Research article

Efficient numerical technique for solution of delay Volterra-Fredholm integral equations using Haar wavelet

Rohul Amin a, Kamal Shah b,*, Muhammad Asif a, Imran Khan a

a Department of Mathematics, University of Peshawar, Peshawar, 25120, Khyber Pakhtunkhwa, Pakistan
b Department of Mathematics, University of Malakand, Dir(L), 18000, Khyber Pakhtunkhwa, Pakistan

ARTICLE INFO

Keywords:
Mathematics
Integral equations
Delay Volterra integral equations
Delay Fredholm integral equations
Haar wavelet
Collocation method

ABSTRACT

In this article, a computational Haar wavelet collocation technique is developed for the solution of linear delay integral equations. These equations include delay Fredholm, Volterra and Volterra-Fredholm integral equations. First we transform the derived estimates for these equations. After that, we transform these estimates to a system of algebraic equations. Finally, we solve the obtained algebraic system by Gauss elimination technique. Numerical examples are taken from literature for checking the validity and convergence of the proposed technique. The maximum absolute and root mean square errors are compared with the exact solution. The convergence rate using distinct numbers of collocation points is also calculated, which is approximately equal to 2. All algorithms for the developed method are implemented in MATLAB (R2009b) software.

1. Introduction

The subject of Integral Equations (IEs) is very important branch of applied mathematics. IEs are classified into two major types depending on the limits of integration. Different types of IEs include Volterra IEs, Fredholm IEs and Volterra-Fredholm IEs. In Volterra IEs one limit of integration is variable and the other is constant, whereas in Fredholm IEs both the limits are constants. Volterra-Fredholm IEs involve both characteristics of Fredholm and Volterra IEs [1]. Many physical phenomena are modeled using delay IEs. Delay Volterra IEs are used for modeling of systems with history, such as electric circuits and mechanical systems. These IEs also play a major role in engineering [2]. Delay Volterra-Fredholm IEs are used in applied sciences for modeling of various phenomena such as dynamical systems, physical models and population growth [3].

Many authors introduced various numerical techniques for the solution of delay IEs. Bellour et al. [3] used collocation technique for the solution of delay IEs using Taylor polynomials. Mosleh and Otadi [4] utilized least squares technique to solve delay Volterra IEs. Bica and Popescu [5] developed an iterative technique to solve nonlinear fuzzy Volterra IEs with fixed delay. Khasi et al. [6] found solution of delay IEs utilizing piecewise collocation technique. Horvat [7] analyzed the global convergence properties of polynomial collocation for the solution of constant delay Volterra IEs. Rao et al. [8] used Walsh series method to remove the inconvenience in Walsh functions and block Pulse functions methods. Balachandran, Palanisamy, Sepehrian, Paul Dhayaaran, Razzaghi and Murugesan utilized the considered method for computation of solutions of many problems of integral/differential equations [9, 10, 11, 12]. Mirzaee and Samadyar [13] used Bernstein collocation method for solution of 2D-mixed Volterra-Fredholm IEs. Mirzaee and Hadadiyan utilized operational matrix [14] for solution of nonlinear class of mixed Volterra-Fredholm IEs, Bell polynomials [15] for solution of nonlinear Fredholm-Volterra IEs and modification of hat functions [16] for solution of Volterra-Fredholm IEs. Mirzaee and Hadadiyan [17] also found the solution of two-dimensional Volterra-Fredholm IEs. Mirzaee and Hoseini [18] used Fibonacci polynomials and collocation points for numerical solution of Volterra-Fredholm IEs. Mirzaee and Hadadiyan [19], applied the modified block-pulse functions for solution of three-dimensional Volterra-Fredholm IEs. Mirzaee et al. [20], used three-dimensional block-pulse functions for numerical solution of three-dimensional nonlinear mixed Volterra-Fredholm IEs.

