ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2015, Vol. 55, No. 9, pp. 1468–1483. © Pleiades Publishing, Ltd., 2015.

Sparse Representation of System of Fredholm Integro-Differential Equations by using Alpert Multiwavelets[1]

Behzad Nemati Saray[a]*, Mehrdad Lakestani[a], and Mohsen Razzaghi[b]*****

[a] *Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran*

[b] *Department of Mathematics and Statistics Mississippi State University, Mississippi State, MS 39762, USA*

*e-mail: *b_nemati@tabrizu.ac.ir; **lakestani@tabrizu.ac.ir, ***razzaghi@math.msstate.edu*

Received September 10, 2013

Abstract—A numerical technique is presented for the solution of system of Fredholm integro-differential equations. The method consists of expanding the required approximate solution as the elements of Alpert multiwavelet functions (see Alpert B. et al. J. Comput. Phys. 2002, vol. 182, pp. 149–190). Using the operational matrix of integration and wavelet transform matrix, we reduce the problem to a set of algebraic equations. This system is large. We use thresholding to obtain a new sparse system; consequently, GMRES method is used to solve this new system. Numerical examples are included to demonstrate the validity and applicability of the technique. The method is easy to implement and produces accurate results.

DOI: 10.1134/S0965542515090031

Keywords: Alpert multiwavelets; Fredholm integro-differential equations; thresholding; sparse matrix; GMRES method.

1. INTRODUCTION

In this paper we solve the system of Fredholm integro-differential equations of the form [1–3]

$$\sum_{j=1}^{n}\left(D_{ij}(x, u_j, u_j^{(1)}, \dots, u_j^{(n_{d_{ij}})}) + \int_a^b k_{ij}(x, t)G_i(u_1(t), \dots, u_n(t))dt\right) = f_i(x), \tag{1.1}$$

$$i = 1, \dots, n, \quad x \in [0, 1],$$

with boundary or the supplementary conditions

$$H_{rj}(u_j^{(0)}(x_\xi), \dots, u_j^{(n_{d_j})}(x_\xi)) = d_{r'},$$

$$j = 1, \dots, n, \quad n_{d_j} = \max_{1 \le i \le n} n_{d_{ij}}, \quad r' = \sum_{j=1}^{n} n_{d_j}, \quad x_\xi \in [0, 1], \tag{1.2}$$

where the functions $k_{ij}(x, t)$ and $f_i(x)$ are analytic functions on the interval [0, 1], G_i are the linear combination of $u_1(x)$, ..., $u_n(x)$ and H_{rj} are the linear combination of $u_j^{(0)}(x_\xi)$, ..., $u_j^{(n_{d_j})}(x_\xi)$. In addition for $i, j = 1, \dots, n$, we suppose the functions

$$D_{ij}(x, u_j, u_j^{(1)}, \dots, u_j^{(n_{d_{ij}})})$$

are analytic in terms of $u_j, u_j^{(1)}, \dots, u_j^{(n_{d_{ij}})}$, as

$$D_{ij}(x, u_j, u_j^{(1)}, \dots, u_j^{(n_{d_{ij}})}) = \sum_{l=0}^{n_{d_{ij}}} \mu_{jl} u_j^{(l)}(x),$$

where μ_{jl} are constants.

[1] The article is published in the original.

The concepts of integro-differential equations have received much attention in recent years. Several numerical methods were used such as the cubic spline method [4], Petrov-Galerkin method [5], the Tau method [6], spline wavelet method, [7], Legendre wavelet method, [8, 9], the Chebyshev and Taylor collocation method, [10, 11].

Systems of integro-differential equations have a strong physical background and many practical applications in scientific fields such as population and polymer rheology [12, 13]. Recently Abbasbandy and Taati [14] have introduced the operational Tau method to solve a system of nonlinear Volterra integro-differential equations with a nonlinear differential part. A simple operational approach, using the Adomian decomposition method, has been proposed for the numerical solution of systems of nonlinear Volterra integro-differential equations in [15]. This method leads to a system of linear algebraic equations. The operational approach to the Tau method is used for the numerical solution of a nonlinear Fredholm integro-differential equations system in [16]. A Sinc-collocation method is considered by Zarebnia et al. [17] to solve system of nonlinear second-order integro-differential equations. Dehghan et al. used Legendre multiwavelet Galerkin method for weakly singular integro-differential equation in [8].

Wavelet theory is relatively new and an emerging area in mathematical research. It has been applied in a wide range of engineering disciplines, as wavelets have advantages over traditional Fourier transforms in accurately approximating functions that have discontinuities and sharp peaks. Wavelets have been applied in signal analysis, numerical analysis, optimal control problems, multi-scale phenomena modeling and pattern recognition [18]. The smooth orthonormal basis obtained by the translation and dilation of a single function in a hierarchical fashion proved very useful to develop compression algorithms for signals and images up to a chosen threshold of relevant amplitudes [19—21].

In this paper we use Alpert multiwavelets [20, 22] to solve Eq. (1.1) with boundary or the supplementary conditions (1.2). The outline of this paper is as follows. In Section 2, we describe the basic formulation of the Alpert multiwavelet systems required for our subsequent development. In Section 3 the proposed method is used to approximate the solution of the problem. As a result a set of algebraic equations are formed and a solution of the considered problem is introduced. Restarted GMRES method is used to solve this system. In Section 4, we report our numerical findings and demonstrate the accuracy of the proposed numerical scheme.

2. ALPERT MULTIWAVELET SYSTEMS

2.1. Multiresolution Analysis

For functions $\phi^m \in L^2(R)$, $m = 0, \ldots, r-1$, let a reference subspace or sample space V_0 be generated as the L^2-closure of the linear span of the integer translates of m, namely:

$$V_0 = \text{clos}_{L^2}\langle\phi^m(.-k) : k \in Z\rangle, \quad m = 0, \ldots, r-1,$$

and consider other subspace

$$V_j = \text{clos}_{L^2}\langle\phi^m_{j,k} : k \in Z\rangle, \quad j \in Z, \quad m = 0, \ldots, r-1,$$

where $\phi^m_{j,k}\,\text{k} = \phi^m(2^jx - k), j, k \in Z, m = 0, \in, r - 1$.

Definition 1 [23]. Functions $\phi^m \in L^2(R)$, is said to generate a multiresolution analysis (MRA) if they generate a nested sequence of closed subspaces V_j that satisfy

$$\text{i)}\ \ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots,$$

$$\text{ii)}\ \text{clos}_{L^2}\Big(\bigcup_{j \in Z} V_j\Big) = L^2(R),$$

$$\text{iii)}\ \bigcup_{j \in Z} V_j = 0, \tag{2.1}$$

$$\text{iv)}\ f(x) \in V_j \Leftrightarrow f(x + 2^{-j}) \in V_j \Leftrightarrow f(2x) \in V_{j+1},$$

$$\text{v)}\ \{\phi^m(.-k)\}_{k \in Z}, \quad \text{form a Riesz basis of } V_0.$$

If ϕ^m generate an MRA, then ϕ^m are called scaling functions. In case the different integer translate of ϕ^m are orthogonal and where is with respect to the standard inner product $\langle f, g\rangle = \int_{-\infty}^{\infty} f(x)\overline{g(x)}dx$ for two functions in $L^2(R)$), denoted by $\phi^m(. - k) \perp \phi^{\tilde{m}}$ for $m \neq \tilde{m}$, $k \neq \tilde{k}$, the scaling functions are called an orthogonal scaling functions.

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

$$V_{j+1} = V_j \oplus W_j, \quad j \in Z,$$

here and in the following $\oplus$ denotes orthogonal sums.

