Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-22
DOI:10.22034/cmde.2025.65501.3016

A novel numerical approach for solving two-dimensional Volterra-Fredholm integral equations

Fatemeh Mohammadi, Farshid Mirzaee*, **Erfan Solhi, and Shiva Mahmoudi**

Department of Mathematics, Faculty of Mathematical Sciences and Statistics, Malayer University, Malayer, Iran.

Abstract

In this article, we propose a numerical method based on alternative Legendre polynomials for approximating the solutions of two-dimensional linear and nonlinear Volterra-Fredholm integral equations. Alternative Legendre polynomials, known for their some special features and simplicity in constructing operational matrices, provide an efficient basis for this method. By employing the integration and product operational matrices of alternative Legendre polynomials, the integral equations are reduced to a system of algebraic equations, simplifying the computational process. Error analysis is conducted to assess the method's accuracy, and several examples are presented to validate the high precision and efficiency of the proposed approach. The results confirm the accuracy and effectiveness of the method in solving complex integral equations.

Keywords. Alternative Legendre polynomials, Two-dimensional integral equations, Operational matrix, Error analysis.
2010 Mathematics Subject Classification. 65R20, 45B05, 45D05, 45G10.

1. Introduction

The study of integral equations, particularly Volterra-Fredholm integral equations, plays a crucial role in the mathematical modeling of various physical and engineering phenomena. These equations are often challenging to solve due to their inherent complexity, especially when dealing with nonlinearities and multiple dimensions. In recent years, numerous numerical techniques have been developed to address such complexities. The primary aim of this article is to introduce a novel numerical method for solving two-dimensional nonlinear Volterra-Fredholm integral equations of the form:

$$\begin{aligned} g(x,y) = f(x,y) &+ \lambda_1 \int_0^x \int_0^y \varphi_1(x,y,t,s)\theta_1(t,s,g(t,s))dtds \\ &+ \lambda_2 \int_0^1 \int_0^1 \varphi_2(x,y,t,s)\theta_2(t,s,g(t,s))dtds, \quad (x,y) \in D = [0,1]\times[0,1], \end{aligned} \tag{1.1}$$

where λ_1 and λ_2 are arbitrary constants, and $g(x,y)$ is the unknown function defined in D. The functions $f(x,y)$, $\varphi_1(x,y,t,s)$, and $\varphi_2(x,y,t,s)$ are known functions defined on D^2 and D^4, respectively. Also, the functions $\theta_\varpi(t,s,g(t,s))$, for $\varpi = 1,2$, are continuous on the domain $D = [0,1]\times(-\infty,+\infty)$ and depend on g in a nonlinear manner. The existence and uniqueness of the solution to two-dimensional nonlinear integral equations can be found in [1, 13, 22, 23, 26].

The problem involves both Volterra and Fredholm components, making it a mixed-type integral equation, which further adds to its complexity. Several methods have been developed for solving two-dimensional (2D) integral equations, each aiming to reduce the integral equation to a system of algebraic equations, facilitating easier computation. Notable contributions in this area include the works of Babolian et al. [2], Mirzaee et al. [12], Ordokhani et al. [20],

Received: 16 January 2025 ; Accepted: 03 September 2025.
* Corresponding author. Email: f.mirzaee@malayeru.ac.ir.

and Rashidinia et al. [16], where various polynomial-based and collocation methods have been employed. These techniques typically involve expanding the unknown function in terms of a suitable basis, which allows for the reduction of the original problem to a finite-dimensional system.

In this paper, we propose a novel approach based on alternative Legendre polynomials (ALPs) for approximating the solutions of two-dimensional nonlinear Volterra-Fredholm integral equations. Polynomials, especially orthogonal ones, are powerful mathematical tools for solving integral equations due to their flexibility, orthogonality properties, and the ease with which they can be manipulated in computational frameworks. In particular, alternative Legendre polynomials have desirable features such as orthogonality over a finite interval and simple recursive relations, which make them ideal for use in operational matrix methods.

The main contribution of this paper lies in the construction and application of operational matrices for integration and multiplication based on ALPs. These operational matrices are employed to transform the integral equations into a system of algebraic equations, significantly reducing the computational complexity of the problem. By doing so, the computational process becomes more efficient, and higher accuracy can be achieved with fewer polynomial terms, making this method particularly attractive for solving high-dimensional problems.

Operational matrices have been extensively utilized in the literature for the numerical solution of various types of differential and integral equations. Examples include the works of Chen and Hsiao [9], Sannuti [21], Razzaghi and Yousefi [24], Hwang and Shih [15], Paraskevopoulos [18], Chan and Wang [7], Yousefi and Behroozifar [6], Paraskevopoulos et al. [19], and Horng and Chou [14]. Among the various classes of polynomials used in these methods, ALPs have shown superior computational efficiency and accuracy, particularly when applied to nonlinear integral equations, making them the focus of this study.

Bazm et al. [5] used alternative Legendre polynomials (ALPs) to solve one-dimensional (1D) Volterra-Fredholm integral equations. We provide a detailed overview of alternative Legendre polynomials, including their definitions, properties, and the construction of the corresponding operational matrices for the numerical solution of one-dimensional (1D) integral equations as a preliminary step. Then, we extend the method to the two-dimensional Volterra-Fredholm integral equation described in Eq. (1.1). Error analysis is conducted to assess the method's accuracy, and several examples are presented to demonstrate the high precision and efficiency of the proposed approach. The results confirm the accuracy and effectiveness of the method in solving complex integral equations.

2. Fundamentals of 1D-ALPs

In this section, we will briefly review the ALPs and some of the operational matrices used in this article, as well as the approximation of a univariate function.

Deifnition 2.1 ([8])**.** The first family of 1D-ALPs is defined as follows:

$$\Lambda l_{m\alpha}(x) = \sum_{\kappa=0}^{m-\alpha} (-1)^{\kappa} \binom{m-\alpha}{\kappa} \binom{m+\alpha+\kappa+1}{m-\alpha} x^{\alpha+\kappa}, \qquad \alpha = 0, 1, \ldots, m. \tag{2.1}$$

According to the weight function $w = 1$, the ALPs are orthogonal on the interval $[0, 1]$, and every term in the set $\mathbf{\Lambda l_m} = \{\Lambda l_{m\alpha}(x)\}_{\alpha=0}^{m}$ has degree m.

One of the key matrices introduced in this article, which plays a central role in reducing the solution of integral equations to systems of algebraic equations, is the diagonal matrix ν_{1D}. This matrix is also utilized in the numerical treatment of two-dimensional integral equations. We begin by defining

$$\Psi(x) = [\Lambda l_{m0}(x), \Lambda l_{m1}(x), \ldots, \Lambda l_{mm}(x)]^T,$$

and, using the function $\Psi(x)$, the following relation is obtained:

$$\nu_{1D_x} = \int_0^1 \Psi(x)\Psi^T(x)\, dx = \mathrm{diag}\left\{\frac{1}{2\iota+1}\right\}_{\iota=0}^{m}, \tag{2.2}$$

where ν_{1D_x} is a diagonal matrix of order $(m+1) \times (m+1)$, and $\Psi(x)$ is an $(m+1) \times 1$ column vector.

Another important matrix is the operational matrix of integration, defined by the relation $\int_0^x \Psi(t)\,dt \simeq T_{1D}\Psi(x)$, where $\Psi(x)$ is the vector of ALPs as defined in (2.2), and $T_{1D} = [\tau_{\alpha r}]_{\alpha,r=0}^{m}$ denotes the operational matrix for one-dimensional ALP integration of order $(m+1)\times(m+1)$. The entries $\tau_{\alpha r}$ are given explicitly by the following formula:

$$\tau_{\alpha r} = (2r+1)\sum_{\kappa=0}^{m-\alpha} \frac{(-1)^{\kappa}\binom{m-\alpha}{\kappa}\binom{m+\alpha+\kappa+1}{m-\alpha}}{\alpha+\kappa+1} \sum_{l=0}^{m-r} \frac{(-1)^{l}\binom{m-r}{l}\binom{m+r+l+1}{m-r}}{\alpha+r+l+\kappa+2}. \tag{2.3}$$

Now, given a univariate function $h_m(x)$, its approximation is computed as follows [10]:

$$h_m(x) = \sum_{\alpha=0}^{m} h_\alpha \Lambda l_{m\alpha}(x), \tag{2.4}$$

where the coefficients h_α are given by

$$h_\alpha = \frac{\langle h(x), \Lambda l_{m\alpha}(x)\rangle}{\langle \Lambda l_{m\alpha}, \Lambda l_{m\alpha}\rangle} = (2\alpha+1)\langle h(x), \Lambda l_{m\alpha}(x)\rangle, \quad \alpha = 0,1,\dots,m. \tag{2.5}$$

3. Fundamentals of 2D-ALPs

We use ALPs to address problems related to integral equations. The extension of ALPs to two dimensions is a novel approach presented in this article. Below, we define and approximate functions of two and four variables. Let $\Psi(x,y)$ be a vector composed of two-variable ALPs:

$$\begin{aligned}\Psi(x,y) = [&\Lambda l_{m0m0}(x,y), \Lambda l_{m0m1}(x,y), \dots, \Lambda l_{m0mm}(x,y), \Lambda l_{m1m0}(x,y), \dots,\\ &\Lambda l_{m1mm}(x,y), \dots, \Lambda l_{mmm0}(x,y), \dots, \Lambda l_{mmmm}(x,y)]^T, \quad (x,y)\in D.\end{aligned} \tag{3.1}$$

The vector $\Psi(x,y)$ is of size $(m+1)^2\times 1$, and it can be obtained via the Kronecker product of $\Psi(x)$ and $\Psi(y)$:

$$\Psi(x,y) = \Psi(x)\otimes\Psi(y), \tag{3.2}$$

where each element of the resulting matrix is defined by:

$$\Lambda l_{m\alpha m\alpha'} = \Lambda l_{m\alpha}(x)\Lambda l_{m\alpha'}(y), \qquad \alpha' = 0,1,\dots,m. \tag{3.3}$$

Now, let us assume that $\mathcal{H} = L^2(D)$, so that the inner product and norm in this space are defined by:

$$\langle h(x,y), b(x,y)\rangle = \int_0^1\int_0^1 h(x,y)b(x,y)\,dx\,dy, \tag{3.4}$$

$$\|h(x,y)\|_2 = \langle h(x,y), h(x,y)\rangle^{\frac{1}{2}} = \left(\int_0^1\int_0^1 |h(x,y)|^2\,dx\,dy\right)^{\frac{1}{2}}. \tag{3.5}$$

Consider the finite-dimensional subspace:

$$\mathcal{H}_m = \operatorname{span}\{\Lambda l_{m0m0}(x,y), \Lambda l_{m0m1}(x,y), \dots, \Lambda l_{mmmm}(x,y)\}.$$

The space $\mathcal{H}_m$ is a closed subspace of $\mathcal{H}$, and for every $h(x,y)\in\mathcal{H}$, there exists a unique best approximation $h_m(x,y)\in\mathcal{H}_m$ satisfying:

$$\|h - h_m\|_2 \le \|h-b\|_2, \qquad \forall b\in\mathcal{H}_m. \tag{3.6}$$

Furthermore, we have the following representation:

$$h_m(x,y) = \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m} h_{\alpha\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y) = \Psi^T(x,y)H, \tag{3.7}$$

where the coefficients are given by:

$$h_{\alpha\alpha'} = \frac{\langle\langle h(x,y), \Lambda l_{m\alpha}(x)\rangle, \Lambda l_{m\alpha'}(y)\rangle}{\langle \Lambda l_{m\alpha}, \Lambda l_{m\alpha}\rangle\langle \Lambda l_{m\alpha'}, \Lambda l_{m\alpha'}\rangle} = (2\alpha+1)(2\alpha'+1)\langle\langle h, \Lambda l_{m\alpha}\rangle, \Lambda l_{m\alpha'}\rangle, \tag{3.8}$$

and thus, based on the ALP basis, any function $h(x,y) \in D$ can be approximated as:

$$h(x,y) \simeq h_m(x,y) = \Psi^T(x,y)H, \tag{3.9}$$

where

$$H = [h_{00}, h_{01}, \dots, h_{0m}, h_{10}, \dots, h_{1m}, \dots, h_{m0}, \dots, h_{mm}]^T. \tag{3.10}$$

The vector H is of dimension $(m+1)^2 \times 1$.

