Received: xxx **Accepted:** xxx **Published:** xxx.
DOI. xxxxxxx
Volume xxx, Issue xxx, p.p. 1-23: xxx
Research Article
Open Access
Control and Optimization in Applied Mathematics - COAM

Optimal Solution of Volterra-Fredholm Integral Equations Based on the Clique and Pell-Lucas Series Collocation Method

Alpha Peter Lukonde[1] **Homan Emadifar**[2,3]

[1]Department of Mathematics, Faculty of Science, Karadeniz Technical University, Trabzon, Turkiye
[2]Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai-602 105, Tamil Nadu, India
[3]Department of Basic Sciences, Technical and Vocational University (TVU), Tehran, Iran

Correspondence:
Homan Emadifar
E-mail:
hemadifar@tuv.ac.ir

How to Cite
Lukonde, A.P., Emadifar, H. (2027). "Optimal solution of Volterra-Fredholm integral equations based on the Clique and Pell-Lucas series collocation method". *Control and Optimization in Applied Mathematics*, 12(x), 1-23. https://doi.org/10.30473/coam.2026.76783.1375

Abstract. This study presents a numerical technique for solving Volterra–Fredholm integral equations of the second kind. The solution is approximated via two independent polynomial bases, namely the Pell–Lucas polynomials and the Clique polynomials, each paired with a collocation discretisation. Both approaches are evaluated under two distinct sets of collocation points: standard equally spaced points and Chebyshev–Gauss–Lobatto points, the latter included for comparative purposes. Solution accuracy is assessed through the L_2 and L_∞ norms, the absolute error, and the root mean squared error (RMSE), while an upper bound error analysis is provided to establish convergence. Numerical experiments confirm that both methods achieve progressively higher accuracy as the degree N of the approximating polynomial series increases. Where applicable, the results are benchmarked against those reported in the existing literature. All numerical findings are presented in tabular and graphical form.

Keywords. Volterra–Fredholm integral equations, Collocation points, Optimization, Pell–Lucas polynomials, Clique polynomials.

MSC. 65R20; 45L05; 65R10.

1 Introduction

Integral equations appear pervasively across engineering and the applied sciences. In stochastic analysis, for instance, stochastic integral equations serve as a natural framework for characterising random disturbances [19]. A broad class of problems in mathematical physics admits an integral-equation formulation, and it is well known that initial value problems (IVPs) can be transformed systematically into equivalent integral equations [4]. The rigorous study of such equations was consolidated during the twentieth century through advances in functional analysis and the calculus of variations [2]; indeed, integral equations of the second kind can be traced to Liouville's 1837 treatment of the Sturm–Liouville eigenvalue problem [2].

Because the majority of integral equations with genuine physical content resist closed-form analysis, a substantial body of numerical methodology has developed around them. Costarelli and Spigler [3] constructed a sigmoidal-function approximation for Volterra integral equations of the second kind, and collocation strategies for integral equations have since been pursued by numerous researchers [1, 6, 9, 10, 16, 17]. Within the Volterra–Fredholm subclass specifically, Dhari, et al. [5] employed a linear programming formulation; Negarchi and Nouri [11] adopted Müntz–Legendre polynomials; Nemati [13] developed a Legendre collocation scheme; and Wang and Wang [18] proposed a Lagrange collocation approach. Amin et al. [14] applied the Haar wavelet method to the same class of equations. More recently, Fermo at al. [7] introduced a global method for second-kind Volterra–Fredholm equations, while Salim et al. [15] combined weighted residual techniques with a non-polynomial spline function. Negarchi and Nouri [12] obtained approximate solutions through Bernstein polynomials in conjunction with a least-squares procedure.

The two polynomial bases employed in the present work each possess attributes that justify their selection. Clique polynomials have been applied successfully to functional differential equations by Mohammad et al. [8] and by Yıldırım and Yüzbaşı [20], consistently producing results that converge to the analytical solution. Pell–Lucas polynomials, meanwhile, were employed by Demir et al. [4] to solve integro-differential equations and have been shown to offer faster convergence relative to other collocation bases, thereby reducing the number of computations required to attain a prescribed accuracy [21]. In addition, the Clique polynomial basis exhibits a structural flexibility that extends its applicability beyond more specialised alternatives. These complementary strengths motivate the combined investigation undertaken here.

The present paper develops a collocation method, based on both Clique and Pell–Lucas polynomials, for the numerical solution of Volterra–Fredholm integral equations of the second kind with a proportional delay.

The remainder of this paper is organised as follows. Section 2 develops the general mathematical formulation of the collocation method, encompassing the derivation of the fundamental matrix equation and its reduction to a linear algebraic system. Section 3 introduces the opera-

tional matrices for the Pell–Lucas and Clique polynomial families, which form the core building blocks of the discretisation. Error estimation techniques — comprising the L_2 norm, the L_∞ norm, the absolute error, and the root mean squared error (RMSE) — are described in Section 4, while a convergence analysis based on an upper bound for the residual function is established in Section 5. Four numerical examples that validate the proposed method and benchmark its performance against results available in the literature are presented in Section 6. The step-by-step computational procedure is summarised as a structured algorithm in Section 7, and concluding remarks on the performance, limitations, and potential extensions of the method are given in Section 8.

2 Mathematical Formulation

This section establishes the mathematical formulation of the problem and derives the fundamental matrix relations underpinning the method. The Volterra–Fredholm integral equation of the second kind is given by

$$A(x)\,y(x) + B(x)\,y(h(x)) = f(x) + I^f + I^v, \tag{1}$$

where

$$I^f = \lambda_1 \int_a^b Q_1(x,t)\,y(h(t))\,dt, \qquad I^v = \lambda_2 \int_a^{h(x)} Q_2(x,t)\,y(t)\,dt.$$

Here, $A(x)$, $B(x)$, $f(x)$, $Q_1(x,t)$, and $Q_2(x,t)$ are known continuous functions on $a \le x,\, t \le b$, and $h(x)$ is a linear function. The parameters λ_1 and λ_2 are physically meaningful eigenvalues. When $h(x)$ is a monomial of degree one, equation (1) is referred to as a second-kind integral equation with a proportional delay [13].

The solution $y(x)$ is approximated by a truncated polynomial series $y_N(x)$ of the form

$$y_N(x) = \sum_{k=0}^{N} c_k\, S_k(x), \tag{2}$$

where $\{S_k(x)\}_{k=0}^{N}$ is a prescribed basis of special polynomials and $\{c_k\}_{k=0}^{N}$ are unknown coefficients to be determined. The row vector of basis polynomials is written as

$$\mathbf{S}(x) = \begin{bmatrix} S_0(x) & S_1(x) & S_2(x) & \cdots & S_N(x) \end{bmatrix}. \tag{3}$$

This vector admits the factored representation

$$\mathbf{S}(x) = \mathbf{X}(x)\,\mathbf{\Omega}, \tag{4}$$

where

$$\mathbf{X}(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^N \end{bmatrix}_{1\times(N+1)}, \tag{5}$$

is the monomial row vector, and $\mathbf{\Omega}$ is the $(N+1)\times(N+1)$ operational matrix associated with the chosen polynomial family. Combining (2) and (4), the approximate solution takes the compact matrix form

$$y_N(x) = \mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C}, \tag{6}$$

where

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_N \end{bmatrix}^T_{(N+1)\times 1}.$$

Substituting (6) into (1) yields

$$A(x)\,\mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C} + B(x)\,\mathbf{X}(h(x))\,\mathbf{\Omega}\,\mathbf{C} = f(x) + I^f + I^v, \tag{7}$$

with

$$I^f = \lambda_1 \int_a^b Q_1(x,t)\,\mathbf{X}(h(t))\,\mathbf{\Omega}\,\mathbf{C}\,dt, \qquad I^v = \lambda_2 \int_a^{h(x)} Q_2(x,t)\,\mathbf{X}(t)\,\mathbf{\Omega}\,\mathbf{C}\,dt.$$

