CrossMark

Applied Mathematics and Nonlinear Sciences 1(2) (2016) 529–546

Applied Mathematics and Nonlinear Sciences

http://journals.up4sciences.org

Modified Wavelet Full-Approximation Scheme for the Numerical Solution of Nonlinear Volterra integral and integro-differential Equations

S. C. Shiralashetti[†] and R. A. Mundewadi.

Department of Mathematics, Karnatak University, Dharwad 580003, Karnataka, INDIA

Submission Info

Communicated by Juan L.G. Guirao
Received 7th February 2016
Accepted 16th November 2016
Available online 16th November 2016

Abstract

In this paper, modified wavelet full-approximation scheme is introduced for the numerical solution of nonlinear Volterra integral and integro-differential equations. Wavelet Prolongation and Restriction operators are developed using Daubechies wavelet filter coefficients. Results show that the proposed scheme offers an efficient and good accuracy with faster convergence in less computation cost, which is justified through the error analysis and CPU time.

Keywords: Daubechies wavelet; filter coefficients; full-approximation scheme; wavelet full-approximation scheme; modified wavelet full-approximation scheme; nonlinear Volterra integral and integro-differential equations.
AMS 2010 codes: 65T60, 97N40, 65R20, 45D05.

1 Introduction

Integral equations arise in various fields of science and engineering. Nowadays integral equations and their applications is an important subject in applied mathematics. In some cases, it is difficult to solve them, especially analytically. There are many analytical approaches were introduced, such as A domian decomposition method, successive substitutions, Laplace transformation method, Picard's method, etc. [1]. The analytical approaches have limited applicability either because the analytical solutions of some of the problems those arise in application do not exists or because they are more time consuming. Therefore, many integral equations arising in mathematical modelling of physical world problem demands efficient numerical techniques. A number of numerical techniques such as Galerkin method [2], collocation method [2], Nystrom interpolation [3], etc. have been proposed by various authors. In numerical analysis, solving integral equations is reduced to solving a system of algebraic equations. Iterative techniques are available to solve a system of algebraic equations, such

[†]Corresponding author.
Email address: shiralashettisc@gmail.com

ISSN 2444-8656 doi:10.21042/AMNS.2016.2.00042

as Newton, method, Jacobi iterative method, Gauss-Seidel method, etc. In the past several years, considerable attention has been made for obtaining solutions of nonlinear integral equations. Nonlinearity is one of the interesting topics among the physicists, mathematicians, engineers, etc. Since most physical systems are inherently nonlinear in nature.

The full-approximation scheme (FAS) is largely applicable in increasing the efficiency of the iterative methods used to solve nonlinear system of algebraic equations. In the historical three decades the development of effective iterative solvers for nonlinear systems of algebraic equations has been a significant research topic in numerical analysis. Nowadays it is recognized that FAS iterative solvers are highly efficient for nonlinear differential equations introduced by Brandt [4]. For a detailed treatment of FAS is given in Briggs et al. [5]. An introduction of FAS is found in Hackbusch and Trottenberg [6], Wesseling [7] and Trottenberg et al. [8]. Many authors applied the FAS to some class of differential equations; Lubrecht [9], Venner and Lubrecht [10], Zargari [11] and others have significant contributions in EHL problems. Lee [12] has introduced a multigrid method for solving the nonlinear Urysohn integral equations. In this paper, we introduce the full-approximation scheme (FAS) for the numerical solution of nonlinear Volterra integral and integro-differential equations.

Wavelet based numerical methods are used for solving the system of equations with faster convergence and lower computational cost. Since 1991, the various types of wavelet based methods have been applied for the numerical solution of different kinds of integral equations, a detailed survey on these papers can be found in [13–15]. In recent years, wavelet analysis is gaining considerable attention in the numerical solution of differential equations. Recently, many authors (Leon [16], Bujurke et al. [17–19], Avudainayagam and Vani [32]) have worked on wavelet multigrid method for the numerical solution of differential equations. Wang et al. [21] have applied a fast wavelet multigrid algorithm for the solution of electromagnetic integral equations. In this paper, we introduce the modified wavelet full-approximation scheme (MWFAS) for the numerical solution of nonlinear Volterra integral and integro-differential equations. Thus, the proposed scheme can be applied to a wide range of science and engineering problems.

The organization of this paper is as follows. In Section 2, Daubechies wavelets is given. In Section 3 intergrid operators are discussed. In Section 4, method of solution is discussed. In Section 5, presents the numerical experiments and results are given. Finally, conclusion of the proposed work is given in Section 6.

2 Daubechies Wavelets

The refinement relation of scaling function $\phi(t)$ is given by,

$$\phi(t) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2t-k) \tag{1}$$

where $\phi(t)$ is normalized: $\int_{-\infty}^{\infty}\phi(t)dt = 1$.
Based on the scaling function $\phi(t)$, the wavelet function can be written as,
The refinement relation of wavelet function $\psi(t)$ is given by,

$$\psi(t) = \sqrt{2}\sum_{k=0}^{L-1} g_k \phi(2t-k) \tag{2}$$

where $\{h_k\}_{k\varepsilon Z}$ and $\{g_k\}_{k\varepsilon Z}$ are known respectively as the low and high pass filter coefficients, and they are related by $g_k = (-1)^k h_{L-k}$ for $k = 0,1,...,L-1$. where, L is an even integer, is the order of the wavelet.
The compactly supported wavelet bases were introduced by Daubechies [22]. They are an orthonormal bases for functions in $L^2(R)$. A family of orthogonal Daubechies wavelets with compactly supported property has been constructed by Daubechies in [23]. Due to excellent properties of orthogonality and minimum compact support, Daubechies wavelets can be useful and convenient in a wide variety of situations.

In this paper, we use Daubechies filter coefficients for $L=4$ which are, $h_0=0.4830$, $h(1)=0.8365$, $h(2)=0.2241$, $h(3)=-0.1294$ are low pass filter coefficients and $g_0=0.1294$, $g(1)=-0.2241$, $g(2)=0.8365$, $g(3)=-0.4830$ are the high pass filter coefficients.

