

Adaptive Sparse Fibonacci Collocation Method for Quadratically Nonlinear Differential Equations

Salih Yalçınbaş 

Department of Mathematics, Manisa Celal Bayar University, Manisa, Türkiye
Email: syalcin35@gmail.com

How to cite this paper: Yalçınbaş, S.
(2026) Adaptive Sparse Fibonacci Collocation Method for Quadratically Nonlinear Differential Equations. *Journal of Applied Mathematics and Physics*, 14, 4080-4101.
<https://doi.org/10.4236/jamp.2026.1410198>

Received: September 8, 2026

Accepted: October 23, 2026

Published: October 26, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This study introduces an adaptive sparse Fibonacci collocation method for the numerical solution of ordinary differential equations with quadratic nonlinearities. Unlike conventional Fibonacci polynomial approaches based on a predefined truncation order and a full set of basis functions, the proposed framework incorporates the selection of significant Fibonacci modes directly into the solution process. The approximate solution is represented by a reduced Fibonacci expansion whose active basis is iteratively determined according to coefficient significance and residual information. Derivative terms are expressed through an operational matrix representation, while quadratic nonlinearities involving the solution and its derivatives are transformed into a nonlinear algebraic system at the collocation points. A residual-controlled enrichment and pruning mechanism is employed to balance approximation accuracy and basis complexity, allowing unnecessary Fibonacci modes to be eliminated without substantially degrading the solution. The accuracy and computational behavior of the method are assessed through benchmark nonlinear differential equations with available exact solutions using maximum absolute errors, residual norms, active basis size, and convergence characteristics. The proposed formulation provides a compact and interpretable extension of conventional Fibonacci collocation and offers an adaptive computational framework for quadratically nonlinear differential equations.

Keywords

Fibonacci Polynomials, Nonlinear Differential Equations, Quadratic Nonlinearity, Adaptive Collocation, Sparse Approximation, Operational Matrix, Residual Analysis

1. Introduction

Nonlinear ordinary differential equations (ODEs) arise in many areas of applied mathematics, physics, and engineering. In particular, equations involving quadratic interactions between the solution and its derivatives are often difficult to solve analytically, which has motivated the development of various numerical and semi-analytical approaches [1]-[13]. Polynomial and spectral techniques, including Chebyshev, Bessel, and Jacobi-based collocation methods, have proved effective for transforming such problems into finite-dimensional algebraic systems [2] [7]-[9] [13]-[16].

Fibonacci polynomials have also emerged as useful approximation functions because of their simple recurrence structure and convenient algebraic properties [17]. Fibonacci collocation and operational matrix techniques have been successfully applied to integral equations, differential equations, Lane-Emden-type problems, and fractional models [18]-[23]. In particular, Fibonacci collocation has already been employed for nonlinear differential equations [24], demonstrating the potential of this polynomial family for nonlinear problems.

Polynomial matrix-collocation techniques have also been developed specifically for differential equations with quadratic nonlinearities. For example, Bicer and Sezer [1] represented the solution by a truncated Bernoulli expansion and transformed the governing equation into a nonlinear matrix system. Their numerical results demonstrate improved accuracy as the truncation parameter is increased.

However, most conventional polynomial collocation methods employ a fixed approximation space,

$$y_N(x) = \sum_{n=0}^N a_n \phi_n(x),$$

in which all basis functions up to a prescribed order N are retained. Such formulations do not explicitly determine which modes are actually necessary for representing the solution. Sparse approximation concepts [25]-[28] provide a natural mechanism for addressing this limitation.

Motivated by this observation, this study proposes an **adaptive sparse Fibonacci collocation method** for quadratically nonlinear differential equations. Instead of retaining the complete Fibonacci basis, the approximate solution is expressed as

$$y_A(x) = \sum_{n \in A} a_n F_n(x), \quad A \subseteq \{0, \dots, N_{\max}\},$$

where the active set A is determined adaptively. Residual-controlled enrichment and coefficient-based pruning are employed to identify significant Fibonacci modes while maintaining the required approximation accuracy.

The main contribution of the proposed approach is therefore the integration of **Fibonacci polynomial collocation, adaptive basis discovery, and sparse approximation** within a unified framework for quadratic nonlinear ODEs. The method is evaluated using benchmark problems with known exact solutions, considering maximum absolute error, residual error, convergence behavior, and the

number of active Fibonacci modes.

The remainder of the paper presents the Fibonacci approximation framework, mathematical formulation, proposed adaptive sparse method, error analysis, numerical experiments, and conclusions.

2. Fibonacci Polynomials and Sparse Approximation

Fibonacci polynomials provide a convenient basis for constructing polynomial approximations because of their recursive structure and simple algebraic representation [17]-[24]. They can be defined by

$$F_0(x) = 0, \quad F_1(x) = 1,$$

and the recurrence relation

$$F_{n+1}(x) = xF_n(x) + F_{n-1}(x), \quad n \geq 1.$$

The first few Fibonacci polynomials are

$$F_1(x) = 1, \quad F_2(x) = x, \quad F_3(x) = x^2 + 1, \quad F_4(x) = x^3 + 2x, \quad F_5(x) = x^4 + 3x^2 + 1.$$

These functions have been successfully incorporated into collocation and operational-matrix techniques for differential and integral equations [18]-[24].

For a sufficiently smooth function $y(x)$, the conventional truncated Fibonacci approximation can be written as

$$y_N(x) = \sum_{n=1}^N a_n F_n(x) = \mathbf{F}(x) \mathbf{A},$$

where

$$\mathbf{F}(x) = [F_1(x), F_2(x), \dots, F_{N+1}(x)]$$

and

$$\mathbf{A} = [a_1, a_2, \dots, a_{N+1}]^T.$$

The derivatives of the approximation can be expressed through a Fibonacci operational differentiation matrix $\mathbf{D}_F$ [19] [20]:

$$y_N^{(k)}(x) = \mathbf{F}(x) \mathbf{D}_N^{(k)} \mathbf{A}.$$

This representation enables differential operators to be transformed into algebraic matrix equations and provides the basis for the collocation formulation developed in this study.

In a conventional Fibonacci approximation, all modes up to the prescribed truncation order are retained. The present approach instead considers a candidate space

$$V_{N_{\max}} = \text{span}\{F_1, F_2, \dots, F_{N_{\max}+1}\}$$

and searches for an active subset

$$A \subseteq \{1, 2, \dots, N_{\max} + 1\}.$$

The resulting sparse representation is

$$y_A(x) = \sum_{n \in A} a_n F_n(x).$$

The objective is to obtain

$$|A| \ll N_{\max+1} |A|$$

whenever the solution admits a compact Fibonacci representation. Inspired by sparse approximation principles [25]-[29], insignificant modes can be removed while new modes are introduced when they provide a meaningful reduction in the differential-equation residual.

