International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

NUMERICAL SOLUTION OF SYSTEMS OF INTEGRO DIFFERENTIAL EQUATIONS USING POLYNOMIAL COLLOCATION METHOD

B. A. Binta[1*], M. R. Odekunle[2], D. Umar[3], H. U. Waniyos[4]

1. Department of Mathematics, Adamawa State Polytechnic, Yola, Nigeria.
2,3. Department of Mathematics, Modibbo Adama University Yola, Nigeria.
4. Department of Statistics, Adamawa State Polytechnic, Yola, Nigeria.
* Corresponding author: binta.abubakar@adamawapoly.edu.ng

Article Info
Received: 14 August 2023 Revised: 18 October 2023
Accepted: 18 October 2023 Available online: 30 October 2023

Abstract

This paper deals with the development of a new numerical scheme to solve systems of linear integro differential equation under mixed conditions. The new method adopted the use of standard collocation points to transform the state equations into linear algebraic equations. These equations are then solved using MATLAB programming through the matrix inversion technique to obtain the unknowns. The convergence of the method is established and numerical examples are solved and compared with existing results to confirm the efficiency of the new method.

Keywords: Integro Differential Equation, Matrix, Mixed Condition, Polynomial Collocation.
MSC2010: 45J05, 47G20, 65D30, 65F05.

1 Introduction

One of the simplest integro differential models studied in literature is the Volterra-Kostitzin model, which is used for describing the evolution of a population in a closed environment, [1], [2], [3], [4]. Integro Differential Equation (IDE) is an equation in which the unknown function $u(x)$ appears under an integral sign and contains an ordinary derivatives, [5]. IDE can be represented in the form

$$u^{(n)}(x) = z(x) + \int_{g(x)}^{h(x)} K(x, t, u(t))\, dt, \tag{1.1}$$

where $h(x)$ and $g(x)$ are the limits of integration and the initial condition is

$$u^{(k)}(a) = u_k, \tag{1.2}$$

where $z(x)$ and $K(x,t)$ are assumed to satisfy the existence and uniqueness theorem. However, $z(x)$ and $K(x,t)$ are given real valued functions which are continuous together with their first derivative. If the limits are fixed then, (1.1) is said to be Fredholm integro differential equation. If at least one limit is a variable then, (1.1) is said to be Volterra integral equation, [5]. IDEs have been used to model heat and mass diffusion processes, biological species coexistence together with increasing and decreasing rate of growth, electromagnetic theory and ocean circulation among others, [6], [7], [8], [9] and [10].

[11] considered operational matrix method for the solution of linear differential difference equations without application to system of differential and integro differential difference equations. [12] applied operational matrix method to solve system of linear Fredholm integro differential equations without applications to Volterra type and differential difference equations. [13] solved Volterra integro differential equations using operational matrix without considering system of differential and difference equations. It should be noted that in general, operational matrix is efficient if the kernel of integration can be expanded in Taylor's series, [13]. Since it is possible to have kernel that is not expandable in Taylor's series, this became a set back for operational matrix. In this work, the works of [11], [12] and [13] are combined and extended so that the operational matrix technique becomes efficient and very easy to implement when the kernel is not expanded.

We consider a system of linear IDE in the form

$$
\begin{aligned}
& {}_{n=0}^{N}{}_{j=1}^{S} P_{ij}^{n}(x)\, u_j^{(n)}(x) + {}_{r=0}^{R}\,{}_{j=1}^{S} q_{ij}^{r}(x)\, u_j^{(r)}(x+\tau) \\
= \;& g_j(x) + \int_0^x {}_{j=1}^{S} K_{ij}(x,t)\, u_j(t)\, dt + \int_0^1 {}_{j=1}^{S} w_{ij}(x,t)\, u_j(t)\, dt,
\end{aligned} \tag{1.3}
$$

subject to the mixed condition

$$
{}_{j=0}^{m-1} a_{ij}^{(n)} u_n^{(i)}(a) + b_{ij}^{(n)} u_n^{(j)}(b) = \lambda_{n,i},\; i = 1,2,...,S, n = 0,1...m-1. \tag{1.4}
$$

Where $u_j : J \to \mathbb{R}^N$, $J \in [0,1]$. The solution to (1.3) and (1.4) is a continuous function to be determined, that is, $u \in C[J, \mathbb{R}^N)$. $P_{ij}^n(x)$, $q_{ij}^r(x)$, $g_j : J \to \mathbb{R}^N$ are given continuous functions. $K, w : J \times J \to \mathbb{R}^N$ are Lipschitz continuous, a_{ij}, b_{ij}, $\lambda_{n,i}$ are real constants. The aim of this work is to develop a new polynomial collocation method that can handle systems of Fredholm, Volterra and Volterra-Fredholm IDEs.

