{-4}$ $\zeta_{u_2} = 6.52 \times 10^{-4}$

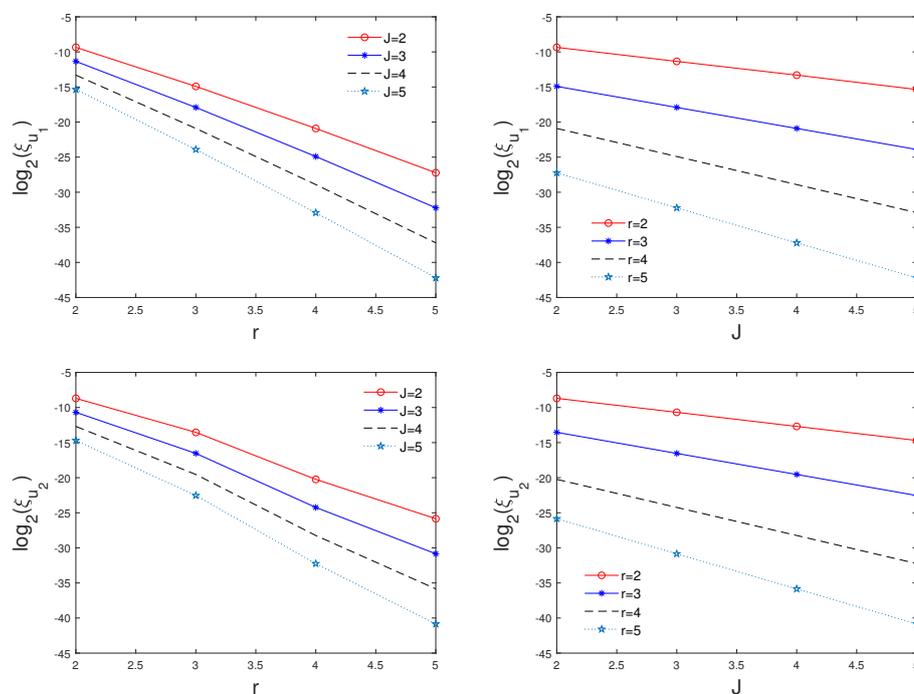


Figure 1. Effects of the refinement level J (left) and the multiplicity parameter r (right) on L^2 error when $r = 5$ and $J = 2$ for Example 1.

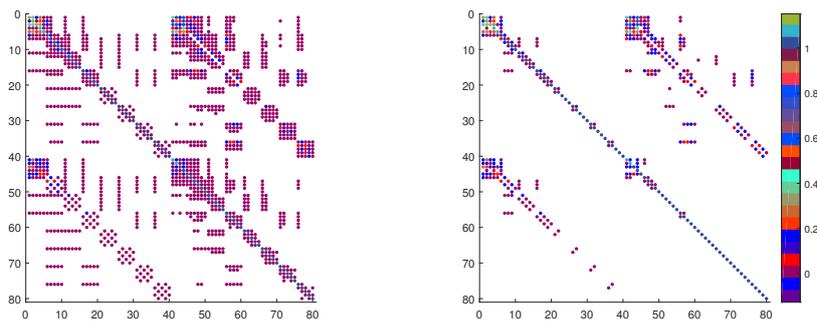


Figure 2. Plot of sparse matrix after thresholding with $\epsilon = 10^{-4}$ (left) and $\epsilon = 10^{-2}$ (right) for Example 1.

Example 2. Let us consider the following system of Volterra integral equation as a further example

$$u_1(x) = -\frac{x^5}{3} - \frac{x^4}{4} + \frac{x^3}{3} + x + \int_0^x (x^2 - s)(u_1(s) + u_2(s))ds$$

$$u_2(x) = \frac{x^3}{2} - \frac{x^4}{3} + x^2 - \int_0^x x(u_1(s) - u_2(s))ds.$$

The solution is reported in [5,30] and is $\mathbf{u} = (x, x^2)$. To illustrate the effect of thresholding on the coefficients matrix obtained from proposed method, the graph in Figure 3 is provided. Figure 4 shows the effect of parameters r and J .

