EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 4, Article Number 6734
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Comparative Analysis of the Differential Transform Method and Generalized Fibonacci Collocation Method for Solving Differential Equations

Maha Abdalla Abdou[1], Amany Saad Mohamed[2,*]

[1] *Department of Mathematics, Faculty of Education, International University of Islamic and Linguistic Sciences, Postgraduate Studies*
[2] *Department of Mathematics, Faculty of Science, Helwan University, Cairo, Egypt*

Abstract. This study investigates two ways of discussing the solution of linear, mixed, and special nonlinear models of second-order differential equations: the differential transform method and the generalized Fibonacci collocation method. The differential transform method uses a step-by-step approach to convert differential equations and their conditions into power series, giving an exact or highly accurate solution. On the other hand, the generalized Fibonacci collocation method transforms the problem into a system of equations with unknown coefficients, which are determined by solving this system using matrix operations. It yields a numerical solution. We discuss convergence and error bounds of generalized Fibonacci polynomials in detail. This study evaluates the solutions of four different problems, focusing on their reliability and accuracy. This comparison shows that our algorithms are efficient.

2020 Mathematics Subject Classifications: 65L05, 65L70, 34A34, 41A10

Key Words and Phrases: Differential transform method, generalized Fibonacci polynomials, collocation method, convergence and error analysis

---

1. Introduction

A wide range of challenges can be modeled in mathematics, physics, chemistry, and related scientific fields using differential or partial differential equations. These equations, which may exhibit linear or non-linear behavior depending on the characteristics of the system, often pose significant difficulties when seeking exact analytical solutions, particularly in practical, real-world contexts. As a result, numerical and semi-analytical approaches have become essential methodologies for obtaining approximate solutions to these complex problems.

The differential transform method (DTM) is much easier to use compared to the typical higher-order Taylor series method, which requires a lot of symbolic math for derivatives.

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i4.6734

Email address: amany.saad78@yahoo.com (A. S. Mohamed)

Although Taylor series can get complicated and demand more resources as you go higher in order, it offers a simpler and more efficient way to find analytic solutions. In addition, it works well with different types of differential equations (ordinary, partial fractional), which can be sorted into various categories.

Different problems were discussed using this method in various studies. For example, to solve RLC circuits problems, linear and nonlinear pantograph equation, non-linear oscillatory systems [1–3]. It is used to solve non-linear and system of partial differential equations [4, 5]. Linear and non-linear fractional ordinary differential equations have also been investigated using the DTM under different conditions [6, 7]. Using this method to solve non-linear and multi-term time fractional partial differential equations [8, 9].

In the last few years, spectral methods have become popular for solving differential equations because they're accurate and only need to deal with a few unknowns. For example, using Telephone polynomials for solving high-order linear, non-linear ordinary differential equations and systems, with homogeneous and nonhomogeneous initial conditions [10]. Treatment of the Tricomi-type time fractional equation using Jacobi shifted polynomials [11]. using Fibonacci polynomials for solving second-order linear differential equations and nonlinear pantograph differential equations [12–14]. a system of ordinary and fractional order differential equations can be solved by Lucas polynomials [15, 16]. Discussion of the time-fractional FitzHugh-Nagumo differential equation using Lucas polynomials shifted [17]. In addition, Fibonacci polynomials solved different kinds of fractional, multiterm fractional initial value problems, and Fractional Integro-Differential Equations[18–20]. While, generalized Fibonacci polynomials solved fractional Bagley-Torvik equation and time-fractional Kuramoto–Sivashinsky equation [21, 22]. Using special cases of generalized Lucas and Fibonacci such as: Fermat polynomials for solving the fractional Burgers' equation [23]. Additionally, second kind Chebyshev collocation method for solving linear and nonlinear ordinary differential equations then comparing with the differential transform method [24]. Chebyshev polynomials with different kinds use for solving various forms for differential equations for example: Third kind discussed the solution of high-order Emden–Fowler Equations [25]. solving the time fractional diffusion wave equation using shifted fourth-kind [26]. Using the seventh kind Chebyshev polynomials for discussing the fractional delay differential equation [27]. As, eighth-kind solved the nonlinear time-fractional generalized Kawahara equation [28]. Solving time-fractional heat equation with nonlocal conditions using rectified Chebyshev [29]. In summary, numerical methods provide reliable and computationally efficient strategies for addressing diverse types of differential equations, particularly in scenarios where obtaining exact analytical solutions is infeasible or analytically intractable.

In this article, we solve linear and nonlinear second-order differential equations using DTM and the generalized Fibonacci collocation method (GFCM). The DTM is simple to apply and works well because it skips complicated algebra. It provides accurate results with fewer calculations, especially for linear differential equations, by creating short series that maintain accuracy. It can handle both initial and boundary-value problems, often reaching quick answers with just a few attempts. It may face convergence restrictions for strongly nonlinear problems or large domains. On the other hand, The GFCM with the generalized Fibonacci polynomials (GFPs), as basis functions to give accurate results, especially for problems with smooth solutions. It can solve different kinds of differential equation, and the error decreases quickly when the collocation points are added. In addition, it requires careful selection of collocation points and may involve a higher computational cost when very high accuracy is required. In this study, we use the DTM to find exact solutions for specific second-order differential equations, including some challenging ones, such as the singular and nonlinear Bratu problem. These methods can be used for different branches of science and engineering. Differential equations are very useful. They appear all the time when we're trying to understand how things change, such as velocity movement, heat flow, or circuits. They can even handle Some tricky nonlinear things.

This paper discussed general nonlinear second-order equations of the form (1), which involve variable coefficients and mixed nonlinear terms.

$$\upsilon_1(\omega)\zeta''(\omega) + \upsilon_2(\omega)\zeta'(\omega) + \upsilon_3(\omega)\zeta'^2(\omega) + \upsilon_4(\omega)\zeta(\omega) = \xi(\omega), \tag{1}$$

or special non-linear models of the form (2),

$$\zeta''(\omega) + f(\zeta) = 0, \tag{2}$$

with the conditions:

$$\zeta(0) = m_1, \quad \zeta'(0) = m_2, \tag{3}$$

or

$$\zeta(0) = m_1, \quad \zeta(1) = m_3. \tag{4}$$

We wrote out eq. (2) to show that our methods aren't just for the most general situation. They also work for some important specific situations. This split makes Section 4 easier to follow. Some problems just naturally fit eq. (1), but others are better suited for eq. (2). Where $\zeta(\omega)$ is the unknown function and m_1,m_2, m_3 are arbitrary real constants. With $\upsilon_j(\omega)(j = 1, ...4)$, $f(\zeta)$, $\xi(\omega)$ are known and continuous on the interval $[0, 1]$. With $\upsilon_1(\omega) \neq 0$

The paper is organized as follows: Section 2 presents the important relations for the DTM, the FCM, and the algorithm of the methods that are useful in the next sections. The convergence is discussed in Section 3. We give some numerical examples and compare them with others in Section 4, the results show the efficiency of the methods. In Section 5, we introduce some conclusions.

