**Volume 1, Issue 1, Pages 31–83, April - June 2026**

**International Journal of Pure, Applied and Computational Mathematics (IJPACM)**

Journal homepage:

https://ijpacm.nobleinkresearch.com/1

ISSN Online- 2839-5875; ISSN Print - 2839-5751

# A Stable Spectral Bernstein Physics-Informed Neural Network for Solving Fractional Allee Population Models with Caputo Derivatives

Pappu Verma[1], Sachin Kumar[1], Paras Varshney[2,*]

[1]Department of Mathematics, Government Degree College,Budaun, Uttar Pradesh, India

[2]Department of Mathematics, Model Public Education College, Chandausi, India.

*Corresponding author: parasshayarmath.chd@gmail.com

**Article history:**

Received: 15 April 2026

Revised: 29 May 2026

Accepted: 14 June 2026

Available online: 30 June 2026

**Article type:**

Research Article

**Keywords:**

Fractional Allee effect

Spectral Bernstein method

Method of Manufactured Solutions

Stability analysis

Population dynamics

**Abstract**

Fractional-order population models have attracted considerable attention because they effectively capture memory and hereditary effects that are absent in classical integer-order models. Among these, the fractional Allee effect model provides a realistic framework for describing population persistence and extinction by incorporating a critical population threshold. In this work, we develop a Stable Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) to solve the Caputo fractional-order Allee effect model. The proposed framework employs a trainable Bernstein polynomial expansion with a logarithmically stable basis and an analytical Bernstein differentiation matrix to evaluate the Caputo fractional derivative efficiently through matrix–vector multiplication, thereby eliminating repeated numerical discretization during training. Chebyshev–Gauss–Lobatto collocation points are used to improve numerical stability, while a hybrid Adam–L-BFGS optimization strategy accelerates convergence. Stability of the equilibrium points is investigated using the Jacobian matrix and Matignon's stability theorem, confirming that the extinction and carrying-capacity equilibria are locally stable, whereas the Allee threshold equilibrium is unstable. The proposed method is validated using the Method of Manufactured Solutions over fractional orders $0.1 \leq \alpha \leq 0.9$. Numerical experiments demonstrate excellent agreement with analytical solutions, achieving residuals of $3.04\times10^{-29}$, maximum absolute errors of $3.55 \times 10^{-15}$, and relative $L_2$ errors of $9.51 \times 10^{-17}$ for $\alpha = 0.9$. The results demonstrate that SBNN-PINN is an accurate, stable, and computationally efficient framework for solving nonlinear fractional population models.

© 2026 Nobleink Research Publishing Private Limited.

DOI: https://doi.org/10.68219/f26md935

# 1. Introduction

Population dynamics is one of the most significant applications of nonlinear differential equations because it provides a quantitative model of the persistence, extinction, invasion, harvesting, and ecosystem stability of species. Mathematical models have become a fundamental tool for studying the evolution of biological populations over time and ecological responses to environmental changes and for conservation and resource management decisions. Classical population models, such as the exponential and logistic growth equations, have been used to explain many ecological phenomena in general by assuming that population growth rates depend only on the current state of the system. Although these integer-order models have provided important theoretical insights, they do not take into account hereditary characteristics, delayed responses, and long-term memory effects which are naturally present in biological and ecological systems. Thus, adding memory effects to mathematical models has become an important research direction in modern theoretical ecology.

Fractional calculus has become a powerful mathematical tool to describe systems with memory and hereditary properties by the derivatives of non-integer order [1–3]. Unlike classical derivatives, fractional derivatives can incorporate historical information over the whole evolution of the system and thus capture a more realistic picture of many biological, physical, engineering, and financial processes [4–6]. In the last 20 years, fractional differential equations (FDEs) have been recognized as a powerful tool to model anomalous diffusion, viscoelastic materials, epidemic transmission, ecological interactions, control systems, and complex dynamics with nonlocal behavior [7]. Of the different types of fractional derivatives, it is the Caputo fractional derivative which is most suitable to describe physical conditions in terms of integer-order derivatives [8]. These initial conditions are known naturally from the experiments (e.g. from the existence of population density) and thus the formula for the Caputo fractional derivatives is one of the best known and well known. The Caputo fractional models were then widely used to study nonlinear population dynamics with memory effects and to retain an intuitive physical interpretation.

A fundamental nonlinear phenomenon in population ecology is the *Allee effect*, originally introduced by W. C. Allee to explain situations in which the per-capita population growth rate decreases when population density becomes sufficiently small [9]. Unlike the classical logistic growth model, populations experiencing an Allee effect exhibit a critical population threshold below which reproduction, survival, and long-term persistence become increasingly difficult [10,11]. This reduction in growth may result from several ecological mechanisms, including mate limitation, cooperative hunting, collective defense, social interactions, and reduced genetic diversity in sparse populations [12, 13]. As a consequence, populations with densities below the Allee threshold tend toward extinction, whereas populations exceeding this critical level may recover and eventually approach the environmental carrying capacity. Because of its ability to capture threshold-dependent dynamics, the Allee effect has become an important component of mathematical models describing endangered species, invasive organisms, fisheries management, forest ecosystems, wildlife conservation, and biological resource management.

The incorporation of fractional-order derivatives into Allee population models provides a more realistic description of ecological systems by accounting for the influence of past population states on current population growth [14, 15]. Such models can reproduce long-memory effects and complex transient dynamics that cannot be adequately described with the standard integer-order equations. These properties have motivated an extensive number of theoretical and numerical studies in fractional ecological models such as equilibrium analysis, stability, bifurcation behavior, optimal harvesting and population persistence. However, the nonlocal nature of fractional derivatives makes the mathematical and computational complexity of the equations much more challenging, and efficient numerical algorithms are an active area of research in the present day.

Many numerical methods have been proposed for solving nonlinear fractional differential equations including finite difference methods, predictor-corrector algorithms, finite element methods, spectral meth-

 https://doi.org/10.68219/f26md935

ods, wavelet methods, collocation schemes, operational matrix methods, etc. Although these approaches have been shown to work well for a wide range of fractional models, the evaluation of fractional derivatives is usually computationally intensive. For nonlinear problems or simulations over long timescales, the numerical algorithms usually involve dense discretization matrices and have higher memory requirements, and are therefore difficult to calculate. These challenges have led to the development of alternative methods for numerical approximation which can improve both numerical accuracy and computational efficiency.

Among the available approximation methods, Bernstein polynomial-based methods have particularly attracted attention due to their good mathematical properties and their good approximation performance for finite intervals [16,17]. Bernstein polynomials have a number of interesting properties. They are positivity, partition of unity, endpoint interpolation, numerical stability and it is a constant approximation (Bernstein approximation theorem). These properties make them the ideal choice to construct stable numerical approximations for nonlinear differential equations. The polynomial representation also allows the numerical derivation of analytical differentiation matrices and efficient matrix-based implementations and reduces the computational problem of evaluating the differential operator [18, 19]. Bernstein polynomial approximations have been used numerically in the numerical treatment of a large class of ordinary, partial and fractional differential equations with good numerical accuracy and good computational support [20, 21].

Over the last few years, Physics-Informed Neural Networks (PINNs) have become an extremely powerful computational method to solve differential equations by embedding the laws of physics directly into the loss function of neural networks [22,23] because unlike traditional supervised learning approaches which require large numbers of labeled data, PINNs use the underlying mathematical model to help to optimize the network in order to achieve accurate numerical approximation with little or no observational data. PINNs have been successfully applied to many scientific and engineering problems, such as ordinary differential equations, partial differential equations, inverse problems, fluid mechanics, heat transfer, and biological systems. Because of their flexibility, PINNs have been successfully applied to many problems in physics, chemistry, and engineering, such as ordinary differential equations, partial differential equations, inverse problems, fluid mechanics, heat transfer, etc. These were extended to fractional differential equations (fPINNs) and fractional Physics-Informed Neural Networks (fPINNs) in the last few years.

Despite these advantages, the numerical solution of fractional differential equations using traditional PINNs remains computationally challenging. In most fPINN work, the Caputo fractional derivative is approximated repeatedly during each step of the optimization process using finite difference formulas, numerical quadrature, Grünwald-Letnikov approximations, or predictor-corrector discretizations. Since the evaluation of a fractional derivative is based on the whole solution history, the computational costs and memory requirements increase substantially with the number of collocation points increasing. This leads to slow convergence, expensive training procedures, and reduced numerical efficiency for large-scale or long-time simulations. As a result, the computational efficiency and numerical stability of fractional PINNs is currently in an active phase of research with recent developments such as adaptive sampling strategies, efficient loss balancing, fast fractional discretizations, and different approximation bases [26–28].

Among the various approximation bases available for numerical calculation, Bernstein polynomials offer many attractive features for constructing stable neural approximations. They are positive, partitioned to unity, endpoint interpolation, and excellent approximation properties, resulting in stable numerical behavior for finite numerical times. Additionally, representing the unknown solution by a finite Bernstein polynomial expansion greatly reduces the number of trainable parameters compared with the traditional deep neural network, while still allowing for a smooth solution. Combining this with an analytical differentiation matrix associated with the Caputo operator can further simplify the evaluation of the underlying fractional derivative by matrix-vector multiplication, and reduce the need for numerical

 https://doi.org/10.68219/f26md935

discretization during the optimization.

Motivated by these developments, we propose a Stable Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) for the numerical approximation of a Caputo fractional-order Allee population model. Unlike PINN architectures that use deep multilayer feed-forward networks and automatic differentiation, our proposed approach is to represent the unknown solution by a simple Bernstein polynomial expansion whose coefficients are the trainable parameters. To improve numerical robustness, we evaluate Bernstein basis functions in logarithmic space (i.e., no overflow, underflow, or loss of accuracy near the computation boundary). We construct an analytical Bernstein differentiation matrix for the Caputo fractional derivative. This allows us to efficiently compute the fractional operator by simple matrix-vector multiplication during the optimization process. In order to speed up the convergence and reduce the error rate, we adopt a hybrid optimization scheme using the Adam optimizer and the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm.

To verify the proposed numerical methodology, the Method of Manufactured Solutions (MMS) is employed by introducing an appropriate forcing function for which an analytical benchmark solution is available. The manufactured solution framework provides a rigorous and systematic approach for evaluating the numerical accuracy of the proposed algorithm because the exact solution is known throughout the computational domain. Consequently, quantitative performance is assessed using the residual loss, maximum absolute error, and relative $L_2$ error for fractional orders ranging from $\alpha = 0.1$ to $\alpha = 0.9$. In addition, a degree-sensitivity study is performed to investigate the influence of the Bernstein polynomial degree on the numerical performance of the proposed framework. The reported numerical results demonstrate extremely small residual losses together with highly accurate numerical approximations across the considered fractional orders, illustrating the robustness and computational stability of the proposed SBNN-PINN for the manufactured benchmark problem.