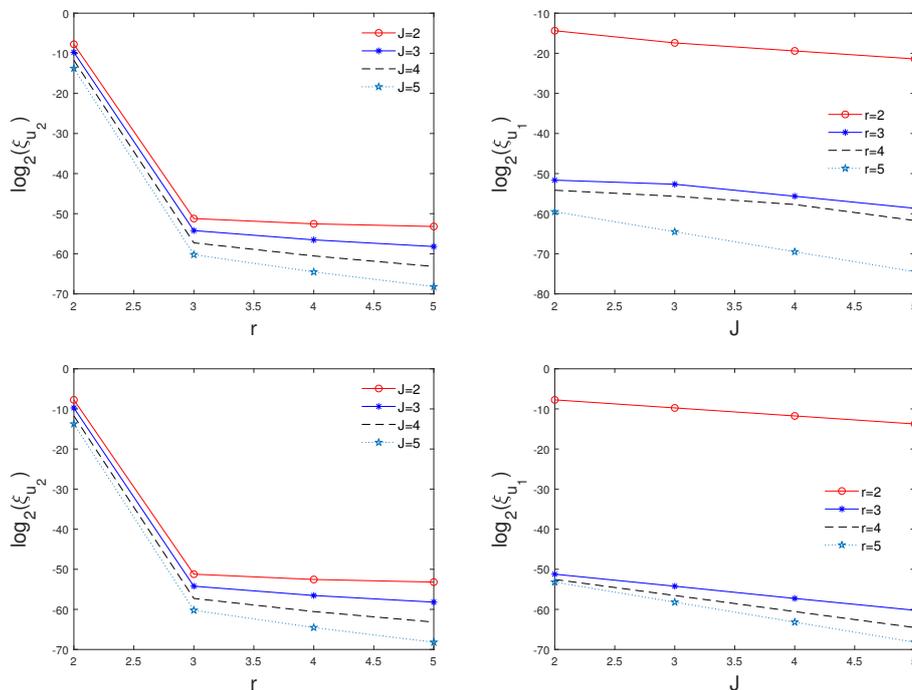


Figure 3. Effects of the refinement level J (left) and the multiplicity parameter r (right) on L^2 error when $r = 4$ and $J = 3$ for Example 2.

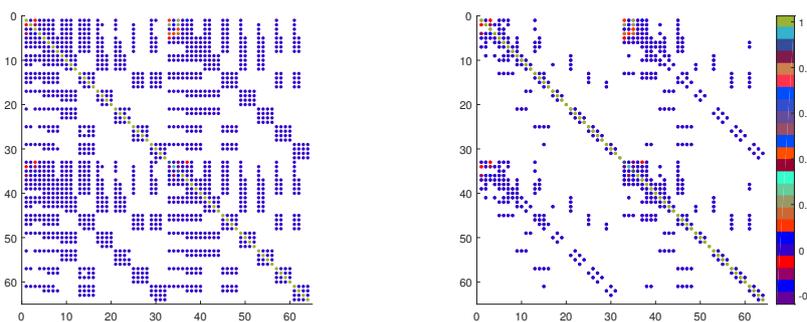


Figure 4. Plot of sparse matrix after thresholding with $\epsilon = 10^{-5}$ (left) and $\epsilon = 10^{-3}$ (right) for Example 2.

Example 3. Let us consider the following nonlinear system

$$u_1(x) = \sin(x) - x + \int_0^x (u_1^2(s) + u_2^2(s))ds$$

$$u_2(x) = \cos(x) - \frac{1}{2} \sin^2(x) + \int_0^x u_1(s)u_2(s)ds.$$

The exact solutions of this system are $u_1 = \sin(x)$ and $u_2 = \cos(x)$ [8].

Table 2 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. In Tables 3 and 4, results are compared with other methods [10,31–33] in terms of the absolute errors. In this paper, J and r are the criteria for the discretization and the degree of polynomials used as a basis, respectively. Taking $r = 7$ and $J = 2$, the results of Tables 3 and 4 indicate that the proposed method solves this equation better than others [10,31–33]. Furthermore, we reported the exact and numerical solution by Figure 5.