2. Description of the methods

This section describes the properties and important relations between the DTM and the GFCM, which are used in the following sections.

2.1. The DTM

This method is an expansion of a function into the Taylor series. The differential transform of a function $\zeta(\omega)$ is defined as follows

$$\Psi(i) = \frac{1}{i!}[\frac{d^i\zeta(\omega)}{d\omega^i}]_{\omega=0}. \tag{5}$$

Then the inverse differential transform is:

$$\zeta(\omega) = \sum_{i=0}^{\infty} \Psi(i)\ \omega^i. \tag{6}$$

The important relations for DTM are:
If $\zeta(\omega) = \sin(b_1\omega + b_2)$, then

$$\Psi(i) \ = \frac{b_1^i}{i\ !}\ \sin\left(\frac{i\ \pi}{2} + b_2\right). \tag{7}$$

If $\zeta(\omega) = \omega^m$, then

$$\Psi(i) \ = \delta(i-m) \ = \begin{cases} 1 & i = m \\ 0 & i \neq m \end{cases} \tag{8}$$

If $\varkappa(\zeta) = \zeta^{(n)}(\omega)$, then

$$\chi(i) = (i+1)\ (i+2)....(i+n)\Psi(i+n). \tag{9}$$

If $\varkappa(\zeta) = e^{c\zeta(\omega)}$, then

$$\chi(i) \ = \begin{cases} e^{c\Psi(0)}, & i = 0 \\ c\sum_{k=0}^{i-1} \frac{k+1}{i}\Psi(k+1)\ \chi(i-k-1)\ , & i \geq 1 \end{cases} \tag{10}$$

If $\varkappa(\omega) = \zeta_1(\omega)\ \zeta_2(\omega)$, then

$$\chi(i) = \sum_{k=0}^{i} \Psi_1(k)\ \Psi_2(i-k). \tag{11}$$

2.2. The GFCM

In this part, we display the important relations between the GFPs, $\phi_i^{\lambda_1,\lambda_2}(\omega)$ of degree i, and the algorithm of these polynomials. The basis used in this paper is the generalized Fibonacci polynomials. These polynomials involve important particular polynomials, such as Fibonacci, Pell, Chebyshev polynomials of the second kind, Fermat and second kind Dickson depend on λ_1, and λ_2. We used the first four polynomials only. The recurrence relations of generalized Fibonacci polynomials have the following form [30]

$$\phi_i^{\lambda_1,\lambda_2}(\omega) = \lambda_1\omega\phi_{i-1}^{\lambda_1,\lambda_2}(\omega) + \lambda_2\phi_{i-2}^{\lambda_1,\lambda_2}(\omega), \quad i \geq 2. \tag{12}$$

with initial values:

$$\phi_0^{\lambda_1,\lambda_2}(\omega) = 1 \quad , \phi_1^{\lambda_1,\lambda_2}(\omega) = \lambda_1\omega.$$

It has the Binet's form [30]

$$\phi_i^{\lambda_1,\lambda_2}(\omega) = \frac{(\lambda_1\omega + \sqrt{\lambda_1^2\omega^2 + 4\lambda_2})^i - (\lambda_1\omega + \sqrt{\lambda_1^2\omega^2 + 4\lambda_2})^i}{2^i\ \sqrt{\lambda_1^2\omega^2 + 4\lambda_2}}, \tag{13}$$

and the analytic form is [30]

$$\phi_i^{\lambda_1,\lambda_2}(\omega) = \sum_{k=0}^{\lfloor \frac{i}{2} \rfloor} \binom{i-k}{k} \lambda_2^k\ (\lambda_1\omega)^{i-2k}. \tag{14}$$

We expand the function in terms of the GFPs:

$$\zeta(\omega) = \sum_{i=0}^{\infty} a_i \phi_i^{\lambda_1,\lambda_2}(\omega). \tag{15}$$

The approximate solution has the following form:

$$\zeta(\omega) \approx \zeta_K(\omega)\ = \sum_{i=0}^{L} a_i\ \phi_i^{\lambda_1,\lambda_2}(\omega) = A^T\ \Phi(\omega), \tag{16}$$

where

$$\Phi(\omega) = \left[\phi_0^{\lambda_1,\lambda_2}(\omega), \phi_1^{\lambda_1,\lambda_2}(\omega), ..., \phi_L^{\lambda_1,\lambda_2}(\omega)\right]^T, \tag{17}$$

and the coefficients

$$A^T = [a_0, a_1, ..., a_L], \tag{18}$$

must be determined.

The first and the second derivatives of $\Phi(\omega)$ are in the following forms [30]

$$\frac{d\Phi(\omega)}{d\omega} = H^{(1)}\Phi(\omega) = \left(h_{\alpha\beta}^{(1)}\right)\Phi(\omega),$$

and

$$\frac{d^2\Phi(\omega)}{d\omega^2} = H^{(2)}\Phi(\omega) = \left(h_{\alpha\beta}^{(1)}\right)^2\Phi(\omega).$$

Where $h_{\alpha\beta}^{(1)}$ is $(L+1) \times (L+1)$ operational matrix. It has the following form

$$h_{\alpha\beta}^{(1)} = \begin{cases} (-1)^{\frac{\alpha-\beta-1}{2}}(\beta+1)\lambda_1\ \lambda_2^{\frac{\alpha-\beta-1}{2}}, & \alpha > \beta, (\alpha+\beta) \text{ odd}, \\ 0, & \textit{otherwise}. \end{cases}$$

If $\alpha=\beta=4$, then $H^{(1)}$ can be written in the form

$$H^{(1)}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0\\ \lambda_1 & 0 & 0 & 0 & 0\\ 0 & 2\lambda_1 & 0 & 0 & 0\\ -\lambda_1\lambda_2 & 0 & 3\lambda_1 & 0 & 0\\ 0 & -2\lambda_1\lambda_2 & 0 & 4\lambda_1 & 0 \end{bmatrix}.$$

From eq. (12) and its initial values, $\Phi(\omega)$ can be written in the form

$$\Phi(\omega)=\begin{bmatrix} 1\\ \lambda_1\omega\\ \lambda_1^2\omega^2+\lambda_2\\ \lambda_1^3\omega^3+2\lambda_1\lambda_2\omega\\ \lambda_1^4\omega^4+3\lambda_1^2\lambda_2\omega^2+\lambda_2^2 \end{bmatrix}.$$

From eq. (16) into eq. (1), we have

$$\upsilon_1(\omega)\sum_{i=0}^{L}a_i\ \left(h^{(1)}_{\alpha\beta}\right)^2\phi_i^{\lambda_1,\lambda_2}(\omega)+\upsilon_2(\omega)\ \sum_{i=0}^{L}a_i\ h^{(1)}_{mn}\phi_i^{\lambda_1,\lambda_2}(\omega)+\upsilon_3(\omega)\ \sum_{i=0}^{L}a_i\ (h^{(1)}_{mn}\phi_i^{\lambda_1,\lambda_2}(\omega))^2+$$

$$\upsilon_4(\omega)\sum_{i=0}^{L}a_i\ \phi_i^{\lambda_1,\lambda_2}(\omega)=\xi(\omega). \tag{19}$$