2 Matrix Formulation of State Equation

Let (1.3) be written in the form

$$
\phi_1(x) + \phi_2(x) = g_j(x) + \phi_3(x) + \phi_4(x) \tag{2.1}
$$

where

$$
\phi_1(x) = \sum_{n=0}^{N} \mathbf{P}_n(x)\, \mathbf{u}^{(n)}(x) \tag{2.2}
$$

and $\mathbf{P}_n(x) = \begin{bmatrix} p_{11}^n & p_{12}^n & p_{13}^n & \cdots & p_{1s}^n \\ p_{21}^n & p_{22}^n & p_{23}^n & \cdots & p_{2s}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{s1}^n & p_{s2}^n & p_{s3}^n & \cdots & p_{ss}^n \end{bmatrix}$, $\mathbf{u}^{(n)}(x) = \begin{bmatrix} u_1^{(n)} & u_2^{(n)} & \cdots & u_s^{(n)} \end{bmatrix}^T$.

Similarly,

$$
\phi_2(x) = \sum_{r=0}^{R} \mathbf{q}_r(x)\, \mathbf{u}^{(r)}(x)\, \mathbf{M} \tag{2.3}
$$

where, $\mathbf{q}_r(x) = \begin{bmatrix} q_{11}^r & q_{12}^r & \cdots & q_{1s}^r \\ q_{21}^r & q_{22}^r & \cdots & q_{2s}^r \\ \vdots & \vdots & \vdots & \vdots \\ q_{s1}^r & q_{s2}^n & \cdots & q_{ss}^r \end{bmatrix}$. $\mathbf{M} = \begin{bmatrix} \mathbf{M}_{-r} & 0 & \cdots & 0 \\ 0 & \mathbf{M}_{-r} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{M}_{-r} \end{bmatrix}$,

International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

where, $\mathbf{M}_{-r} = \begin{bmatrix} \binom{0}{0}(-\tau)^0 & \binom{1}{0}(-\tau)^1 & \binom{2}{0}(-\tau)^2 & \cdots & \binom{N}{0}(-\tau)^N \\ 0 & \binom{1}{1}(-\tau)^0 & \binom{2}{1}(-\tau)^1 & \cdots & \binom{N}{1}(-\tau)^{N-1} \\ 0 & 0 & \binom{2}{2}(-\tau)^0 & \cdots & \binom{N}{2}(-\tau)^{N-2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \binom{N}{N}(-\tau)^0 \end{bmatrix}$.

$$\phi_3(x) = \int_0^x \mathbf{K}(x,t)\,\mathbf{u}(t)\,dt \tag{2.4}$$

where, $\mathbf{K}(x,t) = \begin{bmatrix} K_{11}(x,t) & K_{12}(x,t) & \cdots & K_{1s}(x,t) \\ K_{21}(x,t) & K_{22}(x,t) & \cdots & K_{2s}(x,t) \\ \vdots & \vdots & \vdots & \vdots \\ K_{s1}(x,t) & K_{s2}(x,t) & \cdots & K_{ss}(x,t) \end{bmatrix}$.

$$\phi_4(x) = \int_0^1 \mathbf{w}(x,t)\,\mathbf{u}(t)\,dt \tag{2.5}$$

where, $\mathbf{w}(x,t) = \begin{bmatrix} w_{11}(x,t) & w_{12}(x,t) & \cdots & w_{1s}(x,t) \\ w_{21}(x,t) & w_{22}(x,t) & \cdots & w_{2s}(x,t) \\ \vdots & \vdots & \vdots & \vdots \\ w_{s1}(x,t) & w_{s2}(x,t) & \cdots & w_{ss}(x,t) \end{bmatrix}$.