This give rise to an orthogonal decomposition of $L^2(R)$, namely:

$$L^2(R) = \bigoplus_{j \in Z} W_j.$$

Definition 2 [23]. Functions $\psi\mu \in L^2(R)$ are called wavelets, if they generate the complementary orthogonal subspaces W_j of an MRA, i.e.,

$$W_j = \operatorname{clos}_{L^2} \langle \psi^m_{j,k}, k \in Z \rangle, \quad j \in Z, \quad m = 0, \dots, r-1,$$

where $\psi^m_{j,k} = \psi^m(2^j x - k), j, k \in Z$.

If, $\psi^m_{j,k} \perp \psi^{\tilde{m}}_{\tilde{j},\tilde{k}}$ for $j \neq \tilde{j}$, $m \neq \tilde{m}$ and $k \neq \tilde{k}$ if $< 2^{j/2} \psi^m_{j,k}$, $2^{\tilde{j}/2} \psi^{\tilde{m}}_{\tilde{j},\tilde{k}} >= \delta_{j,\tilde{j}} \delta_{k,\tilde{k}} \delta_{m,\tilde{m}}$ then ψ^m are called orthonormal wavelets.

Now we define Alpert scaling functions and its corresponding multiwavelets according to above MRA.

2.2. Construction of Scaling Functions

Suppose P_r is the Legendre polynomial of order r and r is any fixed nonnegative integer number and let τ_k for $k = 0, \dots, r - 1$ denote the roots of P_r. The interpolating scaling functions (*ISF*) are given by [8, 24, 25]

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

Where ω_k, $k = 0, \dots, r - 1$, are the Gauss-Legendre quadrature weights

$$\omega_k = \frac{2}{r \dot{P}_r(\tau_k) P_{r-1}(\tau_k)}$$

and $L_k(t)$, $k = 0, \dots, r-1$, are the lagrange interpolating polynomials [22]

$$L_k(t) = \prod_{i=0, i \neq k}^{r-1} \left(\frac{t - \tau_i}{\tau_k - \tau_i} \right)$$

that they have characterized by Kronecker property $L_k(\tau_i) = \delta_{ik}$ where

$$\delta_{ki} = \begin{cases} 1, & i = k, \\ 0, & i \neq k. \end{cases}$$

We can expand any polynomial g of degree less than r with the function $\phi^0, \dots, \phi^{r-1}$ that they formed an orthonormal basis on [0, 1)

$$g(t) = \sum_{k=0}^{r-1} d_k \phi^k(t)$$

where the coefficients are given by

$$d_k = \sqrt{\frac{\omega_k}{2}} g(\hat{\tau}_k), \quad k = 0, \dots, r-1,$$

and

$$\hat{\tau}_k = \frac{\tau_k + 1}{2}.$$

Let $\phi^k_{Jl}(t)$, $k = 0, \dots, r - 1$, $l = 0, \dots, 2^J - 1$, be obtained form $\phi^k(t)$ by dilation and translation

$$\phi_{Jl}^{k}(t) = 2^{(J/2)}\phi^{k}(2^{j}t-l) \tag{2.2}$$

where J is any fixed nonnegative integer number.

Note that we have the following orthonormality relation

$$\int_0^1 \phi_{Jl}^{(k)}(t)\phi_{Jl'}^{k'}(t)dt = \delta_{ll'}\delta_{kk'},$$

$$k, k' = 0, \ldots, r-1, \quad l, l' = 0, \ldots, 2^{j}-1,$$

2.3. Construction of Wavelets

The two-scale relations for the r-th order Alpert multiwavelets are in the form [20]:

$$\psi^{i}(x) = \sum_{j=0}^{r-1} h_{i,j}\phi^{j}(2x) + \sum_{j=0}^{r-1} h_{i,r+j+1}\phi^{j}(2x-1). \tag{2.3}$$

As we have $2r^2$ unknown coefficients $\{h\}$ in (2.3), we use the following $2r(r-1)$ vanishing moment conditions and $2r$ orthonormal conditions to determine them.

1. Vanishing moments

$$\int_0^1 \psi^{i}(x)x^{j} = 0, \quad \text{for } i = 0, 1, \ldots, r-1, \quad j = 0, 1, \ldots, i+r-1. \tag{2.4}$$

2. Orthonormality

$$\int_0^1 \psi^{i}(x)\psi^{j}(x) = \delta_{i,j}, \quad \text{for } i, j = 0, 1, \ldots, r-1. \tag{2.5}$$

2.4. Two Scale Relations

The representation of two scale relations is proposed for scaling functions and wavelets as

$$\phi^{k}(x) = \sum_{j=0}^{r-1} g_{k+1,j+1}^{0}\phi^{j}(2x) + g^{2}1_{k+1,j+1}\phi^{j}(2x-1),$$

$$\psi^{k}(x) = \sum_{j=0}^{r-1} h_{k+1,j+1}^{0}\phi^{j}(2x) + h_{k+1,j+1}^{1}\phi^{j}(2x-1).$$

By using the function $\phi^k(x)$ and $\psi^k(x)$ for $k = 0, \ldots, r-1$, we construct the filter coefficients $g_{i,j}^{l}$ and $h_{i,j}^{l} = 0, 1$. In these representation of two scale relation, four matrices $(r \times r)$ is used to show the filter coefficients $g_{i,j}^{l}$ and $h_{i,j}^{l}$, $l = 0, 1$, as

$$G^{0} = \begin{bmatrix} g_{11}^{0} & \ldots & g_{1r}^{0} \\ \vdots & & \vdots \\ g_{r1}^{0} & \ldots & g_{rr}^{0} \end{bmatrix}, \quad G^{1} = \begin{bmatrix} g_{11}^{1} & \ldots & g_{1r}^{1} \\ \vdots & & \vdots \\ g_{r1}^{1} & \ldots & g_{rr}^{1} \end{bmatrix},$$

$$H^{0} = \begin{bmatrix} h_{11}^{0} & \ldots & h_{1r}^{0} \\ \vdots & & \vdots \\ h_{r1}^{0} & \ldots & h_{rr}^{0} \end{bmatrix}, \quad H^{1} = \begin{bmatrix} h_{11}^{1} & \ldots & h_{1r}^{1} \\ \vdots & & \vdots \\ h_{r1}^{1} & \ldots & h_{rr}^{1} \end{bmatrix},$$

The matrices G^0 and G^1 consist of the filter coefficients of two scale relation for scaling functions and their components are given by following equations

$$g^0_{k,k'} = \sqrt{w_{k'}}\phi^k\left(\frac{\hat{\tau}_{k'}}{2}\right), \tag{2.6}$$

$$g^1_{k,k'} = \sqrt{w_{k'}}\phi^k\left(\frac{\hat{\tau}_{k'}+1}{2}\right). \tag{2.7}$$

These equations are obtained by using the interpolating property of scaling functions.

In general, the two scale relation for the neighbour scales J and J + 1 is given by the following matrix form

$$\Phi^r_J(x) = G_J\Phi^r_{J+1}(x), \tag{2.8}$$

where G^J define the transform matrix between two neighbour scales for scaling functions and is getting by

$$G_J = \begin{bmatrix} G & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & G \end{bmatrix}_{r2^J,\, r2^{J+1}}, \tag{2.9}$$

where $\Phi^r_J(x)$ consist of $r2^J$ bases for V^t_J and $G = [G^0G^1]$. We note that the filter coefficients of two scale relation for wavelets is constructed in Subsection 2.3.