Let

$$\theta_i\left(\omega\right)=\upsilon_1(\omega)\left(h^{(1)}_{\alpha\beta}\right)^2\phi_i^{\lambda_1,\lambda_2}(\omega)+\upsilon_2(\omega)\ h^{(1)}_{mn}\phi_i^{\lambda_1,\lambda_2}(\omega)+\upsilon_3(\omega)\ (h^{(1)}_{mn}\phi_i^{\lambda_1,\lambda_2}(\omega))^2+\upsilon_4(\omega)\phi_i^{\lambda_1,\lambda_2}(\omega),$$

then eq. (19) has the form

$$\sum_{i=0}^{L}a_i\ \theta_i\left(\omega\right)=\xi(\omega). \tag{20}$$

The collocation points are taken as the roots of the generalized Fibonacci polynomial of degree L+1, in order to guarantee accuracy and numerical stability. So we have a system of equations

$$\sum_{i=0}^{L}a_i\ \theta_i\left(\omega_\ell\right)=\xi(\omega_\ell),\qquad i,\ell=0,1,...,L.$$

So, the matrix form is:

$$\Theta^T\ A=\Xi,$$

where

$$\Theta=\left(\theta_i\left(\omega_\ell\right)\right),\qquad i,\ell=0,1,....L,$$

and

$$\Xi=[\xi(\omega_0),\xi(\omega_1),...,\xi(\omega_L)]^T.$$

Finally, the constants can be evaluated by the following equation:

$$A = (\Theta^T)^{-1}\Xi, \tag{21}$$

using the following conditions:

$$A^T\Phi(0) = m_1, \qquad A^T\Phi'(0) = m_2,$$

or

$$A^T\Phi(0) = m_1, \qquad A^T\Phi(1) = m_3.$$

Now we summarize the steps of our scheme in Algorithm 1 below

Algorithm 1 Coding algorithm for the proposed scheme

Input $\upsilon_1(\omega)$, $\upsilon_2(\omega)$, $\upsilon_3(\omega)$, $\upsilon_4(\omega)$, $\xi(\omega)$, $f(\zeta)$, and $\zeta(\omega)$.

Step 1. Define generalized Fibonacci polynomials by (13).

Step 2. Compute the basis function of generalized Fibonacci polynomials by (16).

Step 3. Define the basis function vector $\Phi(\omega)$ by (17).

Step 4. Differentiate eq. (16).

Step 5. Substituting eq. (16) and its differentiation into eq. (1).

Step 6. Collocating Eq. (19) in $(L+1)\times(L+1)$ roots of the generalized Fibonacci polynomials.

Step 7. Evaluate the residue of Eq. (1).

Step 8. Compute the matrix $\xi(\omega)$ using (19).

Step 9. Define the $(L+1)\times(L+1)$ unknown vectors A^T.

Step 10. Use *FindRoot* command to solve the system $\Theta^T\ C = \Xi$.

Step 11. Evaluate timeUsed for the program.

Step 12. Plot the absolute error.

Output The absolute error G_i and approximate solution $\Phi(\omega) = A^T\Theta$.

3. The convergence and error analysis

In this section, we evaluate the truncation error with a generalized Fibonacci expansion for the equation. (1).

Theorem 1. *Let*

$$G_L(\omega) = \left|\upsilon_1(\omega)\zeta''_L(\omega) + \upsilon_2(\omega)\zeta'_L(\omega) + \upsilon_3(\omega)\zeta'^2_L(\omega) + \upsilon_4(\omega)\zeta_L(\omega) - \xi(\omega)\ \right|,$$

$$\breve{G}_L = \max_{0\le\omega\le\varrho} G_L(\omega), \qquad \varrho \ge 0,$$

and if $|\upsilon_j(\omega)| \le n_j$. *Where* n_j *are positive constants for* $0 \le j \le 4$. *Then we have the following truncation error:*

$$\breve{G}_L \le 4N\rho^2\{\frac{W^{2+L}}{(1+L)\left(\Gamma\left(1+L\right)\right)^2}\left\{e^W W\left(1+W\right) + (1+L)(1+L+W)\right\} \times$$
$$\left\{-W^{2+L}(1+L+W) + \left(1+2e^W W^2\left(1+W\right)\right)\Gamma\left(1+L\right)\right\} + \frac{W\ e^{WP}(WP)^L}{\Gamma\left(L\right)}\}.$$

Proof. From eq. (1), we have

$$\xi(\omega) = \upsilon_1(\omega)\zeta''(\omega) + \upsilon_2(\omega)\zeta'(\omega) + \upsilon_3(\omega)\zeta'^2(\omega) + \upsilon_4(\omega)\zeta(\omega).$$

From the hypotheses of the theorem, we obtain

$$G_L(\omega) \le n_1 \left|\zeta''(\omega) - \zeta''_L(\omega)\right| + n_2 \left|\zeta^{'}(\omega) - \zeta^{'}_L(\omega)\right| + n_3 \left|\zeta'^2(\omega) - \zeta'^2_L(\omega)\right| + n_4 \left|\zeta(\omega) - \zeta_L(\omega)\right|.$$

From Theorems (5) and (6) [30]

$$|a_i| \le \rho \frac{W^{i+1}}{i!}, \quad \left|\zeta''(\omega) - \zeta''_L(\omega)\right| \le \sum_{i=L+1}^{\infty} |a_i| \left|(\phi_i^{\lambda_1,\lambda_2}(\omega))''\right|.$$

Where $\left|\zeta^{(i)}(0)\right| = Q^i$, $W = \frac{Q}{|\lambda_1|}$, $\rho = \frac{6|\lambda_2|}{Q|\lambda_1|\ W^2 P^2} Li_6(\frac{\lambda_1^2 W^2 P^2}{3|\lambda_2|}) \cosh(\frac{2Q\sqrt{\lambda_2}}{|\lambda_1|})$.