$$g_j(x) = \mathbf{G}(x) \tag{2.6}$$

where, $\mathbf{G}(x) = \begin{bmatrix} g_1(x) & g_2(x) & \ldots & g_s(x) \end{bmatrix}^T$.
Therefore, (3) reduces to

$$\,_{n=0}^{N}\mathbf{P}_n(x)\,\mathbf{u}^{(n)}(x) = -\,_{r=0}^{R}\mathbf{q}_r(x)\,\mathbf{u}^{(r)}(x)\,\mathbf{M} + \mathbf{G}(x) + \int_0^x \mathbf{K}(x,t)\,\mathbf{u}(t)\,dt$$
$$+ \int_0^1 \mathbf{w}(x,t)\,\mathbf{u}(t)\,dt, \tag{2.7}$$

with mixed conditions

$$\,_{j=0}^{m-1}[\mathbf{a}_i\mathbf{u}^{(i)}(a) + \mathbf{b}_i\mathbf{u}^{(j)}(b)] = \lambda_i\ . \tag{2.8}$$

3 Method of Solution

Let $\mathbf{u}_j(x)$ be the approximate solution to (2.7)

$$\mathbf{u}_j(x) = \mathbf{X}_j(x)\,\mathbf{A}_j \tag{3.1}$$

where $\mathbf{X}_j = \begin{bmatrix} 1 & x_j & x_j^2 & \ldots & x_j^n \end{bmatrix}$ and $\mathbf{A}_j = \begin{bmatrix} a_{j0} & a_{j1} & \ldots & a_j^n \end{bmatrix}^T$ are constants to be determined.
Equation (3.1) can be written in matrix form

$$\mathbf{u}(x) = \mathbf{X}(x)\,\mathbf{A}. \tag{3.2}$$

Collocating using

$$x_i = a + \frac{(b-a)\,i}{N}$$

IJMSO

International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

and substituting (3.2) into (2.7) gives

$$\mathbf{D}\left(x_i\right)\mathbf{A}=\mathbf{G}\left(x_i\right), \tag{3.3}$$

where $\mathbf{D}\left(x_i\right)={}_{n=0}^{N}\mathbf{P}_n\left(x_i\right)\mathbf{X}^{(n)}\left(x_i\right)+{}_{r=0}^{R}\mathbf{q}_r\left(x_i\right)\mathbf{u}^{(r)}\left(x_i\right)\mathbf{M}-\int_0^x\mathbf{K}\left(x_i,t\right)\mathbf{X}\left(t\right)dt-\int_0^1\mathbf{w}\left(x,t\right)\mathbf{X}\left(t\right)dt.$
Substituting (3.2) into the mixed condition (2.8)

$$\mathbf{D}_i\mathbf{A}={}_{j=0}^{m-1}a_{ij}\mathbf{X}^{(i)}\left(a\right)+b_{ij}\mathbf{X}^{(j)}\left(b\right) \tag{3.4}$$

and augmenting (3.3) and (3.4) gives

$$\begin{bmatrix}\mathbf{D}\left(x_i\right)\\ \mathbf{D}_i\end{bmatrix}\mathbf{A}=\begin{bmatrix}\mathbf{G}\left(x_i\right)\\ \lambda_i\end{bmatrix}. \tag{3.5}$$

Solving for the unknown constants $\mathbf{A}$ in (3.5) yields

$$\mathbf{A}=\gamma^{-1}\left(x_i\right)\mathbf{V}\left(x_i\right), \tag{3.6}$$

where $\gamma\left(x_i\right)=\begin{bmatrix}\mathbf{D}\left(x_i\right)\\ \mathbf{D}_i\end{bmatrix}^T\begin{bmatrix}\mathbf{D}\left(x_i\right)\\ \mathbf{D}_i\end{bmatrix}$, $\mathbf{V}\left(x_i\right)=\begin{bmatrix}\mathbf{D}\left(x_i\right)\\ \mathbf{D}_i\end{bmatrix}^T\begin{bmatrix}\mathbf{G}\left(x_i\right)\\ \lambda_i\end{bmatrix}$.
Equation (3.6) is then substituted into the approximate solution (3.1) to give the numerical solutions.