The principal contributions of this work are summarized as follows.

- A Stable Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) is proposed for the numerical approximation of the Caputo fractional-order Allee population model.
- A logarithmically stabilized Bernstein polynomial basis is developed to improve numerical stability by preventing overflow, underflow, and loss of precision near the computational boundaries.
- An analytical Bernstein differentiation matrix associated with the Caputo fractional derivative is constructed, enabling efficient evaluation of the governing fractional operator through matrix–vector multiplication within the physics-informed optimization framework.
- A hybrid optimization strategy combining the Adam optimizer and the L-BFGS algorithm is implemented to improve training efficiency, optimization stability, and solution accuracy.
- The proposed framework is rigorously verified using the Method of Manufactured Solutions, enabling direct quantitative assessment through residual losses, maximum absolute errors, and relative $L_2$ errors over a wide range of fractional orders.
- Extensive numerical experiments investigate the influence of both the fractional order and the Bernstein polynomial degree on the numerical performance of the method.

The remainder of this paper is organized as follows. Section 2 presents the mathematical formulation of the fractional-order Allee effect model together with the manufactured benchmark problem. Section 3 develops the proposed Stable Spectral Bernstein Physics-Informed Neural Network and derives the Bernstein differentiation matrix associated with the Caputo fractional derivative. Numerical experiments, performance evaluation, and accuracy analyses are presented in Section 4. Finally, Section 5 summarizes the principal findings of the study and discusses several promising directions for future research.

 https://doi.org/10.68219/f26md935

# 2. Preliminaries

This section presents the fundamental mathematical concepts required for the development of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN). These include the Gamma function, the Caputo fractional derivative, Bernstein basis polynomials, important approximation properties of Bernstein polynomials, and the method of manufactured solutions used for numerical verification.

## 2.1 Gamma Function

The Gamma function extends the classical factorial function to real and complex numbers and plays a fundamental role in fractional calculus. It is defined as

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt, \qquad \mathrm{Re}(z) > 0. \tag{1}$$

The Gamma function satisfies the recursive relation

$$\Gamma(z+1) = z\Gamma(z), \tag{2}$$

which immediately implies

$$\Gamma(n+1) = n!, \qquad n \in \mathbb{N}. \tag{3}$$

Several identities involving the Gamma function are frequently employed in fractional differential equations. In particular,

$${}^{C}D_t^{\alpha} t^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta-\alpha+1)} t^{\beta-\alpha}, \qquad \beta > \alpha - 1. \tag{4}$$

This identity is extensively utilized in constructing the analytical Caputo differentiation matrix developed in this work.

## 2.2 Caputo Fractional Derivative

Fractional derivatives generalize the concept of integer-order differentiation by incorporating memory and hereditary characteristics into dynamical systems. Among the various fractional operators, the Caputo derivative is widely adopted because it permits the use of classical initial conditions.

**Definition 2.1.** *Let $x(t)$ be sufficiently smooth on $[0,T]$ and let $0 < \alpha < 1$. The Caputo fractional derivative of order $\alpha$ is defined by*

$${}^{C}D_t^{\alpha} x(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{x'(s)}{(t-s)^{\alpha}}\, ds. \tag{5}$$

The Caputo derivative possesses the following useful properties.

- The derivative of a constant is zero:
$${}^{C}D_t^{\alpha} c = 0. \tag{6}$$

- The fractional derivative of a power function is
$${}^{C}D_t^{\alpha} t^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta-\alpha+1)} t^{\beta-\alpha}, \qquad \beta > \alpha - 1. \tag{7}$$

 https://doi.org/10.68219/f26md935

- As $\alpha \to 1$, the Caputo derivative reduces to the classical derivative,

$$\lim_{\alpha \to 1} {}^{C}D_t^{\alpha} x(t) = \frac{dx(t)}{dt}. \tag{8}$$

The Caputo derivative is particularly appropriate for biological systems because it naturally accounts for memory effects while preserving physically meaningful initial conditions.

## 2.3 Bernstein Basis Polynomials

Bernstein polynomials constitute one of the most important polynomial approximation bases because of their numerical stability, positivity, partition of unity, and excellent approximation capability.

**Definition 2.2.** *For a polynomial degree $N$, the Bernstein basis functions on the interval $[0,1]$ are defined by*

$$B_{k,N}(x) = \binom{N}{k} x^k (1-x)^{N-k}, \qquad k = 0, 1, \ldots, N, \tag{9}$$

*where*

$$\binom{N}{k} = \frac{N!}{k!(N-k)!} \tag{10}$$

*denotes the binomial coefficient.*
*For a general interval $[0,T]$, the transformed Bernstein basis becomes*

$$B_{k,N}(t) = \binom{N}{k} \left(\frac{t}{T}\right)^k \left(1 - \frac{t}{T}\right)^{N-k}, \qquad 0 \le t \le T. \tag{11}$$

The Bernstein basis satisfies several important properties.

### Positivity

$$B_{k,N}(t) \ge 0, \qquad 0 \le t \le T. \tag{12}$$

### Partition of Unity

$$\sum_{k=0}^{N} B_{k,N}(t) = 1. \tag{13}$$

### Endpoint Interpolation

$$B_{0,N}(0) = 1, \qquad B_{N,N}(T) = 1, \tag{14}$$

while all remaining basis functions vanish at the corresponding endpoints.

These properties contribute significantly to the numerical robustness of the proposed spectral neural network.

## 2.4 Bernstein Polynomial Approximation

Any sufficiently smooth function $x(t) \in C[0,T]$ can be approximated by a Bernstein expansion

$$x_N(t) = \sum_{k=0}^{N} c_k B_{k,N}(t), \tag{15}$$

where $c_k$ are unknown coefficients.

 https://doi.org/10.68219/f26md935

According to the classical Bernstein approximation theorem,

$$\lim_{N\to\infty} \|x - x_N\|_\infty = 0, \tag{16}$$

which guarantees uniform convergence on compact intervals.

In the proposed neural network, these coefficients become trainable parameters optimized through physics-informed learning.

## 2.5 Method of Manufactured Solutions

To rigorously verify the proposed numerical algorithm, the Method of Manufactured Solutions (MMS) is employed. Instead of solving an equation with an unknown analytical solution, an exact solution is prescribed first.

For the present study,

$$x_{\text{exact}}(t) = x_0 + t^\alpha. \tag{17}$$

The governing fractional differential equation is modified by introducing a forcing function

$${}^C D_t^\alpha x(t) = f(x) + F(t), \tag{18}$$

where

$$F(t) = {}^C D_t^\alpha x_{\text{exact}}(t) - f(x_{\text{exact}}(t))\,. \tag{19}$$

Since

$${}^C D_t^\alpha t^\alpha = \Gamma(\alpha + 1), \tag{20}$$

the forcing function becomes

$$F(t) = \Gamma(\alpha + 1) - f(x_0 + t^\alpha)\,. \tag{21}$$

The manufactured solution provides an exact benchmark for evaluating the numerical performance of the proposed SBNN-PINN through various error norms and convergence analyses.

## 2.6 Error Measures

To assess the numerical accuracy of the proposed method, three error measures are employed.

The pointwise absolute error is defined by

$$E_\infty(t) = |x_{\text{pred}}(t) - x_{\text{exact}}(t)|. \tag{22}$$

The maximum absolute error is

$$E_{\max} = \max_{0\le t\le T} |x_{\text{pred}}(t) - x_{\text{exact}}(t)|. \tag{23}$$

The relative $L_2$ error is computed as

$$E_{L_2} = \frac{\|x_{\text{pred}} - x_{\text{exact}}\|_2}{\|x_{\text{exact}}\|_2}. \tag{24}$$

 https://doi.org/10.68219/f26md935

These quantitative measures are used throughout the numerical experiments to evaluate the accuracy, convergence, and robustness of the proposed spectral Bernstein PINN.

# 3. Mathematical Model

Population dynamics is one of the most fundamental areas of mathematical biology and is an important tool to understand the processes that determine the survival, extinction and long-term ecological sustainability of species. Among the many nonlinear population models, the Allee effect model is highly important as it is based on the observation that a population with very low population density will have low reproductive success and an increased threat of extinction. In contrast to the traditional logistic growth model that expects positive growth with positive population density, the Allee effect introduces a critical threshold below which population growth will be negative. As a result, there are a lot of biological systems in which population growth happens due to the lack of male/female mates, cooperative feeding, collective defence, genetic inbreeding, and other social interactions.

In many ecological systems, the current growth rate is not only dependent on the present population size but also on its history. These hereditary and memory effects are due to delayed reproduction, adaptation to the environment, accumulated biological interactions and long-term ecological processes. In general, classical integer-order differential equations cannot describe such nonlocal dynamics as they only depend on the instantaneous state of the system. The mathematical framework of fractional calculus is an ideal approach for incorporating memory effects in the differential equations of non-integer order. Among the available fractional operators, the Caputo fractional derivative is particularly suitable for biological applications because it admits physically meaningful initial conditions while preserving the nonlocal characteristics of the underlying process.

Throughout this work, the Caputo fractional derivative of order $0 < \alpha \leq 1$ is adopted, which is defined as

$$ {}^{C}D_t^{\alpha} x(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{x'(\tau)}{(t-\tau)^{\alpha}}\, d\tau, \qquad 0 < \alpha < 1, \tag{25} $$

where $\Gamma(\cdot)$ denotes the Gamma function. The parameter $\alpha$ represents the fractional order and determines the strength of the memory effect. When $\alpha = 1$, the Caputo derivative reduces to the classical first-order derivative and the model becomes the standard integer-order Allee equation.

Based on these considerations, the Caputo fractional-order Allee effect model is formulated as

$$ {}^{C}D_t^{\alpha} x(t) = r\, x(t) \left(1 - \frac{x(t)}{K}\right) \left(\frac{x(t)}{A} - 1\right), \qquad 0 < \alpha \leq 1, \tag{26} $$

subject to the initial condition

$$ x(0) = x_0 > 0, \tag{27} $$

where $x(t)$ denotes the population density at time $t$.