From Lemma (1) [31], we obtain

$$\left|(\phi_i^{\lambda_1,\lambda_2}(\omega))''\right| \le i^2.$$

Then we have

$$\left|\zeta''(\omega) - \zeta''_L(\omega)\right| = \left|\zeta^{'}(\omega) - \zeta^{'}_L(\omega)\right| \le \rho \sum_{i=L+1}^{\infty} \frac{W^{i+1}}{i!} i^2,$$

and

$$\left|\zeta'^2(\omega) - \zeta'^2_L(\omega)\right| \le \left|\zeta'(\omega) - \zeta'_L(\omega)\right|^2 + 2\left|\zeta'_L(\omega)\right| \ \ \left|\zeta^{'}(\omega) - \zeta^{'}_L(\omega)\right|.$$

From the pervious steps, we obtain

$$G_L(\omega) \le \rho \left\{ (n_1 + n_2)\ + n_3\rho \left(\sum_{i=L+1}^{\infty} \frac{W^{i+1}}{i!} i^2 + 2\sum_{i=0}^{L} \frac{W^{i+1}}{i!} i^2 \right) \right\} \left(\sum_{i=L+1}^{\infty} \frac{W^{i+1}}{i!} i^2 \right) + \rho \frac{n_4 W\ e^{WP} (WP)^L}{(L-1)!}.$$

Where $P = \sqrt{\lambda_1^2\ \varrho^2 + 2\left|\lambda_2\right|}$, If $N = \max(n_1, n_2, n_3, n_4)$, by simplifying, we obtain

$$G_L(\omega) \le 4N\rho^2 \left\{ 1\ + \frac{W^2}{\Gamma(1+L)} \left(-W^L (1 + W + L) + e^W (1 + W) (\Gamma(1+L) + \Gamma(1+L, W)) \right) \right\} \times$$

$$\frac{W^2}{\Gamma(1+L)} \left(-W^L (1 + W + L) + e^W (1 + W) (\Gamma(1+L) - \Gamma(1+L, W)) \right) + \rho \frac{NW\ e^{WP} (WP)^L}{(L-1)!}.$$

Where $\Gamma(.)$ and $\Gamma(.,.)$ are gamma and incomplete gamma functions respectively. So

$$\Gamma(1+L) - \Gamma(1+L, W) \le \int_0^W \omega d\omega = \frac{W^{1+L}}{1+L},$$

and

$$\Gamma\left(1+L\right)+\Gamma\left(1+L,W\right)\leq 2\Gamma\left(1+L\right).$$

Then, we have

$$G_L(\omega)\leq \frac{4N\rho^2W^{2+L}}{(1+L)\left(\Gamma\left(1+L\right)\right)^2}\left\{e^W W\left(1+W\right)+(1+L)(1+L+W)\right\}\times$$

$$\left\{-W^{2+L}(1+L+W)+\left(1+2e^W W^2\left(1+W\right)\right)\Gamma\left(1+L\right)\right\}+\rho\frac{NW\ e^{WP}(WP)^L}{\Gamma\left(L\right)}.$$

So, the result is verified.

For the general second-order problem (1) with initial conditions (3) or boundary conditions (4), the existence and uniqueness can be established if $\upsilon_1(\omega)\neq 0$ and is continuous on the interval of interest. Where $\zeta''(\omega)$ depends on a continuous function, which satisfies a Lipschitz condition in $\zeta'(\omega)$ and $\zeta(\omega)$

4. Numerical examples

This section presents a series of problem solving applications using DTM and GFCM, aimed at demonstrating their practical implementation. By systematically comparing the results with those from established numerical methods, we provide evidence of the enhanced performance of DTM and GFCM in terms of solution accuracy and computational efficiency.

Example 1. *Consider the following linear second-order boundary value problem:[32]*

If $\upsilon_1(\omega)=1,\quad \upsilon_2(\omega)=1-\omega,\quad \upsilon_3(\omega)=0,\quad \upsilon_4(\omega)=2,\quad \xi(\omega)=(1+2\omega-\omega^2)\sin(\omega),$ $m_1=1,\quad m_3=0.$

So, eq. (1) can be written in the form:

$$\zeta^{''}(\omega)+(1-\omega)\zeta^{'}(\omega)+2\zeta(\omega)=(1+2\omega-\omega^2)\sin(\omega),\qquad 0<\ \omega\leq 1 \tag{22}$$

subject to the boundary conditions (4)

$$\zeta(0)=1,\quad \zeta(1)=0. \tag{23}$$

From the definition of DTM, eqs. (7), (8), (9), and (11). So, eq.(22) transforms to:

$$\Psi(i+2)=\frac{1}{(i+1)(i+2)}\{-(i+1)\Psi(i+1)+\sum_{k=0}^{i}\delta(k-1)(i-k+1)\Psi(i-k+1)-2\Psi(i)+$$

$$\frac{1}{i!}\sin(\frac{\pi i}{2})+2\sum_{k=0}^{i}\frac{\delta(k-1)}{(i-k)!}\sin(\frac{\pi(i-k)}{2})-\sum_{k=0}^{i}\frac{\delta(k-2)}{(i-k)!}\sin(\frac{\pi(i-k)}{2})\}, \tag{24}$$

with the conditions:

$$\Psi(0) = 1, \quad \Psi(1) = a. \tag{25}$$

Using eqs. (24), and (25).

when $i = 0$, $\Psi(2) = -\frac{(2+a)}{2}$,

when $i = 1$, then $\Psi(3) = \frac{1}{2}$,

when $i = 2$, then $\Psi(4) = \frac{-1}{24}$.

Then from eq. (6), we have

$$\zeta(\omega) = 1 + a\omega - \frac{(2+a)}{2}\,\omega^2 + \frac{1}{2}\omega^3 + \frac{1}{24}\,\omega^4 - ... \tag{26}$$

By solving conditions (25), so $a = -1$. Then eq.(26) has the form

$$\zeta(\omega) = (1-\omega)\cos(\omega).$$

Which is the exact solution.

Table 1 compares the absolute errors resulting from the GFCM and those obtained using the second-kind Chebyshev wavelets algorithm [32] at i=3,4,5. The data indicate that our method achieves smaller absolute errors, particularly for lower indices i and different values of λ_1 and λ_2. Table 2 provides the computation times (CPU time) for each method and the changes in error values between consecutive steps. Figure 1 visually represent the results corresponding to the same parameter values. The graph of exact and approximate solutions at i=4 is plotted in Figure 2.

Table 1: Comparison between the absolute errors for Example 1

λ_1	λ_2	i	G	i	G	i	G
1	1	3	1.89×10^{-3}	4	2.59×10^{-4}	5	9.08×10^{-6}
2	1		1.89×10^{-3}		2.59×10^{-4}		9.08×10^{-6}
2	-1		1.89×10^{-3}		2.59×10^{-4}		9.08×10^{-6}
3	-2		1.89×10^{-3}		2.59×10^{-4}		9.08×10^{-6}
Method in [32]			2.28×10^{-3}		6.32×10^{-5}		6.23×10^{-6}

Table 2: The CPU time for Example 1

CPU time	G_i
0.25	1.63×10^{-3}
0.282	2.50×10^{-4}
0.297	6.62×10^{-6}

Example 2. *Consider the following non-linear second order boundary [32]*

If $\upsilon_1(\omega) = 4$, $\upsilon_2(\omega) = 0$, $\upsilon_3(\omega) = -2$, $\upsilon_4(\omega) = 1$, $\xi(\omega) = 0$, $m_1 = -1$, $m_2 = 0$.