Hence the wavelet transform matrix [26] between ψ^r_J and Φ^r_J is obtained as

$$\psi^r_J = T_J\Phi^r_J, \tag{2.10}$$

where T_J is a $(r2^J, r2^J)$ matrix which are obtained by the following scheme. Suppose that $H = [H^0H^1]$ and

$$H_J = \begin{bmatrix} H & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & H \end{bmatrix}_{r2^J,\, r2^{J+1}}, \tag{2.11}$$

By using these matrices, we get

$$T_J = \begin{bmatrix} \frac{1}{2^J}(G_0 \times G_1 \times \dots \times G_{J-1}) \\ \frac{1}{2^J}(H_0 \times G_1 \times \dots \times G_{J-1}) \\ \frac{1}{2^{J-1}}(H_1 \times G_2 \times \dots \times G_{J-1}) \\ \vdots \\ \frac{1}{2^2}(H_{J-2} \times G_{J-1}) \\ \frac{1}{2}H_{J-1} \end{bmatrix}. \tag{2.12}$$

2.5. Function Approximation

It can be verified that $V_j \oplus W_j = V_{j+1}$, thus we can write $V_j = V_0 \oplus (\oplus_{i=0}^{j-1} W_i)$ and we have two kind of basis sets for $J \in N$

$$\Phi^r_J(x) = [\phi^0_{J,0}(x), \dots, \phi^{r-1}_{J,0}(x), | \dots, \phi^0_{J,(2^J-1)}(x), \dots, \phi^{r-1}_{J,(2^J-1)}(x)]^{\mathrm{T}}, \tag{2.13}$$

$$\psi_J^r(x) = [\phi_{0,0}^0(x), \ldots, \phi_{0,0}^{r-1}(x), |\psi_{0,0}^0(x), \ldots, \psi_{0,0}^{r-1}(x)|, \ldots \\ \ldots, |\psi_{J-1,0}^0(x), \ldots, \psi_{J-1,0}^{r-1}(x)|, \ldots, \psi_{J-1,2^{J-1}-1}^0(x), \ldots, \psi_{J-1,2^{J-1}-1}^{r-1}(x)]^{\mathrm{T}}. \tag{2.14}$$

Now any function $f(x)$ on [0, 1] can be approximated using scaling functions as

$$f(x) \approx P_J^r f = \sum_{k=0}^{r-1} \sum_{l=0}^{2^J-1} c_{J,l}^k \phi_{J,l}^k(x) = C^{\mathrm{T}} \Phi_J^r(x), \tag{2.15}$$

and the corresponding wavelet functions as

$$f(x) \approx P_J^r f = \sum_{k=0}^{r-1} \left\{ c_{0,0}^k \phi_{0,0}^k + \sum_{j=0}^{J-1} \sum_{l=0}^{2^J-1} d_{j,l}^k \psi_{j,l}^k(x) \right\} = D^{\mathrm{T}} \Psi_J^r(x), \tag{2.16}$$

where

$$c_{J,l}^k = \int_0^1 f(x) \phi_{J,l}^k(x) dx = \int_{h_l}^{h_{l+1}} f(t) \phi_{Jl}^k(t) dt, \tag{2.17}$$

and

$$h_l = \frac{l}{2^J}, \quad l = 0, \ldots, 2^J - 1.$$

These coefficients may be computed using Gauss-Legendre quadrature [24, 27].

$$c_{Jl}^k = 2^{-J/2} \sqrt{\frac{\omega_k}{2}} f(2^{-J}(\hat{\tau}_k + 1)), \quad k = 0, \ldots, r-1, \quad l = 0, \ldots, 2^J - 1. \tag{2.18}$$

Lemma. *Suppose that the function f : [0, 1] ⟶ R is r times continuously differentiable.*

Then $P_J^r f$ approximates f with mean error bounded as follow [20]:

$$\|P_J^r f - f\| \le 2^{-Jr} \frac{2}{4^r r!} \sup_{x \in [0,1]} |f^{(r)}(x)|,$$

Also a function $p(x, t)$ of two independent variables for $0 \le x \le 1$, and $0 \le t \le 1$, may be expanded in terms of interpolating scaling functions as

$$p(x, t) \approx \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} \Phi_i(x) \Phi_j(t) = \Phi_J^{r\mathrm{T}}(x) P \Phi_J^r(t), \tag{3.1}$$

where P is an $N \times N$ matrix as

$$P = \begin{bmatrix} p_{11} & \ldots & p_{1N} \\ \vdots & & \vdots \\ p_{N1} & \ldots & p_{NN} \end{bmatrix},$$

with $N = r2^J$ and

$$p_{i,j} = \int_0^1 \int_0^1 p(x, t) \Phi_i(t) \Phi_j(t) dt dx.$$

We used (2.18) to get

$$p_{i,j} = 2^{-J} \sqrt{\frac{\omega_k}{2}} \sqrt{\frac{\omega_{k'}}{2}} p(2^{-J}(\hat{\tau}_k + 1), 2^{-J}(\hat{\tau}_{k'} + l')), \tag{3.2}$$

where $i = rl + (k + 1)$, $j = rl' + (k' + 1)$, $k, k' = 0, \ldots, r - 1$ and $l, l' = 0, \ldots, 2^J - 1$.

By using Eq. (2.10), the elements of matrix D in Eq. (2.16) are obtained as

$$D^{\mathrm{T}} = C^{\mathrm{T}} T_J^{-1}. \tag{3.3}$$

Also the function $p(x, t)$ is represented by using wavelets in the same manner

$$\hat{P} = T_J^{-1^{\mathrm{T}}} P T_J^{-1}, \tag{3.4}$$

where

$$p(x, t) \approx \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{p}_{ij} \Psi_i^r(x) \Psi_j^r(t) = \Psi_J^{r\mathrm{T}}(x) \hat{P} \Psi_J^r(t),$$

and D and C are $(m \times 1)$ vectors with $m = r2^J$ given by

$$D = [c_{0,0}^0, \ldots, c_{0,0}^r | d_{0,0}^0, \ldots, d_{0,0}^r | \ldots | d_{J-1,0}^0, \ldots, d_{J-1,0}^r |, \ldots, d_{J-1,2^{J-1}-1}^0, \ldots, d_{J-1,2^{J-1}-1}^r]^{\mathrm{T}}, \tag{3.5}$$

$$C = [c_{J,0}^0, \ldots, c_{J,0}^r | \ldots | c_{J,2^J-1}^0, \ldots, c_{J,2^J-1}^r]^{\mathrm{T}}. \tag{3.6}$$

3.1. The Operational Matrix of Integration

The integral of vectors $\Psi_J^r(x)$ and $\Phi_J^r(x)$ can be expressed as

$$\int_0^x \Psi_J^r(t) dt \approx I_\psi \Psi_J^r(x), \tag{3.7}$$

$$\int_0^x \Phi_J^r(t) dt \approx I_\phi \Phi_J^r(x), \tag{3.8}$$

where I_ϕ and I_ψ are $(N \times N)$ operational matrices of integration for Alpert scaling functions and multiwavelets respectively. The matrix I_ψ can be obtained by the following process [27]. Using Eq. (3.8) we have

$$\int_0^x \Phi_J^r(t) dt \approx \sum_{k'=0}^{r-1} \sum_{l'=0}^{2^J-1} [I_\phi]_{lr+(k+1), l'r+(k'+1)} \phi_{Jl'}^{k'}(x), \tag{3.9}$$

$$k = 0, \ldots, r-1, \quad l = 0, \ldots, 2^J - 1.$$

Now we use Eq. (2.18) to obtain

$$[I_\phi]_{lr+(k+1), l'r+(k'+1)} = 2^{\frac{-J}{2}} \sqrt{\frac{\omega_{k'}}{2}} \int_0^{2^{-J}(\hat{\tau}_{k'}+l')} \phi_{Jl}^k(t) dt = 2^{\frac{-J}{2}} \sqrt{\frac{\omega_{k'}}{2}} \int_{\frac{l}{2^J}}^{2^{-J}(\hat{\tau}_{k'}+l')} \phi_{Jl}^k(t) dt. \tag{3.10}$$

To find the entries of matrix I_ϕ we assume the following three cases.