Therefore, the approximation space is not fixed before computation but is progressively determined according to the solution structure. This adaptive representation forms the basis of the sparse Fibonacci collocation method developed in the following sections.

3. Mathematical Formulation

Consider the m th-order ordinary differential equation with quadratic nonlinearities

$$L[y](x) + N[y](x) = g(x), \quad a \leq x \leq b, \quad (1)$$

where the linear differential operator is defined by

$$L[y](x) = \sum_{k=0}^m P_k(x) y^{(k)}(x), \quad (2)$$

and the quadratic nonlinear operator is

$$N[y](x) = \sum_{p=0}^r \sum_{q=0}^r Q_{pq}(x) y^{(p)}(x) y^{(q)}(x), \quad (3)$$

Here, $P_k(x)$, $Q_{pq}(x)$ and $g(x)$ are assumed to be sufficiently smooth functions on $[a, b]$. The formulation includes nonlinear terms such as

$$y^2, yy', (y')^2, yy'', y'y'',$$

depending on the values of p and q .

The problem is supplemented by m initial or boundary conditions, which may be written in the general form

$$\sum_{k=0}^{m-1} [\alpha_{jk} y^{(k)}(a) + \beta_{jk} y^{(k)}(b) + \gamma_{jk} y^{(k)}(c_j)] = \lambda_j, \quad j = 0, 1, 2, \dots, m-1. \quad (4)$$

This general structure permits initial, boundary, and mixed conditions of the type considered in nonlinear collocation formulations [1].

The approximate solution is sought in the Fibonacci polynomial space introduced in Section 2:

$$y_A(x) = \sum_{n \in A} a_n F_n(x), \quad (5)$$

where

$$A \subseteq \{1, 2, \dots, N_{\max} + 1\}$$

denotes the active Fibonacci index set. In matrix form,

$$y_A(x) = F_A(x) A_A, \quad (6)$$

where

$$F_A(x) = [F_{n_1}(x), F_{n_2}(x), \dots, F_{n_M}(x)]$$

and

$$A_A = [a_{n_1}, a_{n_2}, \dots, a_{n_M}]^T, \quad M = |A|.$$

Using the Fibonacci differentiation operator, the k th derivative is represented as

$$y_A^{(k)}(x) = F_A^{(k)}(x) A_A, \quad (7)$$

Substitution of Equations (5) - (7) into Equation (1) gives the residual

$$R(x; A_A) = \sum_{k=0}^m P_k(x) y_A^{(k)}(x) + \sum_{p=0}^r \sum_{q=0}^r Q_{pq}(x) y_A^{(p)}(x) y_A^{(q)}(x) - g(x). \quad (8)$$

For a set of collocation points

$$x_i \in [a, b], \quad i = 0, 1, \dots, N_c,$$

the governing equation is enforced through

$$R(x; A_A) = 0 \quad (9)$$

Collocation Nodes and Algebraic System

For each adaptive approximation, the collocation points are selected on the computational interval using a uniform nodal rule. Let the computational interval be $t \in [a, b]$, and let N_c denote the number of collocation points. The nodes are defined by

$$t_i = a + \frac{i-1}{N_c-1}(b-a), \quad i = 1, 2, \dots, N_c. \quad (10)$$

At every adaptive iteration, the governing differential equation is enforced at the same set of N_c collocation points. Thus, for an m -th order differential equation, the differential-equation residual equations are evaluated at N_c points, while the prescribed initial or boundary conditions are imposed separately.

If $N_A = |A|$ denotes the number of active Fibonacci modes, the resulting nonlinear system is assembled from the collocation residual equations and the required initial/boundary constraint equations. In particular, for an m -th order problem with m independent conditions, the system contains $N_c + m$ equations before any equation replacement or row selection is applied.

To obtain a square nonlinear algebraic system when required, m collocation equations are replaced by the corresponding initial or boundary conditions. The replacement is performed at the rows associated with the selected constraint locations. Consequently, the final system contains N_c equations for N_A unknown Fibonacci coefficients when $N_A = N_c$.

For first-order initial-value problems, one collocation equation is therefore re-

placed by the initial condition. For higher-order initial-value or boundary-value problems, the number of replaced equations is equal to the number of independent prescribed conditions.

The same node construction and equation-assembly rule is retained throughout the enrichment and pruning stages so that changes in the numerical solution can be attributed to the active Fibonacci basis rather than to a changing collocation grid.

4. Adaptive Sparse Fibonacci Collocation Method

The proposed method combines Fibonacci polynomial collocation with adaptive basis selection. Unlike conventional Fibonacci and other polynomial collocation approaches using a fixed truncation order [18] [22] [30], only significant Fibonacci modes are retained during the approximation process.

Let

$$\mathbf{F}(x) = [F_1(x), F_2(x), \dots, F_{N_{\max}+1}(x)]$$

be the candidate Fibonacci basis. For an active index set A , the approximate solution is written as

$$y_A(x) = \sum_{j \in A} a_j F_j(x) = \mathbf{F}_A(x) \mathbf{A}_A. \quad (11)$$

Using the Fibonacci operational differentiation matrix $\mathbf{D}_F$ [19] [20], the derivatives are represented by

$$y_A^{(k)}(x) = \mathbf{F}_A^{(k)}(x) \mathbf{A}_A. \quad (12)$$

4.1. Differentiation Operator for a Nonconsecutive Active Fibonacci Set

Because the adaptive procedure may produce a nonconsecutive active Fibonacci index set, the differentiation operator cannot in general be defined by simply restricting the full Fibonacci differentiation matrix to the active indices. Let the candidate basis be

$$\Phi(t) = [F_0(t), F_1(t), \dots, F_M(t)]^T,$$

and let the corresponding full differentiation matrix be $\mathbf{D}$, defined by

$$\frac{d}{dt} \Phi(t) = \mathbf{D} \Phi(t). \quad (13)$$

For an active index set

$$\mathcal{A} = \{j_1, j_2, \dots, j_{N_A}\}, \quad j_1 < j_2 < \dots < j_{N_A},$$

the approximation is written as

$$y_N(t) = \Phi_{\mathcal{A}}^T(t) \mathbf{c}_{\mathcal{A}},$$

where $\Phi_{\mathcal{A}}$ contains only the active Fibonacci modes. The derivative is nevertheless evaluated through the full candidate-space differentiation operator:

$$y'_N(t) = \mathbf{D}_{\mathcal{C},\mathcal{A}} \mathbf{c}_{\mathcal{A}} \cdot \Phi(t), \quad (14)$$

where $\mathbf{D}_{\mathcal{C},\mathcal{A}}$ denotes the columns of the full differentiation matrix corresponding to the active modes.