Moreover, any four-variable function $\varphi(x,y,t,s) \in L^2(D \times D)$ can be approximated in terms of ALPs as follows:

$$\varphi(x,y,t,s) \simeq \varphi_m(x,y,t,s) = \Psi^T(x,y)\Phi\Psi(t,s), \tag{3.11}$$

where

$$\begin{aligned}\varphi_{\iota\kappa\iota'\kappa'} &= \frac{\langle\langle \varphi(x,y,t,s), \Lambda l_{m\iota m\kappa}\rangle, \Lambda l_{m\iota' m\kappa'}\rangle}{\langle \Lambda l_{m\iota m\kappa}, \Lambda l_{m\iota m\kappa}\rangle\langle \Lambda l_{m\iota' m\kappa'}, \Lambda l_{m\iota' m\kappa'}\rangle} \\ &= (2\iota+1)(2\kappa+1)(2\iota'+1)(2\kappa'+1)\langle\langle \varphi(x,y,t,s), \Lambda l_{m\iota m\kappa}(x,t)\rangle, \Lambda l_{m\iota' m\kappa'}(y,s)\rangle,\end{aligned} \tag{3.12}$$

and $\Phi = [\varphi_{\iota\kappa\iota'\kappa'}]_{0\le\iota,\kappa,\iota',\kappa'\le m}$ is a matrix of dimension $(m+1)^2 \times (m+1)^2$. It can also be shown that the function $\varphi_m(x,y,t,s)$ provides the best unique approximation for the function $\varphi(x,y,t,s)$.

4. Operational Matrices

Theorem 4.1. *Assume that the diagonal matrices ν_{1D_x} and ν_{1D_y} correspond to the vectors $\Psi(x)$ and $\Psi(y)$, respectively. Then, the following relation holds:*

$$\nu_{2D_{(x,y)}} = \nu_{1D_x} \otimes \nu_{1D_y}. \tag{4.1}$$

The diagonal matrix $\nu_{2D_{(x,y)}}$, associated with the vector $\Psi(x,y)$, can thus be constructed from the Kronecker product of these matrix.

Proof. Based on Eq. (2.2), the diagonal matrix corresponding to the bivariate vector $\Psi(x,y)$ is defined as follows:

$$\begin{aligned}\nu_{2D_{(x,y)}} &= \int_0^1\int_0^1 \Psi(x,y)\Psi^T(x,y)\,dx\,dy = \int_0^1\int_0^1 [\Psi(x)\Psi^T(x)] \otimes [\Psi(y)\Psi^T(y)]\,dx\,dy \\ &= \left[\int_0^1 \Psi(x)\Psi^T(x)\,dx\right] \otimes \left[\int_0^1 \Psi(y)\Psi^T(y)\,dy\right],\end{aligned} \tag{4.2}$$

which confirms that $\nu_{2D_{(x,y)}}$ is an $(m+1)^2 \times (m+1)^2$ matrix. □

Theorem 4.2. *Assume that $\Psi(x,y)$ is the 2D-ALPs, defined in Eq. (3.1). Then:*

$$\int_0^x\int_0^y \Psi(t,s)\,dt\,ds \simeq (T_{1D}\Lambda l_{m\alpha}(x)) \otimes (T_{1D}\Lambda l_{m\alpha'}(y)) = (T_{1D} \otimes T_{1D})\Psi(x,y) = T_{2D}\Psi(x,y), \tag{4.3}$$

where T_{1D} is the operational matrix defined by Eq. (2.3), and T_{2D} is the operational matrix of integration 2D-ALPs of order $(m+1)^2 \times (m+1)^2$.

Proof. Considering the integral of the vector $\Psi(x,y)$, we have:

$$\int_0^x\int_0^y \Psi(t,s)\,dt\,ds = \begin{bmatrix} \int_0^x\int_0^y \Lambda l_{m0m0}(t,s)\,dt\,ds \\ \int_0^x\int_0^y \Lambda l_{m0m1}(t,s)\,dt\,ds \\ \vdots \\ \int_0^x\int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds \\ \vdots \\ \int_0^x\int_0^y \Lambda l_{mmmm}(t,s)\,dt\,ds \end{bmatrix}, \tag{4.4}$$

Now, in terms of Eq. (3.3), we can write the following equation:

$$\int_0^x \int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds = \int_0^x \int_0^y \Lambda l_{m\alpha}(t)\Lambda l_{m\alpha'}(s)\,dt\,ds, \tag{4.5}$$

that according to Eq. (2.1), we have:

$$\begin{aligned}\int_0^x \int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds = \int_0^x \int_0^y &\left(\sum_{\kappa=0}^{m-\alpha}(-1)^\kappa \binom{m-\alpha}{\kappa}\binom{m+\alpha+\kappa+1}{m-\alpha} t^{\alpha+\kappa}\right)\\ &\cdot\left(\sum_{\iota=0}^{m-\alpha'}(-1)^\iota \binom{m-\alpha'}{\iota}\binom{m+\alpha'+\iota+1}{m-\alpha'} s^{\alpha'+\iota}\right) dt\,ds, \quad \alpha,\alpha',\iota,\kappa = 0,\dots,m,\end{aligned} \tag{4.6}$$

where

$$\int_0^x \int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds = \left(\sum_{\kappa=0}^{m-\alpha}\frac{(-1)^\kappa\binom{m-\alpha}{\kappa}\binom{m+\alpha+\kappa+1}{m-\alpha}x^{\alpha+\kappa+1}}{\alpha+\kappa+1}\cdot\sum_{\iota=0}^{m-\alpha'}\frac{(-1)^\iota\binom{m-\alpha'}{\iota}\binom{m+\alpha'+\iota+1}{m-\alpha'}y^{\alpha'+\iota+1}}{\alpha'+\iota+1}\right), \tag{4.7}$$

by approximating $x^{\alpha+\kappa+1}$ and $y^{\alpha'+\iota+1}$ with the help of Eq. (2.4), we have:

$$x^{\alpha+\kappa+1} \simeq \sum_{r=0}^{m}(2r+1)\sum_{l=0}^{m-r}\frac{(-1)^l\binom{m-r}{l}\binom{m+r+l+1}{m-r}}{\alpha+r+l+\kappa+2}\Lambda l_{mr}(x), \tag{4.8}$$

$$y^{\alpha'+\iota+1} \simeq \sum_{r'=0}^{m}(2r'+1)\sum_{l'=0}^{m-r'}\frac{(-1)^{l'}\binom{m-r'}{l'}\binom{m+r'+l'+1}{m-r'}}{\alpha'+r'+l'+\iota+2}\Lambda l_{mr'}(y), \tag{4.9}$$

Substituting Eqs. (4.8) and (4.9) into Eq. (4.7), the following equation is obtained:

$$\begin{aligned}\int_0^x \int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds = &\sum_{r=0}^{m}(2r+1)\left[\sum_{\kappa=0}^{m-\alpha}\frac{(-1)^\kappa\binom{m-\alpha}{\kappa}\binom{m+\alpha+\kappa+1}{m-\alpha}}{\alpha+\kappa+1}\sum_{l=0}^{m-r}\frac{(-1)^l\binom{m-r}{l}\binom{m+r+l+1}{m-r}}{\alpha+r+l+\kappa+2}\right]\Lambda l_{mr}(x)\\ &\cdot\sum_{r'=0}^{m}(2r'+1)\left[\sum_{\iota=0}^{m-\alpha'}\frac{(-1)^\iota\binom{m-\alpha'}{\iota}\binom{m+\alpha'+\iota+1}{m-\alpha'}}{\alpha'+\iota+1}\sum_{l'=0}^{m-r'}\frac{(-1)^{l'}\binom{m-r'}{l'}\binom{m+r'+l'+1}{m-r'}}{\alpha'+r'+l'+\iota+2}\right]\Lambda l_{mr'}(y),\end{aligned} \tag{4.10}$$

where

$$\tau_{\alpha r} = (2r+1)\sum_{\kappa=0}^{m-\alpha}\frac{(-1)^\kappa\binom{m-\alpha}{\kappa}\binom{m+\alpha+\kappa+1}{m-\alpha}}{\alpha+\kappa+1}\sum_{l=0}^{m-r}\frac{(-1)^l\binom{m-r}{l}\binom{m+r+l+1}{m-r}}{\alpha+r+l+\kappa+2}, \tag{4.11}$$

$$\tau_{\alpha' r'} = (2r'+1)\sum_{\iota=0}^{m-\alpha'}\frac{(-1)^\iota\binom{m-\alpha'}{\iota}\binom{m+\alpha'+\iota+1}{m-\alpha'}}{\alpha'+\iota+1}\sum_{l'=0}^{m-r'}\frac{(-1)^{l'}\binom{m-r'}{l'}\binom{m+r'+l'+1}{m-r'}}{\alpha'+r'+l'+\iota+2}, \tag{4.12}$$

which according to Eqs. (4.10), (4.11), and (4.12), we have:

$$\int_0^x \int_0^y \Lambda l_{m\alpha m\alpha'}(t,s)\,dt\,ds = \sum_{r=0}^{m}\left(\sum_{r'=0}^{m}(\tau_{\alpha r}\tau_{\alpha' r'})\Lambda l_{mr}(x)\Lambda l_{mr'}(y)\right), \tag{4.13}$$

now by substituting Eq. (4.13) into Eq. (4.4), we can conclude:

$$\int_0^x \int_0^y \Psi(t,s)\,dt\,ds \simeq \begin{bmatrix} \sum_{r=0}^{m}\left(\sum_{r'=0}^{m} \tau_{0r}\tau_{0r'}\Lambda l_{mrmr'}(x,y)\right) \\ \sum_{r=0}^{m}\left(\sum_{r'=0}^{m} \tau_{0r}\tau_{1r'}\Lambda l_{mrmr'}(x,y)\right) \\ \sum_{r=0}^{m}\left(\sum_{r'=0}^{m} \tau_{1r}\tau_{0r'}\Lambda l_{mrmr'}(x,y)\right) \\ \vdots \\ \sum_{r=0}^{m}\left(\sum_{r'=0}^{m} \tau_{mr}\tau_{mr'}\Lambda l_{mrmr'}(x,y)\right) \end{bmatrix}. \tag{4.14}$$