Case 1. $l' < l$. The support of ϕ_{Jl}^k is $\left[\frac{l}{2^J}, \frac{l+1}{2^J}\right]$, and $2^{-J}(\hat{\tau}_{k'} + l') < \frac{l}{2^J}$. Thus we get

$$[I_\phi]_{lr+(k+1), l'r+(k'+1)} = 0. \tag{3.11}$$

Case 2. $l' = l$. Changing the variable $2^J t - l = \hat{\tau}_k x$ we have

$$[I_\phi]_{lr+(k+1), l'r+(k'+1)} = 2^{\frac{-J}{2}} \sqrt{\frac{\omega_{k'}}{2}} \int_0^{\hat{\tau}_k} \phi_{Jl}^k(t)(dt).$$

These coefficients may be computed using the Gauss-Legendre quadrature as

$$[I_\phi]_{lr+(k+1), l'r+(k'+1)} = 2^{-J} \sqrt{\frac{\omega_{k'}}{2}} \hat{\tau}_{k'} \sum_{i=0}^{r-1} \frac{\omega_{k'}}{2} \phi^k(\hat{\tau}_{k'} \hat{\tau}_i). \tag{3.12}$$

Case 3. $l' > l$. Again the support of ϕ_{Jl}^{k} is $[\left[\frac{l}{2^J}, \frac{l+1}{2^J}\right]$ and $2^{-J}(\hat{\tau}_{k'} + l') > \frac{l'}{2^J} > l + \frac{l+1}{2^J}$. Thus we obtain

$$[I_\phi]_{lr+(k+1),\, l'r+(k'+1)} = 2^{\frac{-J}{2}}\sqrt{\frac{\omega_{k'}}{2}}\int\limits_{\frac{l}{2^J}}^{\frac{l+1}{2^J}}\phi_{Jl}^{k}(t)dt = 2^{\frac{-J}{2}}\sqrt{\frac{\omega_{k'}}{2}}\int\limits_{0}^{1}\phi^{k}(t)dt = 2^{-J}\sqrt{\frac{\omega_k}{2}}\sqrt{\frac{\omega_{k'}}{2}}. \tag{3.13}$$

Now we use these three cases to obtain the operational matrix of integration as

$$I_\phi = 2^{-J}\begin{bmatrix} M & P & \dots & \dots & P & P \\ & M & P & \dots & P & P \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & \vdots \\ & & & & M & P \\ & & & & & M \end{bmatrix},$$

where M and P are $r \times r$ matrices which can be obtained by the following equations:

$$[M]_{k+1,\, k'+1} = \sqrt{\frac{\omega_{k'}}{2}}\hat{\tau}_{k'}\sum_{i=0}^{r-1}\frac{\omega_{k'}}{2}\phi^{k}(\hat{\tau}_{k'}\hat{\tau}_i), \quad k, k' = 0, 1, \dots, r-1,$$

$$[P]_{k+1,\, k'+1} = \sqrt{\frac{\omega_k}{2}}\sqrt{\frac{\omega_{k'}}{2}}, \quad k, k' = 0, 1, \dots, r-1,$$

Using Eqs. (2.10), (3.7) and (3.8) we get

$$\int\limits_0^x \Psi_J^r(t)dt = T_J\int\limits_0^x \Phi_J^r(t)dt = T_J I_\phi \Phi_J^r(x) = T_J I_\phi T_J^{-1}\Psi_J^r(x), \tag{3.14}$$

comparing Eqs. (3.7) and (3.14) we get

$$I_\psi = T_J I_\phi T_J^{-1}, \tag{3.15}$$

4. DESCRIPTION OF NUMERICAL METHOD

In this section, we solve the system of integro-differential equations of the form (1.1) with boundary value or supplementary conditions (1.2), by using Alpert multiwavelets.

We know that $u_j^{n_{d_j}}$ for $j = 1, \dots, n$ is the largest derivative of u_j. Assume that we expand $u_j^{n_{d_j}}$ using Alpert multiwavelets as

$$u_j^{n_{d_j}}(x) \approx U_j^{\mathrm{T}}\Psi_J^r(x), \quad j = 1, \dots, n, \tag{4.1}$$

where U_j is an $(N \times 1)$ unknown vector.

By integrating from both sides of Eq. (4.1), and using Eq. (3.7), we get

$$\int\limits_0^x u_j^{n_{d_j}}(t)dt \approx U_j^{\mathrm{T}}\int\limits_0^x \Psi_J^r(t)dt, \quad u_j^{n_{d_j}-1}(x) - u_j^{n_{d_j}-1}(0) \approx U_j^{\mathrm{T}} I_\psi \Psi_J^r(x), \quad j = 1, \dots, n. \tag{4.2}$$

Let $\lambda_{j1} = u_j^{n_{d_j}-1}(0)$, $k = 1, \dots, n_{d_j}$, and $J = 1, \dots, n$, so we have

$$u_j^{n_{d_j}-1}(x) \approx U_j^{\mathrm{T}} I_\psi \Psi_J^r(x) + \lambda_{j1}, \quad j = 1, \dots, n. \tag{4.3}$$

Again by integrating from both sides of Eq. (4.3), we get

$$u_j^{n_{d_j}-1}(x) \approx U_j^{\mathrm{T}} I_\psi \int\limits_0^x \Psi_J^r(t) + \int\limits_0^x \lambda_{j1} dt, \quad j = 1, \dots, n,$$

or

$$u_j^{n_{d_j}-2}(x) - u_j^{n_{d_j}-2}(0) \approx U_j^{\mathrm{T}} I_\psi^2 \Psi_J^r(x) + \lambda_{j1} x, \quad j = 1, \dots, n, \tag{4.4}$$

so

$$u_j^{n_{d_j}-2}(x) \approx U_j^{\mathrm{T}} I_\psi^2 \Psi_J^r(x) + \lambda_{j1} x + \lambda_{j2}, \quad j = 1, \dots, n. \tag{4.5}$$

By continuing this process, we get

$$u_j^{n_{d_j}-\alpha}(x) \approx U_j^{\mathrm{T}} I_\psi^\alpha \Psi_J^r(x) + \sum_{l=1}^{\alpha} \lambda_{jl} \frac{x^{\alpha-l}}{(\alpha-l)!}, \quad 1 \le \alpha \le n_{d_j}, \tag{4.6}$$

and

$$u_j(x) \approx U_j^{\mathrm{T}} I_\psi^{n_{d_j}} \Psi_J^r(x) + \sum_{l=1}^{n_{d_j}} \lambda_{jl} \frac{x^{n_{d_j}-l}}{(n_{d_j}-l)!}, \quad j = 1, \dots, n. \tag{4.7}$$

Now we approximate $\frac{x^{\alpha-l}}{(\alpha-l)!}$ in the Eq. (4.6) as

$$\frac{x^{\alpha-l}}{(\alpha-l)!} \approx \bar{X}_{jl}^{\mathrm{T}} \Phi_j^r(x) = \bar{X}_{jl}^{\mathrm{T}} T_J^{-1} \Psi_j^r(x), \quad l = 1, \dots, \alpha, \quad 1 \le \alpha \le n_{d_j}. \tag{4.8}$$

Where $\bar{X}_{jl}$, $j = 1, \dots, n$, $I = 1, \dots, \alpha$ are the $(m \times 1)$ vectors which their entries can be found using Eq. (2.17). Note that we are not required to approximate all of the $\frac{x^{\alpha-l}}{(\alpha-l)!}$ because these functions repeat for every $j = 1, \dots, n$. Let $X_{il}^{\mathrm{T}} = \bar{X}_{jl}^{\mathrm{T}} T_J^{-1}$. Using Eq. (4.8), in Eq. (4.6), we get

$$u_j^{n_{d_j}-\alpha}(x) \approx U_j^{\mathrm{T}} I_\psi^\alpha \Psi_J^r(x) + \sum_{l=1}^{\alpha} \lambda_{jl} X_{il}^{\mathrm{T}} \Psi_J^r(x), \quad 1 \le \alpha \le n_{d_j}. \tag{4.9}$$