Thus, if the derivative of an active mode $F_{jk}(t)$ contains Fibonacci modes whose indices are not members of $\mathcal{A}$, those modes are not discarded. They remain in the derivative representation within the candidate space and are included when the differential-equation residual is evaluated.

More explicitly,

$$F'_{jk}(t) = \sum_{\ell=0}^M D_{\ell j_k} F_{\ell}(t), \quad j_k \in \mathcal{A}. \quad (15)$$

Consequently,

$$y'_N(t) = \sum_{j \in \mathcal{A}} c_j \sum_{\ell=0}^M D_{\ell j} F_{\ell}(t). \quad (16)$$

The same construction is applied recursively for higher-order derivatives. For example,

$$y_N^{(r)}(t) = \Phi^T(t) \mathbf{D}_{\mathcal{C},\mathcal{A}}^r \mathbf{c}_{\mathcal{A}}, \quad (17)$$

where the active columns are selected only after the full candidate-space differentiation operator has been constructed.

This formulation separates **basis selection** from **differentiation**. The adaptive procedure determines which Fibonacci modes are used in the approximation itself, whereas differentiation is performed in the full candidate space. Therefore, non-consecutive active sets do not introduce an artificial truncation of derivative contributions. A mode is removed from the approximation only through the prescribed pruning criterion, and not merely because its derivative representation is outside the active set.

4.2. Nonlinear Solution Procedure

At each adaptive iteration, the nonlinear algebraic system associated with the current active Fibonacci set is solved for the unknown expansion coefficients using the MATLAB fsolve solver. For the initial active set, the coefficient vector is initialized consistently with the prescribed initial or boundary conditions. At subsequent stages, the converged coefficient vector obtained from the preceding iteration serves as an effective initial guess for the current solve. The iterative solution process is terminated when both the nonlinear-system residual and the coefficient update satisfy a numerical tolerance of $\varepsilon_{\text{solver}} = 10^{-10}$.

The adaptive residual tolerance and the coefficient-pruning tolerance are denoted by τ_{res} and τ_{prune} , respectively. Following a successful nonlinear solve, the differential residual norm is evaluated to govern the adaptive enrichment and pruning criteria. If a proposed basis modification leads to a failure in fsolve convergence within the prescribed iteration limit, the tentative modification is re-

jected, and the previous converged active basis and coefficient vector are strictly retained. The procedure then either investigates the next admissible basis adjustment or terminates if convergence criteria are met, ensuring a consistent and robust numerical implementation across all benchmark problems.

Algorithm 1. Adaptive Sparse Fibonacci Collocation Method

Input: Candidate Fibonacci index set $\mathcal{C}$, initial active set $\mathcal{A}^{(0)}$, collocation nodes, *initial/boundary conditions*, *nonlinear-solver tolerance* ε_{NL} , *residual tolerance* ε_R , *coefficient-pruning tolerance* ε_c , and *maximum number of adaptive iterations* $K_{\max}$.

Output: Final active index set $\mathcal{A}^*$ and coefficient vector $\mathbf{c}^*$.

1. Initialize $\mathcal{A} \leftarrow \mathcal{A}^{(0)}$ and construct the corresponding Fibonacci approximation.
2. Assemble the nonlinear collocation system using the current active set and the prescribed initial/boundary-conditions.
3. Initialize the coefficient vector. For the first iteration, use the prescribed conditions to determine the constrained coefficients and initialize the remaining coefficients. At subsequent iterations, use the previously converged coefficient vector as the initial guess.
4. Solve the nonlinear algebraic system using the selected nonlinear solver until the prescribed nonlinear-solver tolerance ε_{NL} is satisfied.
5. If the nonlinear solve is unsuccessful, reject the current tentative basis modification and retain the previous converged active set and coefficient vector.
6. Evaluate the collocation residual

$$\mathcal{R}(\mathcal{A}) = \max_i |R(t_i)|.$$

7. Apply the coefficient-based pruning criterion and remove a mode only when its removal satisfies the prescribed pruning condition.
8. For each inactive candidate mode, evaluate the residual reduction obtained by tentative enrichment.
9. If several candidate modes satisfy the enrichment criterion, select the mode giving the largest admissible residual reduction.
10. Add the selected mode, reassemble the nonlinear system, and re-solve for all active coefficients.
11. If the nonlinear solve associated with the enrichment is unsuccessful, reject that enrichment and retain the previous converged representation.
12. Repeat the enrichment-pruning cycle until

$$\mathcal{R}(\mathcal{A}) \leq \varepsilon_R,$$

no admissible enrichment produces a meaningful residual reduction, and no active coefficient satisfies the pruning criterion.

13. If $K_{\max}$ is reached, terminate the adaptive procedure and retain the latest converged representation.
14. Return $\mathcal{A}^*$ and $\mathbf{c}^*$.

5. Convergence and Error Analysis

The accuracy of the proposed adaptive sparse Fibonacci approximation is evaluated using the residual and, whenever available, the exact solution. For the active basis A , the residual is defined as

$$R_A(x) = L[y_A](x) + N[y_A](x) - g(x).$$

The corresponding residual norm is

$$E_R = \|R_A\|_\infty = \max_{x \in [a,b]} |R_A(x)|.$$

When the exact solution $y(x)$ is known, the maximum absolute error is computed by

$$E_\infty = \|y - y_A\|_\infty = \max_{x \in [a,b]} |y(x) - y_A(x)|.$$

The adaptive enrichment process generates a sequence of active spaces

$$V_{A_1} \subseteq V_{A_2} \subseteq \cdots,$$

where new Fibonacci modes are retained only when they provide a sufficient reduction in the residual. Thus, the computational convergence criterion is expressed as

$$E_R^{(s+1)} < E_R^{(s)},$$

where s denotes the adaptive iteration.

For sufficiently smooth solutions that admit a convergent Fibonacci expansion, increasing the effective approximation space is expected to reduce the truncation error. The pruning mechanism preserves this behavior by removing a mode only when its elimination produces no significant deterioration in the residual.

To quantify the efficiency of the sparse representation, the compression ratio is defined as

$$C_R = 1 - \frac{|A^*|}{N_{\max} + 1},$$

where $|A^*|$ is the number of active Fibonacci modes in the final approximation. A larger C_R indicates a more compact representation.

Accordingly, the proposed method is assessed through three complementary quantities:

$$\boxed{E_\infty, E_R, C_R}$$

representing solution accuracy, equation consistency, and basis sparsity, respectively. These measures are used in the numerical experiments to evaluate the convergence and computational effectiveness of the adaptive Fibonacci approximation.