Table 2. Effect of the refinement level J and multiplicity parameter r on L_2 -error for Example 3.

	r	$J = 2$	$J = 3$	$J = 4$	$J = 5$
u_1	3	4.20×10^{-5}	5.25×10^{-6}	6.56×10^{-7}	8.20×10^{-8}
	4	4.04×10^{-7}	2.53×10^{-8}	1.58×10^{-9}	9.86×10^{-11}
	5	8.31×10^{-9}	2.60×10^{-10}	8.12×10^{-12}	2.54×10^{-13}
u_2	3	2.50×10^{-5}	3.21×10^{-6}	4.01×10^{-7}	5.02×10^{-8}
	4	6.62×10^{-7}	4.14×10^{-8}	2.59×10^{-9}	1.62×10^{-10}
	5	5.06×10^{-9}	1.58×10^{-10}	4.94×10^{-12}	1.54×10^{-13}

Table 3. The comparison between absolute errors of Example 3 for u_1 .

x	HPM [31] ($n = 5$)	Method Based upon Discretization [32] ($h = 200$)	Simpson’s 3/8 Rule [10] ($h = 0.025$)	Bernstein Collocation Method [33] ($n = 10$)	Present Method ($r = 7, J = 2$)
0.1	1.4×10^{-07}	2.4×10^{-05}	3.0×10^{-10}	5.5×10^{-10}	1.0×10^{-12}
0.2	3.5×10^{-06}	9.8×10^{-05}	1.1×10^{-09}	1.7×10^{-10}	1.1×10^{-12}
0.3	5.5×10^{-05}	2.3×10^{-04}	3.6×10^{-09}	2.6×10^{-10}	1.1×10^{-12}
0.4	3.8×10^{-04}	4.1×10^{-04}	6.0×10^{-09}	1.0×10^{-10}	9.6×10^{-13}
0.5	1.6×10^{-03}	6.6×10^{-04}	8.7×10^{-09}	1.1×10^{-10}	3.2×10^{-12}
0.6		9.7×10^{-04}	1.4×10^{-08}	2.5×10^{-10}	8.4×10^{-13}
0.7		1.4×10^{-03}	1.9×10^{-08}	5.8×10^{-10}	9.2×10^{-13}
0.8		1.8×10^{-03}	2.4×10^{-08}	1.0×10^{-08}	9.2×10^{-13}
0.9		2.5×10^{-03}	3.3×10^{-08}	1.0×10^{-07}	5.0×10^{-12}
1.0		3.2×10^{-03}	4.0×10^{-08}	8.2×10^{-07}	6.7×10^{-11}

Table 4. The comparison between absolute errors of Example 3 for u_2 .

	HPM [31]	Method Based upon Discretization [32]	Simpson’s 3/8 Rule [10]	Bernstein Collocation Method [33]	Present Method ($r = 7, J = 2$)
x	($n = 5$)	($h = 200$)	($h = 0.025$)	($n = 10$)	
0.1	3.2×10^{-07}	2.5×10^{-04}	5.3×10^{-10}	6.5×10^{-11}	1.2×10^{-13}
0.2	1.1×10^{-05}	5.0×10^{-04}	3.0×10^{-10}	1.3×10^{-10}	1.6×10^{-13}
0.3	1.2×10^{-04}	7.5×10^{-04}	6.0×10^{-10}	6.5×10^{-12}	4.0×10^{-13}
0.4	6.3×10^{-04}	1.0×10^{-04}	2.1×10^{-09}	6.5×10^{-11}	3.9×10^{-13}
0.5	2.2×10^{-03}	1.2×10^{-03}	3.1×10^{-09}	9.0×10^{-11}	1.3×10^{-12}
0.6		1.5×10^{-03}	5.3×10^{-09}	4.2×10^{-11}	6.0×10^{-13}
0.7		1.8×10^{-03}	9.4×10^{-09}	4.2×10^{-11}	6.8×10^{-13}
0.8		2.0×10^{-03}	1.4×10^{-08}	3.6×10^{-09}	8.1×10^{-13}
0.9		2.3×10^{-03}	2.0×10^{-08}	3.5×10^{-08}	4.8×10^{-13}
1.0		2.6×10^{-03}	2.9×10^{-08}	2.9×10^{-07}	4.2×10^{-11}