The functions $k_{ij}(x, t)$ and $f_i(x)$ in Eq. (1.1) can be approximated as

$$k_{ij}(x,t) \approx \Phi_J^r(t) \bar{K}_{ij} \Phi_J^{r\mathrm{T}}(x) = \Psi_J^{r\mathrm{T}}(t) \underbrace{T_J^{-1\mathrm{T}} \bar{K}_{ij} T_J^{-1}}_{k_{ij}} \Psi_J^r = \Psi_J^{r\mathrm{T}}(t) K_{ij} \Psi_J^r, \tag{4.10}$$

and

$$f_i(x) = \bar{F}_i^{\mathrm{T}} \Phi_J^r(x) = \underbrace{\bar{F}_i^{\mathrm{T}} T_J^{-1}}_{F_i^{\mathrm{T}}} \Psi_J^r = F_i^{\mathrm{T}} \Psi_J^r, \tag{4.11}$$

where $\bar{K}_{ij}$, $i, j = 1, \dots, n$ are $m \times m$ matrices and $\bar{F}_i$, $i = 1, \dots, n$ are a $m \times 1$ vector, which can be obtained as

$$(K_{ij})_{\iota,\sigma} = \int_0^1 \left[\int_0^1 k_{ij}(x,t) (\Phi_J)_{\sigma,1}^r(t)\, dt \right] (\Phi_J)_{\iota,1}^r(x)\, dx = 2^{-J} \sqrt{\frac{\omega_k}{2}} \sqrt{\frac{\omega_{k'}}{2}} k_{ij}(2^{-J}(\hat{\tau}_k + l), 2^{-J}(\hat{\tau}_{k'} + l')), \tag{4.12}$$

$$(F_i)_{\sigma,1} = \int_0^1 f_i(x) (\Phi_J)_{\sigma,1}^r(x)\, dx = 2^{\frac{-J}{2}} \sqrt{\frac{\omega_{k'}}{2}} f_j(2^{-J}(\hat{\tau}'_k + l')), \tag{4.13}$$

$$\iota = rl + (k+1), \quad \sigma = rl' + (k'+1), \quad l, l' = 0, \dots, 2^J - 1, \quad k, k' = 0, \dots, r-1.$$

Applying Eqs. (4.9)–(4.11) in Eq. (1.1), we get

$$\begin{aligned} &\sum_{j=1}^{n} D_{ij} \left(x, \left(U_j^{\mathrm{T}} I_\psi^{n_{d_j}} \Psi_J^r + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r \right), \dots, \left(U_j^{\mathrm{T}} I_\psi \Psi_J^r + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r \right), U_j^{\mathrm{T}} \Psi_J^r \right) \\ &+ \int_0^1 G_i \left(\left(U_1^{\mathrm{T}} I_\psi^{n_{d_1}} \Psi_J^r(t) + \sum_{l=1}^{n_{d_1}} \lambda_{1l} X_{1l}^{\mathrm{T}} \Psi_J^r(t) \right), \dots, \left(U_n^{\mathrm{T}} I_\psi^{n_{d_n}} \Psi_J^r(t) + \sum_{l=1}^{n_{d_n}} \lambda_{nl} X_{nl}^{\mathrm{T}} \Psi_J^r(t) \right) \right) \Psi_J^{r\mathrm{T}}(t) K_{ij} \Psi_J^r(x)\, dt = F_i^{\mathrm{T}} \Psi_J^r(x), \end{aligned} \tag{4.14}$$

or

$$\begin{aligned}&\sum_{j=1}^{n} D_{ij}\left(x, \left(U_j^{\mathrm{T}} I_{\psi}^{n_{d_j}} \Psi_J^r + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), \ldots, \left(U_j^{\mathrm{T}} I_{\psi} \Psi_J^r + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), U_j^{\mathrm{T}} \Psi_J^r\right) \\ &+ G_i\left(\left(U_1^{\mathrm{T}} I_{\psi}^{n_{d_1}} \int_0^1 \Psi_J^r(t) \Psi_J^{r\mathrm{T}}(t) + \sum_{l=1}^{n_{d_1}} \lambda_{1l} X_{1l}^{\mathrm{T}} \int_0^1 \Psi_J^r(t) \Psi_J^{r\mathrm{T}}(t)\right), \ldots, \left(U_n^{\mathrm{T}} I_{\psi}^{n_{d_n}} \int_0^1 \Psi_J^r(t) \Psi_J^{r\mathrm{T}}(t) + \sum_{l=1}^{n_{d_n}} \lambda_{nl} X_{nl}^{\mathrm{T}} \int_0^1 \Psi_J^r(t) \Psi_J^{r\mathrm{T}}(t)\right)\right) \\ &\times K_{ij} \Psi_J^r(x) dt = F_i^{\mathrm{T}} \Psi_J^r(x),\end{aligned} \tag{4.15}$$

The orthonormality of these multiwavelets implies

$$\begin{aligned}&\sum_{j=1}^{n} D_{ij}\left(x, \left(U_j^{\mathrm{T}} I_{\psi}^{n_{d_j}} \Psi_J^r + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), \ldots, \left(U_j^{\mathrm{T}} I_{\psi} \Psi_J^r + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), U_j^{\mathrm{T}} \Psi_J^r\right) \\ &+ G_i\left(\left(U_1^{\mathrm{T}} I_{\psi}^{n_{d_1}} + \sum_{l=1}^{n_{d_1}} \lambda_{1l} X_{1l}^{\mathrm{T}}\right), \ldots, \left(U_n^{\mathrm{T}} I_{\psi}^{n_{d_n}} + \sum_{l=1}^{n_{d_n}} \lambda_{nl} X_{nl}^{\mathrm{T}}\right)\right) K_{ij} \Psi_J^r(x) dt = F_i^{\mathrm{T}} \Psi_J^r(x).\end{aligned} \tag{4.16}$$

Using wavelet—Galerkin method, we obtain

$$\begin{aligned}&\int_0^1 \left(\sum_{j=1}^{n} D_{ij}\left(x, \left(U_j^{\mathrm{T}} I_{\psi}^{n_{d_j}} \Psi_J^r + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), \ldots, \left(U_j^{\mathrm{T}} I_{\psi} \Psi_J^r + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r\right), U_j^{\mathrm{T}} \Psi_J^r\right)\right. \\ &\left. + G_i\left(\left(U_1^{\mathrm{T}} I_{\psi}^{n_{d_1}} + \sum_{l=1}^{n_{d_1}} \lambda_{1l} X_{1l}^{\mathrm{T}}\right), \ldots, \left(U_n^{\mathrm{T}} I_{\psi}^{n_{d_n}} + \sum_{l=1}^{n_{d_n}} \lambda_{nl} X_{nl}^{\mathrm{T}}\right)\right) K_{ij} \Psi_J^r(x) - F_i^{\mathrm{T}} \Psi_J^r(x)\right) \psi_{j',l}^{k}(x) dx = 0, \\ &j' = -1, \ldots, J-1, \quad k = 0, \ldots, r-1, \quad l = 0, \ldots, (2^{j'} - 1),\end{aligned} \tag{4.17}$$

where we assume $\psi_{-1,l}^{k} = \phi_{0,0}^{k}$. So we have

$$\begin{aligned}E_i &= \sum_{j=1}^{n} D_{ij}\left(\left(U_j^{\mathrm{T}} I_{\psi}^{n_{d_j}} + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}}\right), \ldots, \left(U_j^{\mathrm{T}} I_{\psi} + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}}\right), U_j^{\mathrm{T}}\right) \\ &+ G_i\left(\left(U_1^{\mathrm{T}} I_{\psi}^{n_{d_1}} + \sum_{l=1}^{n_{d_1}} \lambda_{1l} X_{1l}^{\mathrm{T}}\right), \ldots, \left(U_n^{\mathrm{T}} I_{\psi}^{n_{d_n}} + \sum_{l=1}^{n_{d_n}} \lambda_{nl} X_{nl}^{\mathrm{T}}\right)\right) K_{ij} - F_i^{\mathrm{T}} = 0, \\ &i = 1, \ldots, n.\end{aligned} \tag{4.18}$$