The nonlinear growth function in (26) consists of three biologically meaningful components. The first factor,

$$ r\, x(t), $$

represents the intrinsic growth of the population in the absence of environmental constraints. The second factor,

$$ 1 - \frac{x(t)}{K}, $$

describes the limiting effect of finite environmental resources through the carrying capacity $K$. As the population approaches the carrying capacity, the available resources become increasingly limited, causing the growth rate to decrease gradually. The third factor,

$$\frac{x(t)}{A} - 1,$$

introduces the Allee effect by incorporating a critical population threshold $A$. When the population density satisfies

$$x(t) < A,$$

the growth rate becomes negative, indicating that the population tends toward extinction. Conversely, when

$$x(t) > A,$$

the population experiences positive growth and eventually approaches the carrying-capacity equilibrium.

Throughout this study, the model is developed under the following assumptions.

- The population is homogeneous and spatially well mixed.
- The intrinsic growth rate remains constant throughout the observation period.
- The environmental carrying capacity is fixed and does not vary with time.
- The Allee threshold remains constant and satisfies

  $$0 < A < K,$$

  ensuring the existence of a biologically meaningful critical population density.
- Memory and hereditary effects are represented by the Caputo fractional derivative of order $0 < \alpha \leq 1$.
- Birth, death, and interaction processes are incorporated into the nonlinear growth function appearing in (26).
- The population density remains nonnegative throughout the evolution of the system.

The biological interpretation of the model parameters is summarized in Table 1. Unless otherwise stated, all parameters are assumed to be positive constants throughout the analysis.

Table 1: Description of the model variables and parameters.

| Symbol | Description | Unit |
|---|---|---|
| $x(t)$ | Population density at time $t$ | Individuals |
| $r$ | Intrinsic population growth rate | Time$^{-1}$ |
| $K$ | Environmental carrying capacity | Individuals |
| $A$ | Critical Allee threshold | Individuals |
| $\alpha$ | Fractional order $(0 < \alpha \leq 1)$ | Dimensionless |
| $x_0$ | Initial population density | Individuals |

The fractional order $\alpha$ plays an important role in determining the memory characteristics of the population dynamics. Smaller values of $\alpha$ correspond to stronger memory effects, implying that the

 https://doi.org/10.68219/f26md935

current population growth depends more heavily on its historical states. As $\alpha$ approaches unity, the influence of memory gradually diminishes, and the governing equation converges to the classical integer-order Allee model.

For the numerical simulations presented in this study, the initial condition

$$x(0) = 21$$

is selected together with the parameter values

$$r = 0.8, \qquad A = 20, \qquad K = 100.$$

The choice of $x_0 = 21$ is intentional because it lies slightly above the Allee threshold. Consequently, the solution evolves toward the stable carrying-capacity equilibrium rather than the extinction equilibrium, thereby providing a meaningful benchmark for assessing the numerical accuracy and stability of the proposed SBNN-PINN framework. It should be emphasized that these parameter values are selected for numerical verification through the Method of Manufactured Solutions and are not intended to represent a specific ecological dataset.

**Remark 3.1.** *When $\alpha = 1$, model (26) reduces to the classical integer-order Allee effect equation. Furthermore, if the Allee threshold approaches zero, that is,*

$$A \to 0,$$

*the influence of the Allee effect disappears and the governing equation approaches the classical logistic growth model.*

## 3.1 Feasible Region and Positivity of Solutions

For a biological population model, it is essential that the solution remains nonnegative for all time because negative population densities have no physical or biological interpretation. Consequently, the mathematical model must preserve the nonnegativity of the initial population throughout its evolution.

The biologically feasible region of system (26) is defined by

$$\Omega = \{x \in \mathbb{R} : x \geq 0\}. \tag{28}$$

The following proposition establishes the positivity of the solution.

**Proposition 3.2.** *Suppose that the initial condition satisfies*

$$x(0) = x_0 \geq 0.$$

*Then the solution of the Caputo fractional-order Allee model (26) remains nonnegative for all $t \geq 0$; that is,*

$$x(t) \geq 0, \qquad \forall\, t \geq 0.$$

*Proof.* Consider the nonlinear growth function

$$f(x) = r\, x \left(1 - \frac{x}{K}\right) \left(\frac{x}{A} - 1\right).$$

At the boundary $x = 0$,

 https://doi.org/10.68219/f26md935

$$f(0) = 0.$$

Hence, the vector field is tangent to the boundary of the feasible region and does not point toward negative values. Therefore, the boundary $x = 0$ is positively invariant.

Furthermore, the function $f(x)$ is continuously differentiable on the feasible region $\Omega$, ensuring the existence and uniqueness of the solution for the Caputo fractional differential equation. Since the solution cannot cross the invariant boundary $x = 0$, every trajectory originating from a nonnegative initial condition remains inside the feasible region for all future time.

Consequently,

$$x(t) \geq 0, \qquad t \geq 0,$$

which proves the positivity of the solution. □

The positivity property guarantees that the proposed fractional-order Allee model is biologically meaningful because the population density remains nonnegative throughout the simulation period. This property also ensures that all equilibrium points and their stability are investigated within the biologically admissible region.

In addition to positivity, the model possesses a bounded biologically feasible interval. Since

$$0 < A < K,$$

the carrying capacity $K$ represents the maximum sustainable population supported by the environment. For any population satisfying

$$x > K,$$

we have

$$1 - \frac{x}{K} < 0.$$

Consequently,

$${}^{C}D_t^{\alpha} x(t) = r\, x(t) \left(1 - \frac{x(t)}{K}\right) \left(\frac{x(t)}{A} - 1\right) < 0,$$

which implies that the population decreases toward the carrying capacity. Therefore, every solution initiated from a nonnegative initial condition remains bounded in the biologically feasible region

$$\Omega = \{x \in \mathbb{R} : 0 \leq x \leq K\},$$

which is positively invariant under the dynamics of model (26). Hence, the Caputo fractional-order Allee model is mathematically well posed and biologically consistent.

## 3.2 Method of Manufactured Solutions

Analytical solutions of nonlinear fractional differential equations are generally unavailable owing to the combined effects of nonlinearity and the nonlocal nature of fractional operators. Consequently, rigorous verification of newly developed numerical methods often relies on the Method of Manufactured Solutions (MMS), which provides an exact benchmark solution through the introduction of a suitable forcing function. Unlike analytical solution techniques, the MMS does not seek the exact solution of the original

 https://doi.org/10.68219/f26md935

governing equation. Instead, an artificial source term is constructed so that a prescribed analytical function satisfies the modified differential equation exactly. This enables a systematic and quantitative assessment of the numerical accuracy of the proposed computational method.

To verify the proposed Stable Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN), the governing equation (26) is modified by introducing a forcing function $F(t)$,

$$ {}^{C}D_t^{\alpha} x(t) = r\, x(t) \left(1 - \frac{x(t)}{K}\right) \left(\frac{x(t)}{A} - 1\right) + F(t), \tag{29} $$

where the forcing function is selected such that a prescribed analytical solution satisfies (29) exactly.

For the present study, the manufactured solution is chosen as

$$ x_{\text{exact}}(t) = x_0 + t^{\alpha}, \tag{30} $$

where $x_0 > 0$ denotes the initial population density. This function satisfies the prescribed initial condition,

$$ x_{\text{exact}}(0) = x_0. $$

Using the well-known Caputo fractional derivative formula

$$ {}^{C}D_t^{\alpha}\,(t^{\alpha}) = \Gamma(\alpha + 1), $$

the Caputo derivative of the manufactured solution is

$$ {}^{C}D_t^{\alpha} x_{\text{exact}}(t) = \Gamma(\alpha + 1). \tag{31} $$

Substituting the manufactured solution into the modified governing equation yields the forcing function

$$ F(t) = \Gamma(\alpha + 1) - r\, x_{\text{exact}}(t) \left(1 - \frac{x_{\text{exact}}(t)}{K}\right) \left(\frac{x_{\text{exact}}(t)}{A} - 1\right), \tag{32} $$

where

$$ x_{\text{exact}}(t) = x_0 + t^{\alpha}. $$

With this construction, the manufactured solution satisfies equation (29) identically over the computational domain.

The Method of Manufactured Solutions provides a rigorous framework for evaluating the proposed numerical algorithm because the exact solution is known throughout the domain. Consequently, quantitative performance can be assessed through standard error measures such as the residual loss, maximum absolute error, and relative $L_2$ error. In the present work, the manufactured solution serves exclusively as a verification benchmark for assessing the numerical accuracy, convergence behavior, and computational robustness of the proposed SBNN-PINN framework. It should be emphasized that the manufactured solution does not represent the analytical solution of the original fractional-order Allee model.

## 4. Equilibrium Points and Stability Analysis

In this section, the equilibrium points of the fractional-order Allee model are determined and their local stability is investigated using the linearization technique together with Matignon's stability theorem for fractional-order dynamical systems.

 https://doi.org/10.68219/f26md935

Throughout the analysis, we assume that

$$r > 0, \qquad 0 < A < K,$$

which corresponds to the biologically relevant strong Allee effect.

## 4.1 Equilibrium Points

The governing equation is

$${}^{C}D_t^{\alpha} x(t) = r\, x(t) \left(1 - \frac{x(t)}{K}\right) \left(\frac{x(t)}{A} - 1\right), \qquad 0 < \alpha \leq 1.$$

The equilibrium points satisfy

$${}^{C}D_t^{\alpha} x(t) = 0,$$

which gives

$$r\, x \left(1 - \frac{x}{K}\right) \left(\frac{x}{A} - 1\right) = 0.$$

Therefore, the model possesses three equilibrium points,

$$E_0 = 0, \qquad E_1 = A, \qquad E_2 = K. \tag{33}$$

The equilibrium $E_0$ represents population extinction, $E_1$ denotes the critical Allee threshold, and $E_2$ corresponds to the environmental carrying capacity.