3 Intergrid operators

3.1 Full-Approximation Scheme (FAS) operators

Brandt [4] was one of the first to introduce nonlinear multigrid, which seeks to use concepts from the linear multigrid iteration and apply them directly in the nonlinear setting. Since the early application to elliptic partial differential equations, multigrid methods have been applied successfully to a large and growing class of problems. Classical multigrid begins with a two-grid process. First, iterative relaxation is applied, whose effect is to smooth the error. In this paper, we describe how to apply multigrid to nonlinear problems. Applying multigrid method directly to the nonlinear problems by employing the method so-called Full Approximation Scheme (FAS). Full approximation scheme suitable for nonlinear problems is the FAS [5,8] which treats directly the nonlinear equations on finer and coarser grids. In FAS, a nonlinear iteration, such as the nonlinear Gauss-Seidel method is applied to smooth the error. In FAS, the residual is passed from the fine grids to the coarser grids. Vectors from fine grids are transferred to coarser grids with Restriction operator (R), while vectors are transferred from coarse grids to the finer grids with a Prolongation operator (P) respectively given in Section 4.1. The detail survey on FAS is given in [6].

3.2 Wavelet Full-Approximation Scheme (WFAS) operators

Discrete wavelet transform (DWT) matrix: The DWT matrix, which play an important part in the wavelet method. This is highly expedient and informative, particularly for the numerical computations. As we already know about the DWT matrix and its applications in the wavelet method and is given in [17] as,

$$D_1 = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & \dots & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & \dots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & \dots & 0 & 0 \\ \vdots & & \ddots & & \cdots & & \cdots & 0 & 0 \\ h_2 & h_3 & 0 & 0 & \cdots\cdots & & 0 & h_0 & h_1 \\ g_2 & g_3 & 0 & 0 & \cdots\cdots & & 0 & g_0 & g_1 \end{pmatrix}_{N\times N}$$

Using this matrix author's [24] used the restriction operator and prolongation operators respectively given in Section 4.2.

3.3 Modified Wavelet Full-Approximation Scheme (MWFAS) operators

Modified Discrete wavelet transform (MDWT) matrix: Here, we defined modified DWT matrix from DWT matrix in which we have added rows and columns consecutively with diagonal element as 1, which is built as,

$$D_2 = \begin{pmatrix} h_0 & 0 & h_1 & 0 & h_2 & 0 & h_3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & & & \cdots & 0 & 0 \\ g_0 & 0 & g_1 & 0 & g_2 & 0 & g_3 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & \cdots & 0 & 0 \\ \vdots & & \ddots & & \cdots & & \cdots & & & \cdots & 0 & 0 \\ g_2 & 0 & g_3 & 0 & \cdots & & \cdots & 0 & g_0 & 0 & g_1 & 0 \\ 0 & 0 & \cdots & & \cdots & & \cdots & & \cdots & & 0 & 1 \end{pmatrix}_{N\times N}$$

Using this matrix we defined the new restriction operator and new prolongation operators respectively given in Section 4.3.

4 Method of solution

4.1 Full-Approximation Scheme (FAS)

Consider the Nonlinear Volterra integral equation of the second kind,

$$u(t) = f(t) + \int_0^t K(t,s,u(s))ds, 0 \leq t,s \leq 1, \tag{3}$$

where $k(t,s,u(s))$ is a nonlinear function defined on $[0,1] \times [0,1]$. The known function $k(t,s,u(s))$ is called the kernel of the integral equation, while the unknown function $u(t)$ represents the solution of the integral equation. After discretizing the integral equation through the trapezoidal discretization method (TDM) [25], we get the system of nonlinear equations of the form,

$$A(u) = b, \tag{4}$$

It has N equations with N unknowns. Solving (4) through the iterative method that is Gauss Seidel (GS), we get approximate solution v.

Now, we are deliberating about the Full-Approximation Scheme (FAS) of solutions given by Briggs et. al [5] is as follows the procedure,

Step 1: From the system (4), we get the approximate solution v for u . Now we find the residual as,

$$r_{N\times 1} = b_{N\times 1} - A(v)_{N\times 1} \tag{5}$$

Reduce the matrices in the finer level to coarsest level using Restriction operator, i.e.,

$$R = \frac{1}{4}\begin{pmatrix} 1\,2 & 1 & 0 & 0 \cdots 0\,0 \\ 0\,0 & 1 & 2 & 1 \cdots 0\,0 \\ \vdots & \ddots & & \cdots 0\,0 \\ 0\,0 & & \cdots & \cdots 1\,2 \end{pmatrix}_{(N/2)\times N}$$

and then construct the matrices back to finer level from the coarsest level using Prolongation operator, i.e.,

$$P = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & \vdots \\ 0 & 2 & \vdots & \cdots & \vdots \\ 0 & 1 & & & \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix}_{(N)\times(N/2)}$$

Step 2:

$$r_{(N/2)\times 1} = R_{(N/2)\times N} r_{N\times 1}$$

Similarly,

$$v_{(N/2)\times 1} = R_{(N/2)\times N} v_{N\times 1}$$

and

$$A(v_{(N/2)\times 1} + e_{(N/2)\times 1}) - A(v_{(N/2)\times 1}) = r_{(N/2)\times 1} \tag{6}$$

Solving (6) with initial guess 0, we get $e_{(N/2)\times 1}$.
Step 3:

$$r_{(N/4)\times 1} = R_{(N/4)\times(N/2)} r_{(N/2)\times 1}$$

Similarly,

$$v_{(N/4)\times 1} = R_{(N/4)\times(N/2)} v_{(N/2)\times 1}$$

and

$$A(v_{(N/4)\times 1} + e_{(N/4)\times 1}) - A(v_{(N/4)\times 1}) = r_{(N/4)\times 1} \tag{7}$$