Since $h(x)$ is the linear function

$$h(x) = \beta x, \tag{8}$$

direct evaluation of (5) at $h(x)$ gives

$$\mathbf{X}(h(x)) = \begin{bmatrix} 1 & \beta x & (\beta x)^2 & \cdots & (\beta x)^N \end{bmatrix}_{1\times(N+1)}. \tag{9}$$

Factoring the powers of x from (9) and simplifying, one obtains

$$\mathbf{X}(h(x)) = \mathbf{X}(x)\,\mathbf{D} \qquad \text{and} \qquad \mathbf{X}(h(t)) = \mathbf{X}(t)\,\mathbf{D}, \tag{10}$$

where $\mathbf{D}$ is the diagonal scaling matrix

$$\mathbf{D} = \operatorname{diag}\left(1,\, \beta^1,\, \beta^2,\, \ldots,\, \beta^N\right). \tag{11}$$

Substituting (10) into (7) produces

$$A(x)\,\mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C} + B(x)\,\mathbf{X}(x)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C} = f(x) + I^f + I^v, \tag{12}$$

with

$$I^f = \lambda_1 \int_a^b Q_1(x,t)\,\mathbf{X}(t)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}\,dt, \qquad I^v = \lambda_2 \int_a^{h(x)} Q_2(x,t)\,\mathbf{X}(t)\,\mathbf{\Omega}\,\mathbf{C}\,dt.$$

To convert the kernel functions $Q_1(x,t)$ and $Q_2(x,t)$ into matrix form, they are approximated by truncated two-variable Taylor series following the procedure in [4]:

$$Q_1(x,t) = \mathbf{X}(x)\,\mathbf{K}_1\,\mathbf{X}^T(t) \qquad \text{and} \qquad Q_2(x,t) = \mathbf{X}(x)\,\mathbf{K}_2\,\mathbf{X}^T(t), \tag{13}$$

where

$$\mathbf{K}_i = \left[k_{pq}^{(i)}\right], \qquad k_{pq}^{(i)} = \frac{1}{p!\,q!}\,\frac{\partial^{p+q} Q_i(0,0)}{\partial x^p\,\partial t^q}, \quad p,q = 0,1,\ldots,N, \quad i = 1,2. \tag{14}$$

Inserting (13) into (12) transforms equation (1) to

$$A(x)\,\mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C} + B(x)\,\mathbf{X}(x)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C} = f(x) + I^f + I^v, \tag{15}$$

with

$$I^f = \lambda_1 \int_a^b \mathbf{X}(x)\,\mathbf{K}_1\,\mathbf{X}^T(t)\,\mathbf{X}(t)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}\,dt,$$

$$I^v = \lambda_2 \int_a^{h(x)} \mathbf{X}(x)\,\mathbf{K}_2\,\mathbf{X}^T(t)\,\mathbf{X}(t)\,\mathbf{\Omega}\,\mathbf{C}\,dt.$$

Extracting all factors independent of t from beneath the integral signs gives

$$A(x)\,\mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C} + B(x)\,\mathbf{X}(x)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C} = f(x) + I^f + I^v, \tag{16}$$

where

$$I^f = \lambda_1\,\mathbf{X}(x)\,\mathbf{K}_1\left(\int_a^b \mathbf{X}^T(t)\,\mathbf{X}(t)\,dt\right)\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C},$$

$$I^v = \lambda_2\,\mathbf{X}(x)\,\mathbf{K}_2\left(\int_a^{h(x)} \mathbf{X}^T(t)\,\mathbf{X}(t)\,dt\right)\mathbf{\Omega}\,\mathbf{C}.$$

Evaluating these integrals entry by entry yields the matrix $\mathbf{M} = [m_{ij}]$ with

$$m_{ij} = \int_a^b t^{i+j}\,dt = \frac{b^{i+j+1} - a^{i+j+1}}{i+j+1}, \quad i,j = 0,1,\ldots,N, \tag{17}$$

and the x-dependent matrix $\mathbf{V}(x) = [v_{ij}(x)]$ with

$$v_{ij}(x) = \int_a^{h(x)} t^{i+j}\,dt = \frac{(h(x))^{i+j+1} - a^{i+j+1}}{i+j+1}, \quad i,j = 0,1,\ldots,N. \tag{18}$$

Substituting (17) and (18) into (16) produces the algebraic equation

$$A(x)\,\mathbf{X}(x)\,\mathbf{\Omega}\,\mathbf{C} + B(x)\,\mathbf{X}(x)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C} = f(x) + \lambda_1\,\mathbf{X}(x)\,\mathbf{K}_1\,\mathbf{M}\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C} + \lambda_2\,\mathbf{X}(x)\,\mathbf{K}_2\,\mathbf{V}(x)\,\mathbf{\Omega}\,\mathbf{C}. \tag{19}$$

The collocation points are now introduced. The standard equally spaced points are

$$x_i = a + \frac{(b-a)}{N}\,i, \tag{20}$$

while the Chebyshev–Gauss–Lobatto points are

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{\pi i}{N}\right), \quad i = 0,1,\ldots,N. \tag{21}$$

Applying either (20) or (21) to equation (19) at each node x_i generates the linear system

$$A(x_i)\,\mathbf{X}(x_i)\,\mathbf{\Omega}\,\mathbf{C}+B(x_i)\,\mathbf{X}(x_i)\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}=f(x_i)+\lambda_1\,\mathbf{X}(x_i)\,\mathbf{K}_1\,\mathbf{M}\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}+\lambda_2\,\mathbf{X}(x_i)\,\mathbf{K}_2\,\mathbf{V}(x_i)\,\mathbf{\Omega}\,\mathbf{C}. \tag{22}$$

To write (22) in compact matrix form, the following fundamental matrices are introduced:

$$\mathbf{A}=\operatorname{diag}\big(A(x_0),\,A(x_1),\,\ldots,\,A(x_N)\big),$$

$$\mathbf{B}=\operatorname{diag}\big(B(x_0),\,B(x_1),\,\ldots,\,B(x_N)\big),$$

$$\mathbf{F}=\begin{bmatrix} f(x_0) & f(x_1) & \cdots & f(x_N)\end{bmatrix}^{T},$$

$$\overline{\mathbf{V}}=\begin{bmatrix}\mathbf{V}(x_0)\\ \mathbf{V}(x_1)\\ \vdots\\ \mathbf{V}(x_N)\end{bmatrix}\in\mathbb{R}^{(N+1)^2\times(N+1)},$$

$$\mathbf{\Phi}=\begin{bmatrix}\mathbf{X}(x_0)\\ \mathbf{X}(x_1)\\ \vdots\\ \mathbf{X}(x_N)\end{bmatrix}=\begin{bmatrix}1 & x_0 & x_0^2 & \cdots & x_0^N\\ 1 & x_1 & x_1^2 & \cdots & x_1^N\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & x_N & x_N^2 & \cdots & x_N^N\end{bmatrix},$$

$$\overline{\mathbf{K}}_2=\operatorname{diag}\underbrace{\big(\mathbf{K}_2,\,\mathbf{K}_2,\,\ldots,\,\mathbf{K}_2\big)}_{N+1}\in\mathbb{R}^{(N+1)^2\times(N+1)^2},$$

$$\overline{\mathbf{X}}=\operatorname{diag}\big(\mathbf{X}(x_0),\,\mathbf{X}(x_1),\,\ldots,\,\mathbf{X}(x_N)\big).$$