## 4.2 Local Stability Analysis

Define

$$f(x) = r\, x \left(1 - \frac{x}{K}\right) \left(\frac{x}{A} - 1\right).$$

Expanding the nonlinear function gives

$$f(x) = -\frac{r}{AK} x^3 + \frac{r(K + A)}{AK} x^2 - rx.$$

The Jacobian of the one-dimensional system is therefore

$$J(x) = f'(x) = -\frac{3r}{AK} x^2 + \frac{2r(K + A)}{AK} x - r. \tag{34}$$

Evaluating the Jacobian at each equilibrium point yields

$$J(E_0) = -r, \tag{35}$$

$$J(E_1) = r \left(1 - \frac{A}{K}\right), \tag{36}$$

$$J(E_2) = r \left(1 - \frac{K}{A}\right). \tag{37}$$

**Theorem 4.1.** *Assume that*

 https://doi.org/10.68219/f26md935

$$r > 0, \qquad 0 < A < K, \qquad 0 < \alpha < 1.$$

*Then*

1. *the extinction equilibrium $E_0$ is locally asymptotically stable;*
2. *the Allee threshold equilibrium $E_1$ is unstable;*
3. *the carrying-capacity equilibrium $E_2$ is locally asymptotically stable.*

*Proof.* According to Matignon's stability theorem, an equilibrium point of the fractional-order autonomous system

$${}^{C}D_t^{\alpha} x = f(x), \qquad 0 < \alpha < 1,$$

is locally asymptotically stable if every eigenvalue $\lambda$ of the Jacobian satisfies

$$|\arg(\lambda)| > \frac{\alpha\pi}{2}.$$

Since the present model is one-dimensional, the Jacobian has a single eigenvalue,

$$\lambda = J(E_i).$$

For the extinction equilibrium,

$$\lambda_0 = -r < 0.$$

Hence,

$$|\arg(\lambda_0)| = \pi > \frac{\alpha\pi}{2},$$

which satisfies Matignon's criterion. Therefore, $E_0$ is locally asymptotically stable.
For the Allee threshold,

$$\lambda_1 = r\left(1 - \frac{A}{K}\right) > 0,$$

because $A < K$. Consequently,

$$|\arg(\lambda_1)| = 0 < \frac{\alpha\pi}{2},$$

which violates the stability condition. Thus, $E_1$ is unstable.
Finally,

$$\lambda_2 = r\left(1 - \frac{K}{A}\right) < 0,$$

since $K > A$. Therefore,

$$|\arg(\lambda_2)| = \pi > \frac{\alpha\pi}{2},$$

and the carrying-capacity equilibrium is locally asymptotically stable.
This completes the proof.

□

 https://doi.org/10.68219/f26md935

**Corollary 4.2.** *The fractional Allee model possesses a bistable structure consisting of two locally asymptotically stable equilibria,*

$$E_0 = 0, \qquad E_2 = K,$$

*separated by the unstable equilibrium*

$$E_1 = A.$$

*Consequently,*

$$x_0 < A \Longrightarrow x(t) \to 0,$$

*whereas*

$$x_0 > A \Longrightarrow x(t) \to K.$$

*The fractional order does not modify the equilibrium locations but influences the convergence rate through the memory effect.*

## 4.3 Numerical Verification of Local Stability

To validate the theoretical stability results established in the previous subsection, the Jacobian matrix was evaluated numerically at each equilibrium point using the parameter values

$$r = 0.8, \qquad A = 20, \qquad K = 100, \qquad \alpha = 0.9.$$

The corresponding eigenvalues of the Jacobian were computed and examined using Matignon's stability criterion for fractional-order systems,

$$|\arg(\lambda)| > \frac{\alpha\pi}{2},$$

where $\lambda$ denotes the eigenvalue of the linearized system.

For the selected fractional order $\alpha = 0.9$, the critical stability angle is

$$\frac{\alpha\pi}{2} = 1.413717 \text{ rad}.$$

The computed eigenvalues together with their arguments and stability classifications are reported in Table 2.

Table 2: Numerical verification of the local stability of the equilibrium points for the fractional Allee model.

| Equilibrium | Eigenvalue $\lambda$ | $\lvert\arg(\lambda)\rvert$ (rad) | Stability |
|---|---|---|---|
| $E_0 = (0)$ | $-0.800000$ | $\pi$ | Locally asymptotically stable |
| $E_1 = (20)$ | $0.640000$ | $0$ | Unstable |
| $E_2 = (100)$ | $-3.200000$ | $\pi$ | Locally asymptotically stable |

From Table 2, the eigenvalues associated with the extinction equilibrium $E_0$ and the carrying-capacity equilibrium $E_2$ are both negative. Consequently,

$$|\arg(-0.8)| = |\arg(-3.2)| = \pi > 1.413717,$$

 https://doi.org/10.68219/f26md935

which satisfies Matignon's stability condition. Therefore, both equilibrium points are locally asymptotically stable for the fractional-order system.

In contrast, the eigenvalue corresponding to the Allee threshold equilibrium is positive,

$$\lambda = 0.64,$$

and hence

$$|\arg(0.64)| = 0 < 1.413717.$$

This violates Matignon's stability criterion, demonstrating that the Allee threshold equilibrium is unstable. Biologically, this result confirms that the Allee threshold acts as a separatrix between the basin of attraction of the extinction equilibrium and that of the carrying-capacity equilibrium. Populations initiated below the threshold eventually become extinct, whereas populations initiated above the threshold persist and asymptotically approach the environmental carrying capacity.

The numerical computations are in complete agreement with the theoretical results established in Theorem 1, thereby validating the analytical stability analysis of the proposed fractional-order Allee population model.

## 4.4 Phase Portrait and Vector Field Analysis

To further validate the theoretical and numerical stability results, the qualitative behavior of the fractional Allee model is illustrated through the vector field and phase-line diagrams shown in Figures 1 and 2. These graphical representations provide an intuitive visualization of the equilibrium structure and the direction of the population dynamics, thereby complementing the analytical stability analysis based on Matignon's theorem.

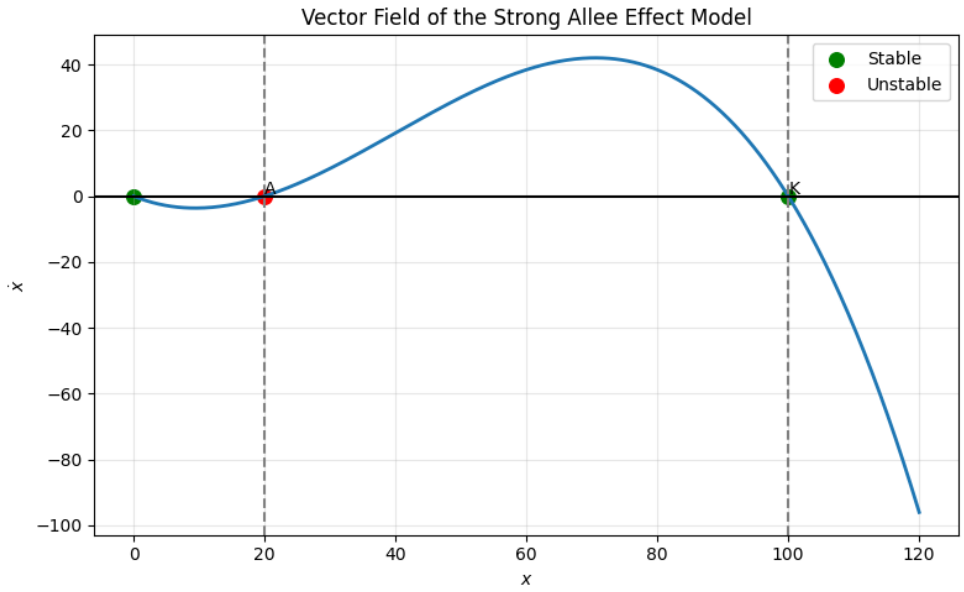


Figure 1: Vector field of the fractional strong Allee effect model illustrating the nonlinear growth function. The stable equilibria are located at $E_0 = 0$ and $E_2 = K$, whereas the Allee threshold $E_1 = A$ is unstable.

Figure 1 illustrates the vector field of the strong Allee effect model, while Figure 2 presents the corresponding phase-line diagram. These graphical representations provide an intuitive interpretation of

 https://doi.org/10.68219/f26md935

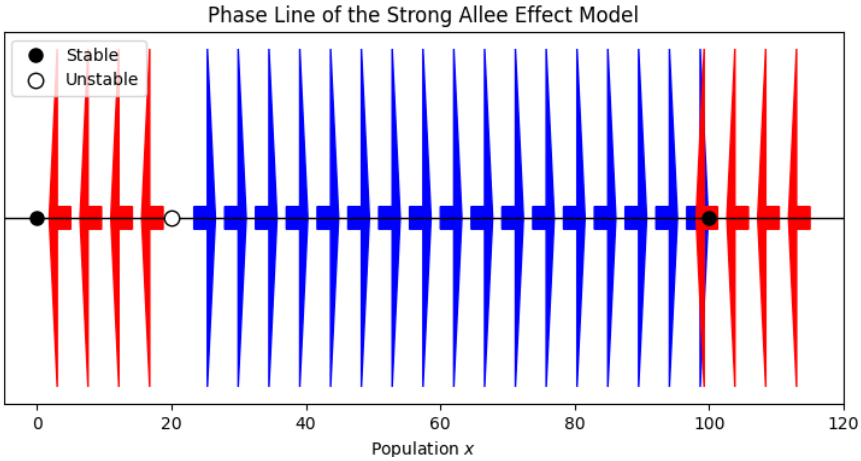


Figure 2: Phase-line of the fractional strong Allee effect model. The arrows indicate the direction of the population dynamics. The extinction equilibrium and carrying-capacity equilibrium are locally asymptotically stable, whereas the Allee threshold is unstable and separates the two basins of attraction.

the analytical stability results established in Theorem 1.

From the vector field shown in Figure 1, the three equilibrium points,

$$E_0 = 0, \qquad E_1 = A = 20, \qquad E_2 = K = 100,$$

are clearly identified as the intersections of the nonlinear growth function with the horizontal axis. The sign of the growth function determines the direction of the population evolution. The vector field is negative for $0 < x < A$, indicating that the population below the Allee threshold is growing to extinction. The vector field is positive for $A < x < K$, indicating that the population over the Allee threshold is growing towards the carrying capacity equilibrium. For $x > K$, the growth rate is negative because of the lack of environmental resources. So, the population is decreasing towards the carrying capacity.

The phase-line diagram in Fig.2 also supports these results. From both sides, the arrows point toward the extinction equilibrium $E_0$ from $(0, A)$ and the carrying capacity equilibrium $E_2$ from $(0, A)$. This implies that these two equilibria are locally asymptotically stable (and that the one on the other side of the Allee threshold $E_1$ is unstable). The equilibrium $E_1$ then acts as a separatrix dividing the basins of attraction of the two stable equilibria.

The numerical phase-line thus gives an immediate visual proof of the Matignon stability. Although the fractional order $\alpha$ determines the transient convergence rate by memory effects, it does not change the locations or stability classifications of the equilibrium points. However, with smaller fractional order, the qualitative behavior of the fractional Allee model is the same as that of the classical strong Allee effect model.