Many numerical methods have been introduced to solve IEs numerically but every method has its own shortcomings. Most of the numerical methods convert the model equations containing IEs into discretized model equations that appear in the form of a set of algebraic linear or nonlinear equations. In case of direct solvers to solve this set of algebraic equations the computational cost of a large system gets worst due to large computational time and memory requirement needed for mathe-
metrical operations. Therefore, it is still a challenging task to introduce a simple and efficient technique for IEs. In the “Haar Wavelet Collocation” (HWC) method, we focus on improving the efficiency of the direct solvers. Direct solvers are useful for solving problems for which iterative solvers struggle. Here, we utilized HWC method for the solution of delay IEs. The HWC technique has been applied for solving several mathematical problems in engineering and applied science. In the literature the researchers used HWC method for solution of boundary value problems [21], partial differential equations [22], evolution equations [23], fractional partial differential equations [24] and delay partial differential equations [25]. Maleknajad and Mirzaee, utilized rationalized Haar wavelet for the solution of linear IEs [26].

The main advantages of HWC method are its simplicity and less computation cost: it is due to the sparsity of the transform matrices and to the small number of significant wavelet coefficients. In comparison the existing numerical schemes to solve the IEs, the Haar wavelet technique is an improvement over other techniques in terms of accuracy. It is worth mentioning that Haar solution provides good results even for small values of $N$. For larger values of $N$ we can obtain the results closer to the real values. The reason of using HWC method is it has sparse matrix representation, fast transformation and possibility of implementation of fast and efficient algorithms. However, HWC technique has disadvantages too. Haar wavelets are piecewise constant functions having jump discontinuity and its derivatives of all orders vanish. Therefore, these wavelets can not be applied directly for solving IEs. To overcome this drawback we approximate the unknown function appearing in the IEs.

The solution of delay Volterra IEs, Fredholm IEs and Volterra-Fredholm IEs with constant delay $\xi > 0$ will be studied in this article. The delay Volterra IEs [27] with both continuous and discrete delays $\xi > 0$ are:

$$u(x) = \begin{cases} a(x)w(x-\xi) + \int_{-\xi}^{0} G_1(t,s)w(s)ds + f(x), & x \in [-\xi,0), \\ \Phi(x), & x \in [-\xi,0), \end{cases}$$

The delay Fredholm IE with delay $\xi > 0$ is:

$$u(x) = \begin{cases} a(x)w(x-\xi) + \int_{0}^{d} G_1(t,s)w(s-\xi)ds + f(x), & x \in [-\xi,0), \\ \Phi(x), & x \in [-\xi,0), \end{cases}$$

and delay Volterra-Fredholm IE with delay $\xi > 0$ is:

$$u(x) = \begin{cases} a(x)w(x-\xi) + \int_{-\xi}^{0} G_1(t,s)w(s-\xi)ds + \int_{0}^{d} G_2(t,s)w(s)ds + f(x), & x \in [-\xi,0), \\ \Phi(x), & x \in [-\xi,0), \end{cases}$$

with the initial condition $u(0) = u_0$, where $G_1, G_2, G_i$ are smooth functions, known as kernels of the integral and $\Phi(x)$ is the delay condition or prestory function.

The paper is organized as: In Section 2, Haar functions are defined. Numerical HWC technique for delay Volterra IEs, Fredholm IEs and Volterra-Fredholm IEs is given in Section 3. In Section 4, some test problems from literature are given for validation of HWC method. Conclusion is given in Section 6.