□

Theorem 4.3. *Suppose that* $\Gamma = [\gamma_{00}, \gamma_{01}, \ldots, \gamma_{0m}, \gamma_{10}, \ldots, \gamma_{1m}, \ldots, \gamma_{m0}, \ldots, \gamma_{mm}]^T$ *is a* $(m+1)^2 \times 1$ *vector. Then we have:*

$$\Psi(x,y)\Psi^T(x,y)\Gamma \simeq \hat{\Gamma}\Psi(x,y), \tag{4.15}$$

where $\hat{\Gamma} = [\hat{\gamma}_{\iota\alpha\iota'\alpha'}]_{0\le\iota,\alpha,\iota',\alpha'\le m}$ *is the operational matrix of the product for 2D-ALPs and of order* $(m+1)^2 \times (m+1)^2$*, in which:*

$$\hat{\gamma}_{\iota\alpha\iota'\alpha'} = (2\alpha+1)(2\alpha'+1)\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m} \gamma_{\kappa\kappa'}\varsigma_{\iota\kappa\alpha\iota'\kappa'\alpha'}, \tag{4.16}$$

where

$$\varsigma_{\iota\kappa\alpha\iota'\kappa'\alpha'} = \langle \Lambda l_{m\kappa m\kappa'}\Lambda l_{m\iota m\iota'}, \Lambda l_{m\alpha m\alpha'}\rangle = \int_0^1\int_0^1 \Lambda l_{m\kappa m\kappa'}(x,y)\Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\alpha m\alpha'}(x,y)\,dx\,dy, \tag{4.17}$$

Proof. At first, we assume:

$$\Psi(x,y)\Psi^T(x,y)\Gamma = \begin{bmatrix} \sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m} \gamma_{\kappa\kappa'}\Lambda l_{m0m0}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y) \\ \sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m} \gamma_{\kappa\kappa'}\Lambda l_{m0m1}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y) \\ \sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m} \gamma_{\kappa\kappa'}\Lambda l_{m1m0}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y) \\ \vdots \\ \sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m} \gamma_{\kappa\kappa'}\Lambda l_{mmmm}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y) \end{bmatrix}, \tag{4.18}$$

which, according to Eqs. (3.7) and (3.8), leads to the following approximation:

$$\Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y) \simeq \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m} b_{\iota\kappa\alpha\iota'\kappa'\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y), \tag{4.19}$$

in which

$$b_{\iota\kappa\alpha\iota'\kappa'\alpha'} = (2\alpha+1)(2\alpha'+1)\langle \Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y), \Lambda l_{m\alpha m\alpha'}(x,y)\rangle, \tag{4.20}$$

Substituting Eq. (4.20) into Eq. (4.19), we get:

$$\Lambda l_{m\iota m\iota'}\Lambda l_{m\kappa m\kappa'} \simeq \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m} (2\alpha+1)(2\alpha'+1)\langle \Lambda l_{m\iota m\iota'}, \Lambda l_{m\kappa m\kappa'}, \Lambda l_{m\alpha m\alpha'}\rangle \Lambda l_{m\alpha m\alpha'}, \tag{4.21}$$

which, according to Eqs. (4.18) and (4.21), leads to:

$$\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\Lambda l_{m\iota m\iota'}\Lambda l_{m\kappa m\kappa'}=\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}(2\alpha+1)(2\alpha'+1)\langle\Lambda l_{m\iota m\iota'},\Lambda l_{m\kappa m\kappa'},\Lambda l_{m\alpha m\alpha'}\rangle\Lambda l_{m\alpha m\alpha'}, \tag{4.22}$$

where

$$\varsigma_{\iota\kappa\alpha\iota'\kappa'\alpha'}=\langle\Lambda l_{m\iota m\iota'}\Lambda l_{m\kappa m\kappa'},\Lambda l_{m\alpha m\alpha'}\rangle=\int_0^1\int_0^1\Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y)\Lambda l_{m\alpha m\alpha'}(x,y)\,dx\,dy, \tag{4.23}$$

Substituting Eq. (4.23) into Eq. (4.22), the following equation is obtained:

$$\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y)=\sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\left[(2\alpha+1)(2\alpha'+1)\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\varsigma_{\iota\kappa\alpha\iota'\kappa'\alpha'}\right]\Lambda l_{m\alpha m\alpha'}(x,y), \tag{4.24}$$

Assuming

$$\hat{\gamma}_{\iota\alpha\iota'\alpha'}=(2\alpha+1)(2\alpha'+1)\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\varsigma_{\iota\kappa\alpha\iota'\kappa'\alpha'}, \tag{4.25}$$

we can rewrite Eq. (4.24) as:

$$\sum_{\kappa=0}^{m}\sum_{\kappa'=0}^{m}\gamma_{\kappa\kappa'}\Lambda l_{m\iota m\iota'}(x,y)\Lambda l_{m\kappa m\kappa'}(x,y)=\sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{\iota\alpha\iota'\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y), \tag{4.26}$$

Finally, substituting Eq. (4.26) into Eq. (4.18), we obtain:

$$\Psi(x,y)\Psi^T(x,y)\Gamma=\begin{bmatrix}\sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{0\alpha0\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y)\\ \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{0\alpha1\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y)\\ \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{1\alpha0\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y)\\ \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{1\alpha1\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y)\\ \vdots\\ \sum_{\alpha=0}^{m}\sum_{\alpha'=0}^{m}\hat{\gamma}_{m\alpha m\alpha'}\Lambda l_{m\alpha m\alpha'}(x,y)\end{bmatrix}. \tag{4.27}$$

□

5. Numerical Method

To solve Eq. (1.1), we propose a numerical method utilizing multiplication, integral, and diagonal operational matrices. By employing 2D-ALPs operational matrices, we transform the nonlinear integral equations into an algebraic system. To achieve this, we assume the following:

$$\eta_1(x,y)=\theta_1(x,y,g(x,y)), \tag{5.1}$$

$$\eta_2(x,y)=\theta_2(x,y,g(x,y)). \tag{5.2}$$

Substituting Eqs. (5.1) and (5.2) into Eq. (1.1) yields:

$$g(x,y)=f(x,y)+\lambda_1\int_0^x\int_0^y\varphi_1(x,y,t,s)\eta_1(t,s)\,dt\,ds+\lambda_2\int_0^1\int_0^1\varphi_2(x,y,t,s)\eta_2(t,s)\,dt\,ds, \tag{5.3}$$

where $\eta_1(x,y)$ and $\eta_2(x,y)$ are the unknown functions to be determined. By substituting Eq. (5.3) into Eqs. (5.1) and (5.2), we obtain:

$$\eta_1(x,y) = \theta_1\left(x, y, f(x,y) + \lambda_1 \int_0^x \int_0^y \varphi_1(x,y,t,s)\eta_1(t,s)\,dt\,ds + \lambda_2 \int_0^1 \int_0^1 \varphi_2(x,y,t,s)\eta_2(t,s)\,dt\,ds\right), \tag{5.4}$$

and

$$\eta_2(x,y) = \theta_2\left(x, y, f(x,y) + \lambda_1 \int_0^x \int_0^y \varphi_1(x,y,t,s)\eta_1(t,s)\,dt\,ds + \lambda_2 \int_0^1 \int_0^1 \varphi_2(x,y,t,s)\eta_2(t,s)\,dt\,ds\right). \tag{5.5}$$

To find the unknown functions $\eta_1(x,y)$ and $\eta_2(x,y)$, we approximate these functions using 2D-ALPs as follows:

$$\begin{aligned}
\eta_1(x,y) &\simeq \Psi^T(x,y)q_1,\\
\eta_2(x,y) &\simeq \Psi^T(x,y)q_2,\\
\varphi_1(x,y,t,s) &\simeq \Psi^T(x,y)\Phi_1\Psi(t,s),\\
\varphi_2(x,y,t,s) &\simeq \Psi^T(x,y)\Phi_2\Psi(t,s),
\end{aligned} \tag{5.6}$$

where q_1 and q_2 are unknown vectors of size $(m+1)^2 \times 1$. Substituting these approximations and Eqs. (4.1), (4.3), and (4.15) into Eqs. (5.4) and (5.5) the integral terms in Eq. (5.3) can be expressed as follows:

$$\begin{aligned}
\int_0^x \int_0^y \varphi_1(x,y,t,s)\eta_1(t,s)\,dt\,ds &\simeq \int_0^x \int_0^y \Psi^T(x,y)\Phi_1\Psi(t,s)\Psi^T(t,s)q_1\,dt\,ds\\
&= \Psi^T(x,y)\Phi_1 \int_0^x \int_0^y \Psi(t,s)\Psi^T(t,s)q_1\,dt\,ds\\
&\simeq \Psi^T(x,y)\Phi_1 \int_0^x \int_0^y \hat{q}_1\Psi(t,s)\,dt\,ds\\
&\simeq \Psi^T(x,y)\Phi_1\hat{q}_1 \int_0^x \int_0^y \Psi(t,s)\,dt\,ds\\
&\simeq \Psi^T(x,y)\Phi_1\hat{q}_1 T_{2D}\Psi(x,y),
\end{aligned} \tag{5.7}$$

and

$$\begin{aligned}
\int_0^1 \int_0^1 \varphi_2(x,y,t,s)\eta_2(t,s)\,dt\,ds &\simeq \int_0^1 \int_0^1 \Psi^T(x,y)\Phi_2\Psi(t,s)\Psi^T(t,s)q_2\,dt\,ds\\
&= \Psi^T(x,y)\Phi_2 \int_0^1 \int_0^1 \Psi(t,s)\Psi^T(t,s)q_2\,dt\,ds\\
&\simeq \Psi^T(x,y)\Phi_2\nu_{2D}q_2,
\end{aligned} \tag{5.8}$$

where ν_{2D} is defined by Eq. (4.1). Substituting Eqs. (5.6), (5.7), and (5.8) into Eqs. (5.4) and (5.5) results in:

$$\Psi^T(x,y)q_1 = \theta_1\left(x, y, f(x,y) + \lambda_1\Psi^T(x,y)\Phi_1\hat{q}_1 T_{2D}\Psi(x,y) + \lambda_2\Psi^T(x,y)\Phi_2\nu_{2D}q_2\right), \tag{5.9}$$

$$\Psi^T(x,y)q_2 = \theta_2\left(x, y, f(x,y) + \lambda_1\Psi^T(x,y)\Phi_1\hat{q}_1 T_{2D}\Psi(x,y) + \lambda_2\Psi^T(x,y)\Phi_2\nu_{2D}q_2\right). \tag{5.10}$$

We use the set of Gauss-Chelyshkov collocation points for $l = 0, \dots, m$:

$$(x_l, y_l) = (\Lambda l_{m+1,0}(x_l) = 0, \Lambda l_{m+1,0}(y_l) = 0)$$

By applying the collocation method to Eqs. (5.8) and (5.9) at these points, we obtain:

$$\Psi^T(x_l,y_l)q_1 = \theta_1\left(x_l, y_l, f(x_l,y_l) + \lambda_1\Psi^T(x_l,y_l)\Phi_1\hat{q}_1 T_{2D}\Psi(x_l,y_l) + \lambda_2\Psi^T(x_l,y_l)\Phi_2\nu_{2D}q_2\right), \tag{5.11}$$

$$\Psi^T(x_l,y_l)q_2 = \theta_2\left(x_l, y_l, f(x_l,y_l) + \lambda_1\Psi^T(x_l,y_l)\Phi_1\hat{q}_1 T_{2D}\Psi(x_l,y_l) + \lambda_2\Psi^T(x_l,y_l)\Phi_2\nu_{2D}q_2\right). \tag{5.12}$$

These equations constitute a system of $(m+1)^2 \times 2$ nonlinear algebraic equations, containing the same number of unknowns. After solving this system, the approximate solution to Eq. (1.1) can be derived by combining the equations:

$$g_m(x,y) = f(x,y) + \lambda_1 \Psi^T(x,y)\Phi_1 \hat{q}_1 T_{2D}\Psi(x,y) + \lambda_2 \Psi^T(x,y)\Phi_2 \nu_{2D} q_2. \tag{5.13}$$

6. Convergence Analysis

Let $C^{m,m}(D \times D)$ denote the space of functions $h : D \times D \to \mathbb{R}$ with continuous partial derivatives. Specifically, let

$$h^{(\iota,\kappa)}(x,y) = \frac{\partial^{\iota+\kappa}}{\partial x^{\iota}\partial y^{\kappa}} h(x,y), \qquad (x,y) \in D, \qquad \iota,\kappa = 0,\dots,m,$$

The purpose of this section is to obtain an upper bound for any function approximated by 2D-ALPs.