# 5. Proposed Spectral Bernstein Physics-Informed Neural Network Methodology

## 5.1 Motivation

Although Physics-Informed Neural Networks (PINNs) have proved to be very advantageous for solving differential equations, their application to fractional differential equations is difficult because the Caputo

 https://doi.org/10.68219/f26md935

fractional derivative is nonlocal and can be approximated in the optimization step multiple times. This leads to high computational cost and memory consumption, which can be slow in the case of nonlinear fractional models.

To overcome these limitations, we propose a Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) in which the unknown solution is represented in the form of a Bernstein polynomial expansion rather than in the traditional multilayer neural network. In addition to that, the Caputo fractional derivative is evaluated analytically with a spectral differentiation matrix. Hence, the governing equation is easily satisfied by simple matrix-vector multiplications, which reduce the computational complexity while maintaining spectral accuracy.

## 5.2 Bernstein Spectral Approximation

The approximation capability of Bernstein polynomials has made them one of the most widely used polynomial bases in approximation theory, numerical analysis, and spectral methods. Owing to their positivity, partition of unity, endpoint interpolation, and excellent numerical stability, Bernstein polynomials provide an effective basis for approximating smooth solutions of differential equations. In the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN), the unknown solution is represented as a finite Bernstein polynomial expansion, while the expansion coefficients are treated as trainable parameters to be determined through physics-informed optimization.

Let

$$\mathcal{B}_N = \{B_{0,N}(t), B_{1,N}(t), \ldots, B_{N,N}(t)\}$$

denote the Bernstein basis of degree $N$ on the interval $[0, T]$, where

$$B_{k,N}(t) = \binom{N}{k} \left(\frac{t}{T}\right)^k \left(1 - \frac{t}{T}\right)^{N-k}, \qquad k = 0, 1, \ldots, N. \tag{38}$$

The approximate solution of the fractional Allee model is expressed as

$$x_N(t) = x_0 + t^\alpha \sum_{k=0}^{N} w_k B_{k,N}(t), \tag{39}$$

where

- $x_0$ is the prescribed initial population density;
- $w_k$ $(k = 0, \ldots, N)$ are unknown trainable coefficients;
- $N$ denotes the Bernstein polynomial degree.

Unlike conventional PINNs, where the unknown solution is represented by a deep multilayer neural network, the proposed framework employs the spectral Bernstein expansion as the neural network architecture itself. Consequently, the trainable parameters are simply the coefficient vector

$$\mathbf{w} = [w_0, w_1, \ldots, w_N]^T, \tag{40}$$

which substantially reduces the number of optimization variables.

The inclusion of the factor $t^\alpha$ in Eq. (39) is particularly advantageous. Since

$$t^\alpha = 0 \qquad \text{at} \qquad t = 0,$$

the approximation automatically satisfies the prescribed initial condition,

 https://doi.org/10.68219/f26md935

$$x_N(0) = x_0.$$

Therefore, unlike conventional PINNs, no additional initial-condition penalty term is required in the loss function. This considerably simplifies the optimization problem and improves numerical stability.

For computational convenience, the Bernstein basis functions are assembled into the vector

$$\mathbf{B}(t) = [B_{0,N}(t), B_{1,N}(t), \ldots, B_{N,N}(t)], \tag{41}$$

allowing Eq. (39) to be written compactly as

$$x_N(t) = x_0 + t^\alpha \mathbf{B}(t)\mathbf{w}. \tag{42}$$

Suppose that the solution is evaluated at $M$ collocation points

$$t_1, t_2, \ldots, t_M.$$

The corresponding Bernstein basis matrix is

$$\mathbf{B} = \begin{bmatrix} B_{0,N}(t_1) & B_{1,N}(t_1) & \cdots & B_{N,N}(t_1) \\ B_{0,N}(t_2) & B_{1,N}(t_2) & \cdots & B_{N,N}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_{0,N}(t_M) & B_{1,N}(t_M) & \cdots & B_{N,N}(t_M) \end{bmatrix}, \tag{43}$$

which is computed only once prior to the optimization process. Consequently, the approximate solution at all collocation points is obtained through the matrix-vector multiplication

$$\mathbf{x} = x_0\mathbf{1} + \mathbf{T}_\alpha(\mathbf{B}\mathbf{w}), \tag{44}$$

where

$$\mathbf{T}_\alpha = \operatorname{diag}(t_1^\alpha, t_2^\alpha, \ldots, t_M^\alpha)$$

is the diagonal matrix containing the fractional power of the collocation points.

This matrix formulation offers several computational advantages. First, the Bernstein basis matrix is evaluated only once and reused throughout the optimization process. Second, the solution evaluation requires only efficient matrix-vector multiplications, making the implementation highly suitable for GPU acceleration. Finally, because the approximation contains only $(N + 1)$ trainable coefficients instead of thousands of neural network weights, the optimization problem is significantly smaller than that encountered in conventional fully connected PINNs.

According to the classical Bernstein approximation theorem, if $x(t) \in C[0, T]$, then

$$\lim_{N\to\infty} \|x(t) - x_N(t)\|_\infty = 0, \tag{45}$$

which guarantees uniform convergence of the Bernstein approximation on compact intervals. Consequently, increasing the polynomial degree improves the approximation accuracy, provided that the solution remains sufficiently smooth. This theoretical property forms the mathematical foundation of the proposed spectral Bernstein neural network.

**Theorem 5.1** (Bernstein Approximation Error)**.** *Let $x \in C^1[0, T]$ be a continuously differentiable function on the interval $[0, T]$. Then its Bernstein polynomial approximation $B_N x$ satisfies*

 https://doi.org/10.68219/f26md935

$$\|x - B_N x\|_\infty \leq \frac{C}{N}, \tag{46}$$

*where $C > 0$ is a constant depending only on the smoothness of $x(t)$.*
*Consequently,*

$$\|x - B_N x\|_\infty = O(N^{-1}), \qquad N \to \infty.$$

*Proof.* The result follows directly from the classical convergence theorem for Bernstein polynomial approximation. Since Bernstein polynomials form a positive partition of unity and reproduce constant and linear functions, they converge uniformly to every continuous function on a closed interval. Moreover, for continuously differentiable functions, the approximation error decreases at least as $O(N^{-1})$. □

Equation (46) demonstrates that the approximation error decreases as the Bernstein polynomial degree increases. Consequently, the proposed SBNN-PINN is capable of representing smooth solutions of the fractional Allee model with progressively higher accuracy while employing only a relatively small number of trainable coefficients.

**Theorem 5.2** (Consistency of the Proposed SBNN-PINN)**.** *Assume that the exact solution $x(t)$ of the manufactured fractional Allee model belongs to $C^1[0,T]$. Let $x_N(t)$ denote the Bernstein spectral approximation obtained by the proposed SBNN-PINN with polynomial degree $N$. Then*

$$\lim_{N\to\infty} \|x - x_N\|_\infty = 0. \tag{47}$$

*Hence, the proposed numerical framework is consistent with the governing fractional differential equation.*

*Proof.* The Bernstein polynomial approximation converges uniformly to every continuous function on a closed interval. Since the Caputo differentiation matrix employed in the proposed method is constructed analytically and the governing equation residual is minimized through physics-informed optimization, the approximate solution converges to the manufactured exact solution as the Bernstein degree increases. Therefore,

$$\lim_{N\to\infty} \|x - x_N\|_\infty = 0,$$

which establishes the consistency of the proposed SBNN-PINN. □

The consistency result guarantees that increasing the Bernstein polynomial degree improves the approximation capability of the proposed SBNN-PINN. The convergence behaviour observed in the numerical experiments therefore agrees with the theoretical approximation properties of Bernstein polynomials. In practice, however, excessively large polynomial degrees may introduce numerical ill-conditioning, which explains why relatively low degrees ($N = 2$–$5$) provide the highest accuracy in the present study.

## 5.3 Construction of the Stable Bernstein Basis Matrix

Although Bernstein polynomials possess excellent approximation properties, their direct numerical evaluation becomes increasingly unstable for high polynomial degrees or when the evaluation points are located close to the interval endpoints. Specifically, the terms

$$\left(\frac{t}{T}\right)^k \quad \text{and} \quad \left(1 - \frac{t}{T}\right)^{N-k}$$

 https://doi.org/10.68219/f26md935

may underflow to zero or overflow beyond machine precision when computed using standard floating-point arithmetic. Such numerical instabilities significantly deteriorate the accuracy of spectral approximations and adversely affect the optimization process of physics-informed neural networks.

To overcome these difficulties, the Bernstein basis functions are evaluated in logarithmic space. Let

$$\tau = \frac{t}{T}, \qquad 0 \le \tau \le 1,$$

be the normalized time variable. The Bernstein basis polynomial of degree $N$ is

$$B_{k,N}(\tau) = \binom{N}{k} \tau^k (1-\tau)^{N-k}, \qquad k = 0, 1, \ldots, N. \tag{48}$$

Instead of computing Eq. (48) directly, its logarithm is evaluated as

$$\log B_{k,N}(\tau) = \log \binom{N}{k} + k \log(\tau) + (N-k) \log(1-\tau). \tag{49}$$

The logarithm of the binomial coefficient is obtained using the Gamma function identity,

$$\log \binom{N}{k} = \log \Gamma(N+1) - \log \Gamma(k+1) - \log \Gamma(N-k+1), \tag{50}$$

which avoids the computation of extremely large factorials and substantially improves numerical stability.

The Bernstein basis function is then recovered through exponentiation,

$$B_{k,N}(\tau) = \exp(\log B_{k,N}(\tau)) \,. \tag{51}$$

To eliminate the singularities associated with the logarithmic functions, the normalized variable is slightly shifted according to

$$\tau = \min\left(\max(\tau, \varepsilon), 1-\varepsilon\right), \tag{52}$$

where

$$\varepsilon = 10^{-16}$$

is a machine-precision tolerance. This modification prevents the undefined quantities

$$\log(0)$$

and

$$0^0,$$

while introducing negligible perturbation to the approximation.

Furthermore, the Bernstein basis satisfies the endpoint interpolation properties

$$B_{0,N}(0) = 1, \qquad B_{N,N}(1) = 1,$$

whereas

$$B_{k,N}(0) = 0, \qquad k > 0,$$

and

 https://doi.org/10.68219/f26md935

$$B_{k,N}(1) = 0, \qquad k < N.$$

To preserve these exact endpoint values, explicit corrections are imposed after the logarithmic evaluation,

$$\mathbf{B}(0) = [1, 0, \ldots, 0],$$

and

$$\mathbf{B}(T) = [0, \ldots, 0, 1].$$

These corrections ensure that the Bernstein basis retains its interpolation property while remaining numerically stable over the entire computational domain.