Solving (7) with initial guess 0, we get $e_{(N/4)\times 1}$.
Step 4: The procedure is continue up to the coarsest level, we have,

$$r_{1\times 1} = R_{1\times 2} r_{2\times 1}$$

Similarly,

$$v_{1\times 1} = R_{1\times 2} v_{2\times 1}$$

and

$$A(v_{1\times 1} + e_{1\times 1}) - A(v_{1\times 1}) = r_{1\times 1} \tag{8}$$

Solving (8), we get $e_{1\times 1}$.
Step 5: Interpolate error upto the finer level, i.e.,

$$e_{2\times 1} = P_{2\times 1} e_{1\times 1},$$
$$e_{4\times 1} = P_{4\times 2} e_{2\times 1},$$

and so on we have,

$$e_{N\times 1} = P_{N\times(N/2)} e_{(N/2)\times 1}.$$

Step 6: Correct the solution with error,

$$u_{N\times 1} = v_{N\times 1} + e_{N\times 1}.$$

This is the required solution of the given integral equation.

4.2 Wavelet Full-Approximation Scheme (WFAS)

In this paper, we applied WFAS for the numerical solution of nonlinear Volterra integral equations. The same procedure is applied as explained in the FAS in the above Section 4.1. But replacing W and W^T as the restriction and prolongation operators in place of R and P, i.e.,

$$W = \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & \cdots & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & \cdots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \cdots & & \cdots & 0 & 0 \\ 0 & 0 & \cdots & g_0 & g_1 & g_2 & g_3 & \cdots & 0 \end{pmatrix}_{(N/2)\times N}$$

and W^T respectively.

4.3 Modified Wavelet Full-Approximation Scheme (MWFAS)

In this paper, we introduced a modified wavelet full-approximation scheme (MWFAS) for the numerical solution of nonlinear Volterra integral equations. The same procedure is applied as explained in the FAS in Section 4.1, using MW and MW^T as the new restriction and new prolongation operators in place of R and P, i.e.,

$$MW = \begin{pmatrix} h_0 & 0 & h_1 & 0 & h_2 & 0 & h_3 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & & & \cdots & 0 & 0 \\ g_0 & 0 & g_1 & 0 & g_2 & 0 & g_3 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & \cdots & 0 & 0 \\ \vdots & & \ddots & & \cdots & & \cdots & & & \cdots & 0 & 0 \\ 0 & \cdots & g_0 & 0 & g_1 & 0 & g_2 & 0 & g_3 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}_{(N/2)\times N}$$

and MW^T respectively.

5 Illustrative problems

In this section, we implemented FAS, WFAS and MWFAS for the numerical solution of nonlinear Volterra integral and integro-differential equations and subsequently presented the efficiency of the MWFAS in the form of tables and figures, here error analysis is considered as,

$$E_{max} = max|u_e - u_a|,$$

where u_e and u_a are the exact and approximate solution respectively.

Test problem 5.1. Let us consider the nonlinear Volterra integral equation [28],

$$u(t) = t - t^2 - \frac{t^5}{4} + \frac{2t^6}{5} + \frac{t^7}{6} + \int_0^t tsu^2(s)ds, \qquad 0 \le t \le 1 \tag{9}$$

which has the exact solution $u(t) = t - t^2$. After discretizing the (9) through the trapezoidal discretization method (TDM), we get system of nonlinear algebraic equations of the form (for N = 8),

$$[A]_{8\times 8}[u]_{8\times 1} = [b]_{8\times 1} \tag{10}$$

Solving (10) through the iterative method, we get the approximate solution v of u . i.e., $u = e + v \Rightarrow v = u - e$, where e is (8×1 matrix) error to be determined. The implementation of MWFAS is discussed in Section 4 is as follows,
From (10), we find the residual as

$$r_{8\times 1} = [b]_{8\times 1} - [A]_{8\times 8}[v]_{8\times 1} \tag{11}$$

we get, = $r_{8\times 1}$[0 0 2.30e-06 2.01e-05 6.32e-05 8.89e-05 3.98e-05 -2.03e-07]
we reduce the matrices in the finer level to coarsest level using Restriction operator MW and then construct the matrices back to finer level from the coarsest level using Prolongation operator MW^T.
From (11),

$$r_{4\times 1} = [MW]_{4\times 8}[r]_{8\times 1} \tag{12}$$

Similarly,

$$v_{4\times 1} = [MW]_{4\times 8}[v]_{8\times 1}$$

t	Exact	FAS	WFAS	MWFAS
0	0	0	0	0
0.1428	0.1224	0.1224	0.1224	0.1224
0.2857	0.2040	0.2043	0.2043	0.2043
0.4285	0.2448	0.2457	0.2457	0.2457
0.5714	0.2448	0.2463	0.2463	0.2463
0.7142	0.2040	0.2055	0.2055	0.2055
0.8571	0.1224	0.1231	0.1231	0.1231
1	0	0.0001	0.0001	0.0001

Table 1 Numerical results of test problem 5.1, for N = *8*.

and

$$A(v_{4\times1}+e_{4\times1})-A(v_{4\times1})=r_{4\times1}. \tag{13}$$

Solving (13) with initial guess 0, we get $e_{4\times1}$.
From (12),

$$r_{2\times1}=[MW]_{2\times4}[r]_{4\times1} \tag{14}$$

Similarly,

$$v_{2\times1}=[MW]_{2\times4}[v]_{4\times1}$$

and

$$A(v_{2\times1}+e_{2\times1})-A(v_{2\times1})=r_{2\times1}. \tag{15}$$

Solving (15) with initial guess 0, we get $e_{2\times1}$.
From (14),

$$r_{1\times1}=[MW]_{1\times2}[r]_{2\times1} \tag{16}$$

Similarly,

$$v_{1\times1}=[MW]_{1\times2}[v]_{2\times1}$$

and

$$A(v_{1\times1}+e_{1\times1})-A(v_{1\times1})=r_{1\times1}. \tag{17}$$

Solving (17), we get $e_{1\times1}$.
From $e_{1\times1}$, Interpolate error up to the finer level, i.e.,