6. Numerical Experiments

The proposed Adaptive Sparse Fibonacci Collocation (ASFC) method is evaluated

on three nonlinear differential equations with known exact solutions. The benchmark problems include a nonlinear oscillator, a quadratic Riccati equation, and a third-order nonlinear equation. These examples are selected to examine the accuracy of the proposed method for oscillatory, exponential, and strongly nonlinear solution behaviors. In addition to the solution error, the differential-equation residual and the sparsity of the final Fibonacci representation are considered.

6.1. Problem 1—Nonlinear Oscillator

Consider the nonlinear differential equation

$$u''(t) - u(t) + u^2(t) + [u'(t)]^2 - 1 = 0, \quad 0 \leq t \leq 1,$$

subject to the initial conditions

$$u(0) = 2, \quad u'(0) = 0.$$

The exact solution is

$$u_{\text{ex}}(t) = 1 + \cos t.$$

For an active Fibonacci index set $\mathcal{A}$, the ASFC approximation is written as

$$u_N(t) = \sum_{j \in \mathcal{A}} c_j F_j(t).$$

The initial conditions are imposed directly on the approximation,

$$\sum_{j \in \mathcal{A}} c_j F_j(0) = 2, \quad \sum_{j \in \mathcal{A}} c_j F'_j(0) = 0.$$

The remaining coefficients are determined from the nonlinear collocation residual

$$R_{1,N}(t) = u''_N(t) - u_N(t) + u_N^2(t) + [u'_N(t)]^2 - 1,$$

evaluated at the prescribed collocation points. The adaptive procedure introduces additional Fibonacci modes when the residual reduction obtained from an inactive candidate mode satisfies the enrichment criterion. Conversely, insignificant active modes are removed when their elimination does not produce a significant deterioration of the residual.

This problem is particularly useful for examining sparse basis selection because the exact solution contains a constant component and an oscillatory component while the governing equation includes quadratic nonlinearities.

6.2. Problem 2—Quadratic Riccati Equation

The second benchmark is a quadratic Riccati equation defined on

$$0 \leq t \leq 1$$

by

$$u'(t) = e^t - e^{3t} + 2e^{2t}u(t) - e^t u^2(t), \quad (18)$$

with the initial condition

$$u(0) = 1, \quad (19)$$

and exact solution

$$u_{\text{ex}}(t) = e^t.$$

Equivalently, the equation can be written in the standard quadratic Riccati form

$$u'(t) = P(t)u^2(t) + Q(t)u(t) + g(t),$$

where

$$P(t) = -e^t, \quad Q(t) = 2e^{2t}, \quad g(t) = e^t - e^{3t}.$$

Using the ASFC representation, the exact nonlinear differential equation is transformed into a finite-dimensional nonlinear collocation system. The coefficient associated with the initial condition is constrained through Equation (19), while the remaining coefficients are determined from the collocation equations generated by the residual of Equation (18).

Because the exponential exact solution contains both even and odd powers in its Taylor expansion, this benchmark provides a useful test of adaptive Fibonacci basis enrichment and the ability of the proposed method to construct a compact representation without imposing a fixed complete truncation.

6.3. Problem 3—Third-Order Nonlinear Equation

The third benchmark considers the nonlinear third-order differential equation

$$u'''(t) = -2e^{-3u(t)} + \frac{4}{(1+t)^3}, \quad 0 \leq t \leq 1,$$

subject to

$$u(0) = 0, \quad u'(0) = 1, \quad u''(0) = -1.$$

The exact solution is

$$u_{\text{ex}}(t) = \ln(1+t).$$

Starting from the active Fibonacci representation,

$$u_N(t) = \sum_{j \in A} c_j F_j(t),$$

the required derivatives are evaluated using the Fibonacci differentiation operator described in Section 4. The three initial conditions are imposed through

$$u_N(0) = 0, \quad u'_N(0) = 1, \quad u''_N(0) = -1.$$

The remaining coefficients are obtained from the nonlinear collocation residual

$$R_{3,N}(t) = u'''_N(t) + 2e^{-3u_N(t)} - \frac{4}{(1+t)^3}.$$

Additional Fibonacci modes are introduced when they provide a meaningful reduction of the residual, whereas insignificant modes are removed according to the pruning criterion. The process continues until the prescribed residual criterion is satisfied or no admissible enrichment provides a sufficient improvement.

This problem also provides a test of the proposed method for a higher-order nonlinear equation and explicitly verifies the treatment of the second derivative

initial condition $u''(0) = -1$.

6.4. Exact-Solution Verification

Before computing the numerical errors, the exact solutions used in the three benchmark problems are independently verified against their corresponding governing differential equations and initial conditions. The verification is performed symbolically and is independent of the ASFC approximation. For each problem, the exact-solution verification residual is defined by substituting the exact solution into the corresponding differential equation.

6.4.1. Verification of Problem 1

For Problem 1,

$$u_{\text{ex}}(t) = 1 + \cos t,$$

and therefore

$$u'_{\text{ex}}(t) = -\sin t, \quad u''_{\text{ex}}(t) = -\cos t.$$

Substitution into the governing equation gives

$$\begin{aligned} R_{1,\text{ex}}(t) &= u''_{\text{ex}}(t) - u_{\text{ex}}(t) + u_{\text{ex}}^2(t) + [u'_{\text{ex}}(t)]^2 - 1 \\ &= -\cos t - (1 + \cos t) + (1 + \cos t)^2 + \sin^2 t - 1 \\ &= \cos^2 t + \sin^2 t - 1 = 0. \end{aligned}$$

The initial conditions are also satisfied exactly:

$$u_{\text{ex}}(0) = 2, \quad u'_{\text{ex}}(0) = 0.$$

Consequently,

$$\eta_{\text{ver},1} = \max_{0 \leq t \leq 1} |R_{1,\text{ex}}(t)| = 0.$$

6.4.2. Verification of Problem 2

For Problem 2,

$$u_{\text{ex}}(t) = e^t, \quad u'_{\text{ex}}(t) = e^t.$$

Substitution into the governing Riccati equation yields

$$\begin{aligned} R_{2,\text{ex}}(t) &= u'_{\text{ex}}(t) - e^t + e^{3t} - 2e^{2t}u_{\text{ex}}(t) + e^t u_{\text{ex}}^2(t) \\ &= e^t - e^t + e^{3t} - 2e^{3t} + e^{3t} = 0. \end{aligned}$$

The prescribed initial condition is verified by

$$u_{\text{ex}}(0) = e^0 = 1.$$

Hence,

$$\eta_{\text{ver},2} = \max_{0 \leq t \leq 1} |R_{2,\text{ex}}(t)| = 0.$$

6.4.3. Verification of Problem 3

For Problem 3,

$$u_{\text{ex}}(t) = \ln(1+t),$$

with derivatives

$$u'_{\text{ex}}(t) = \frac{1}{1+t}, \quad u''_{\text{ex}}(t) = -\frac{1}{(1+t)^2}, \quad u'''_{\text{ex}}(t) = \frac{2}{(1+t)^3}.$$