The system (22) then reduces to the fundamental matrix equation

$$\mathbf{A}\,\mathbf{\Phi}\,\mathbf{\Omega}\,\mathbf{C}+\mathbf{B}\,\mathbf{\Phi}\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}=\mathbf{F}+\lambda_1\,\mathbf{\Phi}\,\mathbf{K}_1\,\mathbf{M}\,\mathbf{D}\,\mathbf{\Omega}\,\mathbf{C}+\lambda_2\,\overline{\mathbf{X}}\,\overline{\mathbf{K}}_2\,\overline{\mathbf{V}}\,\mathbf{\Omega}\,\mathbf{C}. \tag{23}$$

The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$, $\mathbf{\Phi}$, $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{M}$ are each of dimension $(N+1)\times(N+1)$. Isolating $\mathbf{F}$ in (23) yields

$$\Big[\mathbf{A}\,\mathbf{\Phi}+\mathbf{B}\,\mathbf{\Phi}\,\mathbf{D}-\big(\lambda_1\,\mathbf{\Phi}\,\mathbf{K}_1\,\mathbf{M}\,\mathbf{D}+\lambda_2\,\overline{\mathbf{X}}\,\overline{\mathbf{K}}_2\,\overline{\mathbf{V}}\big)\Big]\mathbf{\Omega}\,\mathbf{C}=\mathbf{F}. \tag{24}$$

Defining the composite matrix

$$\mathbf{W}=\Big[\mathbf{A}\,\mathbf{\Phi}+\mathbf{B}\,\mathbf{\Phi}\,\mathbf{D}-\big(\lambda_1\,\mathbf{\Phi}\,\mathbf{K}_1\,\mathbf{M}\,\mathbf{D}+\lambda_2\,\overline{\mathbf{X}}\,\overline{\mathbf{K}}_2\,\overline{\mathbf{V}}\big)\Big]\mathbf{\Omega},$$

equation (24) takes the concise form

$$\mathbf{W}\,\mathbf{C}=\mathbf{F}. \tag{25}$$

Provided $\operatorname{rank}(\mathbf{W})=N+1$, the system (25) is uniquely solvable, giving

$$\mathbf{C}=\mathbf{W}^{-1}\mathbf{F}, \tag{26}$$

where $\mathbf{C}$ in (26) collects the polynomial coefficients of the series (2). The desired approximate solution to equation (1) is thereby determined completely by the polynomial collocation procedure.

3 Operational Matrices for Pell–Lucas and Clique Polynomials

3.1 Pell–Lucas Polynomials

The Pell–Lucas polynomials are defined by the three-term recurrence relation

$$Q_n(x) = 2x\,Q_{n-1}(x) + Q_{n-2}(x), \quad n \geq 2,$$

with initial values $Q_0(x) = 2$ and $Q_1(x) = 2x$ [4].

The truncated Pell–Lucas polynomial series is expressed in matrix form as

$$\mathbf{Q}(x) = \begin{bmatrix} Q_0(x) & Q_1(x) & Q_2(x) & \cdots & Q_N(x) \end{bmatrix}_{1\times(N+1)}.$$

The corresponding operational matrix relation is

$$\mathbf{Q}(x) = \mathbf{X}(x)\,\mathbf{L},$$

where $\mathbf{L}^T$ takes the form

$$\mathbf{L}^T = \begin{bmatrix} 2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 2^1\frac{1}{1}\binom{1}{0} & 0 & 0 & \cdots & 0 \\ 2^0\frac{2}{1}\binom{1}{1} & 0 & 2^2\frac{2}{2}\binom{2}{0} & 0 & \cdots & 0 \\ 0 & 2^1\frac{3}{2}\binom{2}{1} & 0 & 2^3\frac{3}{3}\binom{3}{0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 2^1\frac{N}{N+1}\binom{\frac{N+1}{2}}{\frac{N-1}{2}} & 0 & 2^3\frac{N+3}{2}\binom{\frac{N+3}{2}}{\frac{N-3}{2}} & \cdots & 2^N\frac{N}{N}\binom{N}{0} \end{bmatrix}_{(N+1)\times(N+1)} \tag{27}$$

when N is odd, and

$$\mathbf{L}^T = \begin{bmatrix} 2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 2^1\frac{1}{1}\binom{1}{0} & 0 & 0 & \cdots & 0 \\ 2^0\frac{2}{1}\binom{1}{1} & 0 & 2^2\frac{2}{2}\binom{2}{0} & 0 & \cdots & 0 \\ 0 & 2^1\frac{3}{2}\binom{2}{1} & 0 & 2^3\frac{3}{3}\binom{3}{0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 2^0\frac{N}{N}\binom{\frac{N}{2}}{\frac{N}{2}} & 0 & 2^2\frac{N}{N+2}\binom{\frac{N+2}{2}}{\frac{N-2}{2}} & 0 & \cdots & 2^N\frac{N}{N}\binom{N}{0} \end{bmatrix}_{(N+1)\times(N+1)} \tag{28}$$

when N is even. The expressions in (27) and (28) were established by Duygu et al. [4]. In the Pell–Lucas collocation scheme, the matrix $\mathbf{\Omega}$ appearing in (24) is set equal to $\mathbf{L}$ as defined in (27) or (28), according to whether N is odd or even, respectively.

3.2 Clique Polynomials

Yıldırım and Yüzbaşı [20] introduced the Clique polynomials via the recurrence relation

$$C_{n+1}(x) = (1+x)\, C_n(x),$$

with $C_0(x) = 1$ and $C_1(x) = 1 + x$. The Clique polynomial series is expressed in matrix form as

$$\mathbf{C}_N(x) = \mathbf{X}(x)\, \mathbf{M}_N,$$

where

$$\mathbf{M}_N = \begin{bmatrix} \binom{0}{0} & \binom{1}{0} & \binom{2}{0} & \cdots & \binom{N}{0} \\ 0 & \binom{1}{1} & \binom{2}{1} & \cdots & \binom{N}{1} \\ 0 & 0 & \binom{2}{2} & \cdots & \binom{N}{2} \\ 0 & 0 & 0 & \cdots & \binom{N}{3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \binom{N}{N} \end{bmatrix}, \tag{29}$$

and $\mathbf{X}(x)$ is as defined in (5). When Clique polynomials are employed, $\mathbf{\Omega}$ in (24) is taken as $\mathbf{M}_N$ from (29). All remaining matrices in the formulation are unchanged; the sole modification between the two methods is the substitution of the appropriate operational matrix $\mathbf{\Omega}$.

4 Error Estimation

Four measures are employed to assess the accuracy and consistency of the numerical solution: the root mean squared error (RMSE), the L_2 norm, the L_∞ norm, and the absolute error. Since the true solution $y(x)$ is approximated by $y_N(x)$, a residual function $E_R(x)$ necessarily arises, defined pointwise by

$$E_R(x) = y(x) - y_N(x).$$

At the collocation nodes, this residual takes the values

$$E_R(x_i) = y(x_i) - y_N(x_i), \quad i = 0, 1, \ldots, N, \tag{30}$$

and satisfies $E_R(x) \to 0$ as $N \to \infty$. The absolute error is

$$\mathrm{AE}(x) = |E_R(x)|, \tag{31}$$

from which the L_∞ norm follows as

$$L_\infty = \max_{1 \le i \le N+1} \mathrm{AE}(x_i). \tag{32}$$

The RMSE and L_2 norm are defined, respectively, by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N+1} \sum_{i=0}^{N} \big(E_R(x_i)\big)^2}, \qquad L_2 = \sqrt{\sum_{i=0}^{N} \big(\mathrm{AE}(x_i)\big)^2}. \tag{33}$$

5 Convergence Analysis

Substituting the approximate solution $y_N(x)$ into equation (1) gives

$$A(x)\, y_N(x) + B(x)\, y_N(h(x)) - f(x) - I^f - I^v = R_N(x), \tag{34}$$

where

$$I^f = \lambda_1 \int_a^b K_1(x,t)\, y_N(h(t))\, dt, \qquad I^v = \lambda_2 \int_a^{h(x)} K_2(x,t)\, y_N(t)\, dt.$$

The quantity $R_N(x)$ is the residual incurred by the approximation; it satisfies $R_N(x) \to 0$ as N increases without bound [22].