Applying boundary or supplementary conditions (1.2), we get

$$H_{rj}\left(\left(U_j^{\mathrm{T}} I_{\psi}^{n_{d_j}} \Psi_J^r(x_{\xi}) + \sum_{l=1}^{n_{d_j}} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r(x_{\xi})\right), \ldots, \left(U_j^{\mathrm{T}} I_{\psi}^{1} \Psi_J^r(x_{\xi}) + \sum_{l=1}^{1} \lambda_{jl} X_{jl}^{\mathrm{T}} \Psi_J^r(x_{\xi})\right), U_j^{\mathrm{T}} \Psi_J^r(x_{\xi})\right) = d_{r'}, \tag{4.19}$$

where we have r unknown elements with r equations. Thus we obtain the square system as

$$\begin{aligned}\Gamma U &= D, \\ U &= [U_1^{\mathrm{T}}, \ldots, U_n^{\mathrm{T}}, |\lambda_{11}, \ldots, \lambda_{1n_{d_1}}|, \ldots, |\lambda_{n1}, \ldots, \lambda_{n, n_{d_n}}]^{\mathrm{T}}\end{aligned} \tag{4.20}$$

also D is a $(nN + r' \times 1)$ vector that $D_{i\kappa,1}$, $\kappa = 1, \ldots, N$, $i = 1, \ldots, n$ are constant elements of E_i and $D_{\kappa,1} = d_{\kappa}$, $\kappa = nN + 1, \ldots, nN + r'$.

4.1. Truncation of Coefficients

Let us assume that a function $f(x) \in L^2[0, 1]$ is approximated by its projection on some scale $J + 1$, so that $\|f - P_{J+1}^r f\|_2 \le \epsilon \|f\|_2$. Where ϵ is the desired accuracy of the approximation. We now seek to approximate f on the next coarse scale, J and consider the resulting error introduced by the coarsening. We divide [0, 1] into 2^J subintervals and give the error on each subintervals. By using orthogonality the error on some subintervals l is $\|P^r_{J+1}f - P^r_J f_l\|_2 = \|d^J_l\|_2$. Therefore we have [20]

$$\left\|P^r_{J+1}f - P^r_J f\right\|_2 \le \epsilon \left\|P^r_{J+1}f\right\|_2. \tag{4.21}$$

We truncate the $J + 1$-scale representation when

$$\left\|d^J_l\right\|_2 \le \epsilon 2^{-J/2}\left\|P^r_{J+1}f\right\|_2. \tag{4.22}$$

By using this equation, set to zero all coefficients which satisfy that constraint. By this method, we can reduce the number of coefficients in the representation, while maintaining the specified accuracy ϵ, also, the linear system of equation (4.20) is reduced to a new sparse linear system.

4.2. Restarted GMRES Method

Recently, a method called GMRES has received a considerable attention by numerical linear algebraists in the context of solving large and sparse linear systems. The GMRES method by Saad and Shultz [28] is one of the popular iterative methods to solve 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 [29] for this purpose, firstly we represent Arnoldi's algorithm. Arnoldi's procedure is an algorithm for building an orhogonal basis of the Krylov subspace κ_m. The N-th Krylov subspace define as follow

$$\kappa_m(\Gamma, v) = \operatorname{span}\{v_1, \Gamma v_1, \ldots, \Gamma^{m-1}v_1\}. \tag{4.23}$$

Next, Arnoldi and Restarted (GMRES) algorithms will be stated

Algorithm 4.1. *Arnoldi*

```
1. Choose a vector $v_1$, such that $\|v_1\|_2 = 1$
2. For $j = 1, 2, \ldots, m$ do:
3. compute $h_{i,j} = (\Gamma v_j, v_i)$ for $i = 1, 2, \ldots, j$
4. $w_j = (\Gamma v_j, v_i)$ for $i = 1, 2, \ldots, j$
5. $h_{j+1,j} = \|w_j\|_2$
6. If $h_{2j+1,j} = 0$ then stop
7. $v_{j+1} = w_j/h_{j+1,j}$
8. End do.
```

Where V_m is a $N \times m$ matrix with column vectors $v_1, \ldots, v_m$. Also $\overline{H}_m$ is a $(m + 1) \times m$ Hessenberg matrix whose nonzero entries $h_{i,j}$ are defined by Algorithm 1.

Algorithm 4.2. *Restarted* (*GMRES*)

```
1. compute $r_0 = D - \Gamma U_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$
2. Generate the Arnoldi basis and the matrix $\overline{H}_m$ using the Arnoldi algorithm starting with $v_1$
3. Compute $y_m$ the minimizer of $\|\beta e_1 - \overline{H}_m y\|_2$ and $U_m = U_0 + V_m y_m$
4. If satisfied, then stop, else set $U_0 = U_m$ and goto 1
```

5. NUMERICAL EXAMPLES

In this section, four numerical examples are presented to illustrate the validity and the merits of this technique. As mentioned before one main merit of this technique is the generation of a sparse matrix. This advantage is illustrated in example 1 and 3. Typically one "thresholds" the elements of a wavelet matrix by setting to zero all elements that are less than some small positive number ε multiplied by the largest matrix element namely

$$\epsilon 2^{-J/2}\left\|P^r_{J+1}f\right\|_2 = \varepsilon.$$

The matrix sparsity S_ε is then defined by [30]

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

where N_0 is the total number of elements and N_ε is the number of elements remaining after thresholding. We define the maximum error as $L^\infty = \max|u_{\text{ex}} - u_{\text{ap}}|$ to calculate the errors.

Table 1. Sparsity and L^∞ error

x	Threshold parameter (ε)	Sparsity (S_ε)	L^∞ error u_1	L^∞ error u_2
$J = 2$	0	0%	4.8×10^{-15}	2.0×10^{-15}
	10^{-14}	80.53%	2.6×10^{-4}	9.3×10^{-5}
	10^{-3}	84.04%	4.3×10^{-3}	2.1×10^{-3}
$J = 3$	0	0%	2.0×10^{-15}	1.0×10^{-15}
	10^{-14}	90.46%	2.6×10^{-4}	9.3×10^{-5}
	10^{-3}	93.01%	4.3×10^{-3}	2.1×10^{-3}
method in [1]	0	0%	3.0×10^{-14}	3.1×10^{-14}

Example 1. Consider the following system of integro-differential equations [1]

$$u_1''(x) + u_2'(x) + \int_0^1 2xt(u_1(t) - 3u_2(t))dt = f_1(x),$$

$$u_1'(x) + u_2''(x) + \int_0^1 3(2x + t^2)(u_1(t) - 2u_2(t))dt = f_2(x),$$

where

$$f_1(x) = \frac{3x}{10} + 3x^2 + 8, \quad f_2(x) = 21x + \frac{4}{5},$$

with supplementary conditions

$$u_1(0) + u_1'(0) = 1, \quad u_1(1) + u_1'(1) = 10,$$
$$u_2(0) + u_2'(0) = 1, \quad u_2(1) + u_2'(1) = 7.$$

The exact solution of this equation is

$$u_1(x) = 3x^2 + 1, \quad u_2(x) = x^3 + 2x - 1.$$

Table 1 shows the sparsity and L^∞ error for $r = 5$, $J = 2$, 3 and different values of thresholding parameter, using the presented method together with the method in [1]. Figure 1 shows the plot of the sparsity of the matrix of coefficient for $r = 5$ and $J = 3$ after thresholding.