$$e_{2\times1}=[MW^T]_{2\times1}[e]_{1\times1},$$

$$e_{4\times1}=[MW^T]_{4\times2}[e]_{2\times1},$$

and lastly we have,

$$e_{8\times1}=[MW^T]_{8\times4}[e]_{4\times1}, \tag{18}$$

we get $e_{8\times1}=[1.31e-06 \quad 0 \quad 2.28e-06 \quad 2.06e-05 \quad 3.04e-05 \quad 0 \quad -1.75e-05 \quad 0]$
From (18), correct the solution with error $u_{8\times1}=v_{8\times1}+e_{8\times1}$.
Lastly, we get $u_{8\times1}$ is the required solution of (9). The numerical solutions of the given equation is obtained through the method as explained in Section 4 and are presented in comparison with the exact solution are shown in Table 1 and in Figure 1 for N = 64. Maximum error analysis and CPU time is shown in Table 2.

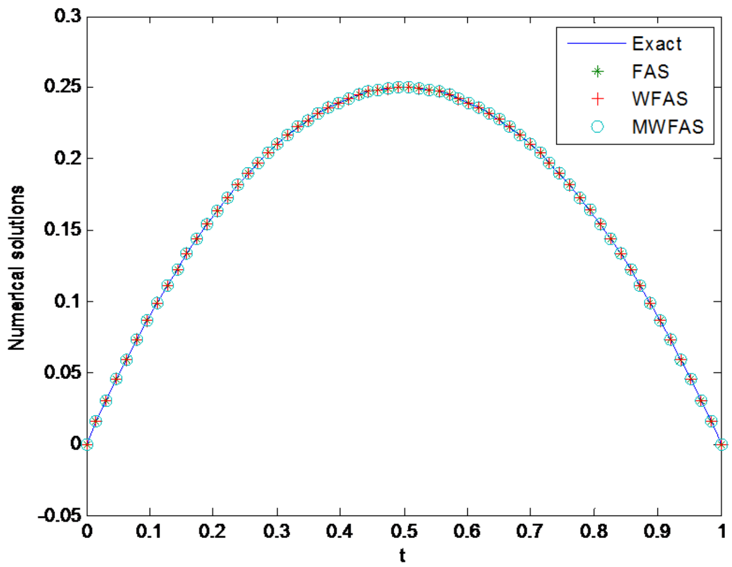

Fig. 1 Comparison of numerical solutions with exact solution of test problem 5.1, for N=64.

N	Methods	E_{max}	Setup time	Running time	Total time
16	FAS	$7.52e-04$	0.0054	0.0321	0.0375
	WFAS	$7.35e-04$	0.0031	0.0264	0.0295
	MWFAS	$7.35e-04$	0.0019	0.0254	0.0273
32	FAS	$3.66e-04$	0.0099	0.1459	0.1558
	WFAS	$3.62e-04$	0.0031	0.1479	0.1510
	MWFAS	$3.62e-04$	0.0019	0.1470	0.1489
64	FAS	$1.80e-04$	0.1261	0.1581	1.2842
	WFAS	$1.79e-04$	0.0031	0.0369	0.0400
	MWFAS	$1.79e-04$	0.0019	0.0357	0.0376
128	FAS	$8.99e-05$	0.1307	0.2185	0.3492
	WFAS	$8.96e-05$	0.0085	0.1419	0.1504
	MWFAS	$8.96e-05$	0.0019	0.1162	0.1182

Table 2 Maximum error and CPU time (in seconds) of the methods of test problem 5.1.

t	Exact	FAS	WFAS	MWFAS
0	0	0	0	0
0.0666	0.0666	0.0667	0.0667	0.0667
0.1333	0.1329	0.1332	0.1332	0.1332
0.2000	0.1986	0.1994	0.1994	0.1994
0.2666	0.2635	0.2648	0.2648	0.2648
0.3333	0.3271	0.3291	0.3291	0.3291
0.4000	0.3894	0.3922	0.3922	0.3922
0.4666	0.4499	0.4537	0.4537	0.4537
0.5333	0.5084	0.5133	0.5130	0.5130
0.6000	0.5646	0.5708	0.5703	0.5703
0.6666	0.6183	0.6259	0.6253	0.6253
0.7333	0.6693	0.6784	0.6776	0.6776
0.8000	0.7173	0.7280	0.7270	0.7270
0.8666	0.7621	0.7746	0.7734	0.7734
0.9333	0.8036	0.8174	0.8164	0.8164
1	0.8414	0.8449	0.8441	0.8441

Table 3 Numerical results of test problem 5.2, for N = *16*.

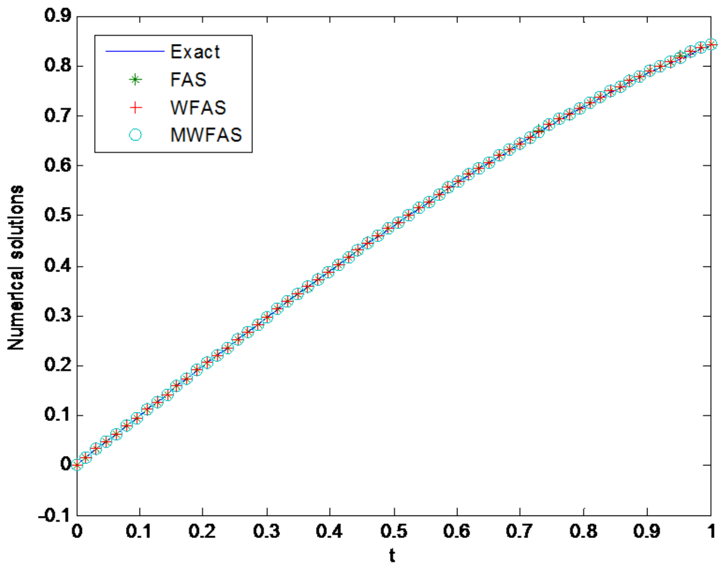

Fig. 2 Comparison of numerical solutions with exact solution of test problem 5.2, for N=64.