4 Convergence of Method

Convergence of solution. Let $\mathbf{u}_N\left(x\right)$ be the approximate solution and $\mathbf{u}\left(x\right)$ be the exact solution to (11). If $\mathbf{G}\left(x\right)$ be a continuous function defined on $J=[0,1]$, $\mathbf{K}\left(x,t\right)$, $\mathbf{w}\left(x,t\right)$ are continuous functions, then the method converges iff

$$\frac{\left\|{}_{n=0}^{N}\mathbf{P}_n\mathbf{E}^{(n)}\right\|_\infty+\left|\mathbf{M}\right|\left\|{}_{r=0}^{R}q_r\left(x\right)\mathbf{E}^{(r)}\right\|_\infty}{\left\|\mathbf{E}\right\|_\infty}\leq\mathbf{K}^*+\mathbf{w}^*$$

where $\mathbf{K}^*=\sup_{x,t\in J}\int_0^x\left|\mathbf{K}(x,t)\right|dt$, $\mathbf{w}^*=\sup_{x,t\in J}\int_0^1\left|\mathbf{w}(x,t)\right|dt$

Proof. Substituting the approximate solution $\mathbf{u}_N\left(x\right)$ into (2.7),

$$\begin{aligned}&{}_{n=0}^{N}\mathbf{P}_n\left(x\right)\mathbf{u}_N^{(n)}\left(x\right)=-{}_{r=0}^{R}\mathbf{q}_r\left(x\right)\mathbf{u}_N^{(r)}\left(x\right)\mathbf{M}+\mathbf{G}\left(x\right)\\&+\int_0^x\mathbf{K}\left(x,t\right)\mathbf{u}_N\left(t\right)dt+\int_0^1\mathbf{w}\left(x,t\right)\mathbf{u}_N\left(t\right)dt.\end{aligned} \tag{4.1}$$

Subtracting (2.7) from (4.1) gives

$$\begin{aligned}&{}_{n=0}^{N}\mathbf{P}_n\left(x\right)(\mathbf{u}_N^{(n)}\left(x\right)-\mathbf{u}^{(n)}\left(x\right))=-{}_{r=0}^{R}\mathbf{q}_r\left(x\right)(\mathbf{u}_N^{(r)}\left(x\right)-\mathbf{u}^{(r)}\left(x\right))\mathbf{M}\\&+\int_0^x\mathbf{K}\left(x,t\right)(\mathbf{u}_N\left(t\right)-\mathbf{u}\left(t\right))dt+\int_0^1\mathbf{w}\left(x,t\right)(\mathbf{u}_N\left(t\right)-\mathbf{u}\left(t\right))dt.\end{aligned}$$

Taking the absolute values:

$$\begin{aligned}&\left|{}_{n=0}^{N}\mathbf{P}_n\left(x\right)(\mathbf{u}_N^{(n)}\left(x\right)-\mathbf{u}^{(n)}\left(x\right))\right|\leq\left|-{}_{r=0}^{R}\mathbf{q}_r\left(x\right)(\mathbf{u}_N^{(r)}\left(x\right)-\mathbf{u}^{(r)}\left(x\right))\mathbf{M}\right|\\&+\left|\int_0^x\mathbf{K}\left(x,t\right)(\mathbf{u}_N\left(t\right)-\mathbf{u}\left(t\right))dt\right|+\left|\int_0^1\mathbf{w}\left(x,t\right)(\mathbf{u}_N\left(t\right)-\mathbf{u}\left(t\right))dt\right|\end{aligned}$$

$$\leq \quad - \sup_{x,t\in J} {}_{r=0}^{R} \left| \mathbf{q}_r(x) \left(\mathbf{u}_N^{(r)}(x) - \mathbf{u}^{(r)}(x) \right) \right| |\mathbf{M}| + \sup_{x,t\in J} \int_0^x |\mathbf{K}(x,t)| \sup_{t\in J} |(\mathbf{u}_N(t) - \mathbf{u}(t))| \, dt$$
$$+ \sup_{x,t\in J} \int_0^1 |\mathbf{w}(x,t)| \sup_{t\in J} |(\mathbf{u}_N(t) - \mathbf{u}(t))| \, dt$$

$$\left\| {}_{n=0}^{N} \mathbf{P}_n (\mathbf{u}_N^{(n)} - \mathbf{u}^{(n)}) \right\|_\infty \leq -{}_{r=0}^{R} \left\| \mathbf{q}_r (\mathbf{u}_N^{(r)} - \mathbf{u}^{(r)}) \right\|_\infty |\mathbf{M}| + \mathbf{K}^* \|\mathbf{u}_N - \mathbf{u}\|_\infty + \mathbf{w}^* \|\mathbf{u}_N - \mathbf{u}\|_\infty$$