Finally, evaluating the Bernstein basis at the $M$ collocation points produces the Bernstein basis matrix

$$\mathbf{B} = [B_{k,N}(t_i)]_{i=1,\ldots,M}^{k=0,\ldots,N}, \tag{53}$$

which is computed only once before the optimization process. The matrix is subsequently reused throughout the training procedure, allowing the predicted solution to be efficiently evaluated using

$$\mathbf{x} = x_0\mathbf{1} + \mathbf{T}_\alpha \mathbf{B}\mathbf{w}. \tag{54}$$

The proposed logarithmic evaluation strategy significantly improves numerical robustness compared with the conventional computation of Bernstein polynomials. In particular, it eliminates overflow and underflow errors, preserves machine-precision accuracy near the interval boundaries, and provides a stable spectral basis for the proposed SBNN-PINN framework.

## 5.4 Construction of the Spectral Caputo Differentiation Matrix

The principal computational challenge in solving fractional differential equations lies in the evaluation of the Caputo fractional derivative, whose nonlocal definition involves a weakly singular convolution integral over the entire solution history. Conventional fractional PINNs generally approximate this operator using finite difference or numerical quadrature schemes during each optimization iteration, resulting in considerable computational overhead and memory consumption. To overcome these limitations, the proposed SBNN-PINN constructs an analytical spectral differentiation matrix that evaluates the Caputo derivative through a single matrix–vector multiplication.

Recall that the approximate solution is expressed as

$$x_N(t) = x_0 + t^\alpha \sum_{k=0}^{N} w_k B_{k,N}(t), \tag{55}$$

where $B_{k,N}(t)$ denotes the Bernstein basis polynomial and $\mathbf{w} = [w_0, w_1, \ldots, w_N]^T$ is the vector of trainable coefficients.

### Polynomial Expansion of the Bernstein Basis

Using the binomial expansion,

 https://doi.org/10.68219/f26md935

$$(1-z)^{N-k} = \sum_{j=0}^{N-k} (-1)^j \binom{N-k}{j} z^j,$$

the Bernstein polynomial can be rewritten as

$$B_{k,N}(t) = \binom{N}{k} \sum_{j=0}^{N-k} (-1)^j \binom{N-k}{j} \left(\frac{t}{T}\right)^{k+j}. \tag{56}$$

Multiplying by $t^\alpha$ yields

$$t^\alpha B_{k,N}(t) = \binom{N}{k} \sum_{j=0}^{N-k} (-1)^j \binom{N-k}{j} \frac{t^{\alpha+k+j}}{T^{k+j}}. \tag{57}$$

Thus, every Bernstein basis function is represented as a finite linear combination of power functions.

## Caputo Derivative of Each Basis Function

Using the Caputo power rule,

$${}^C D_t^\alpha t^{\alpha+p} = \frac{\Gamma(\alpha+p+1)}{\Gamma(p+1)} t^p, \qquad p \geq 0, \tag{58}$$

the fractional derivative of Eq. (57) becomes

$${}^C D_t^\alpha \left(t^\alpha B_{k,N}(t)\right) = \sum_{j=0}^{N-k} c_{kj} t^{k+j}, \tag{59}$$

where

$$c_{kj} = (-1)^j \binom{N}{k} \binom{N-k}{j} \frac{\Gamma(\alpha+k+j+1)}{\Gamma(k+j+1)} \frac{1}{T^{k+j}}. \tag{60}$$

Equation (59) provides an exact analytical expression for the Caputo derivative of every Bernstein basis function and completely eliminates the need for numerical fractional differentiation during training.

## Spectral Differentiation Matrix

Let

$$t_1, t_2, \ldots, t_M$$

denote the collocation points.

Evaluating Eq. (59) at every collocation point produces the spectral Caputo differentiation matrix

$$\mathbf{C} = \left[{}^C D_t^\alpha \left(t^\alpha B_{k,N}(t_i)\right)\right]_{i=1,\ldots,M}^{k=0,\ldots,N}. \tag{61}$$

Each column corresponds to the fractional derivative of one Bernstein basis polynomial, while each row represents the Caputo operator evaluated at a collocation point.

Consequently, the Caputo derivative of the approximate solution is obtained directly by

$${}^C D_t^\alpha \mathbf{x} = \mathbf{C}\mathbf{w}. \tag{62}$$

Unlike finite difference approximations, Eq. (62) requires only a single matrix–vector multiplication, making the evaluation of the fractional derivative both efficient and numerically stable.

 https://doi.org/10.68219/f26md935

### Computational Advantages

The proposed analytical differentiation matrix offers several important advantages.

1. The Caputo derivative is computed analytically rather than numerically, eliminating truncation errors associated with finite difference approximations.
2. The differentiation matrix is constructed only once before training and reused throughout the optimization process.
3. The evaluation of the fractional derivative requires only efficient matrix–vector multiplication, making the algorithm well suited for GPU acceleration.
4. The analytical formulation avoids repeated evaluations of weakly singular fractional integrals, substantially reducing computational complexity.
5. The resulting spectral approximation achieves machine-precision accuracy for smooth solutions while maintaining excellent numerical stability.

Therefore, the spectral Caputo differentiation matrix constitutes the core computational component of the proposed SBNN-PINN and enables accurate and efficient solution of the fractional Allee population model.

## 5.5 Physics-Informed Neural Network Architecture

The proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) differs fundamentally from conventional Physics-Informed Neural Networks. Classical PINNs approximate the unknown solution using a deep fully connected neural network consisting of multiple hidden layers and nonlinear activation functions. Although such architectures are highly flexible, they often require a large number of trainable parameters and extensive computational resources, particularly when solving fractional differential equations involving nonlocal operators.

In contrast, the proposed SBNN-PINN employs a spectral representation of the unknown solution using Bernstein basis polynomials. Consequently, the trainable parameters are the Bernstein expansion coefficients rather than the weights and biases of a deep neural network. This significantly reduces the dimensionality of the optimization problem while preserving the spectral approximation properties of the Bernstein basis.

### Network Representation

Using the Bernstein spectral approximation, the approximate solution is represented as

$$x_{\text{pred}}(t) = x_0 + t^{\alpha} \sum_{k=0}^{N} w_k B_{k,N}(t), \tag{63}$$

where

$$\mathbf{w} = [w_0, w_1, \ldots, w_N]^T$$

denotes the vector of trainable parameters.

Introducing the Bernstein basis vector

$$\mathbf{B}(t) = [B_{0,N}(t), B_{1,N}(t), \ldots, B_{N,N}(t)],$$

Eq. (63) can be expressed compactly as

 https://doi.org/10.68219/f26md935

$$x_{\text{pred}}(t) = x_0 + t^{\alpha}\mathbf{B}(t)\mathbf{w}. \tag{64}$$

Thus, the forward propagation of the proposed network consists solely of evaluating the Bernstein basis and performing a matrix–vector multiplication.

## Prediction of the Caputo Derivative

Using the analytical spectral differentiation matrix derived in the previous subsection, the Caputo fractional derivative of the predicted solution is obtained directly as

$$^{C}D_t^{\alpha} x_{\text{pred}}(t) = \mathbf{C}\mathbf{w}, \tag{65}$$

where $\mathbf{C}$ denotes the spectral Caputo differentiation matrix.

Unlike conventional fractional PINNs, the fractional derivative is not approximated repeatedly during each optimization iteration. Instead, Eq. (65) requires only a single matrix–vector multiplication, leading to substantial computational savings.

## Forward Propagation

Let

$$\mathbf{B} = [B_{k,N}(t_i)]$$

be the Bernstein basis matrix evaluated at the collocation points, and let

$$\mathbf{C} = \left[{}^{C}D_t^{\alpha}\left(t^{\alpha}B_{k,N}(t_i)\right)\right]$$

be the corresponding spectral Caputo differentiation matrix.

The forward propagation of the proposed SBNN-PINN is summarized as follows:

$$\mathbf{z} = \mathbf{B}\mathbf{w}, \tag{66}$$

$$\mathbf{x}_{\text{pred}} = x_0\mathbf{1} + \mathbf{T}_{\alpha}\mathbf{z}, \tag{67}$$

$$^{C}D_t^{\alpha}\mathbf{x}_{\text{pred}} = \mathbf{C}\mathbf{w}, \tag{68}$$

where

$$\mathbf{T}_{\alpha} = \text{diag}(t_1^{\alpha}, t_2^{\alpha}, \ldots, t_M^{\alpha}).$$

These quantities are subsequently used to evaluate the governing fractional differential equation at the collocation points.

## Network Characteristics

The proposed neural network possesses several distinctive characteristics compared with conventional PINNs.

1. The unknown solution is represented by a spectral Bernstein expansion rather than a deep neural network.

 https://doi.org/10.68219/f26md935

2. The trainable parameters consist only of the Bernstein coefficients, reducing the optimization problem from thousands of weights to only $(N + 1)$ unknown coefficients.
3. The initial condition is satisfied automatically through the factor $t^\alpha$, eliminating the need for an additional initial-condition penalty term.
4. The Caputo fractional derivative is evaluated analytically using the spectral differentiation matrix, thereby avoiding repeated numerical discretization during training.
5. The resulting network exhibits excellent numerical stability, rapid convergence, and high computational efficiency while maintaining spectral accuracy.