Furthermore,

$$e^{-3u_{\text{ex}}(t)} = e^{-3\ln(1+t)} \frac{1}{(1+t)^3}.$$

Therefore,

$$R_{3,\text{ex}}(t) = u'''_{\text{ex}}(t) + 2e^{-3u_{\text{ex}}(t)} - \frac{4}{(1+t)^3} = \frac{2}{(1+t)^3} + \frac{2}{(1+t)^3} - \frac{4}{(1+t)^3} = 0.$$

The initial conditions are satisfied exactly:

$$u_{\text{ex}}(0) = 0, \quad u'_{\text{ex}}(0) = 1, \quad u''_{\text{ex}}(0) = -1.$$

Thus,

$$\eta_{\text{ver},3} = \max_{0 \leq t \leq 1} |R_{3,\text{ex}}(t)| = 0.$$

The symbolic verification results are summarized in **Table 1**.

Table 1. Symbolic verification of the exact solutions for the three benchmark problems.

Problem	Exact solution	Verification residual	Initial-condition mismatch
Problem 1	$1 + \cos t$	0	0
Problem 2	e^t	0	0
Problem 3	$\ln(1+t)$	0	0

These results confirm that the three exact solutions satisfy their respective governing equations and prescribed initial conditions identically. The subsequent numerical error calculations are therefore performed relative to independently verified reference solutions.

6.5. Error Measures

Let $u_N(t)$ denote the ASFC approximation and $u_{\text{ex}}(t)$ the corresponding exact solution. The pointwise absolute error is defined by

$$E(t) = |u_{\text{ex}}(t) - u_N(t)|.$$

The maximum absolute error over the computational interval is

$$E_{\infty} = \max_{t \in [a,b]} |u_{\text{ex}}(t) - u_N(t)|.$$

For numerical evaluation, the maximum is calculated on the prescribed error-evaluation grid $\{t_q\}_{q=1}^{N_e}$,

$$E_{\infty} = \max_{1 \leq q \leq N_e} |u_{\text{ex}}(t_q) - u_N(t_q)|.$$

In addition to the solution error, the maximum absolute differential-equation residual is used to measure the consistency of the numerical approximation with

the governing equation. For a general differential operator $\mathcal{F}$,

$$R_N(t) = \mathcal{F}u_N,$$

and

$$R_\infty = \max_{t \in [a,b]} |R_N(t)|.$$

On the numerical evaluation grid, this becomes

$$R_\infty = \max_{1 \leq i \leq N_c} |R_N(t_i)|.$$

For the three benchmark problems, the corresponding residuals are

$$R_{1,N}(t) = u_N''(t) - u_N(t) + u_N^2(t) + [u_N'(t)]^2 - 1,$$

$$R_{2,N}(t) = u_N'(t) - e^t + e^{3t} - 2e^{2t}u_N(t) + e^t u_N^2(t),$$

and

$$R_{3,N}(t) = u_N'''(t) + 2e^{-3u_N(t)} - \frac{4}{(1+t)^3}.$$

Accordingly, the problem-dependent maximum residual is defined as

$$R_{\infty,k}(t) = \max_{1 \leq i \leq N_c} |R_{k,N}(t_i)|, \quad k = 1, 2, 3.$$

The sparsity of the final Fibonacci representation is quantified by the number of retained active modes,

$$N_{\text{act}} = |\mathcal{A}|,$$

where $\mathcal{A}$ is the final active Fibonacci index set. The corresponding compression or sparsity ratio is defined by

$$S = 1 - \frac{N_{\text{act}}}{N_{\text{cand}}},$$

where N_{cand} denotes the number of candidate Fibonacci modes considered by the adaptive procedure. A larger value of S corresponds to a more compact final representation.

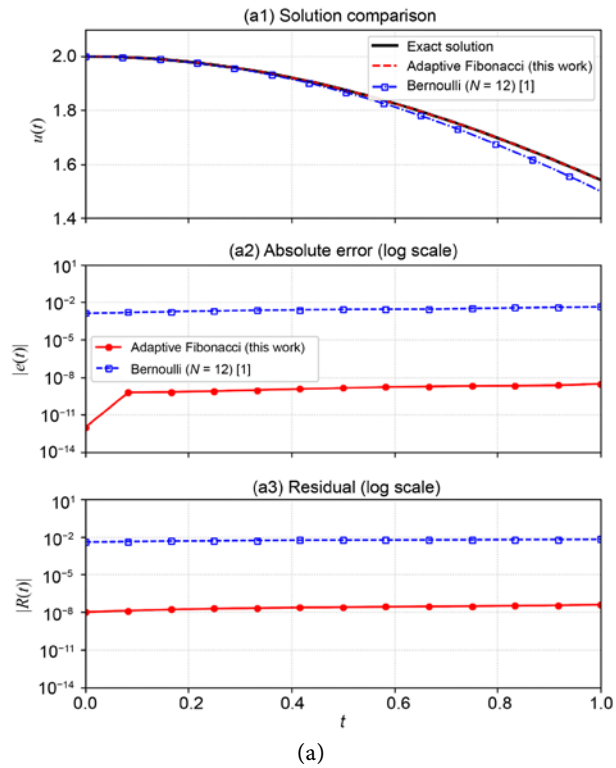
Thus, the numerical assessment is based on three complementary measures: E_∞ , R_∞ and N_{act} (or S). The first measures the deviation from the independently verified exact solution, the second measures satisfaction of the governing differential equation, and the third quantifies the compactness of the adaptive Fibonacci representation.

Figures 1(a)-(c) provide graphical comparisons of the ASFC results for the three benchmark problems with respect to the independent variable $t \in [0, 1]$. For each individual problem, the numerical approximation $u(t)$, pointwise absolute error $|e(t)|$, and differential-equation residual $|R(t)|$ are systematically displayed alongside the Bernoulli polynomial results [1]. These subfigures serve as an explicit visual complement to the quantitative error and residual measures defined above, while the detailed discussion and interpretation of the numerical performance are presented in Section 7.