Therefore,

$$\frac{\| {}_{n=0}^{N} \mathbf{P}_n \mathbf{E}^{(n)} \|_\infty + |\mathbf{M}| \, \| {}_{r=0}^{R} q_r \mathbf{E}^{(r)} \|_\infty}{\|\mathbf{E}\|_\infty} \leq \mathbf{K}^* + \mathbf{w}^*,$$

where $\mathbf{E} = \mathbf{u}_N - \mathbf{u}$. This implies that the error is bounded. □

5 Illustrative Examples

In this section, we present numerical examples to test the efficiency and simplicity of the method. Let $\mathbf{u}_N(x)$ and $\mathbf{u}(x)$ be the approximate and exact solutions respectively, e_N is the error function, then $abs - e_N = |\mathbf{u}_N - \mathbf{u}|$ is the absolute error at N. All results are presented in tables except where the $abs - e_N = 0$. All computations are done with the aid of program written in MATLAB (2015a).

Example 5.1. *: [12]. We considered a second order system of initial value integro differential equations with variable coefficients*

$$y_1^{(2)}(x) - x y_2^{(1)}(x) - y_1(x) = (x-2)\sin x + \int_0^1 (x\cos t \; y_1(t) - x \sin t \; y_2(t)) \, dt$$

$$y_2^{(2)}(x) - 2x y_1^{(1)}(x) + y_2(x) = -2x\cos x + \int_0^1 (\sin x \cos t \; y_1(t) - \sin x \sin t \; y_2(t)) \, dt$$

subject to the conditions $y_1(0) = 0, y_1^{(1)}(0) = 1, y_2(0) = 1, y_2^{(1)}(0) = 0.$

The exact solution is $y_1(x) = \sin x$, $y_2(x) = \cos x$. When $N = 2$, (2.7) gives, $\mathbf{p}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{p}_1 = \begin{bmatrix} 0 & -x \\ -2x & 0 \end{bmatrix}$, $\mathbf{p}_0 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} (x-2)\sin x \\ -2x\cos x \end{bmatrix}$, $\mathbf{K} = 0$, $\mathbf{q}_r = 0$ $\mathbf{w} = \begin{bmatrix} x\cos t & -x\sin t \\ \sin x\cos t & -\sin x \sin t \end{bmatrix}$, $x_i = \begin{bmatrix} 0 & \frac{1}{2} & 1 \end{bmatrix}$. (3.5) gives

$$\mathbf{D}(x_i) = \begin{bmatrix} -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ -1.42 & -0.691 & 1.63 & 0.23 & -0.349 & -0.388 \\ -0.403 & -1.18 & -1.11 & 1.22 & 0.644 & 2.36 \\ -1.84 & -1.38 & 0.761 & 0.46 & -0.699 & -1.78 \\ -0.708 & -2.32 & -4.2 & 1.39 & 1.25 & 3.19 \end{bmatrix},$$

$$\mathbf{D}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \lambda = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T.$$

$$\mathbf{G}(x_i) = \begin{bmatrix} 0 & 0 & -0.719 & -0.878 & -0.841 & -1.08 \end{bmatrix}^T$$

International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

and (3.6) gives $\mathbf{V} = \begin{bmatrix} -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, $\gamma = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^T$,
$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & \frac{-1}{2} \end{bmatrix}^T$. Hence, $\mathbf{u}(x) = \begin{bmatrix} x & 1-0.5x^2 \end{bmatrix}^T$.
For comparison with [12], we use $N = 3, 7$ and 12 so that

$$y_{3,1}(x) = -0.1628939579524x^3 + x$$

$$y_{3,2}(x) = 0.0276492396912x^3 - 0.5x^2 + 1$$

$$y_{7,1}(x) = \begin{array}{c} -0.0001849912351x^7 - 0.00002016875697x^6 + 0.008346135045x^5 \\ -0.000003935762126x^4 - 0.1666661537x^3 + x \end{array}$$

$$y_{7,2}(x) = \begin{array}{c} 6.916118637e - 5x^7 - 0.001466408531x^6 + 4.372835616e - 5x^5 \\ +0.04165402127x^4 + 1.581552737e - 6x^3 - 0.5x^2 + 1 \end{array}$$