Test problem 5.2 Next, consider [29]

$$u(t) = sin(t) + \frac{1}{8}sin(2t) - \frac{1}{4}t + \frac{1}{2}\int_0^t u^2(s)ds, \qquad 0 \le t \le 1 \tag{19}$$

which has the exact solution $u(t) = sin(t)$. The numerical solutions of (19) are presented in Table 3 for N = 16 and in Figure 2 for N = 64. Maximum error analysis and CPU time is shown in Table 4.

N	Methods	E_{max}	Setup time	Running time	Total time
16	FAS	$1.38e-02$	0.0624	0.0636	0.1260
	WFAS	$1.28e-02$	0.0048	0.0443	0.0491
	MWFAS	$1.28e-02$	0.0019	0.0256	0.0275
32	FAS	$7.08e-03$	0.0099	0.1457	0.1556
	WFAS	$6.80e-03$	0.0031	0.1469	0.1499
	MWFAS	$6.80e-03$	0.0019	0.1475	0.1494
64	FAS	$3.56e-03$	0.2463	1.1761	1.4225
	WFAS	$3.49e-03$	0.0031	1.1385	1.1416
	MWFAS	$3.49e-03$	0.0019	1.1456	1.1475
128	FAS	$1.79e-03$	0.1819	10.2591	10.4410
	WFAS	$1.77e-03$	0.0125	9.5301	9.5426
	MWFAS	$1.77e-03$	0.0019	9.3220	9.3239

Table 4 Maximum error and CPU time (in seconds) of the methods of test problem 5.2.

t	Exact	FAS	WFAS	MWFAS
0	1	1.0357	1.0357	1.0357
0.0666	0.9977	0.9972	0.9972	0.9972
0.1333	0.9911	0.9901	0.9901	0.9901
0.2000	0.9800	0.9786	0.9786	0.9786
0.2666	0.9646	0.9628	0.9628	0.9628
0.3333	0.9449	0.9427	0.9427	0.9427
0.4000	0.9210	0.9185	0.9185	0.9185
0.4666	0.8930	0.8903	0.8903	0.8903
0.5333	0.8611	0.8582	0.8582	0.8582
0.6000	0.8253	0.8223	0.8223	0.8223
0.6666	0.7858	0.7829	0.7829	0.7829
0.7333	0.7429	0.7400	0.7400	0.7400
0.8000	0.6967	0.6939	0.6939	0.6939
0.8666	0.6473	0.6448	0.6448	0.6448
0.9333	0.5951	0.5929	0.5929	0.5929
1	0.5403	0.5383	0.5383	0.5383

Table 5 Numerical results of test problem 5.3, for N = *16*.

Test problem 5.3 Next, consider the Nonlinear Volterra-Hammerstein integral equations [30],

$$u(t) = 1 + sin^2(t) - \int_0^t 3sin(t-s)u^2(s)ds, \qquad 0 \le t \le 1 \tag{20}$$

which has the exact solution $u(t) = cos(t)$. The numerical solutions of (20) are presented in Table 5 for N = 16 and in Figure 3 for N = 64. Maximum error analysis and CPU time is shown in Table 6.

Test problem 5.4 Next, consider [31]

$$u(t) = -\frac{15}{56}t^8 + \frac{13}{14}t^7 - \frac{11}{10}t^6 + \frac{9}{20}t^5 + t^2 - t + \int_0^t (t+s)u^3(s)ds, \qquad 0 \le t \le 1 \tag{21}$$

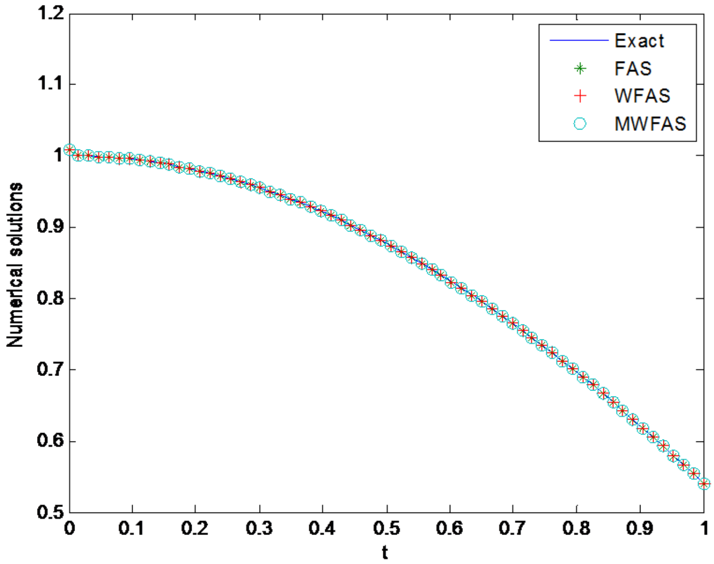

Fig. 3 Comparison of numerical solutions with exact solution of test problem 5.3, for N=64.

N	Methods	E_{max}	Setup time	Running time	Total time
	FAS	$3.57e-02$	0.0054	0.0121	0.0175
16	WFAS	$3.57e-02$	0.0031	0.0121	0.0152
	MWFAS	$3.57e-02$	0.0019	0.0108	0.0127
	FAS	$1.66e-02$	0.0099	0.0172	0.0271
32	WFAS	$1.66e-02$	0.0031	0.0164	0.0195
	MWFAS	$1.66e-02$	0.0019	0.0156	0.0175
	FAS	$8.06e-03$	0.0281	0.0370	0.0651
64	WFAS	$8.06e-03$	0.0031	0.0360	0.0390
	MWFAS	$8.06e-03$	0.0019	0.0357	0.0375
	FAS	$3.96e-03$	0.1009	0.1168	0.2177
128	WFAS	$3.96e-03$	0.0044	0.1329	0.1373
	MWFAS	$3.96e-03$	0.0020	0.1300	0.1320

Table 6 Maximum error and CPU time (in seconds) of the methods of test problem 5.3.