2. Haar wavelet

The Haar functions are piecewise constant functions having three values 1, −1 and 0. The Haar scaling function on interval $[\gamma_1, \gamma_2]$ is given by [25]

$$h_1(y) = \begin{cases} 1 & \text{for } y \in [\gamma_1, \gamma_2), \\ -1 & \text{for } y \in (\frac{\gamma_1 + \gamma_2}{2}, \gamma_2), \\ 0 & \text{otherwise}. \end{cases}$$

The mother wavelet for the HW functions on $[\gamma_1, \gamma_2]$ is

$$h_2(y) = \begin{cases} 1 & \text{for } y \in [\gamma_1, \frac{\gamma_1 + \gamma_2}{2}), \\ -1 & \text{for } y \in (\frac{\gamma_1 + \gamma_2}{2}, \gamma_2), \\ 0 & \text{otherwise}. \end{cases}$$

All functions in the Haar wavelet family except the scaling function are represented as

$$h_i(y) = \begin{cases} 1 & \text{for } y \in [\rho_1, \rho_2), \\ -1 & \text{for } y \in [\rho_2, \rho_3), \\ 0 & \text{otherwise}, \end{cases}$$

where

$$\rho_1 = \gamma_1 + (\gamma_2 - \gamma_1)\frac{\xi}{d},$$

$$\rho_2 = \gamma_1 + (\gamma_2 - \gamma_1)\frac{\xi + 0.5}{d},$$

$$\rho_3 = \gamma_1 + (\gamma_2 - \gamma_1)\frac{\xi + 1}{d}.$$}

In the above definition integer $d = 2^r, r = 0, 1, \ldots, r'$ shows the level of the wavelet and the integer $\xi = 0, 1, \ldots, d - 1$ is the translation parameter. The index $i$ in Eq. (6) is computed by utilizing the formula $i = d + \xi + 1$. In the interval $[0,1]$, $\rho_1$, $\rho_2$ and $\rho_3$ are defined as

$$\rho_1 = \frac{\xi}{d}, \quad \rho_2 = \frac{\xi + 0.5}{d}, \quad \rho_3 = \frac{\xi + 1}{d}.$$}

Any member of $L^2[0,1)$ can be written as:

$$f(y) = \sum_{i=1}^{N} \lambda_i h_i(y).$$

This series is truncated at finite terms $N$ for approximation purpose, i.e.

$$f(y) \approx \sum_{i=1}^{N} \lambda_i h_i(y).$$

We use the following symbol

$$p_i(y) = \int_{0}^{y} h_i(x)dx,$$

and the value of the above integral is calculated by definition of $h_i$ and is given by

$$p_i(y) = \begin{cases} y - \rho_1 & \text{for } y \in [\rho_1, \rho_2), \\ \rho_2 - y & \text{for } y \in [\rho_2, \rho_3), \\ 0 & \text{elsewhere}. \end{cases}$$

In general

$$p_{i,n}(y) = \int_{0}^{y} p_{i,n-1}(x)dx.$$}

Thus $p_{i,n}(y)$ is obtained as under [25],

$$p_{i,n}(y) = \begin{cases} 0 & \text{for } y \in [0, \rho_1), \\ \frac{1}{\rho_2 - \rho_1}(y - \rho_1)^n & \text{for } y \in [\rho_1, \rho_2), \\ \frac{1}{\rho_3 - \rho_2}(y - \rho_2)^n - 2(\rho_1 - \rho_2)^n(y - \rho_2)^n & \text{for } y \in [\rho_2, \rho_3), \\ \frac{1}{\rho_3 - \rho_2}(y - \rho_2)^n - 2(\rho_1 - \rho_2)^n + (y - \rho_2)^n & \text{for } y \in [\rho_3, 1). \end{cases}$$

For HWC technique, the interval $[a, \beta]$ is discretized using the formula:

$$t_m = a + (\beta - a)m - 1/2 \frac{M}{2} \quad m = 1, 2, 3, 4, \ldots, 2M.$$}

Eq. (13) gives the collocation points (CPs). Some of the recent work using HWC technique in literature can be seen in the references [28, 29, 30, 31, 32, 33, 34].
3. Numerical scheme for delay IEs