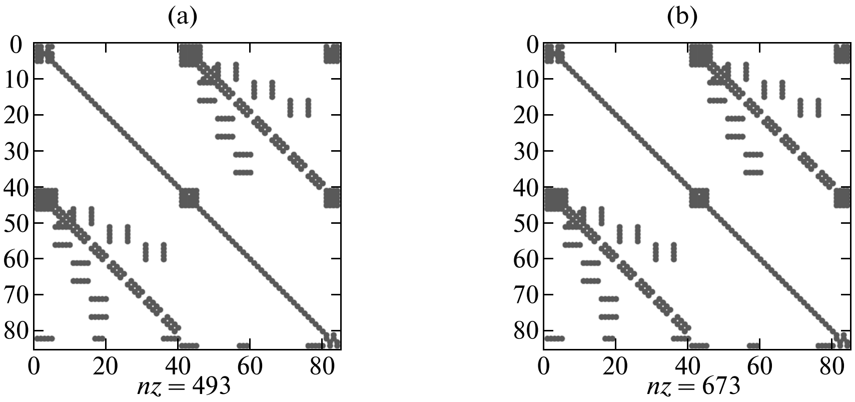

Fig. 1. Plots of sparse matrix after thresholding with $\varepsilon = 10^{-3}$ (a) $\varepsilon = 10^{-4}$ (b).

Table 2. Sparsity and L^∞ error

x	Threshold parameter (ε)	Sparsity (S_ε)	L^∞ error u_1	L^∞ error u_2
$J = 2$	0	0%	1.7×10^{-14}	2.2×10^{-14}
	10^{-5}	89.77%	1.3×10^{-7}	2.2×10^{-6}
	10^{-4}	91.27%	1.2×10^{-4}	1.2×10^{-5}
	10^{-3}	92.10%	1.0×10^{-4}	4.4×10^{-4}
$J = 3$	0	0%	1.6×10^{-14}	2.2×10^{-14}
	10^{-5}	96.63%	1.3×10^{-7}	2.2×10^{-6}
	10^{-4}	97.04%	1.2×10^{-4}	1.2×10^{-5}
	10^{-3}	97.26%	1.0×10^{-4}	4.4×10^{-4}

Example 2. Consider the following system of integro-differential equations [3]

$$\begin{aligned} u_1''(x) - \int_0^1 \left(\frac{u_1(t)}{3} + \frac{u_2(t)}{4}\right) dt &= \frac{8}{9}, \\ u_2''(x) - \int_0^1 x^2\left(\frac{u_1(t)}{6} - \frac{u_2(t)}{3}\right) dt &= 6x - \frac{x^2}{18}, \end{aligned} \tag{5.1}$$

for this example we have these conditions

$$u_1(0) = 0, \quad u_1'(0) = \frac{1}{3}, \quad u_2(0) = 1, \quad u_2'(0) = \frac{-1}{2},$$

also the exact solutions of problem (5.1) are

$$u_1(x) = \frac{x^2}{2} + \frac{x}{3}, \quad u_2(x) = x^3 - \frac{x}{2}.$$

Table 2 shows the sparsity and L^∞ error for $r = 5$, $J = 2$, 3 and different values of thresholding parameter, by using the present method. Figure 2 shows the plot of the sparsity of the matrix of coefficient for $r = 5$ and $J = 3$ after thresholding.

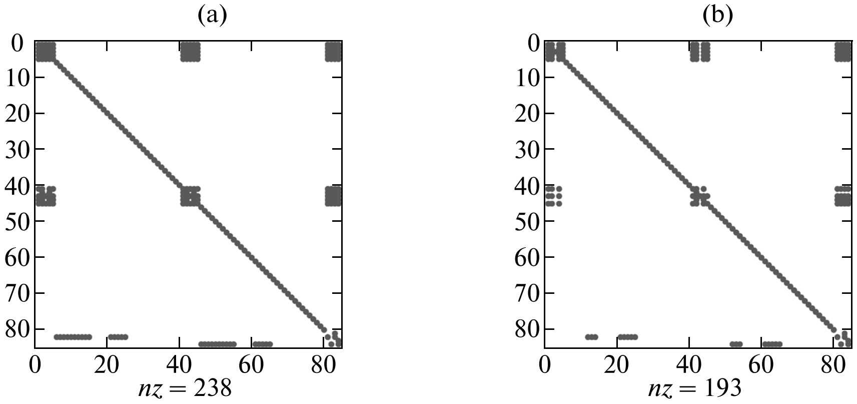

Fig. 2. Plots of sparse matrix after thresholding with $\varepsilon = 10^{-5}$ (a) $\varepsilon = 10^{-3}$ (b).

Table 3. Sparsity and L^∞ error

x	Threshold parameter (ε)	Sparsity (S_ε)	L^∞ error u_1	L^∞ error u_2
$J = 2$	0	0%	4.0×10^{-7}	1.3×10^{-7}
	10^{-4}	81.06%	2.6×10^{-5}	3.8×10^{-5}
	10^{-3}	86.75%	2.5×10^{-4}	8.6×10^{-5}
	10^{-2}	89.07%	1.7×10^{-2}	2.0×10^{-2}
$J = 3$	0	0%	1.2×10^{-8}	4.0×10^{-9}
	10^{-4}	92.31%	2.6×10^{-5}	3.8×10^{-5}
	10^{-3}	94.83%	2.5×10^{-4}	8.5×10^{-5}
	10^{-2}	96.27%	1.7×10^{-2}	2.0×10^{-2}

Example 3. Consider the following system of integro-differential equations

$$\begin{aligned} u_1'''(x) + 2u_2''(x) + u_3'(x) + \int_0^1 (x+t)^2(u_1(t) + u_2(t) + 2u_3(t))dt &= f_1(x), \\ u_1'(x) + u_2''(x) + u_3(x) + \int_0^1 (x^3 + t^2 + 5)(2u_1(t) + u_2(t) - u_3(t))dt &= f_2(x), \\ u_1'(x) + u_2'(x) + u_3'(x) + \int_0^1 (5x^2 + t^2 + 5)(u_1(t) - u_2(t) - u_3(t))dt &= f_3(x), \end{aligned} \tag{5.2}$$

with supplementary conditions

$$u_1(0) + u_1'(0) + u_1''(0) = 3, \quad u_1(1) + u_1'(1) + u_1''(1) = \sin(1) + 4\cos(1),$$

$$u_2(0) + u_2'(0) = 0, \quad u_2(1) + u_2'(1) = -\cos(1),$$

$$u_3(0) = 8, \quad u_3(1).$$

Here the forcing functions f_i, $i = 1, 2, 3$, are selected so that

$$u_1(x) = (1+x)\sin(x), \quad u_2(x) = (1-x)\cos(x), \quad u_3(x) = (1+x)^3,$$

are the exact solutions. Table 3 shows the sparsity and L^∞ error for $r = 5$, $J = 2$, 3 and different values of thresholding parameter, by using the presented method in the previous section. Figure 3 shows the plot of the sparsity of the matrix of coefficient for $r = 5$ and $J = 3$ after thresholding.