So, eq. (1) can be written in the form:

$$4\zeta''(\omega) - 2(\zeta'(\omega))^2 + \zeta(\omega) = 0, \quad 0 < \omega \le 1 \tag{27}$$

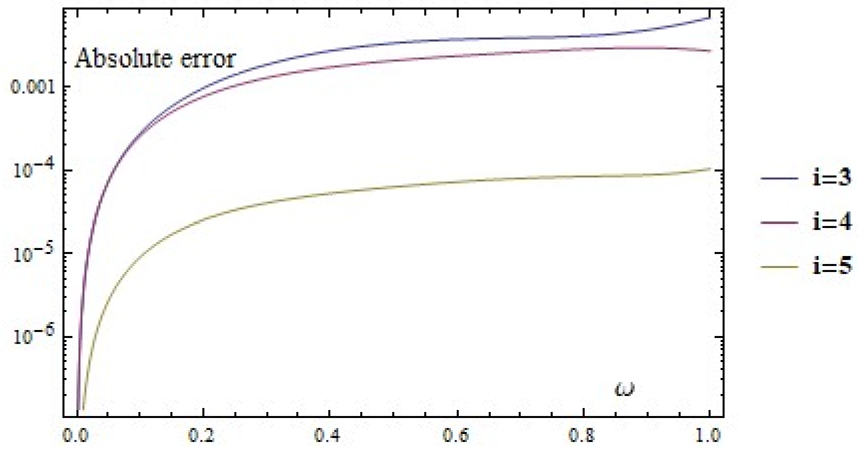

Figure 1: Graph of the error at i=3, 4, and 5 for Example 1

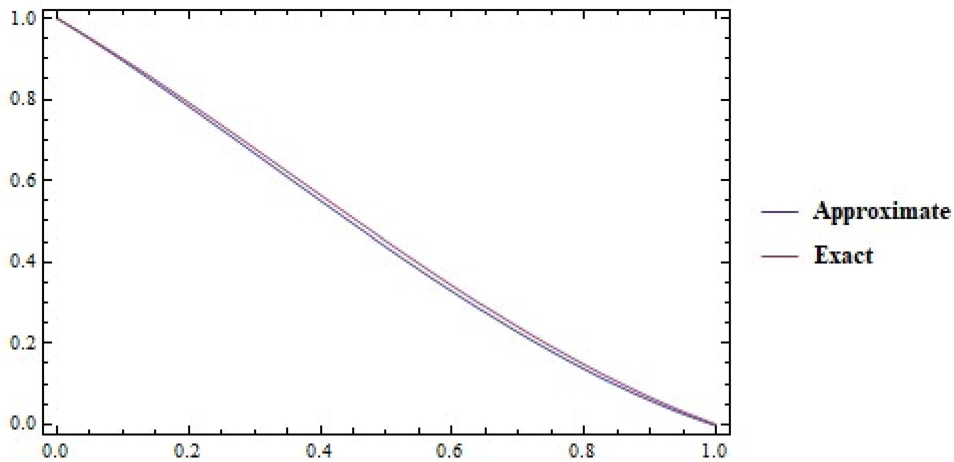

Figure 2: Approximate and exact solutions at i=4, $\lambda_1 = 3, \lambda_2 = -2$ for Example 1

with the conditions (3)

$$\zeta(0) = -1, \quad \zeta'(0) = 0. \tag{28}$$

From the definition of DTM, eqs. (9) and (11). So, eq. (27) transforms to

$$\Psi(i+2) = \frac{1}{4(i+1)(i+2)} \left\{ 2\sum_{k=0}^{i}(k+1)(i-k+1)\Psi(k+1)\Psi(i-k+1) - \Psi(i) \right\}, \tag{29}$$

with the conditions.

$$\Psi(0) = -1, \quad \Psi(1) = 0. \tag{30}$$

From eqs. (29) and (30).

When $i = 0$, then $\Psi(2) = 1$,
when $i = 1$, then $\Psi(3) = 1$,
when $i = 2$, then $\Psi(4) = 0$,
when $i = 3$, then $\Psi(5) = 0$,
when $i = 4$, then $\Psi(6) = 0$.
Then from eq. (6), we have

$$\zeta(\omega) = \frac{\omega^2}{8} - 1.$$

Which is the exact solution.

In Table 3, we compare the absolute errors generated by the GFCM for i=2,3, various values of λ_1 and λ_2. Table 4 provides the CPU times needed for the calculations. The results show that the proposed method achieves the best errors especially for smaller values of i. Figure 3 visualizes our results for i=3,4, and $\lambda_1 = \lambda_2 = 1$. The graph of exact and approximate solutions at i=5 is plotted in Figure 4.

Table 3: Comparison between the absolute errors for Example 2

λ_1	λ_2	i	G	i	G
1	1	2	0	3	2.3×10^{-17}
2	1		0		2.5×10^{-18}
2	-1		0		5.8×10^{-18}
3	-2		0		8.7×10^{-18}

Table 4: The CPU time for Example 2

CPU time	G_i
0.235	2.5×10^{-18}
0.266	1.1×10^{-18}

Example 3. *Consider the non-linear second-order initial value problem: [33]*
If $f(\zeta) = e^{-2\zeta(\omega)}$, $m_1 = 1$, $m_2 = \frac{1}{e}$.

So, eq. (2) can be written in the form:

$$\zeta''(\omega) + e^{-2\zeta(\omega)} = 0, \tag{31}$$

with the conditions (3)

$$\zeta(0) = 1, \quad \zeta'(0) = \frac{1}{e}. \tag{32}$$

From the definition of DTM, eqs. (9) and (10). So, eq. (32) transforms to

$$\Psi(i+2) = \frac{-1}{(i+1)(i+2)} F_1(i), \tag{33}$$

where $F_1(i)$ is the differential transform of $f(\zeta)$, using the following conditions

$$\Psi(0) = 1, \quad \Psi(1) = \frac{1}{e}. \tag{34}$$

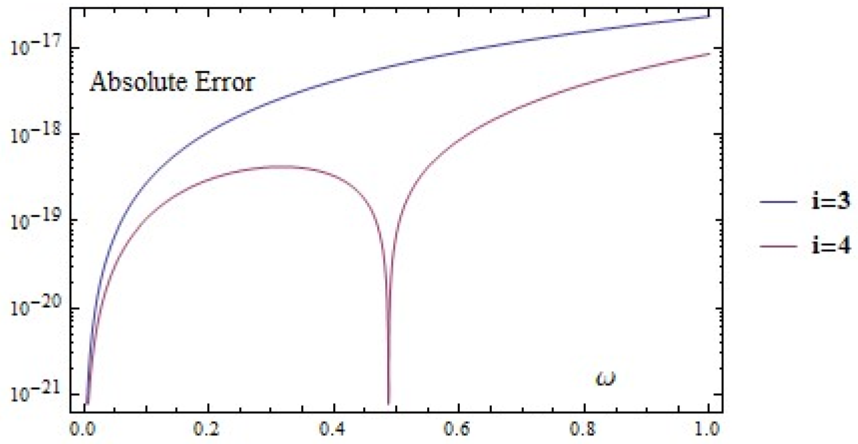

Figure 3: Graph of the error at i=3, 4, and $\lambda_1 = \lambda_2 = 1$ for Example 2