t	Exact	FAS	WFAS	MWFAS
0	0	0	0	0
0.0666	-0.0622	-0.0622	-0.0622	-0.0622
0.1333	-0.1155	-0.1155	-0.1155	-0.1155
0.2000	-0.1600	-0.1600	-0.1600	-0.1600
0.2666	-0.1955	-0.1957	-0.1957	-0.1957
0.3333	-0.2222	-0.2224	-0.2224	-0.2224
0.4000	-0.2400	-0.2403	-0.2403	-0.2403
0.4666	-0.2488	-0.2493	-0.2493	-0.2493
0.5333	-0.2488	-0.2494	-0.2494	-0.2494
0.6000	-0.2400	-0.2405	-0.2405	-0.2405
0.6666	-0.2222	-0.2227	-0.2227	-0.2227
0.7333	-0.1955	-0.1959	-0.1959	-0.1959
0.8000	-0.1600	-0.1602	-0.1602	-0.1602
0.8666	-0.1155	-0.1156	-0.1156	-0.1156
0.9333	-0.0622	-0.0622	-0.0622	-0.0622
1	0	0	0	0

Table 7 Numerical results of test problem 5.4, for N = *16*.

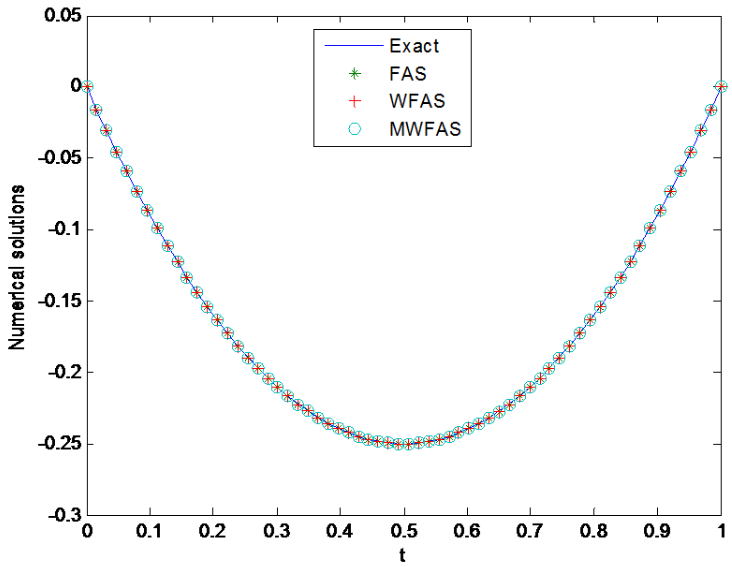

Fig. 4 Comparison of numerical solutions with exact solution of test problem 5.4, for N=64.

which has the exact solution $u(t) = t^2 - t$. The numerical solutions of (21) are presented in Table 7 for N = 16 and in Figure 4 for N = 64. Maximum error analysis and CPU time is shown in Table 8.
Test problem 5.5 Lastly, consider the Nonlinear Volterra integro-differential equation [32],

$$u'(t) = -1 + \int_0^t u^2(s)ds, u(0) = 0, \quad 0 \le t \le 1 \tag{22}$$

which has the exact solution $u(t) = -t + \frac{t^4}{12} - \frac{t^7}{252} + \frac{t^10}{6048} - \frac{t^13}{157248}$.

N	Methods	E_{max}	Setup time	Running time	Total time
	FAS	$5.72e-04$	0.0054	0.0404	0.0458
16	WFAS	$5.60e-04$	0.0030	0.0334	0.0365
	MWFAS	$5.60e-04$	0.0018	0.0315	0.0333
	FAS	$2.81e-04$	0.0105	0.2084	0.2189
32	WFAS	$2.77e-04$	0.0030	0.2056	0.2086
	MWFAS	$2.76e-04$	0.0018	0.2057	0.2075
	FAS	$1.38e-04$	0.0306	0.0484	0.0790
64	WFAS	$1.37e-04$	0.0030	0.0471	0.0502
	MWFAS	$1.37e-04$	0.0018	0.0456	0.0474
	FAS	$6.87e-05$	0.1114	0.1810	0.2924
128	WFAS	$6.85e-05$	0.0044	0.1780	0.1823
	MWFAS	$6.85e-05$	0.0018	0.1804	0.1822

Table 8 Maximum error and CPU time (in seconds) of the methods of test problem 5.4.

We convert the Volterra integro-differential equation to equivalent Volterra integral equation by using the well-known formula, which converts multiple integrals into a single integral. i.e.,

$$\underbrace{\int_0^t \int_0^t \dots \int_0^t u(t)dt^n}_{n-times} = \frac{1}{(n-1)!}\int_0^t (t-s)^{n-1}u(s)ds \tag{23}$$

Integrating (22) on both sides from 0 to t and using the initial condition and also converting the double integral to the single integral, we obtain

$$u(t) = f(t) + \int_0^t k(t,s)u^2(s)ds, \tag{24}$$

where $k(t,s) = (t-s)$ and $f(t) = -t$. The numerical solutions of (22) are presented in Table 9 for N = 32 and in Figure 5 for N = 64. Maximum error analysis and CPU time is shown in Table 10.