Theorem 1. Let $R_N(x)$ be an integrable residual function on $[a, b]$, and let $\overline{R}_N$ denote the upper bound of $|R_N|$ with respect to N. Then the absolute residual is bounded according to

$$|R_N(t)| \le \frac{\int_a^b |R_N(x)|\, dx}{b-a} = \overline{R}_N, \qquad t \in [a, b], \tag{35}$$

so that the pointwise error does not exceed the normalised area under the absolute residual curve.

Proof. The argument follows the approach of Oğuz and Sezer [22], originally stated on $[0, b]$; here the interval is extended to $[a, b]$, the natural domain of the approximating polynomials. Both the Pell–Lucas and Clique polynomials are continuous on $\mathbb{R}$, as is true of all polynomials, and continuity of $y_N(x)$ then implies continuity of $R_N(x)$ on $[a, b]$.

Applying the triangle inequality for integrals yields

$$\left|\int_a^b R_N(x)\,dx\right| \le \int_a^b |R_N(x)|\,dx. \tag{36}$$

Since R_N is continuous on $[a,b]$, the mean value theorem for integrals guarantees the existence of a point $t \in [a,b]$ such that

$$R_N(t) = \frac{\displaystyle\int_a^b R_N(x)\,dx}{b-a}. \tag{37}$$

Taking absolute values on both sides of (37) gives

$$|R_N(t)| = \frac{\left|\displaystyle\int_a^b R_N(x)\,dx\right|}{b-a}, \tag{38}$$

and applying (36) to the numerator of (38) yields

$$|R_N(t)| \le \frac{\displaystyle\int_a^b |R_N(x)|\,dx}{b-a} = \overline{R}_N, \qquad t \in [a,b],$$

which completes the proof. □

6 Numerical Results

Example 1. Consider the Volterra–Fredholm integral equation [11]

$$y(x) + y\left(\tfrac{x}{2}\right) = f(x) + \lambda_1 \int_0^{h(x)} \left(x - \tfrac{t}{2}\right) y(t)\,dt + \lambda_2 \int_0^1 xt\, y\left(\tfrac{t}{2}\right) dt,$$

where $h(x) = \frac{x}{2}$, $\lambda_1 = 1$, $\lambda_2 = 1$,

$$f(x) = -\tfrac{1}{112}\,x + \tfrac{33}{16}\,x^5 - \tfrac{11}{2688}\,x^7,$$

and the exact solution is $y(x) = 2x^5$.

The numerical results for Example 1 are displayed in Tables 1–2 and Figures 1–2. Table 1 compares the Pell–Lucas and Clique polynomial approximations with the exact solution and the results reported in [11]. Table 2 provides a direct comparison of the error measures for both methods. All computations were performed in Maple 2024 at 32-digit precision, with the exception of Figure 2, for which the default Maple precision was used.

Table 1: Comparison of the exact solution with collocation methods for Example 1.

Point x_i	**Exact** $y(x_i)$	**Pell–Lucas** $N = 20$	**Clique** $N = 20$	**Method in [11]** $N = 20$
0.0	0	-8.63×10^{-20}	0	0.00000
0.1	0.00002	0.000020	0.000020	0.000018
0.2	0.00064	0.000639	0.000639	0.000598
0.3	0.00486	0.004860	0.004860	0.004813
0.4	0.02048	0.020480	0.020480	0.020431
0.5	0.06250	0.062500	0.062500	0.062434
0.6	0.15552	0.155520	0.155520	0.155462
0.7	0.33614	0.336140	0.336140	0.336224
0.8	0.65536	0.655360	0.655360	0.655085
0.9	1.18098	1.180980	1.180980	1.180420
1.0	2.000000	2.000000	2.000000	1.999538

Table 2: Comparison of error measures for the Pell–Lucas and Clique polynomial methods using standard collocation points for Example 1.

Error	**Pell–Lucas** $N = 5$	**Pell–Lucas** $N = 10$	**Pell–Lucas** $N = 30$	**Clique** $N = 5$	**Clique** $N = 10$	**Clique** $N = 30$
L_2	2.09×10^{-7}	1.19×10^{-15}	1.67×10^{-26}	1.26×10^{-3}	5.80×10^{-10}	2.80×10^{-29}
L_∞	2.00×10^{-7}	1.12×10^{-15}	4.22×10^{-27}	7.96×10^{-4}	5.20×10^{-10}	1.00×10^{-29}
RMSE	8.56×10^{-8}	3.59×10^{-16}	3.01×10^{-27}	3.79×10^{-4}	1.75×10^{-10}	5.03×10^{-30}

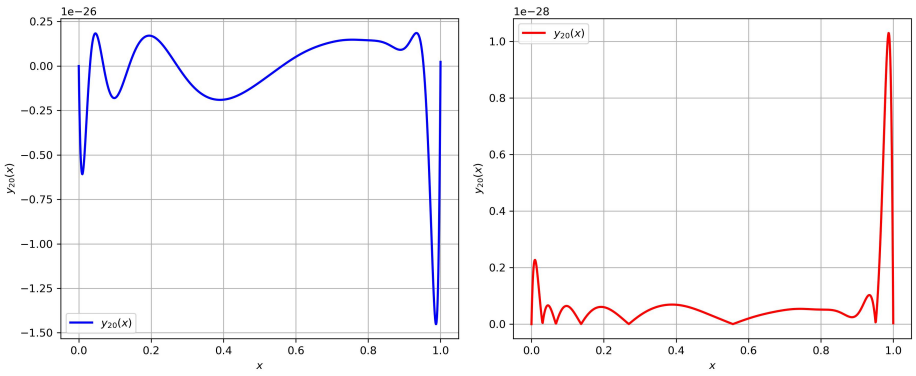

Figure 1: Absolute errors of the Pell–Lucas (right) and Clique polynomial (left) solutions for Example 1 using standard collocation points with $N = 20$.

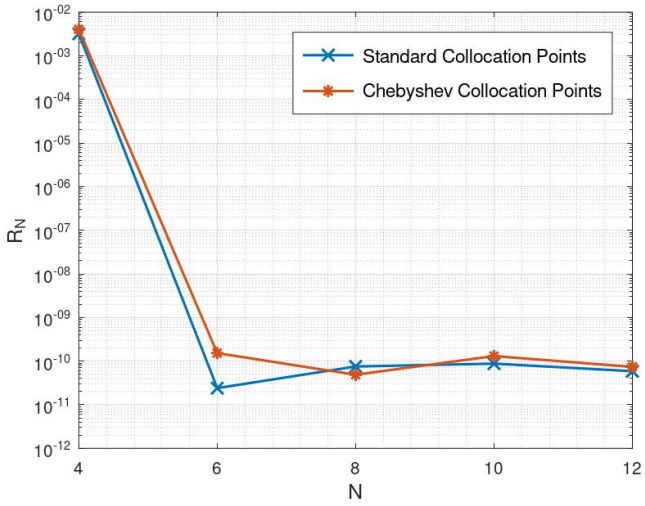

Figure 2: Upper bound errors of the Pell–Lucas solutions under standard and Chebyshev–Gauss–Lobatto collocation points for Example 1. The quantity R_N in the figure denotes $\overline{R}_N$. Computations were performed in Maple 2024 at default precision.