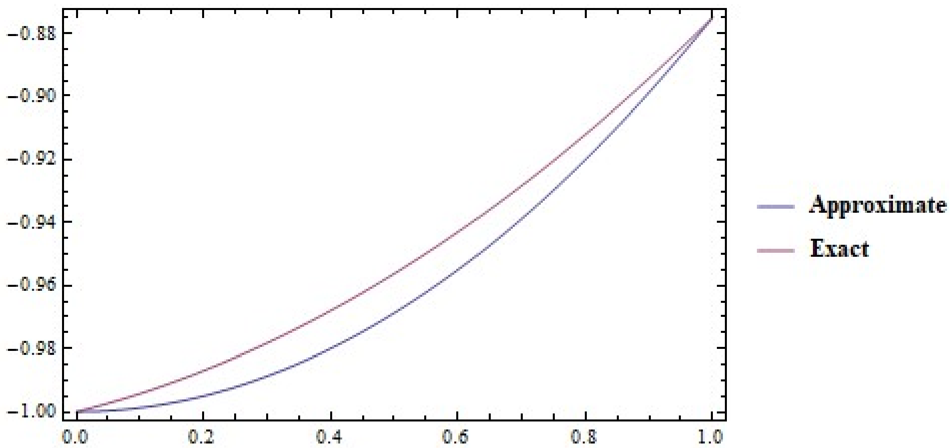

Figure 4: Approximate and exact solutions at i=5, $\lambda_1 = \lambda_2 = 1$ for Example 2

From eqs. (33), and (34).

When $i = 0$, then $\Psi(2) = -\frac{e^{-2}}{2}$,
when $i = 1$, then $\Psi(3) = \frac{e^{-3}}{3}$,
when $i = 2$, then $\Psi(4) = \frac{-e^{-4}}{4}$,
when $j = 3$, then $\Psi(5) = \frac{e^{-5}}{5}$.
So from eq. (6), we have

$$\zeta\omega) = ln(e + \omega).$$

Which is the exact solution.

From GFCM, Table 5 compares the absolute errors of the proposed method with modified Adomian decomposition method and C^2-spline method [33]. The results at i=14, different values of ω, λ_1 and λ_2. The CPU times for these methods are provided in Table 6. It is evident that the proposed method achieves the lowest absolute errors. These errors are also visually represented in Figure 5. The graph of exact and approximate solutions at i=8 is plotted in Figure 6.

Table 5: Results of absolute errors for Example 3

ω	Modified ADM [33]	C^2-spline method [33]
0.2	3.91×10^{-11}	2.04×10^{-14}
0.4	1.66×10^{-9}	4.99×10^{-14}
0.6	1.30×10^{-8}	8.50×10^{-14}
0.8	3.17×10^{-7}	1.25×10^{-13}
1	2.20×10^{-6}	1.68×10^{-13}

ω	$(\lambda_1, \lambda_2) = (1, 1)$	$(\lambda_1, \lambda_2) = (2, 1)$	$(\lambda_1, \lambda_2) = (2, -1)$	$(\lambda_1, \lambda_2) = (3, -2)$
0.2	1.15×10^{-14}	1.24×10^{-14}	1.35×10^{-14}	1.11×10^{-14}
0.4	2.40×10^{-14}	2.62×10^{-14}	2.84×10^{-14}	2.35×10^{-14}
0.6	3.62×10^{-14}	3.95×10^{-14}	4.71×10^{-14}	3.62×10^{-14}
0.8	4.95×10^{-14}	5.35×10^{-14}	5.84×10^{-14}	4.91×10^{-14}
1	6.17×10^{-14}	6.73×10^{-14}	7.39×10^{-14}	6.44×10^{-14}

Table 6: The CPU time for Example 3

CPU time	G_i
0.578	4.44×10^{-14}

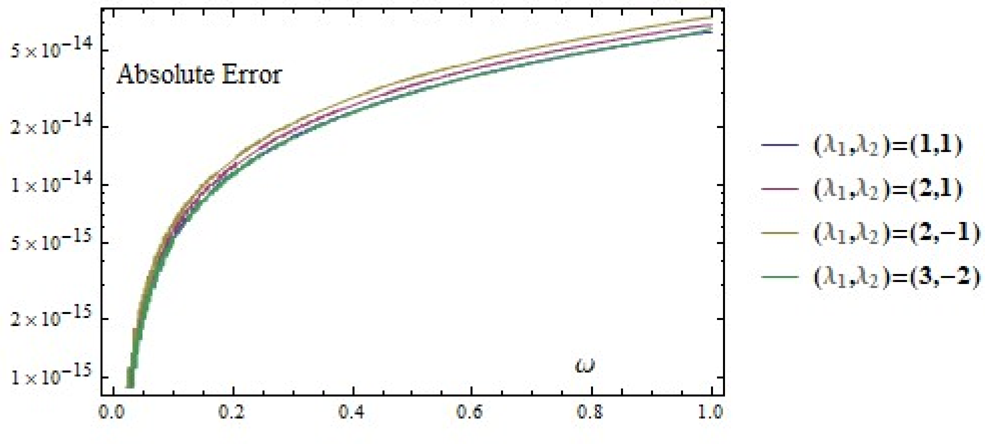

Figure 5: Graph of the error at i=14, different values of λ_1 and λ_2 for Example 3

Example 4. *Consider the nonlinear second-order Bratu problem [34]*
If $f(\zeta) = -2e^{\zeta(\omega)}, \quad m_1 = 0, \quad m_2 = 0.$

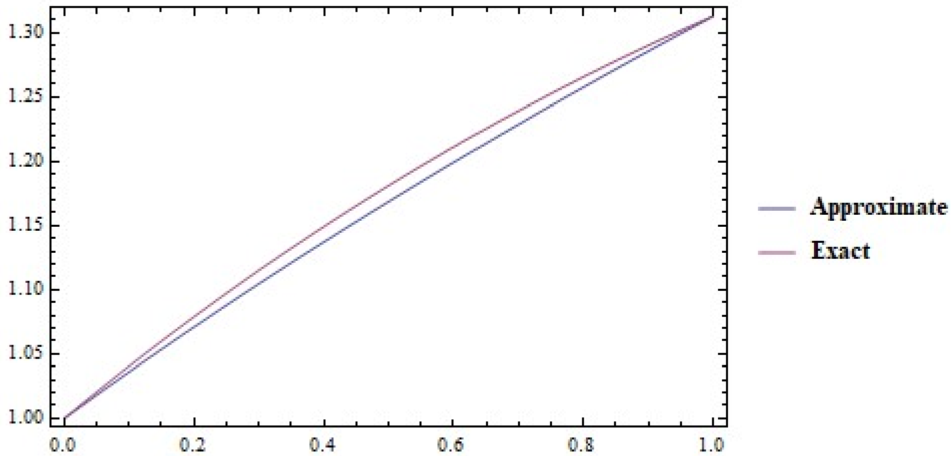

Figure 6: Approximate and exact solutions at i=8, $\lambda_1 = 2, \lambda_2 = -1$ for Example 3

So, eq. (2) can be written in the form:

$$\zeta''(\omega) - 2e^{\zeta(\omega)} = 0, \tag{35}$$

with the conditions (3)

$$\zeta(0) = 0, \quad \zeta'(0) = 0. \tag{36}$$

From the definition of DTM, eqs (9) and (10). So, eq. (35) transforms to

$$\Psi(i+2) = \frac{2}{(i+1)(i+2)} F_2(i), \tag{37}$$

where $F_2(i)$ is the differential transform of $f(\zeta)$, with the conditions

$$\Psi(0) = 1, \quad \Psi(1) = 0. \tag{38}$$

From eqs. (37), and (38).