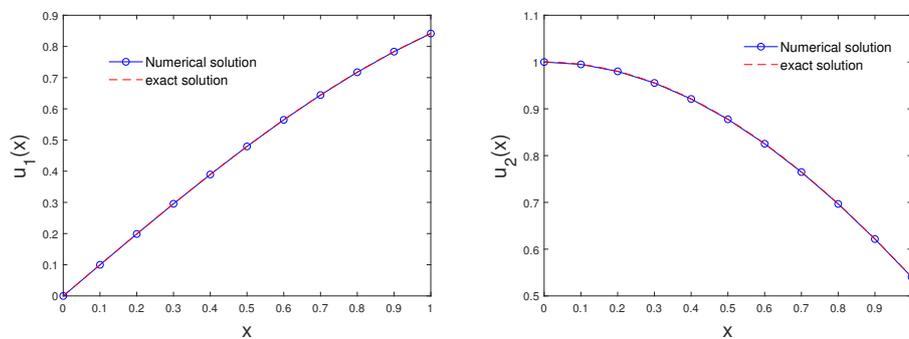


Figure 5. Plot of the exact and numerical solution taking $r = 6$ and $J = 2$ for Example 3.

5. Conclusions

In this paper, we proposed the multi-wavelets Galerkin method to solve the linear and nonlinear Volterra integral equation. The convergence analysis and numerical simulations indicate that the proposed method gives a satisfactory approximation to the exact solution. Thresholding can be used to increase sparsity for a lower computational cost, without affecting the error in L^2 . Using the interpolation property of Alpert’s multi-wavelets, the proposed method turns out to be fast and very competitive against state-of-the-art techniques. The main advantages of the proposed method are the lower computational cost and lower complexity.

Author Contributions: Conceptualization, H.V.L. and H.B.J.; methodology, software, H.B.J. and S.T.; validation, formal analysis, H.B.J.; writing–original draft preparation, investigation, funding acquisition, H.V.L., H.B.J. and S.T.; writing–review and editing, H.V.L., H.B.J. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported by Researchers Supporting Project number (RSP-2020/210), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jerri, A. *Introduction to Integral Equations with Applications*; Wiley: New York, NY, USA, 1999.
2. Li, Y.; Xu, R.; Lin, J. Global dynamics for a class of infection-age model with nonlinear incidence. *Nonlinear Anal. Model. Control* **2018**, *24*, 47–72. [CrossRef]
3. Nordbo, A.; Wyller, J.; Einevoll, G.T. Neural network firing-rate models on integral form. *Biol. Cybern.* **2007**, *97*, 195–209. [CrossRef] [PubMed]