Theorem 6.1. *Assume that $h(x,y) \in C^{m+1,m+1}(D)$ and that*

$$h_m(x,y) = \Psi^T(x,y)H,$$

is the best approximation of h in terms of its expansion using ALPs. In this case, we have:

$$\|h - h_m\|_2 \leq \frac{1}{(m+1)!2^{2m+1}} \left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}} \right), \tag{6.1}$$

where the constants ω_1, ω_2, and ω_3 satisfy the following relations:

$$\max_{(x,y)\in D\times D} \left| \frac{\partial^{m+1} h(x,y)}{\partial x^{m+1}} \right| \leq w_1, \tag{6.2}$$

$$\max_{(x,y)\in D\times D} \left| \frac{\partial^{m+1} h(x,y)}{\partial y^{m+1}} \right| \leq w_2, \tag{6.3}$$

$$\max_{(x,y)\in D\times D} \left| \frac{\partial^{2m+2} h(x,y)}{\partial x^{m+1}\partial y^{m+1}} \right| \leq w_3. \tag{6.4}$$

Proof. Let $P_{m,m}(x,y)$ be the interpolating polynomial of h at the points(x_l, y_k), where $l, k = 0,\dots,m$ and $x_l = y_k$ are the roots of the shifted Chebyshev polynomial of degree $m+1$ in $[0,1]$. For every$(x,y) \in D$, we have:

$$\begin{aligned} h(x,y) - P_{m,m}(x,y) = {} & \frac{\partial^{m+1} h(\zeta,y)}{\partial x^{m+1}(m+1)!} \prod_{l=0}^{m} (x - x_l) + \frac{\partial^{m+1} h(x,\varrho)}{\partial y^{m+1}(m+1)!} \prod_{k=0}^{m} (y - y_k) \\ & - \frac{\partial^{2m+2} h(\zeta',\varrho')}{\partial x^{m+1}\partial y^{m+1}(m+1)!^2} \prod_{l=0}^{m} (x - x_l) \prod_{k=0}^{m} (y - y_k), \end{aligned} \tag{6.5}$$

where $\zeta, \varrho, \zeta', \varrho' \in [0,1]$.

According to the estimate for the Chebyshev interpolation points, we have:

$$|h(x,y) - P_{m,m}(x,y)| \leq \frac{1}{(m+1)!2^{2m+1}} \left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}} \right), \tag{6.6}$$

and since h_m is the best unique approximation of h in $\mathcal{H}_m$, we have:

$$\begin{aligned} \|h - h_m\|_2^2 \leq \|h - P_{m,m}\|_2^2 &= \int_0^1 \int_0^1 |o(x,y) - P_{m,m}(x,y)|^2 \, dx\, dy \\ &\leq \int_0^1 \int_0^1 \left(\frac{1}{(m+1)!2^{2m+1}} \left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}} \right) \right)^2 dx\, dy \\ &= \left(\frac{1}{(m+1)!2^{2m+1}} \left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}} \right) \right)^2. \end{aligned} \tag{6.7}$$

Hence, the following relation is obtained:

$$\|h - h_m\|_2 \leq \frac{1}{(m+1)!2^{2m+1}} \left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}} \right).$$

□

Now, let us assume

$$(m+1)!2^{2m+1} = \upsilon, \tag{6.8}$$

$$\left(\omega_1 + \omega_2 + \frac{\omega_3}{(m+1)!2^{2m+1}}\right) = \omega'.$$

Then

$$\|h - h_m\|_2 \leq \frac{\omega'}{\upsilon}. \tag{6.9}$$

Theorem 6.2. *Suppose that* $\varphi(x,y,t,s) \in C^{m+1,m+1,m+1,m+1}(D \times D)$ *and that*

$$\varphi_m(x,y,t,s) = \Psi^T(x,y)\Phi\Psi(x,y)$$

is the approximation of φ *in terms of its expansion using ALPs. In this case, we have:*

$$\|\varphi - \varphi_m\|_2 \leq \frac{1}{(m+1)!2^{2m+1}}\left(\omega_1 + \omega_2 + \omega_3 + \omega_4 + \frac{\omega_5 + \omega_6 + \omega_7 + \omega_8 + \omega_9 + \omega_{10}}{(m+1)!2^{2m+1}} + \frac{\omega_{11} + \omega_{12} + \omega_{13} + \omega_{14}}{(m+1)!^2 2^{4m+2}} + \frac{\omega_{15}}{(m+1)!^3 2^{6m+3}}\right). \tag{6.10}$$

where the constants $\omega_1, \omega_2, \ldots, \omega_{15}$ *satisfy the following relation:*

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{m+1}\varphi(x,y,t,s)}{\partial x^{m+1}}\right| \leq w_1,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{m+1}\varphi(x,y,t,s)}{\partial y^{m+1}}\right| \leq w_2,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{m+1}\varphi(x,y,t,s)}{\partial t^{m+1}}\right| \leq w_3,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{m+1}\varphi(x,y,t,s)}{\partial s^{m+1}}\right| \leq w_4,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial x^{m+1}\partial y^{m+1}}\right| \leq w_5,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial x^{m+1}\partial t^{m+1}}\right| \leq w_6,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial x^{m+1}\partial s^{m+1}}\right| \leq w_7,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial y^{m+1}\partial t^{m+1}}\right| \leq w_8,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial y^{m+1}\partial s^{m+1}}\right| \leq w_9,$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{2m+2}\varphi(x,y,t,s)}{\partial t^{m+1}\partial s^{m+1}}\right| \leq w_{10},$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{3m+3}\varphi(x,y,t,s)}{\partial x^{m+1}\partial y^{m+1}\partial t^{m+1}}\right| \leq w_{11},$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{3m+3}\varphi(x,y,t,s)}{\partial x^{m+1}\partial y^{m+1}\partial s^{m+1}}\right| \leq w_{12},$$

$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{3m+3}\varphi(x,y,t,s)}{\partial x^{m+1}\partial t^{m+1}\partial s^{m+1}}\right|\leq w_{13},$$
$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{3m+3}\varphi(x,y,t,s)}{\partial y^{m+1}\partial t^{m+1}\partial s^{m+1}}\right|\leq w_{14},$$
$$\max_{(x,y,t,s)\in D\times D}\left|\frac{\partial^{4m+4}\varphi(x,y,t,s)}{\partial x^{m+1}\partial y^{m+1}\partial t^{m+1}\partial s^{m+1}}\right|\leq w_{15}.$$

Proof. Let $P_{m,m,m,m}(x,y,t,s)$ be the interpolating polynomial of $\varphi(x,y,t,s)$ at the points $(x_l,y_k,t_{l'},s_{k'})$, where $l,k,l',k'=0,\dots,m$ and $x_l=y_k=t_{l'}=s_{k'}$ are the roots of the shifted Chebyshev polynomial of degree $m+1$ in $[0,1]$. For every $(x,y,t,s)\in D\times D\times D\times D$, we have:

$$
\begin{aligned}
\varphi(x,y,t,s)-P_{m,m,m,m}(x,y,t,s) &= \frac{\partial^{m+1}\varphi(\zeta_1,y,t,s)}{\partial x^{m+1}(m+1)!}\prod_{l=0}^{m}(x-x_l)+\frac{\partial^{m+1}\varphi(x,\varrho_1,t,s)}{\partial y^{m+1}(m+1)!}\prod_{k=0}^{m}(y-y_k)\\
&+\frac{\partial^{m+1}\varphi(x,y,\Delta_1,s)}{\partial t^{m+1}(m+1)!}\prod_{l'=0}^{m}(t-t_{l'})+\frac{\partial^{m+1}\varphi(x,y,t,\sigma_1)}{\partial s^{m+1}(m+1)!}\prod_{k'=0}^{m}(s-s_{k'})\\
&-\frac{\partial^{2m+2}\varphi(\zeta_2,\varrho_2,t,s)}{\partial x^{m+1}\partial y^{m+1}(m+1)!^2}\prod_{l=0}^{m}(x-x_l)\prod_{k=0}^{m}(y-y_k)\\
&-\frac{\partial^{2m+2}\varphi(\zeta_3,y,\Delta_2,s)}{\partial x^{m+1}\partial t^{m+1}(m+1)!^2}\prod_{l=0}^{m}(x-x_l)\prod_{l'=0}^{m}(t-t'_l)\\
&-\frac{\partial^{2m+2}\varphi(\zeta_4,y,t,\sigma_2)}{\partial x^{m+1}\partial s^{m+1}(m+1)!^2}\prod_{l=0}^{m}(x-x_l)\prod_{k'=0}^{m}(s-s_{k'})\\
&-\frac{\partial^{2m+2}\varphi(x,\varrho_3,\Delta_3,s)}{\partial y^{m+1}\partial t^{m+1}(m+1)!^2}\prod_{k=0}^{m}(y-y_k)\prod_{l'=0}^{m}(t-t_{l'})\\
&-\frac{\partial^{2m+2}\varphi(x,\varrho_4,t,\sigma_3)}{\partial y^{m+1}\partial s^{m+1}(m+1)!^2}\prod_{k=0}^{m}(y-y_\kappa)\prod_{k'=0}^{m}(s-s_{k'})\\
&-\frac{\partial^{2m+2}\varphi(x,y,\Delta_4,\sigma_4)}{\partial t^{m+1}\partial s^{m+1}(m+1)!^2}\prod_{l'=0}^{m}(t-t'_l)\prod_{k'=0}^{m}(s-s_{k'})\\
&+\frac{\partial^{3m+3}\varphi(\zeta_5,\varrho_5,\Delta_5,s)}{\partial x^{m+1}\partial y^{m+1}\partial t^{m+1}(m+1)!^3}\prod_{l=0}^{m}(x-x_l)\prod_{k=0}^{m}(y-y_k)\prod_{l'=0}^{m}(t-t_{l'})\\
&+\frac{\partial^{3m+3}\varphi(\zeta_6,\varrho_6,s,\sigma_5)}{\partial x^{m+1}\partial y^{m+1}\partial s^{m+1}(m+1)!^3}\prod_{l=0}^{m}(x-x_l)\prod_{k=0}^{m}(y-y_k)\prod_{k'=0}^{m}(s-s_{k'})\\
&+\frac{\partial^{3m+3}\varphi(\zeta_7,y,\Delta_6,\sigma_6)}{\partial x^{m+1}\partial t^{m+1}\partial s^{m+1}(m+1)!^3}\prod_{l=0}^{m}(x-x_l)\prod_{l'=0}^{m}(t-t_{l'})\prod_{k'=0}^{m}(s-s_{k'})\\
&+\frac{\partial^{3m+3}\varphi(x,\varrho_7,\Delta_7,\sigma_7)}{\partial y^{m+1}\partial t^{m+1}\partial s^{m+1}(m+1)!^3}\prod_{k=0}^{m}(y-y_k)\prod_{l'=0}^{m}(t-t_{l'})\prod_{k'=0}^{m}(s-s_{k'})\\
&-\frac{\partial^{4m+4}\varphi(\zeta_8,\varrho_8,\Delta_8,\sigma_8)}{\partial x^{m+1}\partial y^{m+1}\partial t^{m+1}\partial s^{m+1}(m+1)!^4}\prod_{l=0}^{m}(x-x_l)\prod_{k=0}^{m}(y-y_k)\prod_{l'=0}^{m}(t-t_{l'})\prod_{k'=0}^{m}(s-s_{k'})\\
&\leq\frac{\omega_1+\omega_2+\omega_3+\omega_4}{(m+1)!2^{2m+1}}+\frac{\omega_5+\omega_6+\omega_7+\omega_8+\omega_9+\omega_{10}}{(m+1)!^2 2^{4m+2}}
\end{aligned}
$$

$$+\frac{\omega_{11}+\omega_{12}+\omega_{13}+\omega_{14}}{(m+1)!^3 2^{6m+3}}+\frac{\omega_{15}}{(m+1)!^4 2^{8m+4}}, \tag{6.11}$$

where $\zeta_1-\zeta_8, \varrho_1-\varrho_8, \Delta_1-\Delta_8, \sigma_1-\sigma_8 \in [0,1]$. Now, according to Eq. (6.8), we have:

$$\omega^{''}=\sum_{l=1}^{4}\omega_l+\sum_{l=5}^{10}\frac{\omega_l}{\upsilon}+\sum_{l=11}^{14}\frac{\omega_l}{\upsilon^2}+\frac{\omega_{15}}{\upsilon^3}. \tag{6.12}$$