When $i = 0$, then $\Psi(2) = 1$,
when $i = 1$, then $\Psi(3) = 0$,
when $i = 2$, then $\Psi(4) = \frac{1}{6}$,
when $i = 3$, then $\Psi(5) = 0$,
when $i = 4$, then $\Psi(6) = \frac{2}{45}$.

From eq. (6), we have

$$\zeta(\omega) = -2\log(\cos(\omega)).$$

Which is the exact solution.

Table 7 shows a comparison of the absolute errors obtained using the proposed method and the Legendre wavelet method [34] at i=10. The associated CPU times are detailed in Table 8. The data indicate that the proposed method produces the lowest absolute errors at different values of ω. These results are also depicted graphically in Figure 7. The graph of exact and approximate solutions at i=3 is plotted in Figure 8.

Table 7: Comparison between the absolute errors for Example 4

ω	$(\lambda_1, \lambda_2) = (1, 1)$	$(\lambda_1, \lambda_2) = (2, 1)$	$(\lambda_1, \lambda_2) = (2, -1)$	$(\lambda_1, \lambda_2) = (3, -2)$
0.2	1.62×10^{-5}	1.62×10^{-5}	1.62×10^{-5}	1.62×10^{-5}
0.4	3.57×10^{-5}	3.57×10^{-5}	3.57×10^{-5}	3.57×10^{-5}
0.6	5.86×10^{-5}	5.86×10^{-5}	5.86×10^{-5}	5.86×10^{-5}
0.8	8.88×10^{-5}	8.88×10^{-5}	8.88×10^{-5}	8.88×10^{-5}
1	1.29×10^{-4}	1.29×10^{-4}	1.29×10^{-4}	1.29×10^{-4}
LWM [34]		1.38×10^{-4}		

Table 8: The CPU time for Example 4

CPU time	G_i
0.484	7.75×10^{-5}

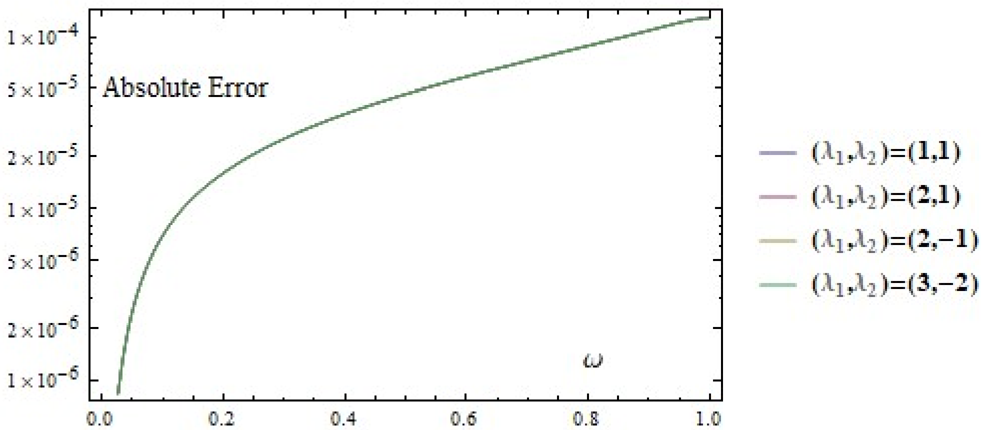

Figure 7: Graph of the error at at i=10, different values of λ_1 and λ_2 for Example 4

5. Conclusion

This research looked at solving differential equations using two main techniques. The useful of our work is that we're making better use of the generalized Fibonacci polynomials with a collocation method. We've also got some fresh ways to look at how well it works and where it might be bad. Plus, we're putting the different technique, DTM, to tackle all sorts of tricky math problems – straight-up ones, weird ones, and even those with singularities. It looks like our approach is more accurate and faster than what people are already doing. We compared our methods with others, which showed that the accuracy, and found them to be effective and easy to use in different types of differential equations. The truncation error of the main equation is discussed. In the future studies, the proposed

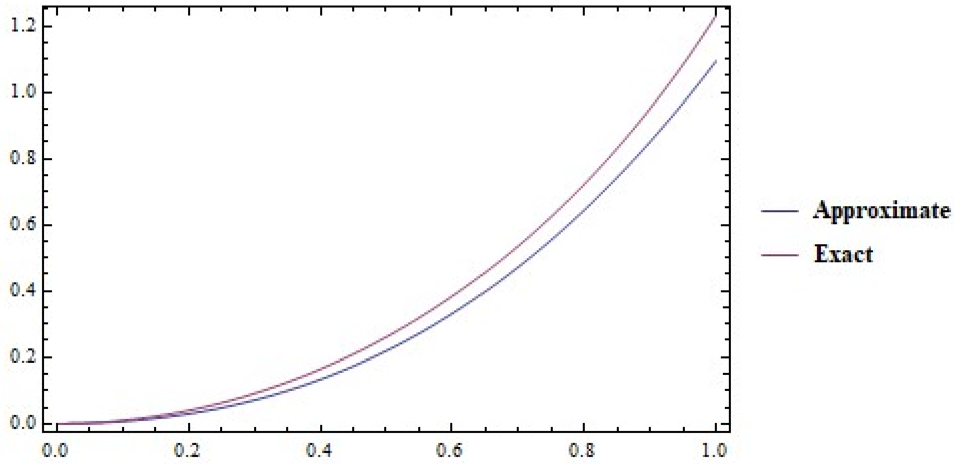

Figure 8: Approximate and exact solutions at i=3, $\lambda_1 = 2, \lambda_2 = 1$ for Example 4

methods may solve higher-order, coupled, and fractional differential equations. It would also be interesting to explore their computational efficiency on large-scale problems and to apply them to practical models in physics and engineering.

Acknowledgements

The authors are very grateful to the anonymous referees for careful reviewing and crucial comments, which enabled us to improve the manuscript.

Competing interests

The authors declares that they have no competing interest.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' contributions

The authors confirm that this is their own work.

References

[1] R. A. Ibrahim and M. Saad. Application of differential transform method with a domain polynomial for solving RLC circuits problems and higher order differential equations. *Engineering Research Journal*, 51(4):89–95, 2022.

[2] S. R. M. Noori and N. Taghizaheh. Modified differential transform method for solving linear and nonlinear pantograph type of differential and Volterra integro-differential equations with proportional delays. *Advances in Difference Equations*, 2020:649, 2020.

[3] M. El-Shahed. Application of differential transform method to non-linear oscillatory systems. *Communications in Nonlinear Science and Numerical Simulation*, 13:1714–1720, 2008.