Example 2. Consider the Volterra–Fredholm integral equation [13]

$$x^2 y(x) + e^x y(2x) = f(x) + \lambda_1 \int_0^{2x} e^{x+t}\, y(t)\, dt + \lambda_2 \int_0^1 e^{x-2t}\, y(2t)\, dt,$$

where $h(x) = 2x$, $\lambda_1 = 1$, $\lambda_2 = -1$,

$$\begin{aligned} f(x) = & -\tfrac{e^x}{4} - \tfrac{e^{x-2}}{4}\cos(2) + \tfrac{1}{2}e^{3x}\cos(2x) - \tfrac{1}{4}e^{x-2}\sin(2) \\ & + x^2\sin(x) + e^x\sin(2x) - \tfrac{1}{2}e^{3x}\sin(2x), \end{aligned}$$

and the exact solution is $y(x) = \sin(x)$.

The numerical results for Example 2 are presented in Table 3 and Figures 3–4. Table 3 compares the error measures for both methods under Chebyshev–Gauss–Lobatto collocation points. All computations were performed in Maple 2024 at default precision.

Example 3. Consider the Volterra–Fredholm integral equation [15]

$$y(x) = f(x) + \lambda_1 \int_0^x (x+t)\, y(t)\, dt + \lambda_2 \int_0^1 (x-t)\, y(t)\, dt,$$

where $\lambda_1 = 1$, $\lambda_2 = 1$,

$$f(x) = -\tfrac{9}{10}\, x^5 + 2x^3 - \tfrac{3}{2}\, x^2 - \tfrac{3}{2}\, x + \tfrac{19}{10},$$

and the exact solution is $y(x) = 2x^3 + 1$.

Table 3: Comparison of error measures for the Pell–Lucas and Clique polynomial methods using Chebyshev–Gauss–Lobatto collocation points for Example 2.

Error	Pell–Lucas $N=5$	Pell–Lucas $N=15$	Pell–Lucas $N=45$	Clique $N=5$	Clique $N=15$	Clique $N=45$
L_2	4.46×10^{-2}	2.018×10^{-10}	5.55×10^{-46}	9.19×10^{-2}	2.23×10^{-10}	5.55×10^{-46}
L_∞	3.58×10^{-2}	9.97×10^{-11}	1.47×10^{-46}	6.63×10^{-2}	9.96×10^{-11}	1.47×10^{-46}
RMSE	1.823×10^{-2}	5.047×10^{-11}	7.29×10^{-47}	3.75×10^{-2}	5.59×10^{-11}	8.19×10^{-47}

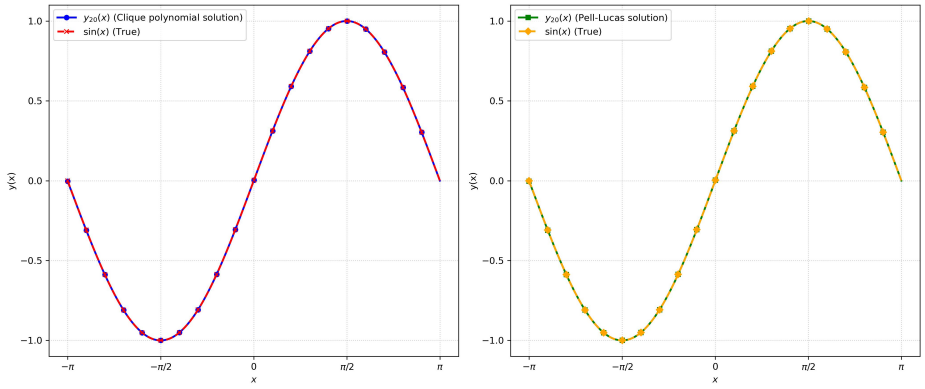

Figure 3: Pell–Lucas and Clique polynomial solutions plotted against the exact solution for Example 2 with $N = 20$ using Chebyshev–Gauss–Lobatto collocation points.

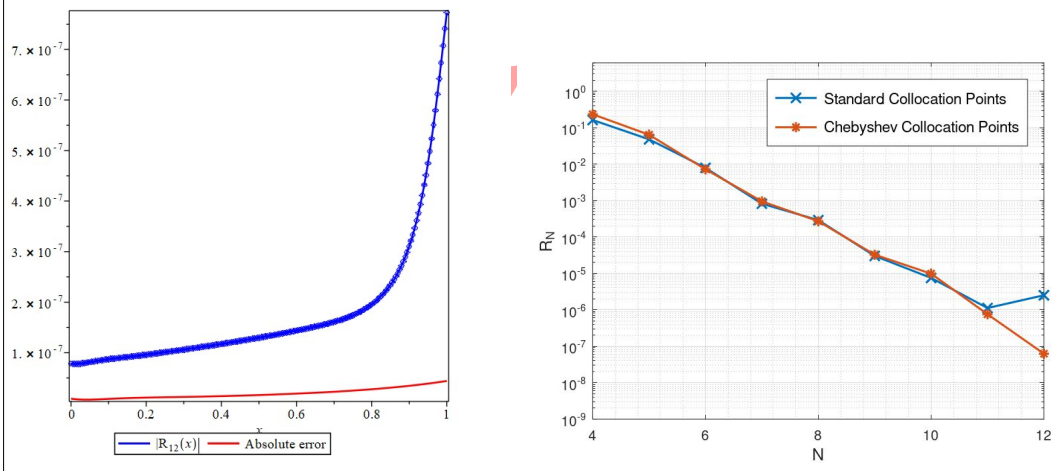

Figure 4: Left: residual and absolute error functions of the Clique polynomial solution for Example 2 at $N = 12$. Right: upper bound errors for selected values of N via the Clique polynomial method. The quantity R_N in both panels denotes $\overline{R}_N$.

The results for Example 3 are presented in Table 4 and Figures 5–6. At $N = 4$, the Clique polynomial collocation method yields

$$y_4(x) = -1.03773166127075 \times 10^{-13}(x+1)^4 + 2.00000000000063\,(x+1)^3 - 6.00000000000138\,(x+1)^2 + 6.00000000000133\,x + 5.00000000000086.$$

The leading coefficient of the quartic term converges to zero, and after straightforward algebraic simplification the remaining terms reduce to a cubic polynomial that converges to the exact solution. The Pell–Lucas collocation method at the same truncation order gives

$$y_4(x) = 7.55328323462513 \times 10^{-15}\,x^4 + 4.98217787888186 \times 10^{-15}\,x^2 + 2.99999999999999\,x^3 - 2.22044604925031 \times 10^{-16}\,x + 1.0,$$

in which the coefficients of x^4, x^2, and x each converge to zero, again leaving a cubic polynomial that converges to the exact solution. All computations were performed in Maple 2024 at default precision.

Table 4: Comparison of the exact solution with numerical methods for Example 3.