Thus,

$$\|\varphi-\varphi_m\|_2 \le \frac{\omega^{''}}{\upsilon}, \tag{6.13}$$

where $\zeta_1-\zeta_8, \varrho_1-\varrho_8, \Delta_1-\Delta_8, \sigma_1-\sigma_8 \in [0,1]$. Now, according to Eq. (6.8), we have:

$$\omega^{''}=\sum_{l=1}^{4}\omega_l+\sum_{l=5}^{10}\frac{\omega_l}{\upsilon}+\sum_{l=11}^{14}\frac{\omega_l}{\upsilon^2}+\frac{\omega_{15}}{\upsilon^3}. \tag{6.14}$$

Thus,

$$\|\varphi-\varphi_m\|_2 \le \frac{\omega^{''}}{\upsilon}. \tag{6.15}$$

□

Theorem 6.3. *Suppose $g(x,y)$ is the exact solution of Eq. (1.1) and $g_m(x,y)$ is its approximate solution obtained from the 2D-ALPs method. Also, Assume further that the nonlinear term satisfies a Lipschitz condition, as follows:*

$$\|\eta_\varpi(x,y)-\eta_{\varpi,m}(x,y)\| \le L_\varpi \|g(x,y)-g_m(x,y)\|, \qquad \varpi=1,2,$$

and

$$1-|\lambda_1|L_1\left(M_1+M_1''\right)-|\lambda_2|L_2\left(M_2+M_2''\right)>0.$$

Then, we obtain the following upper error bound:

$$\|g(x,y)-g_m(x,y)\| \le \frac{N+|\lambda_1|L_1M_1'M_1''+|\lambda_2|L_2M_2'M_2''}{1-|\lambda_1|L_1\left(M_1+M_1''\right)-|\lambda_2|L_2\left(M_2+M_2''\right)},$$

where:

(1) *For all* $(x,y,t,s)\in D\times D$, $\max|\varphi_\varpi(x,y,t,s)|=M_\varpi, \qquad \varpi=1,2,$
(2) *For all* $(x,y)\in D$, $\max|\eta_\varpi(x,y)|=M'_\varpi, \qquad \varpi=1,2,$
(3) $\max|f(x,y)-f_m(x,y)|=N,$
(4) $\max|\varphi_\varpi(x,y,t,s)-\varphi_{\varpi,m}(x,y,t,s)|=M''_\varpi, \qquad \varpi=1,2.$

Proof. According to Eq. (1.1) and approximating the functions $f(x,y), g(x,y), \varphi_1(x,y,t,s)$ $,\varphi_2(x,y,t,s), \theta_1(t,s,g(t,s)), \theta_2(t,s,g(t,s))$ by using ALPs, we have:

$$\begin{aligned}\|g(x,y)-g_m(x,y)\| \le\ & \|f(x,y)-f_m(x,y)\|\\ &+|\lambda_1|\|x\|\|y\|\|\varphi_1(x,y,t,s)\eta_1(t,s)-\varphi_{1,m}(x,y,t,s)\eta_{1,m}(t,s)\|\\ &+|\lambda_2|\|\varphi_2(x,y,t,s)\eta_2(t,s)-\varphi_{2,m}(x,y,t,s)\eta_{2,m}(t,s)\|,\end{aligned}$$

Now, considering $\|x\|\|y\|\le 1$, we have:

$$\begin{aligned}\|g(x,y)-g_m(x,y)\| \le\ & \|f(x,y)-f_m(x,y)\|\\ &+|\lambda_1|\|\varphi_1(x,y,t,s)\eta_1(t,s)-\varphi_{1,m}(x,y,t,s)\eta_{1,m}(t,s)\|\\ &+|\lambda_2|\|\varphi_2(x,y,t,s)\eta_2(t,s)-\varphi_{2,m}(x,y,t,s)\eta_{2,m}(t,s)\|,\end{aligned}$$

which implies:

$$\|f(x,y)-f_m(x,y)\|+|\lambda_1|\|\varphi_1(x,y,t,s)\eta_1(t,s)-\varphi_1(x,y,t,s)\eta_{1,m}(t,s)$$

$$
\begin{aligned}
&+ \varphi_1(x,y,t,s)\eta_{1,m}(t,s) - \varphi_{1,m}(x,y,t,s)\eta_{1,m}(t,s)\| \\
&+ |\lambda_2|\|\varphi_2(x,y,t,s)\eta_2(t,s) - \varphi_2(x,y,t,s)\eta_{2,m}(t,s) \\
&+ \varphi_2(x,y,t,s)\eta_{2,m}(t,s) - \varphi_{2,m}(x,y,t,s)\eta_{2,m}(t,s)\| \\
&\le N + |\lambda_1| \left(\|\varphi_1(x,y,t,s)\|\|\eta_1(t,s) - \eta_{1,m}(t,s)\|\right) \\
&+ \|\varphi_1(x,y,t,s) - \varphi_{1,m}(x,y,t,s)\| \\
&\cdot(\|\eta_{1,m}(t,s) - \eta_1(t,s)\| + \|\eta_1(t,s)\|)) \\
&+ |\lambda_2| \left(\|\varphi_2(x,y,t,s)\|\|\eta_2(t,s) - \eta_{2,m}(t,s)\|\right) \\
&+ \|\varphi_2(x,y,t,s) - \varphi_{2,m}(x,y,t,s)\| \\
&\cdot(\|\eta_{2,m}(t,s) - \eta_2(t,s)\| + \|\eta_2(t,s)\|)\,,
\end{aligned}
$$

which by using assumptions $\{1\}$–$\{4\}$, we have:

$$
\begin{aligned}
\|g(x,y) - g_m(x,y)\| \le N &+ |\lambda_1| \left(M_1 L_1 \|g(x,y) - g_m(x,y)\| + M_1'' L_1 (\|g(x,y) - g_m(x,y)\| + M_1')\right) \\
&+ |\lambda_2| \left(M_2 L_2 \|g(x,y) - g_m(x,y)\| + M_2'' L_2 (\|g(x,y) - g_m(x,y)\| + M_2')\right).
\end{aligned}
$$

□

7. Numerical examples

In this study, we have implemented a numerical algorithm using MATLAB to evaluate the performance of the proposed method through a series of examples. These examples enable us to compare the effectiveness of the proposed method against other existing methods. The accuracy of the method is quantified using the following relations, which define error terms:

$$
\begin{aligned}
e_m(x,y) &= |g(x,y) - g_m(x,y)|, \qquad (x,y) \in D, \\
\|e_m\|_\infty &= \max\{e_m(x_l, y_l)\},
\end{aligned}
\tag{7.1}
$$

where g represents the exact solution, and g_m is the approximate solution obtained via the proposed method. The points (x_l, y_l) correspond to the selected collocation points used in the calculations.

Remark 7.1. In this article, we consider both dimensions equal ($m = n$).

Example 7.2 ([25])**.** Consider the following nonlinear integral equation:

$$
g(x,y) = f(x,y) + \int_0^y \int_0^x (-x - y - t - s) g^2(t,s)\,dt\,ds + \int_0^1 \int_0^1 (-xy - ts^2) g(t,s)\,dt\,ds; \quad (x,y) \in D, \tag{7.2}
$$

where the function $f(x,y)$ is defined as:

$$
f(x,y) = x^2 + \frac{1}{4} + \frac{17}{6}xy + \frac{7}{9}x^3y^4 + \frac{29}{18}x^4y^2 + \frac{6}{5}x^5y^2 + \frac{11}{30}x^6y.
$$

The exact solution of the equation is given by $g(x,y) = x^2 + 2xy$. This Equation (7.2) is classified as a nonlinear Volterra-Fredholm integral equation. To solve this problem, we employ the ALPs method, which utilizes product and integral operational matrices.

We have compared the absolute error and maximum error of the proposed method with the methods reported in [2], [11], and [25]. As presented in Table 1, it is evident that the error of our method is significantly lower. Additionally, to further illustrate the effectiveness of the method, we computed the error at various points, which is summarized in Table 2.

The graphs of the absolute error, approximate solution, and exact solution for $m = 8$ are displayed in Figure 1. Furthermore, in Figure 2, we plot the progression of the absolute error for $m = 2, 4$, and 6.

In Table 3, we provide the CPU time for all computational steps corresponding to the values of $m = 4, 5, 6, 7, 8$, and 9, which demonstrates the efficiency of the proposed method, particularly in handling a high volume of calculations with increasing values of m.

TABLE 1. Absolute errors: $|e_m(x,y)|$ for Example 7.2.

(x,y)	Present method			Method in [2]	Method in [11]	Method in [25]	
	$m=4$	$m=6$	$m=8$	$m=8$	$m=8$	$m=4$	$m=6$
(0.0, 0.0)	1.83E-5	2.18E-8	2.16E-9	4.71E-6	1.73E-3	3.05E-3	8.29E-4
(0.1, 0.1)	1.08E-4	5.29E-9	5.06E-9	4.09E-4	1.65E-3	3.16E-3	8.51E-4
(0.2, 0.2)	1.17E-4	4.29E-9	8.24E-10	1.63E-4	1.43E-3	3.50E-3	8.19E-4
(0.3, 0.3)	2.20E-4	5.03E-9	1.27E-8	8.81E-4	1.35E-3	2.72E-3	6.14E-4
(0.4, 0.4)	4.52E-4	4.52E-9	4.62E-9	7.69E-4	1.65E-3	3.91E-5	2.89E-3
(0.5, 0.5)	1.00E-4	4.73E-9	5.83E-9	1.33E-3	3.15E-4	1.91E-3	1.02E-3
(0.6, 0.6)	6.41E-4	6.94E-9	1.12E-8	2.71E-3	8.16E-3	3.35E-2	5.15E-3
(0.7, 0.7)	7.52E-4	8.68E-9	2.90E-8	6.21E-3	2.53E-2	2.11E-2	1.18E-2
(0.8, 0.8)	6.89E-4	6.12E-9	1.62E-8	1.20E-2	4.38E-2	3.85E-2	8.77E-3
(0.9, 0.9)	1.95E-3	2.92E-8	3.68E-8	1.88E-2	2.99E-2	8.06E-2	2.83E-2
$\|e_m\|_\infty$	1.95E-3	2.92E-8	3.68E-8	1.88E-2	4.38E-2	8.06E-2	2.83E-2