[4] H. Nayar and P. A. Anakira Phiri. Modification of differential transform method with search direction along the spatial axis. *International Journal of Differential Equations*, 2024:22, 2024. Article ID 6777175.

[5] C. Huang, J. Li, and F. Lin. A new algorithm based on differential transform method for solving partial differential equation system with initial and boundary conditions. *Advances in Pure Mathematics*, 10:337–349, 2020.

[6] E. Demirci and N. Ozalp. A method for solving differential equations of fractional order. *Journal of Computational and Applied Mathematics*, 236:2754–2762, 2012.

[7] H. K. Jassim and M. A. Hussein. A new approach for solving nonlinear fractional ordinary differential equations. *Mathematics*, 11:1565, 2023.

[8] A. Khalouta. Closed form solutions to some nonlinear fractional partial differential equations arising in mathematical sciences. *Palestine Journal of Mathematics*, 11:113–126, 2022.

[9] S. Abuasad, I. Hashim, and S. A. Abdul Karim. Modified fractional differential transform method for the solution of multi-term time-fractional diffusion equations. *Advances in Mathematical Physics*, 2019:14, 2019. Article ID 5703916.

[10] Ahmed Gamal Atta, Jomana Farag Soliman, Elaf Wael Elsaeed, Mostafa Wael Elsaeed, and Youssri Hassan Youssri. Spectral collocation algorithm for the fractional bratu equation via hexic shifted chebyshev polynomials. *Computational Methods for Differential Equations*, 13(4):1102–1116, 2025.

[11] Y. H. Youssri, R. M. Hafez, and A. G. Atta. An innovative pseudo-spectral Galerkin algorithm for the time-fractional Tricomi-type equation. *Physica Scripta*, 99(10):105238, 2013.

[12] A. Kurt, S. Yalcinbas, and M. Sezer. Fibonacci collocation method for solving linear differential difference equations. *Mathematical and Computational Applications*, 18(3):448–458, 2013.

[13] A. B. M. I. Koc, M. Cakmak, A. Kurnaz, and K. Uslu. A new fibonacci type collocation procedure for boundary value problems. *Advances in Difference Equations*, 2013:262, 2013.

[14] M. Cakmak. Fibonacci collocation method for solving a class of nonlinear pantograph differential equations. *Mathematical Methods in the Applied Sciences*, 45:11962–11976, 2022.

[15] M. Cetin, M. Sezer, and C. Guler. Lucas polynomials approach for system of high-order linear differential equations and residual error estimation. *Mathematical Problems in Engineering*, 2015:14, 2015. Article ID 625984.

[16] W. M. Abd-Elhameed and Y. H. Youssri. Spectral solutions for fractional differential equations via a novel lucas operational matrix of fractional derivatives. *Romanian*

Journal of Physics, 61(5–6):795–818, 2016.

[17] YH Youssri and SM Sayed. Efficient spectral collocation algorithm for pantograph integro-differential equations involving delay and weakly singular kernels. *Journal of Umm Al-Qura University for Applied Sciences*, pages 1–15, 2025.

[18] W. M. Abd-Elhameed and Y. H. Youssri. A novel operational matrix of Caputo fractional derivatives of fibonacci polynomials: spectral solutions of fractional differential equations. *Entropy*, 18, 2016.

[19] Y. H. Youssri and W. M. Abd-Elhameed. Spectral solutions for multi-term fractional initial value problems using a new fibonacci operational matrix of fractional integration. *Progress in Fractional Differentiation and Applications, An International Journal*, 2(2):141–151, 2016.

[20] Y. H. Youssri and A. G. Atta. Fejér-quadrature collocation algorithm for solving fractional integro-differential equations via fibonacci polynomials. *Contemporary Mathematics*, 5(1):296–308, 2024.

[21] A. G. Atta, G. M. Moatimid, and Y. H. Youssri. Generalized fibonacci operational tau algorithm for fractional Bagley-Torvik equation. *Progress in Fractional Differentiation and Applications, An International Journal*, 6(3):215–224, 2020.

[22] Youssri Hassan Youssri and Ahmed Gamal Atta. Optimal third-kind chebyshev collocation algorithm for solving beam-type micro-and nanoscale bvps. *Journal of Mathematical Modeling*, 13(4):883–898, 2025.

[23] Shorouk Gamal Kamel, Ahmed Gamal Atta, and Youssri Hassan Youssri. A spectral delannoy polynomial-based computational framework for fractional differential problems in complex systems. *International Journal of Modern Physics C*, 2025.

[24] A. S. Mohamed. Exact differential transform method and semi-analytic Chebyshev collocation method for ordinary differential equations. *Boundary Value Problems*, 2025:36, 2025.

[25] W. M. Abd-Elhameed, M. S. Al-Harbi, A. K. Amin, and H. M. Ahmed. Spectral treatment of high-order Emden–Fowler equations based on modified Chebyshev polynomials. *Axioms*, 12(2):99, 2023.

[26] A. G. Atta. Approximate Petrov–Galerkin solution for the time fractional diffusion wave equation. *Mathematical Methods in the Applied Sciences*, 48(12):11670–11685, 2025.

[27] W. M. Abd-Elhameed, M. S. Al-Harbi, A. K. Amin, and H. M. Ahmed. Tau algorithm for fractional delay differential equations utilizing seventh-kind Chebyshev polynomials. *Journal of Mathematical Modeling*, 12(2):277–299, 2024.

[28] W. M. Abd-Elhameed, Y. H. Youssri, A. K. Amin, and A. G. Atta. Eighth-kind Chebyshev polynomials collocation algorithm for the nonlinear time-fractional generalized Kawahara equation. *Fractal and Fractional*, 7:652, 2023.

[29] Y. H. Youssri, M. I. Ismail, and A. G. Atta. Chebyshev Petrov-Galerkin procedure for the time-fractional heat equation with nonlocal conditions. *Physica Scripta*, 99(1):015251, 2023.

[30] W. M. Abd-Elhameed and Y. H. Youssri. Spectral tau algorithm for certain coupled system of fractional differential equations via generalized fibonacci polynomial

sequence. *Iranian Journal of Science and Technology, Transaction A: Science*, 2017.

[31] A. S. Mohamed. Fibonacci collocation pseudo-spectral method of variable-order space-fractional diffusion equations with error analysis. *Mathematics*, 7(8):14323–14337, 2022.

[32] W. M. Abd-Elhameed, E. H. Doha, and Y. H. Youssri. New spectral second kind Chebyshev wavelets algorithm for solving linear and nonlinear second-order differential equations involving singular and Bratu type equations. *Abstract and Applied Analysis*, 2013:9, 2013. Article ID 715756.

[33] K. Al-Khaled and M. N. Anwar. Numerical comparison of methods for solving second-order ordinary initial value problems. *Applied Mathematical Modeling*, 31:292–301, 2007.

[34] S. G. Venkatesh, S. K. Ayyaswamy, and S. Balachandar. The Legendre wavelet method for solving initial value problems of Bratu-type. *Computers and Mathematics with Applications*, 63:1287–1295, 2012.