Point	**Exact**	**Pell–Lucas**	**Clique**	**Method in [15]**
x_i	$y(x_i)$	$N = 4$	$N = 4$	$N = 10$
0.0	1.000000	1.00000	1.000000	1.0089
0.1	1.002	1.002	1.002	1.0417
0.2	1.016	1.016	1.016	1.0204
0.3	1.054	1.054	1.054	1.0566
0.4	1.128	1.128	1.128	1.1291
0.5	1.250	1.250	1.250	1.2491
0.6	1.432	1.432	1.432	1.4262
0.7	1.686	1.686	1.686	1.6682
0.8	2.024	2.024	2.024	1.9848
0.9	2.458	2.458	2.458	2.3924
1.0	3.000000	3.00000	3.00000	2.3924

Example 4. Consider the Volterra–Fredholm integral equation

$$\sin(x)\,y(x) + \cos(x)\,y\left(\tfrac{x}{2}\right) = f(x) + \lambda_1 \int_{-1}^{h(x)} (x+t)\,y(t)\,dt + \lambda_2 \int_{-1}^{1} \cosh(x-t)\,y\left(\tfrac{t}{2}\right)dt,$$

where $h(x) = \frac{x}{2}$, $\lambda_1 = 1$, $\lambda_2 = 1$, and

$$
\begin{aligned}
f(x) = \; & \sin^2(x) + \cos(x)\sin\left(\tfrac{x}{2}\right) \\
& - \Bigg[\frac{\left(-\cos\left(\frac{1}{2}\right) + 2\sin\left(\frac{1}{2}\right)\right)e^{1-x}}{5} + \frac{\left(\cos\left(\frac{1}{2}\right) + 2\sin\left(\frac{1}{2}\right)\right)e^{-1-x}}{5} \\
& + \frac{\left(-\cos\left(\frac{1}{2}\right) - 2\sin\left(\frac{1}{2}\right)\right)e^{x-1}}{5} + \frac{\left(\cos\left(\frac{1}{2}\right) - 2\sin\left(\frac{1}{2}\right)\right)e^{x+1}}{5} \\
& - \tfrac{3}{2}\cos\left(\tfrac{x}{2}\right)x + \sin\left(\tfrac{x}{2}\right) + (x-1)\cos(1) + \sin(1) \Bigg],
\end{aligned}
$$

with exact solution $y(x) = \sin(x)$.

The Pell–Lucas and Clique polynomial results for Example 4 are presented in Tables 5–6 and Figures 7–8. All computations were performed in Maple 2024 at 16-digit precision.

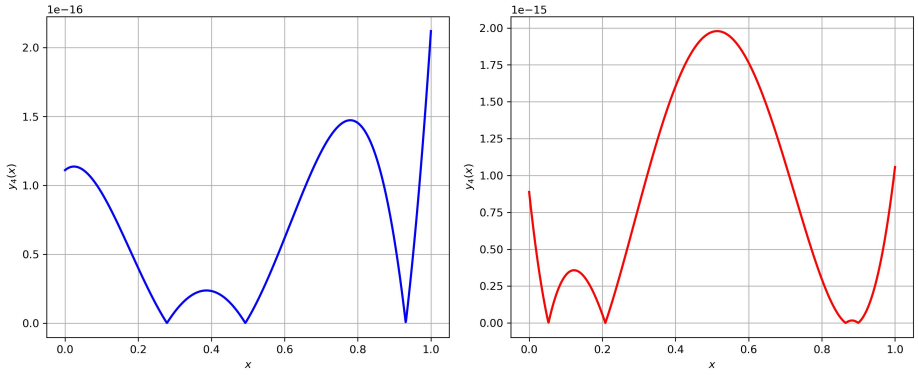

Figure 5: Absolute errors of the Pell–Lucas (right) and Clique polynomial (left) solutions for Example 3 using standard collocation points with $N = 4$.

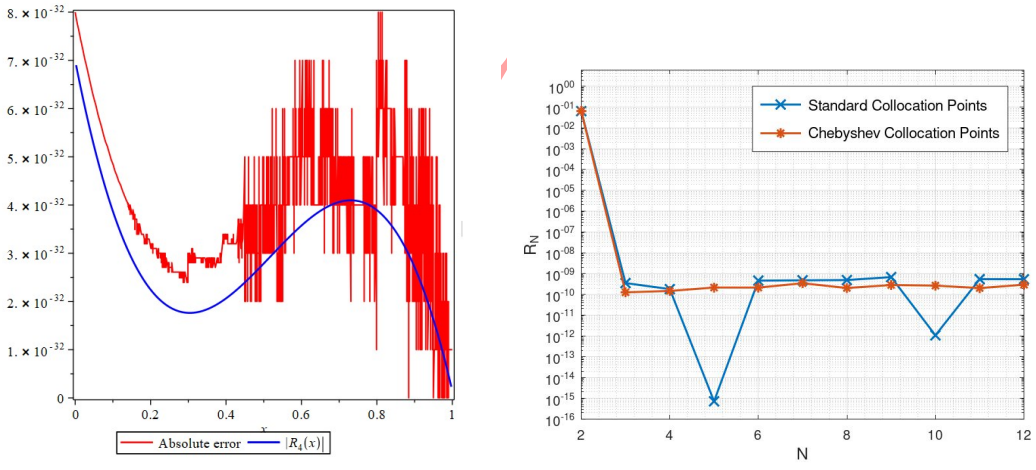

Figure 6: Left: residual and absolute error functions of the Clique polynomial solution for Example 3 at $N = 4$, computed in Maple 2024 at 32-digit precision. Right: upper bound errors for selected values of N via the Pell–Lucas method, computed at default precision. The quantity R_N in both panels denotes $\overline{R}_N$.

Table 5: Comparison of the exact solution with collocation methods for Example 4.

Point	**Exact**	**Pell–Lucas**	**Clique**
x_i	$y(x_i)$	$N = 5$	$N = 5$
0.0	0.000000000000000	0.0000107370512543	0.0000107370512590
0.1	0.0998334166468282	0.0998468911448919	0.0998468911448950
0.2	0.198669330795061	0.198685473618088	0.198685473618087
0.3	0.295520206661340	0.295539467971529	0.295539467971526
0.4	0.389418342308650	0.389441533834613	0.389441533834606
0.5	0.479425538604203	0.479453687838613	0.479453687838600
0.6	0.564642473395035	0.564676984489856	0.564676984489836
0.7	0.644217687237691	0.644261197042898	0.644261197042878
0.8	0.717356090899523	0.717414498373704	0.717414498373677
0.9	0.783326909627483	0.783413141852810	0.783413141852787
1.0	0.841470984807897	0.841611142218513	0.841611142218487

Table 6: Comparison of error measures for the Pell–Lucas and Clique polynomial methods using standard collocation points for Example 4.

Error	**Pell–Lucas**	**Pell–Lucas**	**Pell–Lucas**	**Clique**	**Clique**	**Clique**
	$N = 4$	$N = 8$	$N = 16$	$N = 4$	$N = 8$	$N = 16$
L_2	1.68×10^{-4}	2.15×10^{-6}	1.28×10^{-11}	4.28×10^{-3}	2.15×10^{-6}	5.28×10^{-12}
L_∞	1.40×10^{-4}	1.68×10^{-6}	5.04×10^{-12}	3.36×10^{-3}	1.68×10^{-6}	2.06×10^{-12}
RMSE	6.86×10^{-5}	7.19×10^{-7}	3.11×10^{-12}	1.91×10^{-3}	7.19×10^{-7}	1.28×10^{-12}

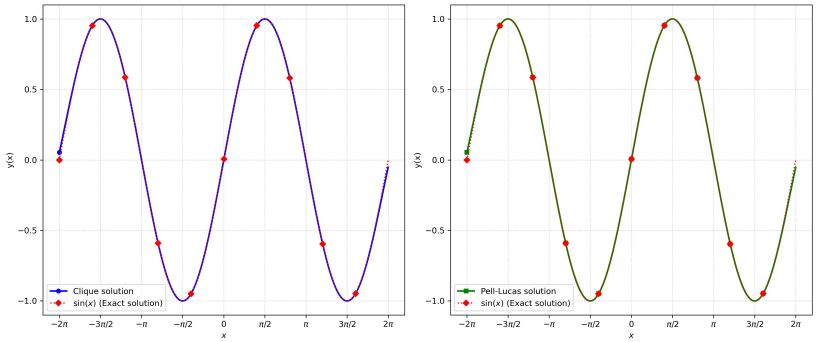

Figure 7: Pell–Lucas and Clique polynomial solutions plotted against the exact solution for Example 4 with $N = 15$.