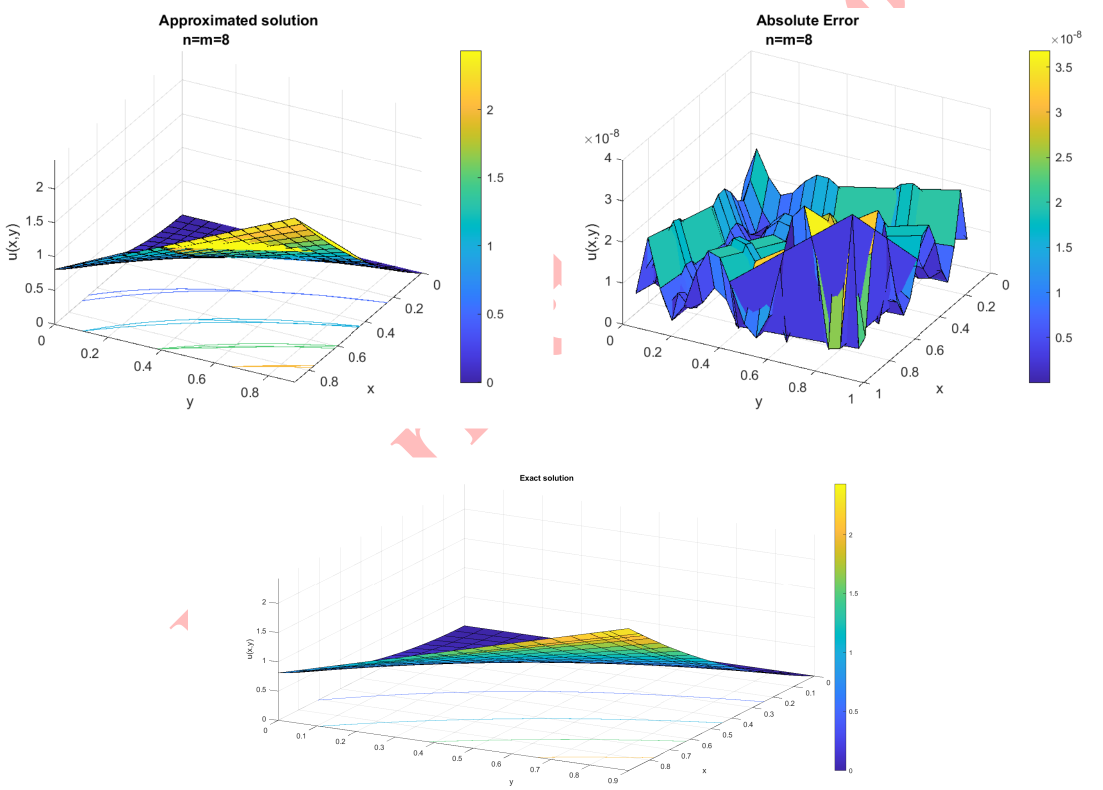

FIGURE 1. Plots of numerical results for Example 7.2.

Example 7.3 ([17]). Consider the following two-dimensional nonlinear integral equation:

$$g(x,y) = f(x,y) + \int_0^y \int_0^x (x+y-s-t)g^2(t,s)dtds; \quad (x,y) \in D, \tag{7.3}$$

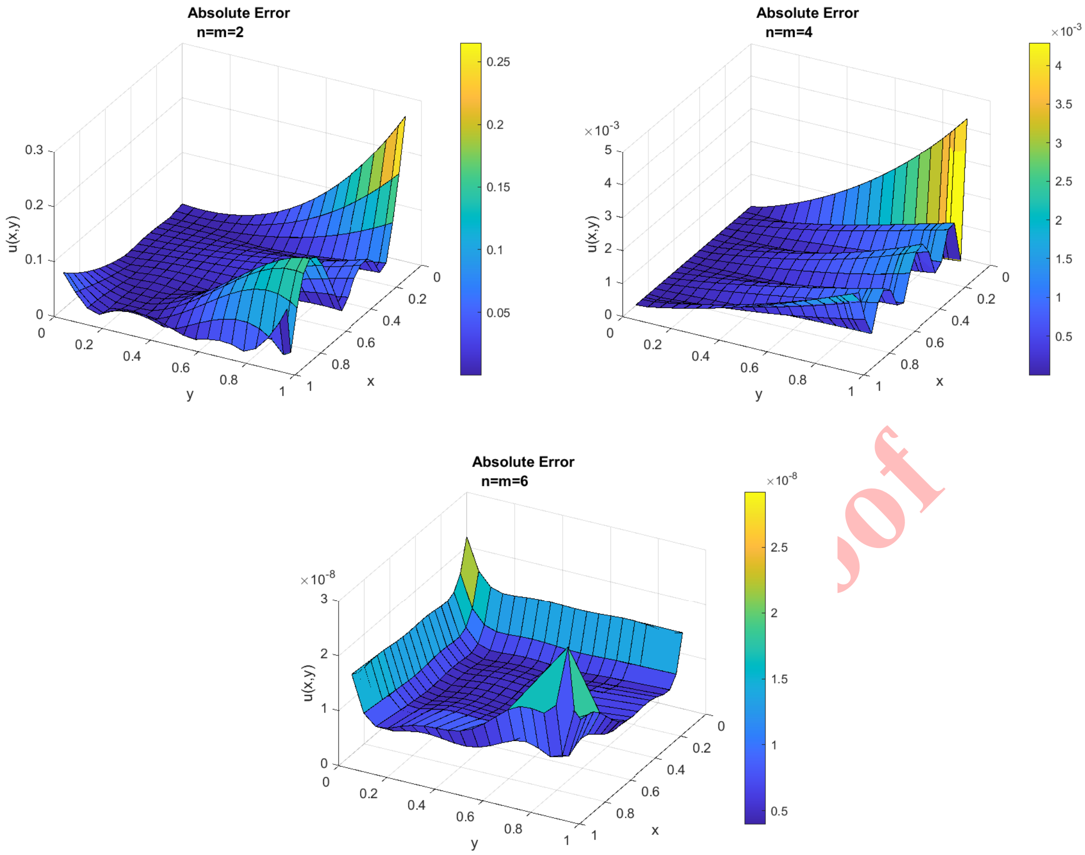

FIGURE 2. Absolute errors for Example 7.2.

TABLE 2. Absolute errors: $|e_m(x,y)|$ for Example 7.2 at different points.

(x,y)	Present method		
	$m=5$	$m=6$	$m=7$
(0.05, 0.05)	2.96E-6	1.02E-8	2.17E-8
(0.15, 0.15)	4.09E-6	4.02E-9	4.91E-9
(0.25, 0.25)	1.42E-5	4.82E-9	4.98E-9
(0.35, 0.35)	8.37E-6	4.85E-9	4.97E-9
(0.45, 0.45)	1.89E-5	4.38E-9	3.07E-9
(0.55, 0.55)	2.19E-5	5.64E-9	4.60E-9
(0.65, 0.65)	1.90E-5	8.18E-9	9.38E-9
(0.75, 0.75)	4.68E-5	7.85E-9	9.68E-9
(0.85, 0.85)	1.80E-5	7.92E-9	2.55E-9
(0.95, 0.95)	3.35E-5	1.15E-7	3.34E-8
$\|e_m\|_\infty$	4.68E-5	1.15E-7	3.34E-8

Table 3. CPU time(s) of proposed method for Example 7.2.

	Compute $\Psi(x,y)$	Compute ς	Compute Φ_1	Compute Φ_2	Compute T_{2D} matrix	Compute ν_{2D}	Total CPU time(s)	Max error
$m=4$	0.000778	0.009883	0.021268	0.022386	0.001766	0.000733	1.472287	6.06E-3
$m=5$	0.023839	0.023839	0.040263	0.041343	0.002087	0.000814	3.837237	2.04E-4
$m=6$	0.001223	0.055802	0.064608	0.065971	0.003638	0.000569	8.588116	3.74E-7
$m=7$	0.001068	0.128283	0.115509	0.116243	0.004143	0.000548	22.689024	3.47E-7
$m=8$	0.001303	0.255771	0.185068	0.178971	0.006797	0.000784	60.744029	2.10E-7
$m=9$	0.001805	0.489955	0.287980	0.326172	0.007817	0.000811	165.824428	2.36E-7

Table 4. Absolute errors: $|e_m(x,y)|$ for Example 7.3.

(x,y)	Legendre polynomials method [17] $m=4$	Chebyshev polynomials method [17] $m=4$	Haar wavelet method [3] $m=32$	Present method $m=4$
(0.5, 0.5)	9.2E-10	1.1E-10	3.1E-2	4.9E-10
(0.25, 0.25)	8.0E-10	1.7E-10	3.1E-2	6.9E-11
(0.125, 0.125)	7.0E-10	8.3E-10	3.1E-2	5.6E-11
(0.0625, 0.0625)	5.3E-10	5.2E-10	3.1E-2	6.3E-13
(0.03125, 0.03125)	8.0E-10	2.5E-10	1.2E-3	1.5E-11
(0.015625, 0.015625)	1.2E-10	1.1E-10	2.2E-9	1.4E-11
$\|e_m\|_\infty$	9.2E-10	8.3E-10	3.1E-2	4.9E-10

where

$$f(x,y) = x + y - \frac{1}{12}xy(x^3 + 4x^2y + 4xy^2 + y^3).$$

The exact solution of the equation is given by $g(x,y) = x + y$. We present the numerical results of this example, comparing the absolute error and maximum error of the proposed method for $m = 4$ in Table 4 with other methods, such as Haar wavelets [3], Chebyshev polynomials [17], and Legendre polynomials [17]. The error of the proposed method is lower compared to these methods. The plots of the absolute error, approximate solution, and exact solution for $m = 4$ are shown in Figure 3. Additionally, we report the CPU time in Table 8, confirming the efficiency of the method.

Example 7.4 ([25])**.** Consider the following two-dimensional integral equation:

$$g(x,y) = f(x,y) + \int_0^1 \int_0^1 (t \cdot \sin(s) + 1)g(t,s)dtds; \quad (x,y) \in D, \tag{7.4}$$

where

$$f(x,y) = x \cdot \cos(y) - \frac{\sin(1)}{6}(\sin(1) + 3).$$

The exact solution of the equation is given by $g(x,y) = x \cdot \cos(y)$. We present the numerical results of this example, comparing the absolute error and maximum error of the proposed method with the method [25] in Table 5. The plots of the absolute error, approximate solution, and exact solution are shown in Figures 4 and 5. Additionally, we report the CPU time in Table 8, confirming the efficiency of the method.