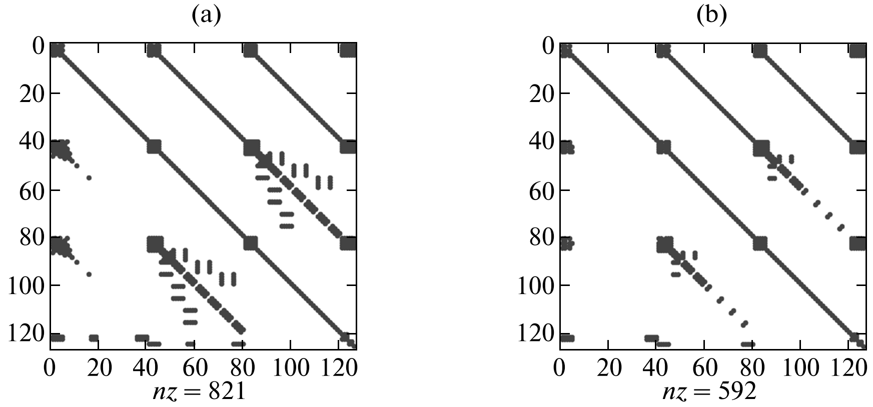

Fig. 3. Plots of sparse matrix after thresholding with $\varepsilon = 10^{-3}$ (a) $\varepsilon = 10^{-2}$ (b).

Table 4. L^∞ error for example 4

	Present method		Method in [2]	
	$N = 12$	$\varepsilon = 10^{-3}$	$N = 50$	$N = 100$
u_1	1.0×10^{-15}	1.0×10^{-15}	1.9×10^{-9}	2.5×10^{-13}
u_2	1.0×10^{-15}	1.0×10^{-15}	2.2×10^{-9}	2.4×10^{-13}
sparsity	0%	90.63%	0%	0%

Example 4. In this example, we consider the system of integral equations as [2]

$$\begin{cases} u_1(x) = \int\limits_0^1 \frac{x+t}{3}(u_1(t)+u_2(t))dt + \frac{x}{18} + \frac{17}{36}, \\ u_2(x) = \int\limits_0^1 xt(u_1(t)+u_2(t))dt + x^2 - \frac{19}{12}x + 1, \end{cases} \tag{5.3}$$

with exact solution $(u_1(t), u_2(t)) = (x + 1, x_2 + 1)$ given in [2].

Table 4 show that the L_∞ error for $r = 3$, $J = 2$ and thresholding parameter $\varepsilon = 10^{-3}$, using the present method together with the method in [2]. As seen in Table 4, the present method is superior to method in [2].

CONCLUSION

In this paper, we presented a numerical scheme for solving system of Fredholm integro-differential equations. This technique is based on Alpert multiwavelets and Galerkin—method. The method tested on several examples taken from the literature to observe the efficiency of the new technique. The numerical results given in the previous section demonstrate the accuracy of this scheme. The obtained results show that this techniques can solve the problem effectively. Because of interpolating property of scaling functions, this system of equations are solved rapidly by using this method. We used Matlab and Maple to solve this system of equations.

REFERENCES

1. J. Pour-Mahmoud and M. Y. Rahimi-Ardabili, S. Shahmorad, “Numerical solution of the system of Fredholm integro-differential equations by the Tau method,” Appl. Math. Comput. **168**, 465–478 (2005).
2. J. Rashidinia and M. Zarebnia, “Convergence of approximate solution of system of Fredholm integral equations,” J. Math. Anal. Appl. **333**, 1216–1227 (2007).
3. A. Davari and M. Khanian, “Solution of system of Fredholm integro-differential equations by Adomian decomposition method,” Austral. J. Basic Appl. Sci. **5** (12), 2356–2361 (2011).
4. P. Oja and D. Saveljeva, “Cubic spline collocation for Volterra integral equations,” Computing, **69**, 319–337 (2001).
5. B. Zhang, T. Lin, Y. Lin and M. Rao, “Defect correction and a posteriori error estimation of Petrov–Galerkin methods for nonlinear Volterra integro-differential equation,” Appl. Math. **45**, 241–263 (2000).
6. E. L. Ortiz and L. Samara, “An operational approach to the tau method for the numerical solution of nonlinear differential equations,” Computing **27**, 15-25 (1981).
7. M. Lakestani, M. Razzaghi, and M. Dehghan, “Semiorthogonal spline wavelets approximation for Fredholm integro-differential equations, Math. Probl. Eng., Article ID 96184, 1–12 (2006).
8. M. Lakestani, B. N. Saray, and M. Dehghan, “Numerical solution for the weakly singular Fredholm integro-differential equations using Legendre multiwavelets,” J. Comput. Appl. Math. **235**, 3291–3303 (2011).
9. M. Razzaghi and S. Yousefi, “Legendre wavelets method for the nonlinear Volterra Fredholm integral equations,” Math. Comput. Simul. **70**, 1–8 (2005).
10. M. Lakestani and M. Dehghan, “Numerical solution of fourth-order integro-differential equations using Chebyshev cardinal functions,” Int. J. Comput. Math. **87** (6), 1389–1394 (2010).
11. Y. Ren, Â. Zhang, and H. Qiao, “A simple Taylor-series expansion method for a class of second kind integral equations,” J. Comput. Appl. Math. **110**, 15–24 (1999).
12. P. Linz, *Analytical and Numerical Methods for Volterra Equations* (SIAM, Philadelphia, PA, 1985).

13. J. Abdul Jerri, *Introduction to Integral Equations with Applications* (Wiley, New York, 1999).
14. S. Abbasbandy and A. Taati, "Numerical solution of the system of nonlinear Volterra integro-differential equations with nonlinear differential part by the operational Tau method and error estimation," J. Comput. Appl. Math. **231**, 106–113 (2009).
15. A. Khani, M. Mohseni Moghadam, and S. Shahmorad, "Numerical solution of special class of system of nonlinear Volterra integro-differential equations by a simple high accuracy method," Bull. Iran. Math. Soc. **34** (2), 141–152 (2008).
16. G. Ebadi, M. Y. Rahimi, and S. Shahmorad, "Numerical solution of the system of nonlinear Fredholm integro-differential equations by the operational Tau method with an error estimation," Sci. Iran. **14**, 546–554 (2007).
17. M. Zarebnia and M. G. Ali Abadi, "Numerical solution of system of nonlinear second-order integro-differential equations," Comput. Math. Appl. **60**, 591–601 (2010).
18. R. Dai and J. E. Cochran Jr., "Wavelet collocation method for optimal control problems," J. Optim. Theory Appl. **143**, 265–287 (2009).
19. I. Daubechies, "Orthonormal bases of compactly supported wavelets," Commun. Pure Appl. Math. **41**, 909–996 (1988).
20. Â. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, "Adaptive solution of partial differential equations in multiwavelet bases," J. Comput. Phys. **182**, 149–190 (2002).
21. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, 1992).
22. M. Shamsi and M. Razzaghi, "Solution of Hallen's integral equation using multiwavelets," Comput. Phys. Commun. **168**, 187–197 (2005).
23. E. G. Quak and N. Weyrich, "Wavelet on the interval," Ed. by S. P. Singh (Toim) *Approximation Theory, Wavelets, and Applications* (Kluwer, 1995), pp. 247–283.
24. M. Shamsi and M. Razzaghi, "Numerical solution of the controlled Duffing oscillator by the interpolating scaling functions," Electromagn. Waves Appl. **18** (5), 691–705 (2004).
25. M. Lakestani and B. N. Saray, "Numerical solution of telegraph equation using interpolating scaling functions," Comput. Math. Appl. **60**, 1964–1972 (2010).
26. G. Hanwei, L. Kecheng, H. Jianguo, Y. Jiaxian, and L. Peiguo, "A novel wavelet transform matrix for efficient solutions of electromagnetic integral equations," *Proceedings of 1999 International Conference on Computational Electromagnetics and Its Applications, ICCEA'99* (1999).
27. M. Dehghan, Â. N. Saray, and Ì. Lakestani, "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. **55**, 1129–1142 (2012).
28. Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual method for solving nonsymmetric linear systems" SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986).
29. Y. Saad, *Iterative Methods for Sparse Linear Systems* (SIAM, Philadelphia, 2003).
30. J. C. Goswami, A. K. Chan, and Ñ. K. Chui, "On solving first-kind integral equations using wavelets on bounded interval," IEEE Trans. Antennas Propag. **43** (6), 614–622 (1995).