In this section, the proposed HWC method is discussed for the solution of delay Volterra IEs, delay Volterra-Fredholm IEs and system of Volterra-Fredholm IEs. In first subsection, the solution of delay Volterra IEs is studied, while in the next subsection, delay Volterra-Fredholm IEs are considered. Here we introduce the notation $\Theta = \sum_{i=1}^{N}$. Let $u(x) \in L_{2}(0,1)$, then

$$u(x) = \Theta \lambda_{i} h_{i}(x).$$

We approximate the integral in Eqs. (1), (2) and (3) with the help of Haar formula

$$\int_{\eta_{1}}^{\eta_{2}} f(x)dx \approx \frac{\eta_{2} - \eta_{1}}{2M} \sum_{k=1}^{2M} f(x_{k}) = \sum_{k=1}^{2M} f \left( \eta_{1} + \frac{(\eta_{2} - \eta_{1})(k - 1/2)}{2M} \right).$$

After obtaining the system of algebraic equations we discretized the IEs by using Eq. (14), Eq. (15) and CPs in the given delay IEs. By solving these linear systems of equations we have obtained the values of $\lambda_{i}$’s. The obtained system is then solved by Gauss technique. By putting the values of constants $\lambda_{i}$’s in Eq. (14), we obtain the required solution of delay IEs.

3.1. Delay Volterra IEs

HWC method is developed for the solution of delay Volterra IE (1) in this section. Putting the Haar approximation and using the delay condition, we get

$$\Theta \lambda_{i} h_{i}(x) = a(x) \Phi(x - \xi) + \int_{x-\xi}^{x} G_{1}(x,s) \Phi(s)ds$$

$$+ \int_{a}^{b} G_{2}(x,s) \Theta \lambda_{i} h_{i}(s)ds + f(x).$$

Applying Haar formula of integration given in Eq. (15), we have

$$\Theta \lambda_{i} h_{i}(x) = a(x) \Phi(x - \xi) + \frac{x-a}{N} \sum_{n=1}^{N} G_{1}(x,s_{n}) \Phi(s_{n})$$

$$+ \frac{x-a}{N} \sum_{n=1}^{N} G_{2}(x,s_{n}) \Theta \lambda_{i} h_{i}(s_{n}) + f(x).$$

Simplifying we obtain

$$\Theta \lambda_{i} h_{i}(x) - \frac{x-a}{N} \sum_{n=1}^{N} G_{2}(x,s_{n}) \Theta \lambda_{i} h_{i}(s_{n})$$

$$= a(x) \Phi(x - \xi) + \frac{x-a}{N} \sum_{n=1}^{N} G_{1}(x,s_{n}) \Phi(s_{n}) + f(x).$$

Plugging the CPs $x_{k}$, $k = 1,2,3,4,\ldots,N$, we obtain

$$\Theta \lambda_{i} h_{i}(x_{k}) - \frac{x_{k}-a}{N} \sum_{n=1}^{N} G_{2}(x_{k},s_{n}) \Theta \lambda_{i} h_{i}(s_{n})$$

$$= a(x_{k}) \Phi(x_{k} - \xi) + \frac{x_{k}-a}{N} \sum_{n=1}^{N} G_{1}(x_{k},s_{n}) \Phi(s_{n}) + f(x_{k}).$$

In matrix notations, we have

$$MA = B,$$

where

$$A = [\lambda_{1},\lambda_{2},\lambda_{3},\ldots,\lambda_{N}]^{T},$$

$$M = [m_{ij}]_{N \times N},$$

$$B = [b_{k}]_{N \times 1}.$$
Thus the unknown Haar coefficients $\lambda_j$’s are obtained as $\mathbf{A} = \mathbf{M}^{-1} \mathbf{B}$. Now plugging these $\lambda_j$’s in Eq. (14), we obtain the required solution at CPs.