t	Exact	FAS	WFAS	MWFAS
0	0	0	0	0
0.0322	-0.0322	-0.0322	-0.0322	-0.0322
0.0645	-0.0645	-0.0645	-0.0645	-0.0645
0.0967	-0.0967	-0.0967	-0.0967	-0.0967
0.1290	-0.1290	-0.1290	-0.1290	-0.1290
0.1612	-0.1612	-0.1612	-0.1612	-0.1612
0.1935	-0.1934	-0.1934	-0.1934	-0.1934
0.2258	-0.2255	-0.2255	-0.2255	-0.2255
0.2580	-0.2577	-0.2577	-0.2577	-0.2577
0.2903	-0.2897	-0.2897	-0.2897	-0.2897
0.3225	-0.3216	-0.3216	-0.3216	-0.3216
0.3548	-0.3535	-0.3535	-0.3535	-0.3535
0.3870	-0.3852	-0.3852	-0.3852	-0.3852
0.4193	-0.4167	-0.4168	-0.4168	-0.4168
0.4516	-0.4481	-0.4481	-0.4481	-0.4481
0.4838	-0.4793	-0.4793	-0.4793	-0.4793
0.5161	-0.5102	-0.5102	-0.5102	-0.5102
0.5483	-0.5409	-0.5409	-0.5409	-0.5409
0.5806	-0.5712	-0.5712	-0.5712	-0.5712
0.6129	-0.6012	-0.6013	-0.6013	-0.6013
0.6451	-0.6309	-0.6309	-0.6309	-0.6309
0.6774	-0.6601	-0.6601	-0.6601	-0.6601
0.7096	-0.6888	-0.6889	-0.6889	-0.6889
0.7419	-0.7171	-0.7172	-0.7172	-0.7172
0.7741	-0.7449	-0.7449	-0.7449	-0.7449
0.8064	-0.7720	-0.7721	-0.7721	-0.7721
0.8387	-0.7986	-0.7986	-0.7986	-0.7986
0.8709	-0.8244	-0.8245	-0.8245	-0.8245
0.9032	-0.8496	-0.8497	-0.8497	-0.8497
0.9354	-0.8740	-0.8741	-0.8741	-0.8741
0.9677	-0.8976	-0.8977	-0.8977	-0.8977
1	-0.9204	-0.9205	-0.9205	-0.9205

Table 9 Numerical results of test problem 5.5, for $N = 32$.

N	Methods	E_{max}	Setup time	Running time	Total time
16	FAS	$2.81e-04$	0.0054	0.0125	0.0180
	WFAS	$2.81e-04$	0.0039	0.0113	0.0152
	MWFAS	$2.81e-04$	0.0025	0.0108	0.0133
32	FAS	$6.56e-05$	0.0100	0.0169	0.0268
	WFAS	$6.56e-05$	0.0030	0.0161	0.0190
	MWFAS	$6.56e-05$	0.0019	0.0158	0.0177
64	FAS	$1.57e-05$	0.0286	0.0368	0.0654
	WFAS	$1.57e-05$	0.0030	0.0358	0.0388
	MWFAS	$1.57e-05$	0.0019	0.0353	0.0372
128	FAS	$3.69e-06$	0.1042	0.1240	0.2281
	WFAS	$3.69e-06$	0.0033	0.1156	0.1189
	MWFAS	$3.69e-06$	0.0019	0.1167	0.1186

Table 10 Maximum error and CPU time (in seconds) of the methods of test problem 5.5.

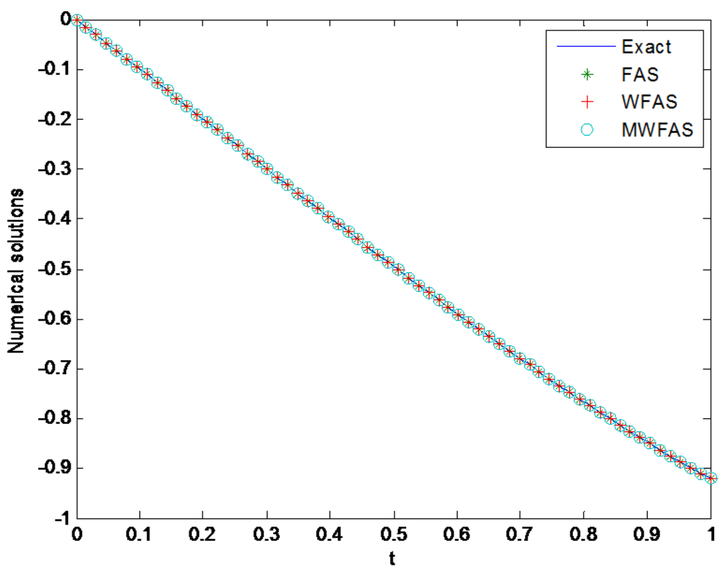

Fig. 5 Comparison of numerical solutions with exact solution of test problem 5.5, for N=64.

6 Conclusion

In this paper, we proposed a modified wavelet full-approximation scheme for the numerical solution of nonlinear Volterra integral and integro-differential equations. The modified wavelet intergrid operators of prolongation and restrictions are used in this paper, a modified wavelet based FAS, has been shown to be effective and versatile. The numerical solutions obtained agree well with the exact ones. Convergence is also observed in the numerical solutions when the calculation is refined by increasing the number N used. The standard FAS and WFAS converge slowly with larger computational cost, whereas MWFAS does ensure such slower convergence with lesser computational cost. Test problems are justified through the error analysis, as the level of resolution N increases, larger the accuracy increases. Hence, the new scheme is very convenient and efficient than the existing standard methods.

Acknowledgements

The authors thank for the financial support of UGC's UPE Fellowship vide sanction letter D. O. No. F. 14-2/2008(NS/PE), dated-19/06/2012 and F. No. 14-2/2012(NS/PE), dated 22/01/2013.