$$y_{12,1}(x) = \begin{array}{c} 8.422674098e - 10x^{12} - 2.704068073e - 8x^{11} + 2.757546666e - 9x^{10} \\ +2.753251326e - 6x^9 + 1.512520071e - 9x^8 - 0.0001984133328x^7 \\ +1.816843037e - 10x^6 + 0.008333333299x^5 + 3.972132716e - 12x^4 \\ -0.1666666667x^3 + x \end{array}$$

$$y_{12,2}(x) = \begin{array}{c} 1.904457069e - 9x^{12} + 5.816757768e - 10x^{11} - 2.764833009e - 7x^{10} \\ +8.733845028e - 10x^9 + 2.480103345e - 5x^8 + 2.382978721e - 10x^7 \\ -0.001388888958x^6 + 1.328481409e - 11x^5 + 0.04166666667x^4 \\ +8.80036864e - 14x^3 - 0.5x^2 + 1 \end{array}$$

Table 1: Comparison of Absolute Error with Existing Method for Example 1

		[12]			**Present Method**		
x_i	**y**	abs-e$_3$	abs-e$_7$	abs-e$_{12}$	abs-e$_3$	abs-e$_7$	abs-e$_{12}$
0.2	y_1	2.0231e-05	1.8518e-09	9.9920e-15	2.7518e-05	7.8235e-10	9.7317e-17
	y_2	1.4732e-04	3.6087e-10	9.9920e-15	1.5461e-04	2.2734e-09	5.0953e-18
0.4	y_1	9.8150e-05	1.9768e-08	5.9952e-14	1.5644e-04	1.8186e-09	2.4382e-16
	y_2	6.5019e-04	1.6751e-08	8.0047e-14	7.0856e-04	4.8438e-09	1.6573e-17
0.6	y_1	2.4311e-05	7.2678e-08	2.1994e-13	1.7243e-04	3.2199e-09	3.6121e-16
	y_2	4.3962e-04	6.8547e-08	2.8999e-13	6.3662e-04	7.4939e-09	1.3188e-17
0.8	y_1	1.2241e-03	1.8555e-07	5.6999e-13	7.5780e-04	5.2379e-09	5.4323e-16
	y_2	3.0173e-03	1.8148e-07	7.1998e-13	2.5503e-03	1.0383e-08	2.4906e-17
1.0	y_1	5.0207e-03	5.0207e-07	1.2126e-12	4.3650e-03	9.9254e-08	1.6014e-14
	y_2	1.3565e-02	6.3006e-07	1.5526e-12	1.2653e-02	2.2203e-07	7.6675e-15

Example 5.2. *: [14], considered system of Volterra integro differential equation*

$$y_1^{(2)}(x) + 2xy_1^{(1)}(x) - y_1(x) = f_1(x) + \int_0^x y_1(t)\,dt - \int_0^x y_2(t)\,dt$$

$$y_2^{(2)}(x) + y_2^{(1)}(x) - 2xy_2(x) = f_2(x) + \int_0^x y_1(t)\,dt + \int_0^x y_2(t)\,dt$$

$f_1(x) = 2 + x - e^x + 2xe^x - \cos x$, $f_2(x) = -\sin x + 2\cos x - 3x - 2x\sin x - e^x$, *subject to the conditions* $y_1(0) = 1, y_1^{(1)}(0) = 1, y_2(0) = 1, y_2^{(1)}(0) = 1$. *The exact solution is* $y_1(x) = e^x$, $y_2(x) = 1 + \sin x$. (11) *gives* $\mathbf{p}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{p}_1 = \begin{bmatrix} 2x & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{p}_0 = \begin{bmatrix} -1 & 0 \\ 0 & -2x \end{bmatrix}$, $\mathbf{K} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$,
$\mathbf{G} = \begin{bmatrix} 2 + x - e^x + 2xe^x - \cos x \\ -\sin x + 2\cos x - 3x - 2x\sin x - e^x \end{bmatrix}$, $\mathbf{q}_r = 0$ $\mathbf{w} = 0$,