**Remark.** Delay Fredholm IEs.

Delay Fredholm IE is a special form of the delay Volterra-Fredholm IE. Repeating the above procedure as discussed in Section 3.1 for solution of Volterra IE to Eq. (2), we get the solution of delay Fredholm IE. If we put $G_1$ and $G_2$ equal to zero then Eq. (3) becomes delay Fredholm IE. Similarly by replacing $G_2$ by zero in Eq. (3), we obtain delay Volterra IE.

4. **Test problems**

In this section, HWC technique is utilized for solution of some numerical problems from literature. For validation of HWC method, the numerical solution is compared with the exact solution.

The notation $L_{\infty}$ is utilized for maximum absolute error while $L_{rms}$ is used for root mean square errors at $N$ collocation points. The $L_{rms}$ error is given by:

$$L_{rms} = \frac{1}{N} \left( \sum_{j=1}^{N} \left( u(x_j) - u_{hyp}(x_j) \right)^2 \right)^{1/2},$$

where $u(x_j)$ and $u_{hyp}(x_j)$ are used for exact and approximate solution on points $x_j$ for $j = 1, 2, 3, ..., N$ at $N$ discrete CPs. We have also computed the experimental rate of convergence $R_e$ which is given by

$$R_e = \log(L_{\infty}(N/2)/L_{\infty}(N)) \frac{1}{\log(2)}.$$

**Problem 1.** Consider delay IE [27]

$$w(t) = \left\{ \begin{array}{ll}
  f(t) + \int_{0}^{t} e^{-t} w(s) ds + \int_{0}^{t-1} t \sin s w(s) ds, & -1 < t \leq 0 \\
  \sin t & \end{array} \right.$$

The $f(t)$ function is so that the analytical solution is

$$w(t) = 1 + \sin t.$$  

**Problem 2.** Consider delay Volterra IE [27]

$$w(t) = \left\{ \begin{array}{ll}
  f(t) + \int_{t-1}^{t} s \cos(s+t) w(s) ds + f(t), & 2 \sin t + 1, \quad -1 \leq t < 0 \\
  \sin t & \end{array} \right.$$

The $f(t)$ function is so that analytical solution is

$$w(t) = 1 + 2 \sin t.$$  

**Problem 3.** Next, we consider delay Volterra IE [27]

$$w(t) = \left\{ \begin{array}{ll}
  \int_{0}^{t} \sin(s+t) w(s) ds + \int_{0}^{t-1} \frac{t}{1+s} w(s) ds + f(t), & 0 \leq t < 1 \\
  \int_{0}^{t} \frac{t}{2} & \end{array} \right.$$  

The analytical solution is $w(t) = \frac{t \sin t}{2}.$

### Table 1. $L_{\infty}$ and $L_{rms}$ errors for Problem 1.

| $T$ | $N = 2^{7}$ | $L_{\infty}$ | $R_e$ | $L_{rms}$ | CPU time (in seconds) |
|-----|-------------|-------------|------|-----------|-----------------------|
| 1   | 4           | 8.364591×10^{-06} | —    | 8.534419×10^{-05} | 0.001659              |
| 2   | 8           | 7.29663×10^{-06}  | 1.70501 | 2.442521×10^{-05} | 0.000854              |
| 3   | 16          | 2.004567×10^{-06} | 1.87242 | 9.66233×10^{-06}  | 0.001950              |
| 4   | 32          | 8.870291×10^{-06} | 3.94111 | 3.114947×10^{-04} | 0.002732              |
| 5   | 64          | 4.515910×10^{-05} | 1.94342 | 1.165751×10^{-04} | 0.001950              |
| 6   | 128         | 2.144391×10^{-05} | 2.01186 | 3.921467×10^{-05} | 4.193567              |
| 7   | 256         | 1.080362×10^{-05} | 1.99289 | 1.396304×10^{-05} | 8.4482617             |
| 8   | 512         | 5.310449×10^{-06} | 1.99649 | 4.928097×10^{-06} | 14.655901             |