References

[1] A.-M. Wazwaz, (2011), *Linear and Nonlinear Integral Equations: Methods and Applications*, Springer-Verlag Berlin Heidelberg. doi 10.1007/978-3-642-21449-3
[2] A. J. Jerri, (1999), *Introduction to Integral Equations with Applications*, John Wiley & Sons, Inc.
[3] K. E. Atkinson, (1997), *The Numerical Solution of Integral Equations of the Second Kind*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York.
[4] A. Brandt, (1977), *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31, No 138, 333-390. doi 10.1090/S0025-5718-1977-0431719-X
[5] W. L. Briggs, V. E. Henson and S. F. McCormick, (2000), *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
[6] W. Hackbusch and U. Trottenberg (Eds.), (1982), *Multigrid Methods*, Proceedings of the Conference Held at Köln-Porz, November 23-27, Springer-Verlag Berlin Heidelberg. doi 10.1007/BFb0069927
[7] P. Wesseling, (1995), *Introduction to Multigrid Methods*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, NASA COntractor Report 195045.
[8] U. Trottenberg, C. Oosterlee and A. Schüller, (2001), *Multigrid*, Academic Press, London-San Diego (CA).
[9] A.A. Lubrecht, (1987), *Numerical solution of the EHL line and point contact problem using multigrid techniques*, PhD Thesis, University of Twente, Enschede, The Netherlands.
[10] C.H. Venner and A.A. Lubrecht, (2000), *Multi-Level Methods in Lubrication*, Elsevier Science B.V., Amsterdam.
[11] E. A. Zargari, P. K. Jimack and M. A. Walkley, (2007), *An investigation of the film thickness calculation for elastohydrodynamic lubrication problems*, International Journal for Numerical Methods in Fluids, 00:1-6.
[12] H. Lee, (1997), *Multigrid method for nonlinear integral equations*, Korean Journal of Computational & Applied Mathematics, 4, No 2, 427-440. doi 10.1007/BF03014490
[13] C. K. Chui, (1997), *Wavelets: A Mathematical Tool for Signal Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
[14] Ü. Lepik and E. Tamme, (2005), *Application of the Haar Wavelets for Solution of Linear Integral Equations*, Proceedings of Dynamical Systems and Applications, Antalya, Turkey, 494-507.
[15] I. Aziz and S.-ul-Islam, (2013), *New algorithms for the numerical solution of nonlinear Fredholm and Volterra integral equations using Haar wavelets*, Journal of Computational and Applied Mathematics, 239, 333-345. doi 10.1016/j.cam.2012.08.031
[16] D. De Leon, (2008), *A new wavelet multigrid method*, Journal of Computational and Applied Mathematics, 220, No 1-2, 674-685. doi 10.1016/j.cam.2007.09.021
[17] N. M. Bujurke, C. S. Salimath, R. B. Kudenatti and S. C. Shiralashetti, (2007), *A fast wavelet-multigrid method to solve elliptic partial differential equations*, Applied Mathematics and Computation, 185, No 1, 667-680. doi 10.1016/j.amc.2006.07.074
[18] N. M. Bujurke, C. S. Salimath, R. B. Kudenatti and S. C. Shiralashetti, (2007), *Wavelet-multigrid analysis of squeeze film characteristics of poroelastic bearings*, Journal of Computational and Applied Mathematics, 203, No 1, 237-248.

doi 10.1016/j.cam.2006.04.001

[19] N. M. Bujurke, C. S. Salimath, R. B. Kudenatti and S. C. Shiralashetti, (2007), *Analysis of modified Reynolds equation using the wavelet-multigrid scheme*, Numerical Methods for Partial Differential Equations, 23, No 3, 692-705. doi 10.1002/num.20199

[20] A. Avudainayagam and C. Vani, (2004), *Wavelet based multigrid methods for linear and nonlinear elliptic partial differential equations*, Applied Mathematics and Computation, 148, No 2, 307-320. doi 10.1016/S0096-3003(02)00845-7

[21] G. Wang, R. Dutton and J. Hou, (2000), *A fast wavelet multigrid algorithm for solution of electromagnetic integral equations*, Microwave and Optical Technology Letters, 24, No 2, 86-91.

[22] I. Daubechies, (1992), *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA. doi 10.1137/1.9781611970104

[23] I. Daubechies, (1988), *Orthonormal bases of compactly supported wavelets*, Communications on Pure and Applied Mathematics, 41, No 7, 909-996. doi 10.1002/cpa.3160410705

[24] M. Griebel and S. Knapek, (1995), *A Multigrid-Homogenization Method*, in: Modeling and Computation in Environmental Sciences, Proceedings of the First GAMM-Seminar at ICA Stuttgart, October 12-13, Stuttgart, Volume 59 of the series Notes on Numerical Fluid Mechanics (NNFM) pp 187-202. doi 10.1007/978-3-322-89565-3_17

[25] I. S. Kotsireas, (2008), *A Survey on Solution Methods for Integral Equations*. Available at: http://www.orcca.on.ca/TechReports/TechReports/2008/TR-08-03.pdf

[26] S. C. Shiralashetti and A. B. Deshi, (2016), *An efficient Haar wavelet collocation method for the numerical solution of multi-term fractional differential equations*, Nonlinear Dynamics, 83, No 1, 293-303. doi 10.1007/s11071-015-2326-4

[27] S.C. Shiralashettia, A.B. Deshia and P.B. Mutalik Desaib, (2016), *Haar wavelet collocation method for the numerical solution of singular initial value problems*, Ain Shams Engineering Journal, 7, No 2, 663-670. doi 10.1016/j.asej.2015.06.006

[28] A. Mohsen and M. El-Gamel, (2010), *On the numerical solution of linear and nonlinear volterra integral and integro-differential equations*, Applied Mathematics and Computation, 217, No 7, 3330-3337. doi 10.1016/j.amc.2010.08.065

[29] A. Shahsavaran, (2011), *Numerical Solution of Nonlinear Fredholm-Volterra Integtral Equations via Piecewise Constant Function by Collocation Method*, American Journal of Computational Mathematics, No 1, 134-138. doi 10.4236/ajcm.2011.12014

[30] M. Razzaghi, Y. Ordokhani, (2000), *Solution of nonlinear Volterra-Hammerstein integral equations via rationalized Haar functions*, Mathematical Problems in Engineering, 7, No 2, 205-219. doi 10.1155/S1024123X01001612

[31] S. Yalçinbaş, (2002), *Taylor polynomial solutions of nonlinear Volterra-Fredholm integral equations*, Applied Mathematics and Computation, 127, No 2-3, 195-206. doi 10.1016/S0096-3003(00)00165-X

[32] A. Avudainayagam and C. Vani , (2000),*Wavelet-Galerkin method for integro-differential equations*, Applied Numerical Mathematics, 32, No 3, 247-254. doi 10.1016/S0168-9274(99)00026-4