IJMSO

International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

$x_i = \begin{bmatrix} 0 & \frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{3}{5} & 1 \end{bmatrix}$. For comparison with [14], we shall consider the case when $N = 5$, so that

$$y_{5,1}(x) = \begin{array}{c} 0.0112827160x^5 + 0.0397852411x^4 + 0.1670814127x^3 + 0.5x^2 \\ +x+1 \end{array}$$

$$y_{5,2}(x) = 0.0079293422x^5 + 0.00031747361x^4 - 0.1667404937x^3 + x + 1$$

Table 3: Comparison of Absolute Error with Existing Method for Example 2

	[14], N=5		Present Method, N=5	
x_i	y1	y2	y1	y2
0.1	1.0000e-09	0.0000e-00	2.5468e-07	4.6099e-08
0.2	9.1000e-08	1.9999e-09	1.1600e-06	2.0940e-07
0.3	1.0580e-06	1.8200e-06	2.0680e-06	3.6015e-07
0.4	6.0310e-06	3.2500e-07	2.5500e-06	4.1012e-07
0.5	2.4101e-05	1.5440e-06	3.0683e-06	4.6628e-07
0.6	7.0800e-05	5.5270e-06	4.2960e-06	6.8982e-07
0.7	8.1290e-05	1.6230e-05	4.9396e-06	7.6664e-07
0.8	4.0126e-04	4.1242e-05	2.0900e-06	1.1003e-06
0.9	8.4486e-04	9.3840e-05	3.5334e-05	9.7621e-06
1.0	1.6152e-03	1.9568e-04	1.3246e-04	3.5337e-05

Example 5.3. *: [15], considered the linear system of Volterra-Fredholm integro differential equations,*

$$y_1^{(2)}(x) = 6x - \frac{3x^2}{4} - \frac{x^6}{5} + \int_0^x xty_1(t)\,dt + \int_0^1 x^2ty_2(t)\,dt$$

$$y_2^{(2)}(x) = 2 - \frac{x^5}{5} - \frac{3}{2}x + \int_o^x ty_1(t)\,dt + \int_0^1 2xty_2(t)\,dt$$

within the interval $0 \le x \le 1$, *subject to initial conditions* $y_1(0) = 0$, $y^{(1)}{}_1(0) = 0$, $y_2^{(1)}(0) = 0$, $y_2(0) = 1$, *with exact solution* $y_1(x) = x^3$, $y_2(x) = x^2 + 1$.

From (3.5), for $N = 3$,

$$\mathbf{D}(x_i) = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ \frac{-1}{54} & \frac{-1}{243} & \frac{1943}{972} & \frac{7289}{3645} & \frac{-1}{18} & \frac{-1}{27} & \frac{-1}{36} & \frac{-1}{45} \\ \frac{-1}{18} & \frac{-1}{81} & \frac{-1}{324} & \frac{-1}{1215} & \frac{-1}{3} & \frac{-2}{9} & \frac{11}{6} & \frac{28}{15} \\ \frac{-4}{27} & \frac{-16}{143} & \frac{478}{243} & \frac{14516}{3645} & \frac{-2}{9} & \frac{-4}{27} & \frac{-1}{9} & \frac{-4}{45} \\ \frac{-2}{9} & \frac{-8}{81} & \frac{-4}{81} & \frac{-32}{1215} & \frac{-2}{3} & \frac{-4}{9} & \frac{5}{3} & \frac{56}{15} \\ \frac{-1}{2} & \frac{-1}{3} & \frac{7}{4} & \frac{29}{5} & \frac{-1}{2} & \frac{-1}{3} & \frac{-1}{4} & \frac{-1}{5} \\ \frac{-1}{2} & \frac{-1}{3} & \frac{-1}{4} & \frac{-1}{5} & -1 & \frac{-2}{3} & \frac{3}{2} & \frac{28}{5} \end{bmatrix},$$

Solving for the unkowns yield

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ \frac{-1}{54} & \frac{-1}{243} & \frac{1943}{972} & \frac{7289}{3645} & \frac{-1}{18} & \frac{-1}{27} & \frac{-1}{36} & \frac{-1}{45} \\ \frac{-1}{18} & \frac{-1}{81} & \frac{-1}{324} & \frac{-1}{1215} & \frac{-1}{3} & \frac{-2}{9} & \frac{11}{6} & \frac{28}{15} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \gamma = \begin{bmatrix} 0 & 2 & 1.9 & 1.5 & 1 & 0 & 0 & 0 \end{bmatrix}^T$$

and $\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & -0.0595 \end{bmatrix}^T$.