Example 7.5 ([25])**.** Consider the following two-dimensional Volterra integral equation:

$$g(x,y) = f(x,y) + \int_0^y \int_0^x (x \cdot y^2 + \cos(s))g^2(t,s)dtds; \quad (x,y) \in D, \tag{7.5}$$

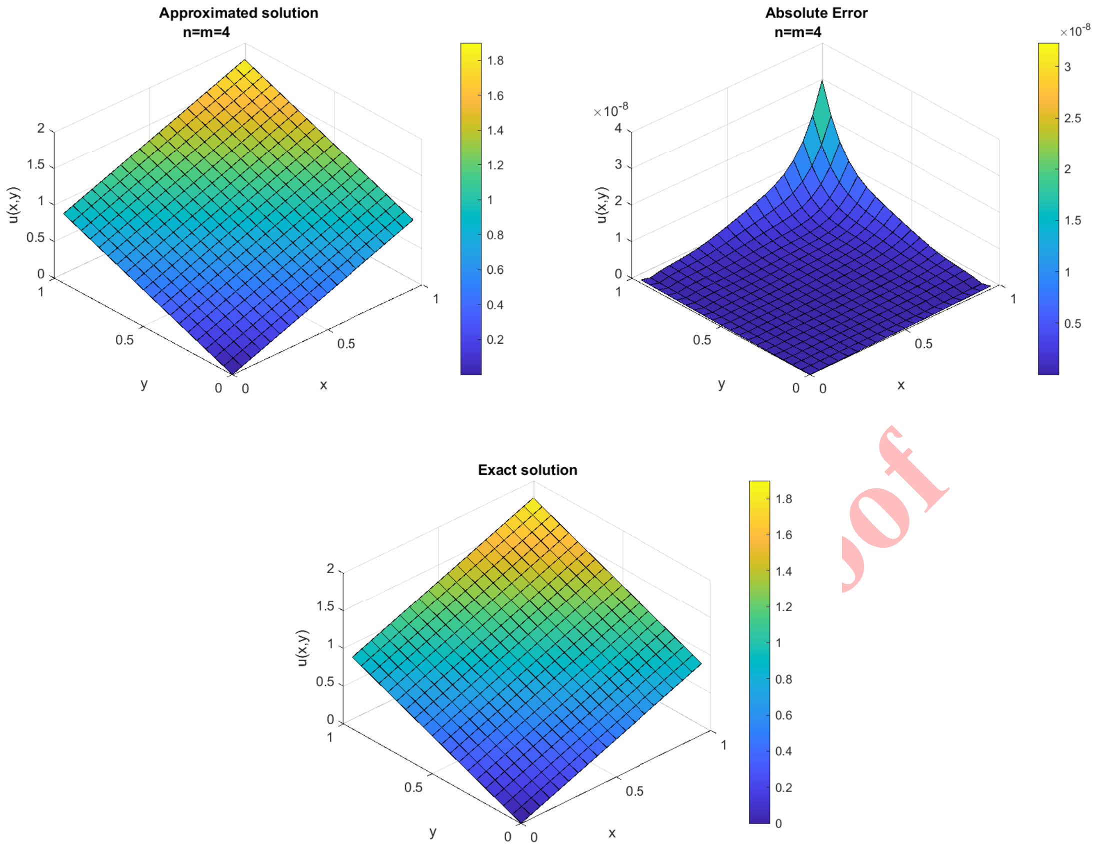

FIGURE 3. Plots of numerical results for Example 7.3.

TABLE 5. Absolute errors: $|e_m(x, y)|$ for Example 7.4.

(x, y)	Present method		Method in [25]	
	$m = 7$	$m = 8$	$m = 8$	$m = 16$
(0.0, 0.0)	7.37E-8	7.57E-9	9.50E-6	5.91E-7
(0.1, 0.1)	1.09E-7	9.74E-10	1.02E-5	4.79E-7
(0.2, 0.2)	1.13E-7	1.38E-8	5.92E-6	2.95E-7
(0.3, 0.3)	1.12E-7	7.16E-9	2.25E-5	1.30E-6
(0.4, 0.4)	1.13E-7	3.14E-9	2.07E-7	3.18E-6
(0.5, 0.5)	1.13E-7	1.70E-8	9.50E-6	5.91E-7
(0.6, 0.6)	1.13E-7	3.14E-9	3.12E-5	4.48E-6
(0.7, 0.7)	1.12E-7	7.16E-9	4.29E-5	2.89E-6
(0.8, 0.8)	1.13E-7	1.38E-8	8.49E-5	5.11E-6
(0.9, 0.9)	1.09E-7	9.69E-10	3.38E-5	1.18E-5
(1.0, 1.0)	7.37E-8	7.57E-9	9.50E-6	5.91E-7
$\|e_m\|_\infty$	1.13E-7	1.38E-8	8.49E-5	1.18E-5

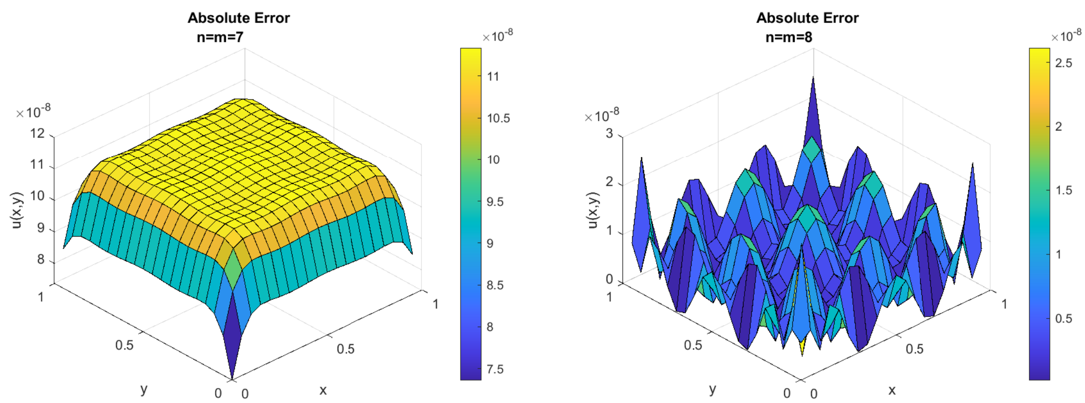

FIGURE 4. Absoulte errors for Example 7.4.

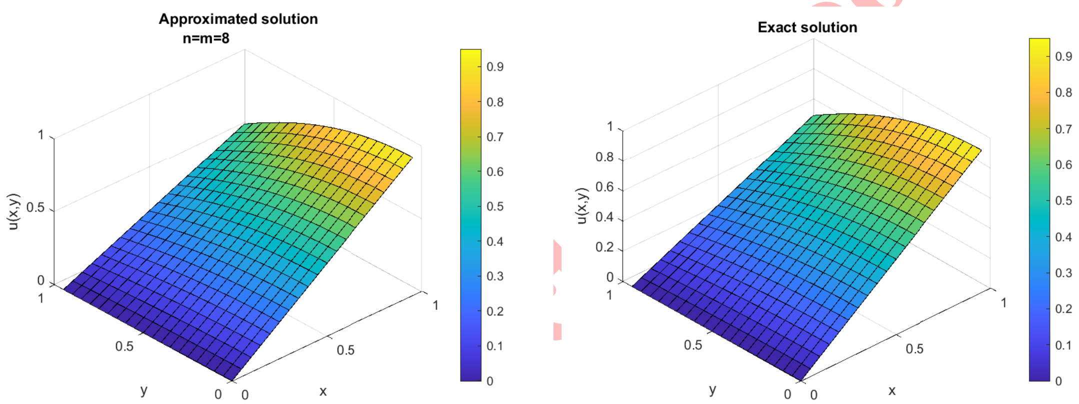

FIGURE 5. Plots of numerical results for Example 7.4.

where

$$f(x,y) = \frac{x^6}{20}(\sin(2y) - 2y) + \frac{x}{9}\sin(y)(9 - x^2 \cdot \sin^2(y)).$$

The exact solution of the equation is given by $g(x, y) = x \cdot \sin(y)$. We present the numerical results of this example, comparing the absolute error and maximum error of the proposed method for $m = 4, 8$ with another methods in Table 6. The plots of the absolute error, approximate solution, and exact solution are shown in Figures 6 and 7. Additionally, we report the CPU time in Table 8, confirming the efficiency of the method.

Example 7.6 ([4]). Consider the following two-dimensional Volterra-Fredholm integral equation:

$$g(x,y) = f(x,y) + \int_0^y \int_0^x xsg^2(t,s)dtds + \int_0^1 \int_0^1 (t-y)g(t,s)dtds; \quad (x,y) \in D, \tag{7.6}$$

where

$$f(x,y) = \frac{-1}{16}(16y + 16\cos(1) - 16\cos(2) - 32\sin(1) + 16\sin(2) - 16\cos(y+x)$$
$$+ 4y^2x^2 - x\sin(2y) - x\sin(2x) - 32y\cos(1) + 16y\cos(2) + x\sin(2y+2x)$$

$$+\,2yx\cos(2y) - 2yx\cos(2y+2x)).$$

The exact solution of the equation is given by $g(x, y) = \cos(x + y)$. We present the numerical results of this example, comparing the absolute error and maximum error of the proposed method for $m = 6$ with another methods in Table 7.

TABLE 6. Absolute errors: $|e_m(x, y)|$ for Example 7.5.

(x, y)	Present method		Method in [25]		Method in [3]	
	$m = 4$	$m = 8$	$m = 4$	$m = 8$	$m = 4$	$m = 8$
(0.5, 0.5)	3.8E-6	2.2E-10	7.7E-6	1.1E-6	1.2E-1	6.0E-2
(0.25, 0.25)	2.8E-8	1.5E-11	4.5E-7	1.2E-7	7.5E-2	3.4E-2
(0.125, 0.125)	1.3E-9	1.0E-12	2.3E-6	2.1E-9	3.8E-5	1.9E-2
(0.0625, 0.0625)	7.0E-9	3.5E-14	3.0E-6	2.0E-8	1.2E-2	2.5E-6
(0.03125, 0.03125)	8.0E-9	3.5E-14	1.2E-6	4.0E-8	1.4E-2	2.9E-3
(0.015625, 0.015625)	1.0E-8	3.1E-15	3.6E-7	1.7E-8	1.5E-2	3.6E-3
$\|e_m\|_\infty$	3.8E-6	2.2E-10	7.7E-6	1.1E-6	1.2E-1	6.0E-2

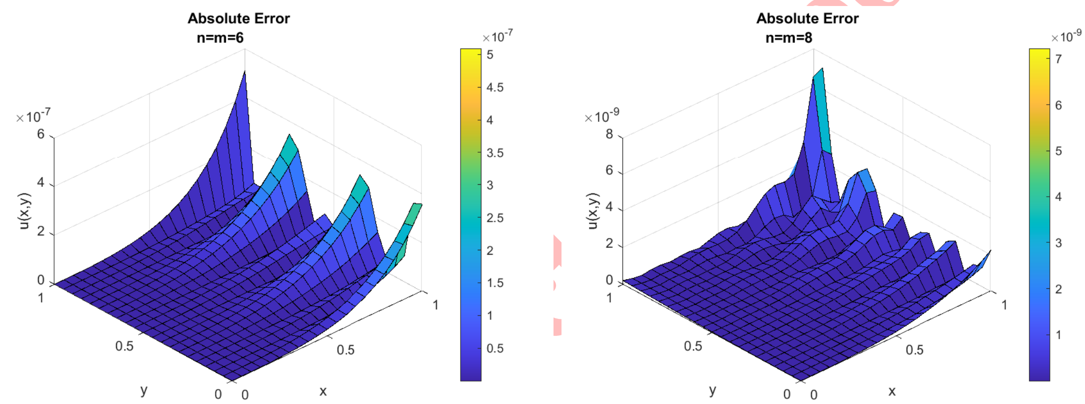

FIGURE 6. Absoule errors for Example 7.5.