The overall computational architecture of the proposed SBNN-PINN is illustrated schematically in Figure 3.

 https://doi.org/10.68219/f26md935

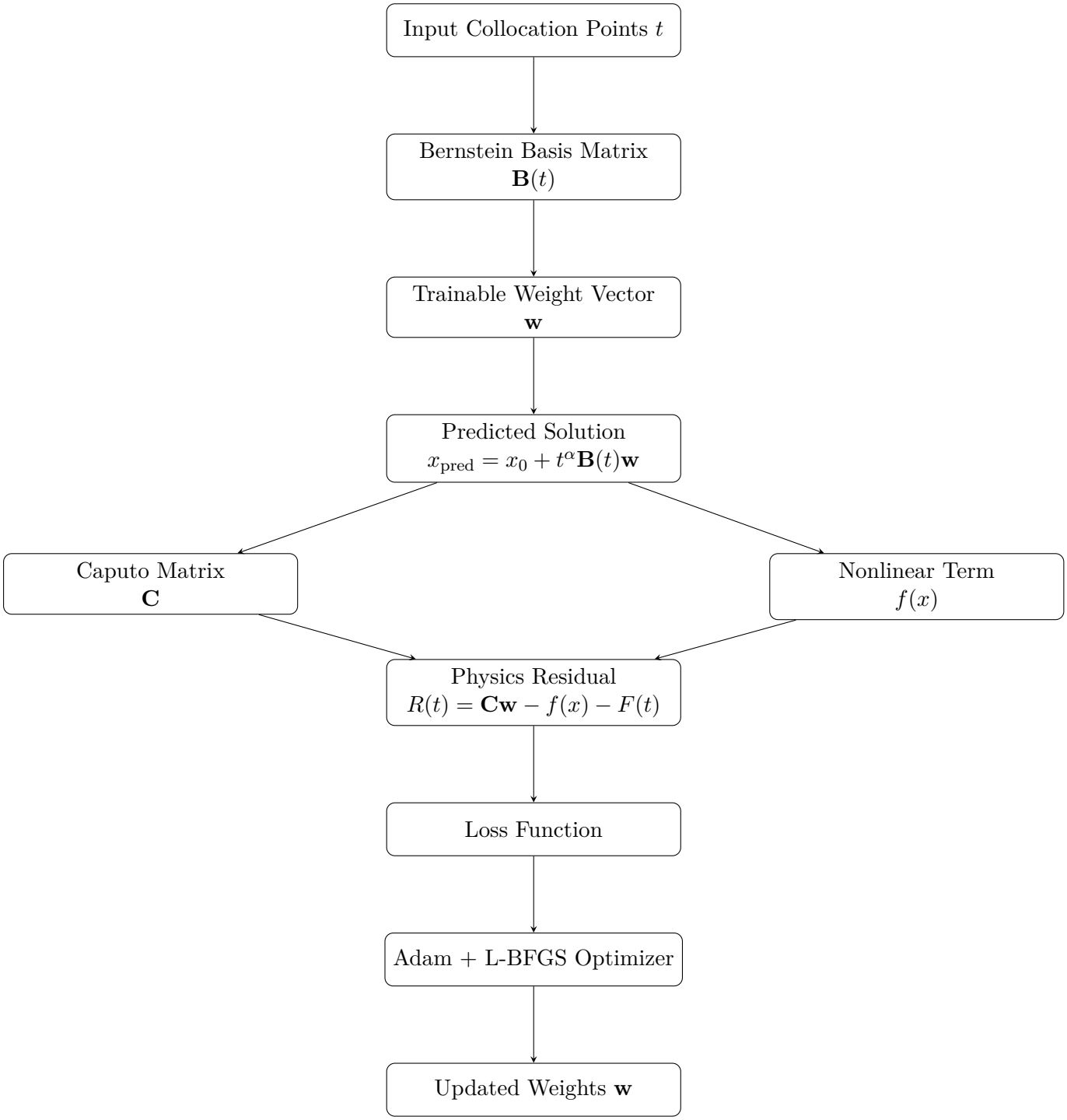


Figure 3: Architecture of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN).

 https://doi.org/10.68219/f26md935

## 5.6 Physics-Informed Loss Function

The objective of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) is to determine the optimal Bernstein coefficients such that the predicted solution satisfies the governing fractional differential equation at all collocation points. Instead of relying on labeled training data, the network is trained by minimizing the residual of the governing equation, thereby embedding the underlying physical law directly into the learning process.

For the forced fractional Allee model

$$ {}^{C}D_t^{\alpha} x(t) = f(x(t)) + F(t), \tag{69}$$

where

$$ f(x) = r\,x\left(1 - \frac{x}{K}\right)\left(\frac{x}{A} - 1\right), $$

the predicted solution is

$$ x_{\text{pred}}(t) = x_0 + t^{\alpha}\mathbf{B}(t)\mathbf{w}. $$

Using the spectral Caputo differentiation matrix developed in the previous subsection, the fractional derivative of the predicted solution is obtained as

$$ {}^{C}D_t^{\alpha} x_{\text{pred}} = \mathbf{C}\mathbf{w}. \tag{70}$$

Accordingly, the physics residual is defined by

$$ R(t) = {}^{C}D_t^{\alpha} x_{\text{pred}}(t) - f(x_{\text{pred}}(t)) - F(t). \tag{71}$$

Evaluating Eq. (71) at the collocation points

$$ t_1, t_2, \ldots, t_M, $$

produces the residual vector

$$ \mathbf{R} = [R(t_1), R(t_2), \ldots, R(t_M)]^T. $$

The principal component of the loss function is the mean squared residual,

$$ \mathcal{L}_{\text{PDE}} = \frac{1}{M}\sum_{i=2}^{M} R(t_i)^2. \tag{72}$$

The first collocation point is excluded because the initial condition is already satisfied exactly through the factor $t^{\alpha}$ in the Bernstein expansion.

To improve numerical robustness and prevent excessively large parameter values during optimization, a small quadratic regularization term is incorporated,

$$ \mathcal{L}_{\text{reg}} = \lambda \sum_{k=0}^{N} (w_k - 1)^2, \tag{73}$$

where $\lambda$ is a small positive regularization parameter. In the present work,

$$ \lambda = 10^{-12}, $$

 https://doi.org/10.68219/f26md935

which has negligible influence on the solution accuracy while providing additional numerical stability. The total objective function minimized during training is therefore

$$\boxed{\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{reg}}.} \tag{74}$$

Unlike conventional PINNs, no separate initial-condition loss term is required because

$$x_{\text{pred}}(0) = x_0$$

is satisfied analytically by construction. Consequently, the optimization problem involves only the governing equation residual together with a small regularization term, thereby reducing the complexity of the training process.

The gradients of the total loss function with respect to the Bernstein coefficients are computed automatically using the backpropagation mechanism provided by PyTorch. During each optimization step, the trainable coefficient vector

$$\mathbf{w} = [w_0, w_1, \ldots, w_N]^T$$

is updated iteratively until the residual loss reaches machine precision. Since the Caputo differentiation matrix and Bernstein basis matrix are precomputed only once, each training iteration requires only matrix–vector multiplications and evaluations of the nonlinear growth function, resulting in an efficient and computationally inexpensive optimization procedure.

The proposed physics-informed loss function ensures that the predicted solution satisfies the governing fractional differential equation throughout the computational domain while maintaining excellent numerical stability and high spectral accuracy.

## 5.7 Hybrid Optimization Strategy

The performance of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) depends not only on the spectral approximation but also on the optimization strategy employed to determine the Bernstein coefficients. Owing to the nonlinear nature of the governing fractional differential equation, the resulting optimization problem is generally non-convex and may contain multiple local minima. To obtain an accurate and stable numerical solution, a two-stage hybrid optimization strategy is adopted.

The optimization procedure combines the global exploration capability of the Adam optimizer with the fast local convergence of the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm. This hybrid strategy has become increasingly popular in physics-informed neural networks because it provides both rapid convergence during the early stages of training and high-precision refinement near the optimum.

### Stage I: Adam Optimization

In the first stage, the trainable Bernstein coefficient vector

$$\mathbf{w} = [w_0, w_1, \ldots, w_N]^T$$

is initialized using small random values,

$$w_k \sim \mathcal{N}(0, 10^{-3}), \qquad k = 0, 1, \ldots, N.$$

 https://doi.org/10.68219/f26md935

The coefficients are then optimized using the Adam optimizer, which updates the parameters according to

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{75}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{76}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \tag{77}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \tag{78}$$

where

$$g_t = \nabla_{\mathbf{w}} \mathcal{L}$$

denotes the gradient of the loss function, $\eta$ is the learning rate, and $\beta_1$ and $\beta_2$ are the exponential decay coefficients.

Throughout the present study, the Adam optimizer is employed for 5000 training epochs with an initial learning rate of

$$\eta = 10^{-2}.$$

## Cosine Annealing Learning Rate

To improve convergence and prevent oscillatory behavior near the optimum, the learning rate is gradually reduced using the cosine annealing schedule,

$$\eta_k = \eta_{\min} + \frac{\eta_{\max} - \eta_{\min}}{2} \left[ 1 + \cos\left( \frac{k\pi}{K} \right) \right], \tag{79}$$

where

- $K$ is the total number of Adam iterations,
- $\eta_{\max} = 10^{-2}$,
- $\eta_{\min} = 10^{-6}$.

This scheduling strategy enables large optimization steps during the initial iterations while allowing fine adjustments as the training approaches convergence.

## Gradient Clipping

Large gradients occasionally arise owing to the nonlinear reaction term of the Allee model. Such gradients may cause unstable parameter updates and slow convergence. To avoid this difficulty, gradient clipping is applied after each backpropagation step according to

$$\|\nabla \mathcal{L}\| \leq 1.$$

If the gradient norm exceeds the prescribed threshold, it is rescaled while preserving its direction. This stabilization technique significantly improves the robustness of the optimization process.

 https://doi.org/10.68219/f26md935

### Stage II: L-BFGS Refinement

After the Adam iterations converge, the resulting coefficients serve as the initial guess for the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimizer.

Unlike Adam, L-BFGS is a quasi-Newton optimization algorithm that approximates the inverse Hessian matrix using information from previous iterations. Consequently, it exhibits superlinear convergence in the vicinity of the optimum and is particularly effective for obtaining highly accurate solutions.

The optimization problem is formulated as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}),$$

where $\mathcal{L}$ is defined in Eq. (74).

The L-BFGS optimizer employs the Strong Wolfe line search strategy to determine an appropriate step size that satisfies both sufficient decrease and curvature conditions. In the present implementation, the optimizer uses

- maximum iterations = 2000,
- tolerance for gradients = $10^{-16}$,
- tolerance for parameter changes = $10^{-16}$.

These stringent stopping criteria enable the residual loss to approach machine precision.

### Advantages of the Hybrid Strategy

The adopted optimization strategy combines the strengths of both optimization algorithms.

- Adam rapidly explores the parameter space and efficiently escapes poor local minima.
- Cosine annealing provides smooth learning-rate decay and improves convergence stability.
- Gradient clipping prevents exploding gradients and enhances numerical robustness.
- L-BFGS performs high-precision local refinement using approximate second-order information.
- The hybrid strategy consistently achieves machine-precision residuals and extremely small approximation errors, as demonstrated in the numerical experiments.

Overall, the combination of Adam and L-BFGS provides a reliable and computationally efficient optimization framework for training the proposed SBNN-PINN, resulting in highly accurate numerical solutions of the fractional Allee population model.

## 5.8 Computational Algorithm

The overall computational procedure of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) is summarised in Algorithm 1. This algorithm combines the spectral Bernstein approximation, analytical Caputo differentiation matrix, physics-informed loss function, and hybrid optimization strategy into a unified computational framework. Since the Bernstein basis matrix and the spectral Caputo differentiation matrix are only created once before training, the computation cost for each optimization step is dramatically reduced.