**Fig. 1.** Approximate and analytical solution comparison for 32 discrete CPs of Problem 1.

### Table 2. Comparison of HWC technique with Results of [27] for Problem 2.

| $T$ | $N = 2^{7}$ | $L_{\infty}$ | $R_e$ | $L_{rms}$ | CPU time (in seconds) |
|-----|-------------|-------------|------|-----------|-----------------------|
| 1   | 4           | 3.92×10^{-06} | —    | 4.440912×10^{-16} | 0.001659              |
| 2   | 8           | 6.97×10^{-06} | 1.894 | 4.40931×10^{-16}  | 0.002059              |
| 3   | 16          | 8.86×10^{-06} | 1.9447| 8.81837×10^{-16}  | 0.007111              |
| 4   | 32          | 9.38×10^{-06} | 1.9717| 8.81829×10^{-16}  | 0.018102              |
| 5   | 64          | 8.48×10^{-06} | 1.9856| 1.332317×10^{-13} | 0.066990             |
| 6   | 128         | 6.28×10^{-06} | 1.9928| 8.818088×10^{-16} | 0.279974              |
| 7   | 256         | 3.14×10^{-06} | 1.9964| 1.332391×10^{-13} | 1.299999              |
| 8   | 512         | 3.91×10^{-07} | 1.9986| 1.776453×10^{-13} | 5.440151              |
| 9   | 1024        | 3.63×10^{-06} | 1.9991| 1.793221×10^{-13} | 16.675396             |
| 10  | 2048        | 5.84×10^{-06} | 1.9992| 1.889947×10^{-13} | 31.400251             |

**Fig. 2.** Approximate and analytical solution comparison for 32 discrete CPs of Problem 2.

### Table 3. $L_{\infty}$ and $L_{rms}$ errors for Problem 3.

| $T$ | $N = 2^{7}$ | $L_{\infty}$ | $R_e$ | $L_{rms}$ | CPU time (in seconds) |
|-----|-------------|-------------|------|-----------|-----------------------|
| 1   | 2           | 3.257415×10^{-02} | —    | 9.399706×10^{-03} | 0.008904              |
| 2   | 4           | 1.877708×10^{-02} | 1.894 | 3.880102×10^{-03} | 0.004944              |
| 3   | 2           | 1.031141×10^{-02} | 1.944 | 1.289138×10^{-03} | 0.004036              |
| 4   | 5           | 4.942115×10^{-03} | 1.971 | 4.790248×10^{-04} | 0.0025743             |
| 5   | 2           | 2.954365×10^{-03} | 1.9855| 1.628386×10^{-04} | 3.960979              |
| 6   | 2           | 1.225109×10^{-03} | 1.992 | 5.961403×10^{-05} | 6.390869              |
| 7   | 2           | 6.149329×10^{-04} | 1.996 | 2.056519×10^{-05} | 13.611565              |
| 8   | 2           | 3.050115×10^{-04} | 1.998 | 7.276324×10^{-06} | 23.519690             |
| 9   | 2           | 1.523437×10^{-04} | 1.999 | 2.578212×10^{-06} | 38.619530             |
Problem 4. Consider delay Volterra IE [27]

\[ u(t) = \begin{cases} 
  f(t) + \int_{-0.5}^{0} w(s)ds, & \text{if } 0.5 \leq t < 0 \\
  e^{t}, & \text{if } t < 0 
\end{cases} \tag{27} \]

The function \( f(t) \) is so that the analytical solution is \( u(t) = e^{t} \).

Problem 5. Consider Volterra-Fredholm integral equation [35]

\[ u(t) = f(t) + \int_{-1}^{1} \left( 2t + 3t^2 \right) w(s)ds + \int_{-1}^{1} (2t-s)w(s)ds, \tag{28} \]

the delay condition is \( \Phi(t) = 3t - 1 \). The analytical solution is \( u(t) = 3t - 1 \).
Declarations

Author contribution statement

K. Shah: Conceived and designed the experiments; Performed the formal analysis; Analyzed and interpreted the data; Numerical interpretation and explanation.