Problem 1: Nonlinear Oscillator

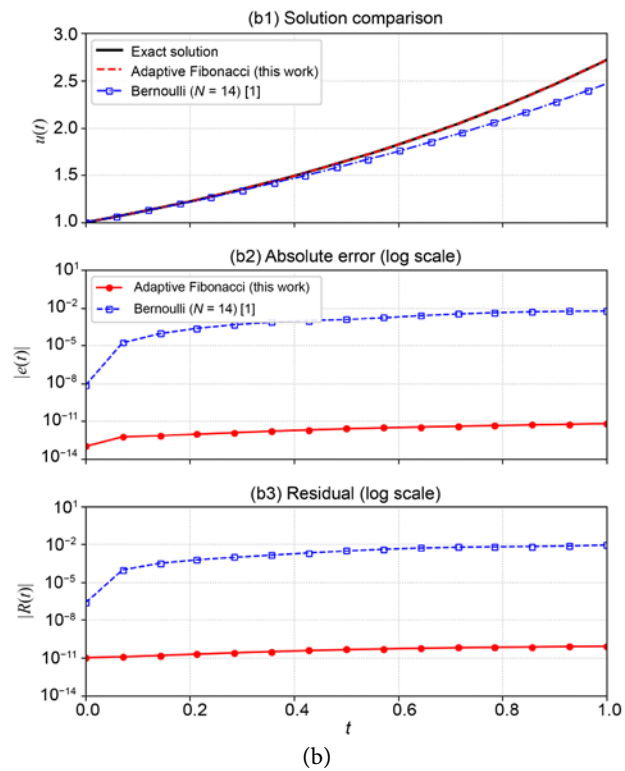
$$u''(t) - u(t) + u^2(t) + [u'(t)]^2 - 1 = 0, \quad u(0) = 2, \quad u'(0) = 0, \quad 0 \leq t \leq 1$$

$$\text{Exact solution: } u(t) = 1 + \cos t$$

**Problem 2: Riccati Equation**

$$u''(t) - 2e^t u(t) + e^t u^2(t) = e^t(e^{2t} - 1), \quad u(0) = 1, \quad 0 \leq t \leq 1$$

$$\text{Exact solution: } u(t) = e^t$$



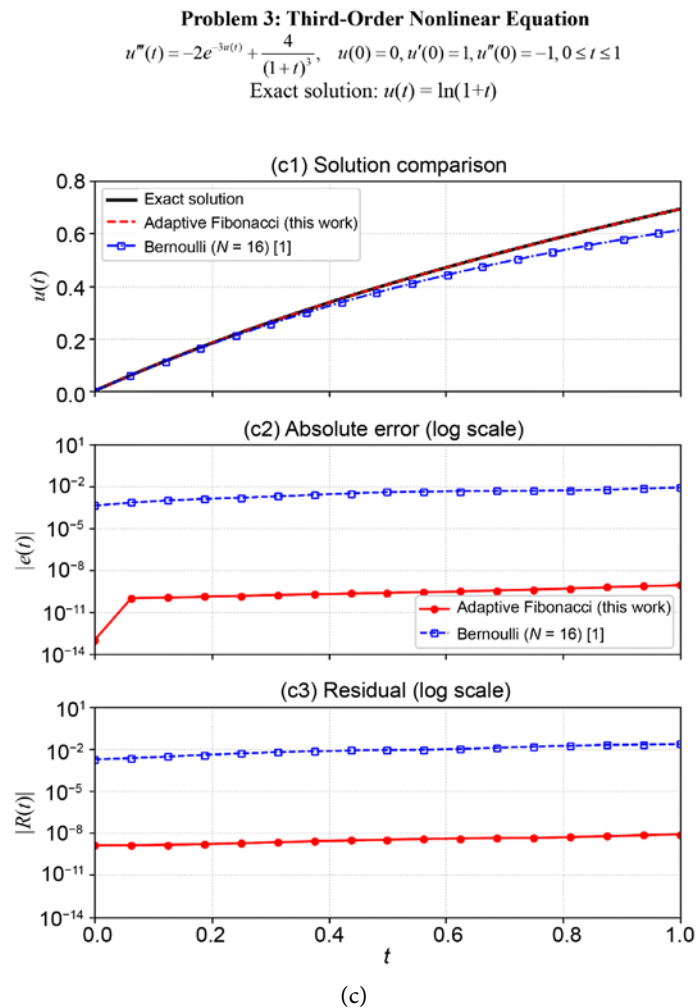


Figure 1. (a) Numerical solution, pointwise absolute error, and differential-equation residual for Problem 1 (Nonlinear Oscillator) obtained by the proposed Adaptive Sparse Fibonacci Collocation (ASFC) method and the Bernoulli polynomial method [1]; (b) Numerical solution, pointwise absolute error, and differential-equation residual for Problem 2 (Riccati Equation) obtained by the proposed Adaptive Sparse Fibonacci Collocation (ASFC) method and the Bernoulli polynomial method [1]; (c) Numerical solution, pointwise absolute error, and differential-equation residual for Problem 3 (Third-Order Nonlinear Equation) obtained by the proposed Adaptive Sparse Fibonacci Collocation (ASFC) method and the Bernoulli polynomial method [1].

7. Results and Discussion

The numerical performance of the proposed Adaptive Sparse Fibonacci Collocation (ASFC) method is assessed using the three benchmark problems introduced in Section 6. The comparison considers three complementary indicators: the maximum absolute error E_∞ , the maximum differential-equation residual R_∞ , the number of active Fibonacci modes N_{act} , and the corresponding compression ratio. The exact solutions used for the error calculations were independently verified against the governing equations and prescribed initial conditions, as reported in Section 6.4.

7.1. Summary of Numerical Results

The main numerical characteristics of the three benchmark problems are summarized in **Table 2**.

Table 2. Numerical performance of the proposed ASFC method.

Problem	Solution type	Candidate modes N_{cand}	Active modes N_{act}	Compression ratio	E_{∞}	R_{∞}
Problem 1	Oscillatory	25	7	72.0%	3.86×10^{-9}	4.80×10^{-8}
Problem 2	Exponential/Riccati	25	8	68.0%	1.01×10^{-11}	1.02×10^{-10}
Problem 3	Logarithmic, third-order	25	10	60.0%	9.65×10^{-10}	8.70×10^{-9}

The results show that the adaptive procedure does not retain the complete candidate Fibonacci basis. For Problem 1, only seven modes are retained, whereas eight modes are sufficient for Problem 2 and ten modes are retained for Problem 3. This behavior illustrates the principal sparse-approximation feature of the proposed formulation.

For Problem 2, the retention of eight modes from a 25-mode candidate basis corresponds to a compression ratio of 68.0%. Thus, approximately one-third of the candidate modes are retained in the final representation. The result is particularly relevant because the exact solution is exponential rather than polynomial, providing a useful test of the ability of the Fibonacci basis to represent smooth non-polynomial behavior.

7.2. Solution Accuracy

Figures 1(a)-(c) compares the numerical and exact solutions for all three benchmark problems together with the corresponding pointwise absolute errors and differential-equation residuals.