Substituting the value of **A** into the approximate solution gives the exact solution.

IJMSO

International Journal of Mathematical Sciences and Optimization: Theory and Applications
Vol. 9, No. 2, pp. 44 - 52
https://doi.org/10.5281/zenodo.10035489

6 Discussion of Results and Conclusion

First, we considered a system of initial value integro differential equation with variable coefficients.We solved this problem for values of $N = 3, 7$, and 12. The absolute errors presented in Table 1 shows that as N increases, the absolute errors approaches zero; this means that the solution converges. The comparison of the Absolute Errors with the existing method as presented in Table 1 for $N = 3$, 7 and 12 shows that our method gives better approximation. The second example is a system of initial value second order Volterra IDE with variable coefficients. The result obtained as shown in Table 2 also has better approximation. Finally, the third example is a Volterra-Fredholm IDE whose approximate solutions at $N = 3$ are exact.

Systems of IDEs are difficult to solve analytically. It is often necessary to approximate the solutions. In this paper we developed a new method that efficiently solved systems of IDEs. In general, error propagation decreases and f_i. There is also great improvement as N increases. The method gives exact solutions if $f_i(x)\,; i = 1, 2, ..., n$ is a polynomial as seen in the third example. The results show that our method gives better approximations than the methods used in [12], [14] and [15]. All computations were performed with MATLAB (2015a).

7 Acknowledgments

The authors wish to thank the reviewers of this paper.

References

[1] Fazeli, S. & Hojjati, G. (2015). Numerical solution of Volterra integro-differential equations by superimplicit multistep collocation methods. *Numerical Algorithm* **68**, 741–768.

[2] Mehdiyeva, G., Imanova, M. & Ibrahim, V. (2011). On an application of hybrid method to the solution of Volterra nonlinear integro differential equation. doi: 10.1109/ICPCI.2012.6486392.

[3] Mohamed, N. A. & Majid, Z. A. (2016). Multistep block method for solving Volterra integro-differential equations. *Maiaysian Journal of Mathematical Science,* **10**(s), 33-48.

[4] Rahman, M. (2007). Integral Equations and their Applications, Southampton, Boston. WIT press.

[5] Wazwaz, A. M. (2011). Linear and Nonlinear Equations. Springer, Berlin.

[6] Agbolade, O. A. & Anake, T. A. (2017). Solution of first order Volterra linear integro differential equations by collocation method. *Journal of Applied Mathematics*, Article ID. 1510267.

[7] Akyaz, A. & Sezer, M. (2009). A Chebyshev collocation method for the solution of linear integro differential equations. *International Journal of Computer Mathematics,* **72, (4)**, 491-507.

[8] Arikoglu, A. & Ozkol, I. (2008). Solutions of integral and integro-differential equation systems by using differential transform method. *Computer and Mathematics with application*, **56**, 2411-2417.

[9] Biazar, J. & Ebrahimi H. (2011). A strong method for solving systems of integro differential equations. *Applied Mathematics*, **2**, 1105 - 1113.

[10] Day, J. T. (1967). Note on the numerical solution of integro-differential equations. *The Computer Journal*, **9**:394–395.

[11] Tugce, A. & Salih, Y. (2012). Boubakar polynomial approach for solving high order linear differential difference equations. *AIP Conf. proc.* **1493**, 26.12.

[12] Yuzbasi, S. Sachin, N. & Sezer, M. (2011). Numerical solution of systems of linear Fredholm integro differential equations with Bessel polynomial bases. *Computer and Mathematics with Application*, **61**, 3079-3096.

[13] Ordokhani, Y. & Dehestani, H. (2014). Numerical solution of the nonlinear Fredholm-Volterra-Hammerstein integral equations via Bessel functions. *Journal of Information and Computing Science*, **9(2)**, 123-131.

[14] Al-Faour, O. M. and Saeed, R. K. (2006). Solution of a system of linear Volterra integral and integro differential equations by spectral method. Journal of *Al-Nahran University,* **6 (2),** 30-46.

[15] Adnan, A. J., Nejmaddin, A. S., & Azad, I. A. (2019). Numerical methods for solving the system of Volterra Fredholm integro differential equations, *Zanco Journal of Pure and Applied Sciences*, doi:10.2127/ZJPAS.31.2.4.2019.