R. Amin: Performed the formal analysis; Analyzed and interpreted the data; Numerical interpretation and explanation.

M. Asif, I. Khan: Numerical interpretation and explanation; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Acknowledgements

The authors would like to express their sincere thanks to the referees for their careful review of this manuscript and their useful suggestions which led to an improved version.

References

[1] A.M. Wazwaz, A First Course in Integral Equations, World Scientific, London, 2015.
[2] Z. Yang, H. Brunner, Blow-up behavior of Hammerstein-type delay Volterra integral equations, Front. Math. China 8 (2) (2013) 261–280.
[3] A. Bellour, M. Bousselal, A Taylor collocation method for solving delay integral equations, Numer. Algorithms 65 (4) (2014) 843–857.
[4] M. Mosleh, M. Onali, Least squares approximation method for the solution of Hammerstein Volterra delay integral equations, Appl. Math. Comput. 258 (2015) 105–110.
[5] A.M. Bica, C. Popescu, Numerical solutions of the nonlinear fuzzy Hammerstein Volterra delay integral equations, Inf. Sci. 223 (2013) 236–255.
[6] M. Khasi, F. Ghorbati, M. Hadizadeh, Numerical analysis of a high order method for state dependent delay integral equations, Numer. Algorithms 66 (1) (2014) 177–201.
[7] V. Horvat, On collocation methods for Volterra integral equations with delay arguments, Math. Commun. 4 (1) (1999) 92–109.
[8] G.P. Rao, K.R. Palanimanik, T. Srinivasan, Extension of computation beyond the limit of initial normal interval in Walsh series analysis of dynamical systems, IEEE Trans. Autom. Control 25 (2) (1980) 317–329.
[9] K. Balachandran, K. Murugesan, Analysis of different systems via single-term Walsh series method, Int. J. Comput. Math. 33 (3) (1990) 171–179.
[10] K. Balachandran, K. Murugesan, Optimal control of singular systems via single-term Walsh series, Int. J. Comput. Math. 43 (3) (1992) 153–159.
[11] K. Murugesan, D.P. Dhanayabaran, E.H. Amirtharaj, A study of second-order state-space systems of time-invariant and time-varying transistor circuits using the stws technique, Int. J. Electron. 89 (4) (2002) 305–315.
[12] A.E. Pushpam, D. Dhayabaran, E. Amirtharaj, Numerical solution of higher order systems of IVPs using generalized STWS technique, Appl. Math. Comput. 180 (1) (2006) 200–205.
[13] F. Mirzazee, N. Samadzay, Convergence of 2D-orthonormal Bernstein collocation method for solving 2D-mixed Volterra-Fredholm integral equations, Trans. A. Razmadze Math. Inst. 172 (3) (2018) 631–641.
[14] F. Mirzazee, E. Hadadiyan, Using operational matrix for solving nonlinear class of mixed Volterra-Fredholm integral equations, Math. Methods Appl. Sci. 40 (10) (2016) 3433–3444.
[15] F. Mirzazee, Numerical solution of nonlinear Fredholm-Volterra integral equations via Bell polynomials, Comput. Methods Differ. Equ. 5 (2) (2017) 88–102.
[16] F. Mirzazee, E. Hadadiyan, Numerical solution of Volterra-Fredholm integral equations via modification of hat functions, Appl. Math. Comput. 280 (2016) 110–123.
[17] F. Mirzazee, E. Hadadiyan, A new numerical method for solving two-dimensional Volterra-Fredholm integral equations, J. Appl. Math. Comput. 52 (1–2) (2016) 489–513.
[18] F. Mirzazee, S.F. Hoseini, Application of Fibonacci collocation method for solving Volterra-Fredholm integral equations, Appl. Math. Comput. 273 (2016) 637–644.