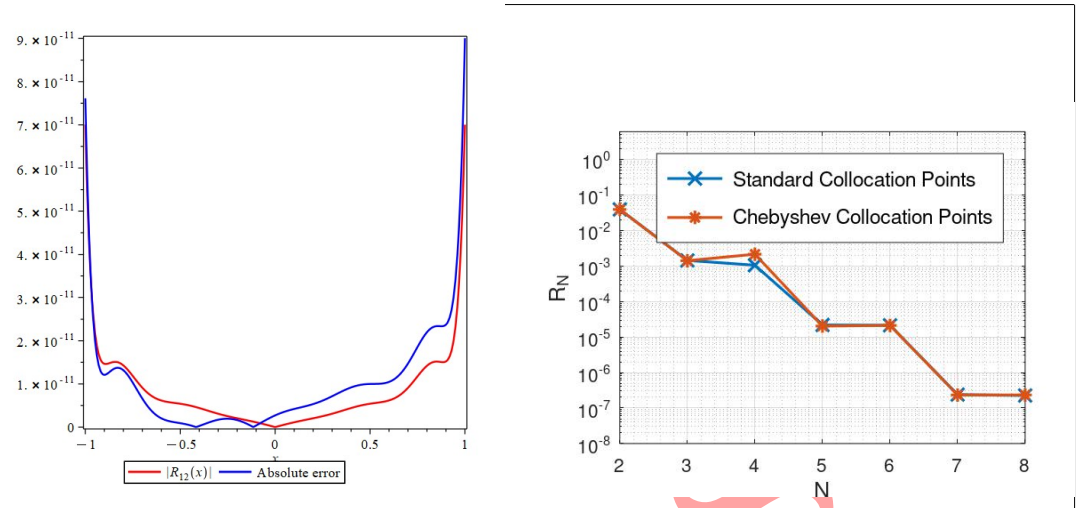

Figure 8: Left: residual and absolute error functions of the Clique polynomial solution for Example 4 at $N = 12$, computed in Maple 2024 at 32-digit precision. Right: upper bound errors for selected values of N via the Pell–Lucas method, computed at default precision. The quantity R_N in both panels denotes $\overline{R}_N$.

7 Algorithm

The method developed in Section 2 is distilled into Algorithm 1 below. The procedure is self-contained and lends itself to direct implementation in any symbolic or numerical computing environment: it walks through the assembly of the system matrix $\mathbf{W}$, the solution of the resulting linear system for the coefficient vector $\mathbf{C}$, and the iterative increase of the truncation order N until the approximate solution $y_N(x)$ meets the prescribed accuracy criterion.

Algorithm 1 Polynomial collocation method for Volterra–Fredholm integral equations

Input: The truncation order N; the interval endpoints a and b; the eigenvalues λ_1 and λ_2; and the functions $A(x)$, $B(x)$, $f(x)$, $h(x)$, $Q_1(x,t)$, and $Q_2(x,t)$.

Step 1. Define the collocation points as either the standard equally spaced nodes

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

or the Chebyshev–Gauss–Lobatto nodes

$$x_i = \frac{1}{2}\left[(a+b) + (b-a)\cos\left(\frac{\pi i}{N}\right)\right], \quad i = 0, 1, \ldots, N.$$

Step 2. Using the collocation points from Step 1, compute the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{\Phi}$, $\mathbf{D}$, $\mathbf{F}$, and $\overline{\mathbf{X}}$.

Step 3. From the kernel functions $Q_1(x,t)$ and $Q_2(x,t)$, compute the coefficient matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ via the truncated Taylor expansion (14).

Step 4. Using $\mathbf{K}_2$ from Step 3, form the block-diagonal matrix $\overline{\mathbf{K}}_2$.

Step 5. Compute the entries of $\mathbf{M}$ using the formula

$$m_{i,j} = \frac{b^{i+j+1} - a^{i+j+1}}{i+j+1}, \quad i, j = 0, 1, \ldots, N.$$

Step 6. Compute the entries of $\mathbf{V}(x)$ using the formula

$$v_{i,j}(x) = \frac{(h(x))^{i+j+1} - a^{i+j+1}}{i+j+1}, \quad i, j = 0, 1, \ldots, N.$$

Step 7. Evaluate $\mathbf{V}(x)$ at each collocation point to assemble the stacked matrix $\overline{\mathbf{V}}$.

Step 8. Select the polynomial basis (Pell–Lucas or Clique) and set its operational matrix $\mathbf{\Omega}$ to $\mathbf{L}$ (equations (27)–(28)) or $\mathbf{M}_N$ (equation (29)), respectively. Then compute the system matrix

$$\mathbf{W} = \left[\mathbf{A\Phi} + \mathbf{B\Phi D} - \left(\lambda_1 \mathbf{\Phi K}_1 \mathbf{M D} + \lambda_2\, \overline{\mathbf{X}}\, \overline{\mathbf{K}}_2\, \overline{\mathbf{V}}\right)\right] \mathbf{\Omega}.$$

Step 9. Solve the linear system $\mathbf{WC} = \mathbf{F}$ for the coefficient vector $\mathbf{C}$.

Step 10. Substitute the computed coefficients $c_0, c_1, \ldots, c_N$ into the series

$$y_N(x) = \sum_{k=0}^{N} c_k\, S_k(x)$$

to obtain the approximate solution.

Iteration. Repeat Steps 1–10 for increasing values of N until either $|y(x) - y_N(x)|$ or $|R_N(x)|$ is sufficiently small for all $x \in [a, b]$.

8 Conclusion

This study has presented a polynomial collocation method for Volterra–Fredholm integral equations of the second kind with a proportional delay, employing two distinct polynomial bases: Pell–Lucas and Clique polynomials. Analysis of the computed error tables confirms that both bases yield highly accurate approximations, offering a competitive alternative to the more conventional orthogonal polynomial families used in the literature. Direct comparison with existing methods — see Examples 1 and 3 — demonstrates that the present approach achieves superior accuracy at comparable or lower truncation orders. The framework is general in structure: to apply a different polynomial basis one need only substitute the corresponding operational matrix $\mathbf{\Omega}$, leaving all remaining matrix relations unchanged. A further advantage is that upper bound errors can be estimated from the residual of the approximate solution alone, without knowledge of the exact solution, thereby providing an independent means of validating the method.

Several conclusions may be drawn from the numerical experiments. The choice of collocation points has no appreciable effect on solution accuracy: the standard equally spaced nodes and the Chebyshev–Gauss–Lobatto nodes produce results that are, in practice, indistinguishable when measured by the reported error norms. Under Chebyshev–Gauss–Lobatto discretisation, moreover, the Pell–Lucas and Clique methods converge at nearly identical rates. A practical remark regarding implementation is also warranted: when computations are carried out in Maple, increasing the working precision substantially reduces truncation errors and is strongly recommended for high-order approximations, as was done throughout this study using Maple 2024.

Limitations

The method is not without limitations. Because the approximating functions are polynomials, achieving higher accuracy requires increasing the truncation order N, which enlarges all system matrices and raises the associated computational cost. In this regime, Chebyshev–Gauss–Lobatto points are preferable, as they suppress the Runge phenomenon that afflicts high-degree polynomial interpolation on equally spaced nodes. A secondary limitation concerns the upper bound error computation: evaluating integrals of absolute-value functions becomes increasingly expensive as N grows, and this cost may be non-negligible in large-scale implementations. Furthermore, because the method relies on a global polynomial representation, its performance may deteriorate for solutions that exhibit sharp gradients or localised behaviour over the domain $[a, b]$.