4. Porter, D.; Stirling, D.S. *Integral Equations: A Practical Treatment from Spectral Theory to Applications*; Cambridge University Press: Cambridge, UK, 2004.
5. Aminikhah, H.; Biazar, J. A new analytical method for solving systems of Volterra integral equations. *Int. J. Comput. Math.* **2010**, *87*, 1142–1157. [[CrossRef](#)]
6. Biazar, J.; Babolian, E.; Islam, R. Solution of a system of Volterra integral equations of the first kind by Adomian method. *Appl. Math. Comput.* **2003**, *139*, 249–258.
7. Saray, B.N. An efficient algorithm for solving Volterra integro-differential equations based on Alpert's multi-wavelets Galerkin method. *J. Comput. Appl. Math.* **2019**, *348*, 453–465. [[CrossRef](#)]
8. Golbabai, A.; Mammadova, M.; Seifollahi, S. Solving a system of nonlinear integral equations by an RBF network. *Comput. Math. Appl.* **2009**, *57*, 1651–1658.
9. Saberi-Nadjafi, J.; Tamangar, M. Modified homotopy perturbation method for solving the system of Volterra integral equations. *Int. Nonlinear Sci. Numer. Simul.* **2008**, *9*, 409–413. [[CrossRef](#)]
10. Kılıçman, A.; Kargaran Dehkordi, L.; Tavassoli Kajani, M. Numerical Solution of Nonlinear Volterra Integral Equations System Using Simpson's 3/8 Rule. *Math. Prob. Eng.* **2012**, *2012*, 1–16. [[CrossRef](#)]
11. Aguilar, M.; Brunner, H. Collocation methods for second-order Volterra integrodifferential equations. *Appl. Numer. Math.* **1988**, *4*, 455–470. [[CrossRef](#)]
12. Górska, K.; Horzela, A. The Volterra type equations related to the non-Debye relaxation. *Commun. Nonlinear Sci.* **2020**, *85*, 105246. [[CrossRef](#)]
13. Alpert, B.; Beylkin, G.; Gines, D.; Vozovoi, L. Adaptive solution of partial differential equations in multiwavelet bases. *J. Comput. Phys.* **2002**, *182*, 149–190. [[CrossRef](#)]
14. Saray, B.N.; Lakestani, M.; Razzaghi, M. Sparse representation of system of Fredholm integro-differential equations by using Alpert multiwavelets. *Comput. Math. Math. Phys.* **2015**, *55*, 1468–1483. [[CrossRef](#)]
15. Alpert, B.; Beylkin, G.; Coifman, R.R.; Rokhlin, V. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM Sci. Stat. Comput.* **1993**, *14*, 159–184. [[CrossRef](#)]
16. Mallat, S.G. *A Wavelet Tour of Signal Processing*; Academic Press: Cambridge, MA, USA, 1999.
17. Meyer, Y. *Wavelets and Operators*; Cambridge University Press: Cambridge, UK, 1993.
18. Dehghana, M.; Saray, B.N.; Lakestani, M. Three methods based on the interpolation scaling functions and the mixed collocation finite difference schemes for the numerical solution of the nonlinear generalized Burgers–Huxley equation. *Math. Comput. Model.* **2012**, *55*, 1129–1142. [[CrossRef](#)]
19. Saray, B.N.; Lakestani, M.; Cattani, C. Evaluation of mixed Crank–Nicolson scheme and Tau method for the solution of Klein–Gordon equation. *Appl. Math. Comput.* **2018**, *331*, 169–181.
20. Shahriari, M.; Saray, B.N.; Lakestani, M.; Manafia, J. Numerical treatment of the Benjamin–Bona–Mahony equation using Alpert multiwavelets. *Eur. Phys. J. Plus.* **2018**, *133*, 1–12. [[CrossRef](#)]
21. Shamsi, M.; Razzaghi, M. Numerical solution of the controlled duffing oscillator by the interpolating scaling functions, *Electromagn. Waves Appl.* **2004**, *18*, 691–705.
22. Shamsi, M.; Razzaghi, M. Solution of Hallen's integral equation using multiwavelets. *Comput. Phys. Comm.* **2005**, *168*, 187–197. [[CrossRef](#)]
23. Seyed, S.; Saray, B.N.; Nobari, M. Using interpolation scaling functions based on Galerkin method for solving non-Newtonian fluid flow between two vertical flat plates. *Appl. Math. Comput.* **2015**, *269*, 488–496.
24. Saray, B.N.; Manafian, J. Sparse representation of delay differential equation of pantograph type using multiwavelets Galerkin method. *Eng. Comput.* **2018**, *35*, 887–903. [[CrossRef](#)]
25. Hovhannisyan, N.; Müller, S.; Schäfer, R. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. *Math. Comput.* **2014**, *83*, 113–151. [[CrossRef](#)]
26. Atkinson KE. *The Numerical Solution of Integral Equations of the Second Kind*; Cambridge University Press: Cambridge, UK, 1997.
27. Lancaster, P.; Farahat, H.K. Norms on direct sums and Tensor products. *Math. Comput.* **1972**, *26*, 401–414. [[CrossRef](#)]
28. Goswami, J.C.; Chan, A.K.; Chui, C.K. On solving first-kind integral equations using wavelets on bounded interval. *IEEE Trans. Antennas Propag.* **1995**, *43*, 614–622. [[CrossRef](#)]
29. Delves, L.M.; Mohamed, J.L. *Computational Methods for Integral Equations*; Cambridge University Press: New York, NY, USA, 1988.
30. Babolian, E.; Biazar, J. Solution of a system of non-linear Volterra integral equations of the second kind. *Far East J. Math. Sci.* **2000**, *2*, 935–945.

31. Biazar, J.; Ghazvini, H. Hes homotopy perturbation method for solving systems of Volterra integral equations of the second kind. *Chaos Soliton Fract.* **2009**, *39*, 770–777. [[CrossRef](#)]
32. Yaghouti, M.R. A numerical method for solving a system of Volterra integral equations. *World Appl. Program.* **2012**, *2*, 18–33.
33. Davaeifar, S.; Rashidinia, J. Approximate Solution of System of Nonlinear Volterra Integro-Differential Equations by Using Bernstein Collocation Method. *Int. J. Math. Model. Comput.* **2017**, *7*, 79–91.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).