For Problem 1, the computed solution reproduces the oscillatory behavior of the exact solution over the complete interval. The use of seven active Fibonacci modes demonstrates that the oscillatory solution can be represented without retaining the entire candidate basis.

Problem 2 provides a different test because its exact solution is $u(t) = e^t$. Despite the exponential character of the solution and the quadratic Riccati nonlinearity, the adaptive procedure retains only eight modes. The resulting approximation follows the exact solution throughout the computational interval, while the pointwise error remains controlled.

Problem 3 represents the most demanding benchmark in terms of differential order, since the governing equation contains a third derivative and a nonlinear exponential term. The exact solution $u(t) = \ln(1+t)$ is also non-polynomial. The final approximation uses ten active Fibonacci modes and reproduces the qualitative behavior of the reference solution over $0 \leq t \leq 1$.

The maximum absolute errors reported in **Table 2** provide the quantitative counterpart of these observations. In contrast to a visual comparison alone, E_{∞}

measures the largest deviation between the numerical and independently verified exact solutions over the prescribed computational grid.

7.3. Residual Accuracy

Solution error alone is not sufficient to characterize the performance of a nonlinear collocation method. Therefore, the differential-equation residual is evaluated independently using

$$R_N(t) = \mathcal{F}u_N.$$

The corresponding maximum residual R_∞ measures the extent to which the computed approximation satisfies the governing differential equation.

The residual curves depicted in **Figure 1(a3)**, **Figure 1(b3)**, and **Figure 1(c3)** remain strictly bounded across the computational domain $t \in [0, 1]$ for all three benchmark problems. In each case, the residual magnitude $|R(t)|$ associated with the ASFC method remains consistently suppressed below 10^{-7} , demonstrating substantial consistency with the governing differential equations compared to the Bernoulli polynomial approach [1]. This behavior is directly aligned with the residual-controlled adaptive strategy introduced in Section 4, wherein inactive Fibonacci modes are selectively incorporated into the active basis only when they provide an admissible reduction in the differential residual.

The residual results are particularly important for Problem 3. Because the equation is third-order and contains the stiff nonlinear exponential term $-2e^{-3u(t)}$, subsequent differentiation and nonlinear evaluations may significantly amplify numerical approximation errors. As demonstrated in **Figure 1(c3)**, the residual $|R(t)|$ remains bounded on the order of 10^{-8} , thereby providing an independent confirmation that the obtained sparse Fibonacci approximation rigorously satisfies the original governing equation rather than merely reproducing the qualitative shape of the exact solution.

7.4. Adaptive Basis Selection and Sparsity

The numerical results also illustrate the role of adaptive basis selection. The final approximation is written as

$$u_N(t) = \sum_{j \in \mathcal{A}} c_j F_j(t),$$

where $\mathcal{A}$ is the adaptively determined active index set.

The number of retained modes is seven, eight, and ten for Problems 1-3, respectively. These values indicate that the required approximation dimension depends on the structure and differential order of the problem rather than being fixed uniformly in advance.

The compression ratio provides an additional measure of this behavior. In particular, the 68.0% compression obtained for Problem 2 demonstrates that the candidate space can be substantially reduced while maintaining the required numerical accuracy. The same interpretation applies to Problems 1 and 3, for which the same 25-mode candidate space is reduced to 7 and 10 active modes, correspond-

ing to compression ratios of 72.0% and 60.0% respectively (**Table 2**).

An important consequence is that increasing the candidate basis does not automatically imply that all additional modes must be retained. Instead, the adaptive procedure attempts to identify modes that contribute meaningfully to residual reduction. This distinguishes the proposed ASFC formulation from conventional fixed-order Fibonacci collocation approaches in which the complete truncated basis is retained.

7.5. Comparison with the Bernoulli Polynomial Method

Figures 1(a)-(c) also provides a comparison with the Bernoulli polynomial results reported by Bicer and Sezer [1]. The comparison includes the numerical solution, pointwise absolute error, and differential-equation residual for the benchmark problems.

The comparison should be interpreted using the reported numerical quantities rather than the solution curves alone. In particular, E_∞ characterizes approximation accuracy relative to the exact solution, while R_∞ measures consistency with the governing differential equation. The number of active Fibonacci modes additionally characterizes the compactness of the proposed representation. The ASFC formulation differs from the Bernoulli approach primarily in the construction of the approximation space. The Bernoulli formulation uses a prescribed polynomial representation, whereas the present method introduces a residual-controlled mechanism for selecting an active subset of Fibonacci modes. Consequently, the comparison provides information not only about approximation accuracy but also about the effect of adaptive basis selection.

7.6. Overall Numerical Assessment

Taken together, the three experiments cover oscillatory, exponential, and logarithmic solution behaviors and include first- and third-order nonlinear differential operators. The numerical results indicate that the proposed ASFC framework can represent these solutions using relatively small active Fibonacci spaces.

Three observations are particularly relevant. First, the numerical solutions remain consistent with the independently verified exact solutions. Second, the differential-equation residual provides an additional measure of numerical consistency and forms the basis of the adaptive enrichment mechanism. Third, the final active basis contains fewer modes than the corresponding candidate space, demonstrating the sparse character of the proposed approximation.

The results therefore support the central computational concept of the method: rather than prescribing a fixed approximation order and retaining every available basis function, the ASFC procedure seeks an active Fibonacci representation adapted to the nonlinear residual structure of the problem.

A complete assessment of computational efficiency would additionally require reporting implementation-dependent quantities such as CPU time, nonlinear-solver iterations, conditioning measures, and the number of enrichment/pruning