[19] F. Mirzazee, E. Hadadiyan, Applying the modified block-pulse functions to solve the three dimensional Volterra-Fredholm integral equations, Appl. Math. Comput. 265 (2015) 759–767.
[20] F. Mirzazee, E. Hadadiyan, S. Bimesl, Numerical solution for three-dimensional nonlinear mixed Volterra-Fredholm integral equations via three-dimensional block-pulse functions, Appl. Math. Comput. 237 (2014) 168–175.
[21] V. Vampa, M.T. Martin, E. Serrano, A hybrid method using wavelets for the numerical solution of boundary value problems, Appl. Math. Comput. 217 (2010) 3355–3367.
[22] G. Chiavasina, M. Guichoa, J. Liandrat, Two adaptive wavelet algorithms for nonlinear parabolic partial differential equations, Comput. Fluids 31 (2002) 467–480.
[23] U. Lepik, Haar wavelet method for solving integral equations and evolution equations, in: European Conference on Computational Mechanics 255(1), 2006, pp. 44.
[24] J.L. Wu, A wavelet operational method for solving fractional partial differential equations numerically, Appl. Math. Comput. 214 (2009) 31–40.
[25] L. Aziz, R. Amin, A numerical solution of a class of delay differential and delay partial differential equations via Haar wavelet, Appl. Math. Model. 40 (2016) 10286–10299.
[26] K. Maleknejad, F. Mirzazee, Using rationalized Haar wavelet for solving linear integral equations, Appl. Math. Comput. 160 (2) (2005) 579–587.
[27] R.G. Sekar, K. Murugesan, Numerical solutions of delay Volterra integral equations using single term Walsh series approach, Int. J. Appl. Comput. Math. 3 (2017) 2409–2421.
[28] R. Reienhofer, S. Bosse, G. Kutyniok, T. Wiegand, A Haar wavelet-based perceptual similarity index for image quality assessment, Signal Process. Image Commun. 61 (2018) 33–43.
[29] B.J. Adiguna, A.G. Buchan, A. Adama, S. Dargavillea, M.A. Goffina, A Haar wavelet method for angularly discretising the Boltzmann transport equation, Proc. Nucl. Energy 108 (2018) 295–309.
[30] Q. Dai, Q. Cao, Y. Chen, Frequency analysis of rotating truncated conical shells using the Haar wavelet method, Appl. Math. Model. 57 (2018) 603–613.
[31] R. Singh, H. Garg, V. Guleria, Haar wavelet(collocation) method for Lane–Emden equations with Dirichlet, Neumann and Neumann–Robin boundary conditions, J. Comput. Appl. Math. 346 (2019) 150–161.
[32] M.M. Khanan, R. Amin, M.I. Syam, A new algorithm for fractional Riccati type differential equations by using Haar wavelet, Mathematics 7 (2019) 545–557.
[33] R. Amin, S. Nazir, I.G. Magarino, Efficient sustainable algorithm for numerical solution of nonlinear delay Fredholm-Volterra integral equations via Haar wavelet for dense sensor networks in emerging telecommunications, Trans. Emerg. Telecomm. Technol. 30 (11) (2020) 1–12.
[34] R. Amin, S. Nazir, I.G. Magarino, A collocation method for numerical solution of nonlinear delay integro-differential equations for wireless sensor network and Internet of things, Sensors 20 (2020) s1862-1973.
[35] F. Gommanjani, M.H. Farahi, N. Pariz, A new approach for numerical solution of a linear system with distributed delays, Volterra delay-integro-differential equations and nonlinear Volterra-Fredholm integral equation by Bezier curves, Comput. Appl. Math. 36 (2017) 1349–1365.
[36] J. Majak, B. Shvartsman, M. Kiris, M. Pohlak, H. Herrana, Convergence theorem for the Haar wavelet based discretization method, Compos. Struct. 126 (2015) 227–232.