Future Work

Several directions for future research present themselves naturally. The method can be extended to nonlinear Volterra–Fredholm integral equations, wherein the nonlinear terms are cast in matrix form through the same polynomial collocation procedure under the assumption that the series satisfies the governing equation. An analogous extension to linear and nonlinear integro-differential equations is equally straightforward, as the differential operator can be replaced by its matrix representation within the present framework. Beyond these immediate generalisations, the method may be applied to two-dimensional integral equations and to fractional integro-differential equations, both of which arise frequently in mathematical physics and engineering. The development of adaptive strategies for selecting the truncation order N and the collocation point distribution, with the aim of minimising computational cost for a prescribed accuracy, also merits investigation. Finally, a rigorous theoretical analysis of the convergence rate as a function of the regularity of the solution and the choice of polynomial basis would further consolidate the mathematical foundations of the method.

Declarations

Data Availability and Computational Details

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author Contributions

Alpha Peter Lukonde contributed to supervision, validation, methodology, review and editing of the manuscript, and the overall scientific guidance of the study. All authors read and approved the final manuscript. **Homan Emadifar** contributed to conceptualization, methodology, formal analysis, software implementation, data curation, writing of the original draft, and preparation of the numerical results.

Artificial Intelligence Statement

Artificial intelligence (AI) tools, including large language models, were used solely for language editing and improving readability. AI tools were not used for generating ideas,

performing analyses, interpreting results, or writing the scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

[1] Ansari, K.J., Usta, F. (2025). "On a modified Bernstein operators approximation method for computational solution of Volterra integral equation". *Journal of Inequalities and Applications*, 8, 1–15. https://doi.org/10.1186/s13660-024-03225-y

[2] Birkhoff, G., Kreyszig, E. (1984). "The establishment of functional analysis". *Historia Mathematica*, 11, 258–321. https://doi.org/10.1016/0315-0860(84)90036-3

[3] Costarelli, D., Spigler, R. (2013). "Solving Volterra integral equations of the second kind using by sigmoidal functions approximation". *Journal of integral equations and applications, 25(2), 193–222*. https://doi.org/10.1216/JIE-2013-25-2-193

[4] Demir, D.D., Lukonde, A.P., Kürkçü, Ö.K., Sezer, M. (2021). "Pell–Lucas series approach for a class of Fredholm-type delay integro-differential equations with variable delays". *Mathematical Sciences*, 15, 55–64. https://doi.org/10.1007/s40096-020-00370-5

[5] Dhari, I.A., Mustafa, N.R., Mustafa, M.M. (2020). "Solving linear Volterra -Fredholm integral equation of the second type using linear programming method". *Baghdad Science Journal*, 17(1), 342–347. https://doi.org/10.21123/bsj.2020.17.1(Suppl.).0342

[6] Khidir, A.A. (2024). "A numerical technique for solving Volterra-Fredholm integral equations using Chebyshev spectral method". *Ricerche mat*, 73, 1559–1577. https://doi.org/10.1007/s11587-022-00692-7

[7] Fermo, L., Mezzanotte, D., Occorsio, D. (2025). "A global method for solving second-kind Volterra–Fredholm integral equations". *BIT Numerical Mathematics*, 65, 18. https://doi.org/10.1007/s10543-025-01061-0

[8] Mohammad, I., Jagdev, S., Samad, N. (2023). "Simulating accurate and effective solutions of some nonlinear nonlocal two-point BVPs: Clique and QLM-clique matrix methods". *Heliyon*, 9, e22267. https://doi.org/10.1016/j.heliyon.2023.e22267

[9] Mundewadi, R.A., Jummannaver, R.B. (2024). "Hosoya polynomial method for the numerical solution of Volterra integral equations". *Journal of Umm Al-Qura University for Applied Sciences*. https://doi.org/10.1007/s43994-024-00191-5

[10] Mustapha, D. (2023). "Gegenbauer Series for Numerical Solution of Fredholm Integral Equations of the Second Kind". *Communications in Mathematics and Applications*, 14(1), 167–173. https://doi.org/10.26713/cma.v14i1.2003

[11] Negarchi, N., Nouri, K. (2018). "Numerical solution of Volterra–Fredholm integral equations using the collocation method based on a special form of the Müntz–Legendre polynomials". *Journal of Computational and Applied Mathematics*, 344, 15–24. https://doi.org/10.1016/j.cam.2018.05.035

[12] Negarchi, N., Nouri, K. (2023). "Solution of the Volterra-Fredholm integral equations via the Bernstein polynomials and least squares approach". *TWMS Journal of Applied and Engineering Mathematics*, 13(1), 291–299.

[13] Nemati, S. (2015). "Numerical solution of Volterra–Fredholm integral equations using Legendre collocation method". *Journal of Computational and Applied Mathematics*, 278, 29–36. https://doi.org/10.1016/j.cam.2014.09.030

[14] Amin, R., Shah, K., Asif, M., Khan, I. (2020). "Efficient numerical technique for solution of delay Volterra-Fredholm integral equations using Haar wavelet". *Heliyon*, 6(10), e05108. https://doi.org/10.1016/j.heliyon.2020.e05108

[15] Salim, S.H., Saeed, R.K., Jwamer, K.H.F. (2025). "Solving Volterra-Fredholm integral equations by non-polynomial spline function based on weighted residual methods". *Bulletin of the Karaganda University, Mathematics Series*, 117(1), 155–169. https://doi.org/10.31489/2025m1/155-169

[16] Kehali, S., Khirani, A. (2024). "Using Hermite polynomials to solve Volterra-Fredholm integral equation of the second kind". *Mathematical Modelling of Engineering Problems*, 11(8), 2227–2234. https://doi.org/10.18280/mmep.110824

[17] Sezer, M. (1994). "Taylor polynomial solution of Volterra integral equations". *International Journal of Mathematical Education in Science and Technology*, 25(5), 625–633. https://doi.org/10.1080/0020739940250501

[18] Wang, K, Wang, Q. (2013). "Lagrange collocation method for solving Volterra–Fredholm integral equations". *Applied Mathematics and Computation*, 219, 10434–10440. https://doi.org/10.1016/j.amc.2013.04.017

[19] Wen, X., Huang, J. (2021). "A combination method for numerical solution of the nonlinear stochastic Itô-Volterra integral equation". *Applied Mathematics and Computation*, 407, 126302. https://doi.org/10.1016/j.amc.2021.126302

[20] Yıdırım, G., Yüzbaşı, Ş. (2025). "Clique collocation method to solve the third-order multisingular (MS) functional differential equations". *Fundamentals of Contemporary Mathematical Sciences*, 6(1), 59–74. https://doi.org/10.54974/fcmathsci.1470784

[21] Yıdırım, G., Yüzbaşı, Ş. (2024). "A collocation technique via Pell-Lucas polynomials to solve fractional differential equation model for HIV/AIDS with treatment compartment". *Computer Modeling in Engineering & Sciences*, 141(1), 281–310. https://doi.org/10.32604/cmes.2024.052181

[22] Oğuz, C., Sezer, M. (2015). "Chelyshkov collocation method for a class of mixed functional integro-differential equations". *Applied Mathematics and Computation*, 25(5), 625–633. http://dx.doi.org/10.1016/j.amc.2015.03.024

Authors Bio-sketches

Alpha Peter Lukonde graduated with a Bsc Mathematics degree from the Department of Mathematics at Manisa Celal Bayar University in 2018 and received his master degree in 2020 from the same university, currently a doctorate degree student at Karadeniz Technical University. His research interests include numerical solutions of the differential equations especially with the boundary element method, and polynomial techniques for solving integro-differential equations

Homan Emadifar has a Ph.D. in Applied Mathematics with a focus on numerical analysis. His research interests include mathematical computing, mathematical analysis, numerical analysis, real and complex analysis, fractional calculus. He is currently affiliated with Technical and Vocational University as a faculty member. Corresponding author: Email: hemadifar@tuv.ac.ir