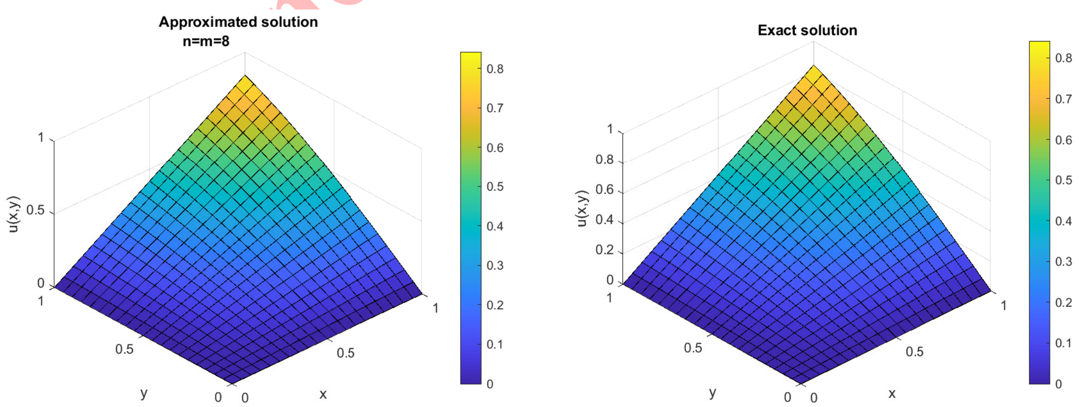

FIGURE 7. Plots of numerical results for Example 7.5.

Table 7. Absolute errors: $|e_m(x,y)|$ for Example 7.6.

(x,y)	Method, $m=6$	
	Present method	Method in [4]
(1.0, 1.0)	3.2E-6	3.6E-6
(0.5, 0.5)	4.0E-8	1.1E-5
(0.25, 0.25)	9.2E-9	4.5E-5
(0.125, 0.125)	8.7E-8	2.2E-5
(0.0625, 0.0625)	9.7E-8	1.8E-5
(0.03125, 0.03125)	9.0E-8	1.8E-5
(0.015625, 0.015625)	8.9E-8	1.8E-5
$\|e_m\|_\infty$	3.2E-6	4.5E-5

Table 8. Total CPU time (s) of examples.

Example	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$
Example 2	0.680725	1.716595	4.080173	9.922513	29.200006	64.32143
Example 3	0.556352	1.549495	5.058304	13.604409	35.530252	88.656782
Example 4	1.437029	7.959578	22.511002	27.06528	53.315282	127.049667
Example 5	1.065567	2.168973	7.279949	18.601045	52.242704	126.166915

The plots of the absolute error, approximate solution, and exact solution are shown in Figures 8 and 9. Additionally, we report the CPU time in Table 8, confirming the efficiency of the method.

8. Conclusion

In this research, we successfully applied 2D-ALPs operational matrices and the collocation method to solve two-dimensional nonlinear integral equations. A key contribution of this work is the development of explicit formulas for product, integration, and diagonal operational matrices, which enable efficient approximation of functions involving two and four variables. One of the significant advantages of the proposed method is that it simplifies the original problem into a nonlinear system of algebraic equations, achieving accurate results with only a small number of basis functions. The effectiveness of the method was demonstrated through several numerical examples, which showed that the approach is both computationally efficient and highly accurate. Additionally, the error analysis of the method was

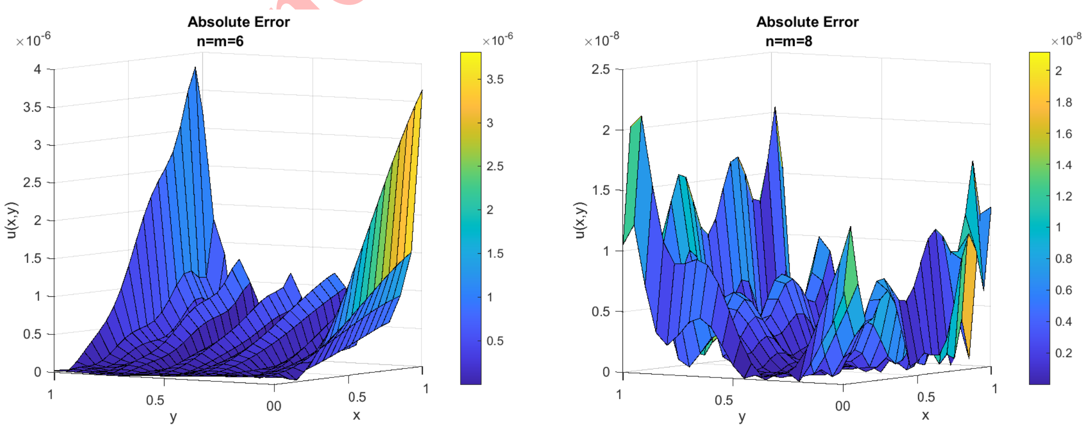

Figure 8. Absolute errors for Example 7.6.

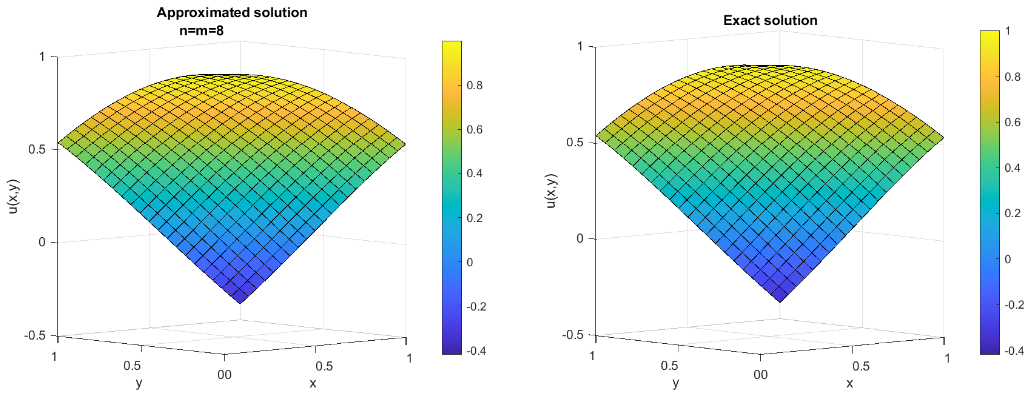

FIGURE 9. Plots of numerical results for Example 7.6.

rigorously examined through three theoretical theorems. Numerical experiments, presented in tabular form within the examples, demonstrate that the proposed numerical solution achieves both high accuracy and computational efficiency. These results validate the reliability and efficiency of the method, highlighting its potential for solving complex nonlinear integral equations.

Ethical Approval

The authors confirm their consent to participate and their consent for the publication of this manuscript.

Availability of Supporting Data

Data sharing is not applicable to this article since no datasets were generated or analyzed during this study.

Competing Interests

The authors declare that there are no conflicts of interest.

Funding

This research did not receive any external funding.

Acknowledgements

The authors would like to express our appreciation to the editor and referees for their valuable comments and constructive suggestions, which have enhanced the quality of this paper.

References

[1] M. A. Abdou, W. G. El-sayed, and E. I. Deebs, *A solution of a nonlinear integral equation*, Journal of Applied Mathematics and Computing, *160*(1) (1988), 1–14.

[2] E. Babolian, K. Maleknejad, M. Roodaki, and H. Almasieh, *Two-dimensional triangular functions and their applications to nonlinear 2D Volterra-Fredholm integral equations*, Computers and Mathematics with Applications, *60*(6) (2010), 1711–1722.

[3] E. Babolian, S. Bazm, and P. Lima, *Numerical solution of nonlinear two-dimensional integral equations using rationalized Haar functions*, Communications in Nonlinear Science and Numerical Simulation, *16*(3) (2011), 1164–1175.

[4] S. Bazm, *Numerical solution of a class of non-linear two-dimensional integral equations using Bernoulli polynomials*, Sahand Communications in Mathematical Analysis, *3*(1) (2016), 37–51.

[5] S. Bazm and A. Hosseini, *Numerical solution of nonlinear integral equations using alternative Legendre polynomials*, Journal of Applied Mathematics and Computing, *4*(1) (2016), 57–77.

[6] M. Behroozifar and S. A. Yousefi, *Operational matrices of Bernstein polynomials and their applications*, International Journal of Systems Science, *41*(6) (2010), 709–716.

[7] R. Y. Chang and M. L. Wang, *Parameter identification via shifted Legendre polynomials*, International Journal of Systems Science, *13*(10) (1982), 1125–1135.

[8] V. S. Chelyshkov, *Alternative orthogonal polynomials and quadratures*, Electronic Transactions on Numerical Analysis, *25*(7) (2006), 17–26.

[9] C. F. Chen and C. H. Hsiao, *A Walsh series direct method for solving variational problems*, Journal of the Franklin Institute, *300*(4) (1975), 265–280.

[10] F. Deutsch, *Best Approximation in Inner Product Spaces*, Springer, New York, 2001.

[11] E. Hadadiyan and F. Mirzaee, *A new numerical method for solving two-dimensional Volterra-Fredholm integral equations*, Journal of Applied Mathematics and Computing, *52*(1) (2015), 489–513.

[12] E. Hadadian and F. Mirzaee, *A new computational method for solving two-dimensional Stratonovich Volterra integral equation*, Mathematical Methods in the Applied Sciences, *40*(16) (2017), 4577–5791.

[13] S. A. Hosseini, S. Shahmorad, and A. Tari, *Existence of an Lp-solution for two dimensional integral equations of the Hammerstein type*, Bulletin of the Iranian Mathematical Society, *40*(4) (2014), 851–862.

[14] I. R. Horng and J. H. Chou, *Shifted Chebyshev direct method for solving variational problems*, International Journal of Systems Science, *16*(7) (1985), 855–861.

[15] C. Hwang and Y. P. Shih, *Parameter identification via Laguerre polynomials*, International Journal of Systems Science, *13*(2) (1982), 209–217.

[16] K. Maleknejad, J. Rashidinia, and T. Eftekhari, *Numerical solutions of two-dimensional nonlinear fractional Volterra and Fredholm integral equations using shifted Jacobi operational matrices via collocation method*, Journal of King Saud University - Science, *33*(1) (2021), 1–11.

[17] O. R. Navid Samadi, J. S. Nadjafi, and E. Tohidi, *Numerical solution of two-dimensional Volterra integral equations by spectral Galerkin method*, Journal of Applied Mathematics and Bioinformatics, *1*(2) (2011), 159–174.

[18] P. N. Paraskevopoulos, *Chebyshev series approach to system identification, analysis and control*, Journal of The Franklin Institute, *316*(2) (1983), 135–157.

[19] P. N. Paraskevopoulos, P. Sklavounos, and G. C. H. Georgiou, *The operation matrix of integration for Bessel functions*, Journal of The Franklin Institute, *327*(2) (1990), 329–341.

[20] M. Razzaghi and Y. Ordokhani, *Solution of nonlinear Volterra-Fredholm-Hammerstein integral equations via a collocation method and rationalized Haar functions*, Applied Mathematics Letters, *21*(1) (2008), 4–9.

[21] P. Sannuti, *Analysis and synthesis of dynamic systems via block-pulse functions*, Proceedings of the IEEE, *124*(6) (1977), 569–571.

[22] A. Tari, *On the Existence Uniqueness and Solution of the Nonlinear Volterra Partial Integro-Differential Equations*, International Journal of Nonlinear Sciences and Numerical Simulation, *16*(2) (2013), 152–163.

[23] A. Tari and S. Shahmorad, *A computational method for solving two-dimensional linear Volterra integral equations of the first kind*, Scientia Iranica, *19*(3) (2012), 829–835.

[24] S. Yousefi and M. Razzaghi, *The Legendre wavelets operational matrix of integration*, International Journal of Systems Science, *32*(4) (2001), 495–502.

[25] N. Zainalian Eskandi and S. Abbasbandy, *Numerical solution of 2D Volterra-Fredholm integral equations via modification of hat functions*, Journal of Mathematical Analysis and Applications, *10*(5) (2019), 41–53.

[26] F. Ziyaee and A. Tari, *Regularization method for the two-dimensional Fredholm integral equations of the first kind*, International Journal of Nonlinear Sciences and Numerical Simulation, *18*(3) (2014), 189–194.