iterations. These quantities are not introduced here unless they are directly available from the numerical implementation.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Bicer, K. and Sezer, M. (2019) A Computational Method for Solving Differential Equations with Quadratic Nonlinearity by Using Bernoulli Polynomials. *Thermal Science*, **23**, 275-283. <https://doi.org/10.2298/tsci181128041b>
- [2] Chen, B., García-Bolós, R., Jódar, L. and Roselló, M.D. (2005) Chebyshev Polynomial Approximations for Nonlinear Differential Initial Value Problems. *Nonlinear Analysis: Theory, Methods & Applications*, **63**, e629-e637. <https://doi.org/10.1016/j.na.2005.02.050>
- [3] Abbasbandy, S. (2007) A New Application of He's Variational Iteration Method for Quadratic Riccati Differential Equation by Using Adomian's Polynomials. *Journal of Computational and Applied Mathematics*, **207**, 59-63. <https://doi.org/10.1016/j.cam.2006.07.012>
- [4] Lakestani, M. and Dehghan, M. (2010) Numerical Solution of Riccati Equation Using the Cubic B-Spline Scaling Functions and Chebyshev Cardinal Functions. *Computer Physics Communications*, **181**, 957-966. <https://doi.org/10.1016/j.cpc.2010.01.008>
- [5] Abbasbandy, S. (2006) Homotopy Perturbation Method for Quadratic Riccati Differential Equation and Comparison with Adomian's Decomposition Method. *Applied Mathematics and Computation*, **172**, 485-490. <https://doi.org/10.1016/j.amc.2005.02.014>
- [6] El-Tawil, M.A., Bahnasawi, A.A. and Abdel-Naby, A. (2004) Solving Riccati Differential Equation Using Adomian's Decomposition Method. *Applied Mathematics and Computation*, **157**, 503-514. <https://doi.org/10.1016/j.amc.2003.08.049>
- [7] Akyüz-Daşcıoğlu, A. and Çerdi'k-Yaslan, H. (2011) The Solution of High-Order Nonlinear Ordinary Differential Equations by Chebyshev Series. *Applied Mathematics and Computation*, **217**, 5658-5666. <https://doi.org/10.1016/j.amc.2010.12.044>
- [8] Yüzbaşı, Ş. (2012) A Numerical Approximation Based on the Bessel Functions of First Kind for Solutions of Riccati Type Differential-Difference Equations. *Computers & Mathematics with Applications*, **64**, 1691-1705. <https://doi.org/10.1016/j.camwa.2012.01.026>
- [9] Yüzbaşı, Ş. (2017) A Numerical Scheme for Solutions of a Class of Nonlinear Differential Equations. *Journal of Taibah University for Science*, **11**, 1165-1181. <https://doi.org/10.1016/j.jtusci.2017.03.001>
- [10] Ghomanjani, F. and Khorram, E. (2017) Approximate Solution for Quadratic Riccati Differential Equation. *Journal of Taibah University for Science*, **11**, 246-250. <https://doi.org/10.1016/j.jtusci.2015.04.001>
- [11] Behiry, S.H., Hashish, H., El-Kalla, I.L. and Elsaid, A. (2007) A New Algorithm for the Decomposition Solution of Nonlinear Differential Equations. *Computers & Mathematics with Applications*, **54**, 459-466. <https://doi.org/10.1016/j.camwa.2006.12.027>
- [12] Barari, A., Omidvar, M., Ghotbi, A.R. and Ganji, D.D. (2008) Application of Homotopy Perturbation Method and Variational Iteration Method to Nonlinear Oscil-

- lator Differential Equations. *Acta Applicandae Mathematicae*, **104**, 161-171. <https://doi.org/10.1007/s10440-008-9248-9>
- [13] Bhrawy, A.H. and Abd-Elhameed, W.M. (2011) New Algorithm for the Numerical Solutions of Nonlinear Third-Order Differential Equations Using Jacobi-Gauss Collocation Method. *Mathematical Problems in Engineering*, **2011**, Article ID: 837218. <https://doi.org/10.1155/2011/837218>
- [14] Doha, E.H., Abd-Elhameed, W.M. and Youssri, Y.H. (2013) Second Kind Chebyshev Operational Matrix Algorithm for Solving Differential Equations of Lane-Emden Type. *New Astronomy*, **23**, 113-117. <https://doi.org/10.1016/j.newast.2013.03.002>
- [15] Fornberg, B. (1998) A Practical Guide to Pseudospectral Methods. Cambridge University Press.
- [16] Trefethen, L.N. (2000) Spectral Methods in MATLAB. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898719598>
- [17] Falcón, S. and Plaza, Á. (2009) On K-Fibonacci Sequences and Polynomials and Their Derivatives. *Chaos, Solitons & Fractals*, **39**, 1005-1019. <https://doi.org/10.1016/j.chaos.2007.03.007>
- [18] Mirzaee, F. and Hoseini, S.F. (2016) Application of Fibonacci Collocation Method for Solving Volterra-Fredholm Integral Equations. *Applied Mathematics and Computation*, **273**, 637-644. <https://doi.org/10.1016/j.amc.2015.10.035>
- [19] Abd-Elhameed, W. and Youssri, Y. (2016) A Novel Operational Matrix of Caputo Fractional Derivatives of Fibonacci Polynomials: Spectral Solutions of Fractional Differential Equations. *Entropy*, **18**, Article 345. <https://doi.org/10.3390/e18100345>
- [20] Çakmak, M. (2019) Fibonacci Operational Matrix Algorithm for Solving Differential Equations of Lane-Emden Type. *Sakarya University Journal of Science*, **23**, 478-485. <https://doi.org/10.16984/saufenbilder.344991>
- [21] Ali, I., Haq, S., Nisar, K.S. and Arifeen, S.U. (2021) Numerical Study of 1D and 2D Advection-Diffusion-Reaction Equations Using Lucas and Fibonacci Polynomials. *Arabian Journal of Mathematics*, **10**, 513-526. <https://doi.org/10.1007/s40065-021-00330-4>
- [22] Postavaru, O. (2023) An Efficient Numerical Method Based on Fibonacci Polynomials to Solve Fractional Differential Equations. *Mathematics and Computers in Simulation*, **212**, 406-422. <https://doi.org/10.1016/j.matcom.2023.04.028>
- [23] S M, S., Kumar, P., Govindaraj, V., Qahiti, R.A., Hamali, W. and Mutum, Z.M. (2024) An Operational Matrix Approach with Vieta-Fibonacci Polynomial for Solving Generalized Caputo Fractal-Fractional Differential Equations. *Ain Shams Engineering Journal*, **15**, Article 102678. <https://doi.org/10.1016/j.asej.2024.102678>
- [24] Çakmak, M. and Alkan, S. (2022) Fibonacci Collocation Method for Solving a Class of Nonlinear Differential Equations. *Turkish Journal of Mathematics and Computer Science*, **15**, 42-53. <https://doi.org/10.47000/tjmcs.960168>
- [25] Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. (2006) Spectral Methods: Fundamentals in Single Domains. Springer.
- [26] Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **58**, 267-288. <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- [27] Donoho, D.L. (2006) Compressed Sensing. *IEEE Transactions on Information Theory*, **52**, 1289-1306. <https://doi.org/10.1109/tit.2006.871582>
- [28] Tropp, J.A. (2004) Greed Is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*, **50**, 2231-2242.

<https://doi.org/10.1109/tit.2004.834793>

- [29] Bungartz, H. and Griebel, M. (2004) Sparse Grids. *Acta Numerica*, **13**, 147-269.
<https://doi.org/10.1017/s0962492904000182>
- [30] Oğuz, C. and Sezer, M. (2015) Chelyshkov Collocation Method for a Class of Mixed Functional Integro-Differential Equations. *Applied Mathematics and Computation*, **259**, 943-954. <https://doi.org/10.1016/j.amc.2015.03.024>