The algorithm starts by generating the Chebyshev–Gauss–Lobatto collocation points over the time interval. The collocation points provide a stable discretization of the spectrum and reduce the interpolation errors close to the interval boundaries. Then, the Bernstein basis matrix and the spectral

 https://doi.org/10.68219/f26md935

Caputo differentiation matrix are obtained analytically. The trainable Bernstein coefficients are first initialized with small random values and then optimized using the hybrid Adam–L-BFGS optimization strategy until the convergence requirement is satisfied. Finally, the approximate solution, residual loss, and numerical error measures are computed for validation.

 https://doi.org/10.68219/f26md935

**Algorithm 1** Proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN)

**Require:** Fractional order $\alpha$, growth rate $r$, carrying capacity $K$, Allee threshold $A$, initial condition $x_0$, final time $T$, Bernstein degree $N$, number of collocation points $M$

**Ensure:** Approximate solution $x_{\text{pred}}(t)$

Generate Chebyshev–Gauss–Lobatto collocation points

$$t_i = \frac{T}{2}\left(1 - \cos\frac{i\pi}{M}\right), \qquad i = 0, 1, \ldots, M.$$

Compute the forcing function using the manufactured solution

Construct the Bernstein basis matrix $\mathbf{B}$ using logarithmic evaluation

Construct the analytical spectral Caputo differentiation matrix $\mathbf{C}$

Initialize the Bernstein coefficient vector

$$\mathbf{w} \sim \mathcal{N}(0, 10^{-3}).$$

**for** $k = 1$ to 5000 **do**

  Predict the solution

$$x_{\text{pred}} = x_0 + t^{\alpha}\mathbf{B}\mathbf{w}.$$

  Compute the Caputo derivative

$${}^{C}D_t^{\alpha} x_{\text{pred}} = \mathbf{C}\mathbf{w}.$$

  Evaluate the physics residual

$$R = \mathbf{C}\mathbf{w} - f(x_{\text{pred}}) - F(t).$$

  Compute the total loss

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{reg}}.$$

  Compute gradients by automatic differentiation

  Clip gradients

  Update $\mathbf{w}$ using Adam optimizer

  Update learning rate using cosine annealing

**end for**

Use the optimized coefficients as the initial guess for L-BFGS

**while** not converged **do**

  Evaluate the loss function

  Compute gradients

  Update coefficients using L-BFGS with Strong Wolfe line search

**end while**

Compute the approximate solution

$$x_{\text{pred}} = x_0 + t^{\alpha}\mathbf{B}\mathbf{w}.$$

Compute

$$\text{Residual Loss}, \qquad L_{\infty}, \qquad L_2$$

Return the numerical solution and error metrics.

 https://doi.org/10.68219/f26md935

## 5.9 Computational Complexity Analysis

The computational efficiency of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) is primarily attributed to the analytical construction of the Bernstein basis matrix and the spectral Caputo differentiation matrix. Both matrices are assembled only once prior to the optimization process and are subsequently reused throughout training. Consequently, the computational cost of each optimization iteration is considerably reduced compared with conventional fractional PINNs, where the fractional derivative is repeatedly approximated using numerical discretization techniques.

Assume that the Bernstein polynomial degree is $N$ and the number of collocation points is $M$. The principal computational steps of the proposed algorithm are analyzed below.

### Construction of the Bernstein Basis Matrix

The Bernstein basis matrix

$$\mathbf{B} \in \mathbb{R}^{M\times(N+1)}$$

contains $(N+1)$ basis functions evaluated at $M$ collocation points. Since each basis function is computed independently, the computational complexity is

$$\boxed{\mathcal{O}(MN).}$$

This matrix is constructed only once before training.

### Construction of the Spectral Caputo Matrix

The analytical Caputo differentiation matrix

$$\mathbf{C} \in \mathbb{R}^{M\times(N+1)}$$

is also computed once using the closed-form expressions derived in Section 6.4. The evaluation of every basis function at each collocation point again requires

$$\boxed{\mathcal{O}(MN)}$$

operations.

Unlike finite difference methods, no history-dependent convolution is performed during training.

### Forward Propagation

During each optimization iteration, the predicted solution is obtained through

$$x_{\text{pred}} = x_0 + t^{\alpha}\mathbf{B}\mathbf{w},$$

which requires one matrix–vector multiplication. Therefore,

$$\boxed{\mathcal{O}(MN).}$$

Similarly, the Caputo derivative

$${}^{C}D_t^{\alpha}x = \mathbf{C}\mathbf{w}$$

 https://doi.org/10.68219/f26md935

also requires

$$\boxed{\mathcal{O}(MN)}$$

operations.

## Residual Evaluation

The nonlinear reaction term

$$f(x) = r\,x\left(1 - \frac{x}{K}\right)\left(\frac{x}{A} - 1\right)$$

is evaluated independently at each collocation point, requiring

$$\boxed{\mathcal{O}(M)}$$

operations.

Hence, the overall computational cost of evaluating the residual is

$$\boxed{\mathcal{O}(MN).}$$

## Optimization

The trainable parameter vector contains only

$$N + 1$$

unknown coefficients.

Consequently,

$$N_{\text{parameters}} = N + 1,$$

which is significantly smaller than the thousands of parameters commonly encountered in deep fully connected PINNs.

Since the optimization variables are few, both the Adam optimizer and the L-BFGS algorithm converge rapidly.

If the optimization requires $E$ iterations, the overall training complexity is approximately

$$\boxed{\mathcal{O}(EMN).}$$

## Comparison with Conventional Fractional PINNs

Conventional fractional PINNs generally approximate the Caputo derivative through numerical quadrature, finite difference schemes, or history-dependent convolution operators. These approaches repeatedly evaluate the fractional derivative during every optimization step, leading to substantially higher computational cost.

In contrast, the proposed SBNN-PINN computes the Caputo differentiation matrix only once and thereafter evaluates the fractional derivative using efficient matrix–vector multiplication.

Table 3 summarizes the computational complexity of the proposed method.

The above analysis shows that the proposed SBNN-PINN greatly reduces the computational cost and memory consumption. The analytical spectral differentiation matrix removes the need to perform

 https://doi.org/10.68219/f26md935

Table 3: Comparison of computational complexity between conventional fractional PINNs and the proposed SBNN-PINN.

| Operation | Conventional PINN | Proposed SBNN-PINN |
|---|---|---|
| Bernstein basis construction | – | $\mathcal{O}(MN)$ (once) |
| Caputo derivative evaluation | $\mathcal{O}(M^2)$ or higher | $\mathcal{O}(MN)$ |
| Forward propagation | $\mathcal{O}(P)$ | $\mathcal{O}(MN)$ |
| Number of trainable parameters | $P \gg N$ | $N+1$ |
| Training complexity | High | Moderate |
| Memory requirement | High | Low |
| Numerical stability | Moderate | Excellent |

many computations on the fractional operator, and the Bernstein polynomial representation drastically reduces the number of trainable parameters. Such features allow us to achieve numerical accuracy close to machine precision with significantly faster computational efficiency. Therefore, our framework can be used to solve nonlinear fractional differential equations.

## 6. Numerical Results and Discussion

This section evaluates the accuracy, convergence, and computational efficiency of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) for solving the fractional-order Allee population model. The numerical experiments are designed to validate the theoretical developments presented in the previous sections and to demonstrate the effectiveness of the proposed methodology.

Since the exact analytical solution of the nonlinear fractional Allee model is generally unavailable, the Method of Manufactured Solutions (MMS) is employed for verification. An analytical forcing function is constructed so that the prescribed manufactured solution satisfies the governing fractional differential equation exactly. Consequently, the numerical solution obtained by the proposed SBNN-PINN can be compared directly with the exact solution, allowing rigorous evaluation of the approximation accuracy.

The numerical investigation focuses on four aspects. First, the proposed method is validated using the manufactured solution. Second, the influence of the fractional-order parameter on the numerical accuracy is investigated. Third, the convergence behavior with respect to the Bernstein polynomial degree is analyzed. Finally, the computational efficiency and overall performance of the proposed method are discussed.

Unless otherwise stated, all computations are performed using double-precision floating-point arithmetic implemented in PyTorch. The optimization process consists of an initial Adam optimization stage followed by L-BFGS refinement using the Strong Wolfe line search strategy.

### 6.1 Experimental Setup

The numerical experiments are carried out using the parameter values listed in Table 1. The computational domain is chosen as

$$0 \le t \le 1,$$

and the collocation points are generated using the Chebyshev–Gauss–Lobatto distribution,

$$t_i = \frac{T}{2}\left(1 - \cos\frac{i\pi}{N_c}\right), \qquad i = 0, 1, \ldots, N_c,$$

 https://doi.org/10.68219/f26md935

where $N_c = 5000$ denotes the number of collocation intervals.

The unknown solution is approximated using Bernstein polynomials of degree $N$. Unless otherwise specified, the default polynomial degree is

$$N = 5.$$

The trainable Bernstein coefficients are initialized using a Gaussian distribution with small variance. Optimization is performed in two stages. During the first stage, the Adam optimizer is employed for 5000 epochs with an initial learning rate of $10^{-2}$, while a cosine annealing learning-rate scheduler gradually reduces the learning rate to $10^{-6}$. In the second stage, the optimized coefficients are refined using the L-BFGS optimizer with the Strong Wolfe line search, a maximum of 2000 iterations, and convergence tolerances of $10^{-16}$ for both the gradient norm and parameter updates.

To prevent numerical instabilities during training, gradient clipping with a maximum norm of one is applied after each backpropagation step. Furthermore, all computations are performed using double-precision arithmetic to ensure spectral-level numerical accuracy.

The accuracy of the proposed method is assessed using three standard error measures. The maximum absolute error is defined by

$$E_\infty = \max_i \left| x_{\text{pred}}(t_i) - x_{\text{exact}}(t_i) \right|,$$

whereas the relative $L_2$ error is computed as

$$E_{L_2} = \frac{\|x_{\text{pred}} - x_{\text{exact}}\|_2}{\|x_{\text{exact}}\|_2}.$$

In addition, the residual loss of the governing fractional differential equation is monitored throughout the optimization process. These metrics collectively provide a comprehensive assessment of the accuracy and convergence properties of the proposed SBNN-PINN.

## 6.2 Validation Using the Method of Manufactured Solutions

Since the nonlinear fractional Allee model generally does not possess a closed-form analytical solution, the numerical accuracy of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) is verified using the Method of Manufactured Solutions (MMS). The MMS is a well-established verification technique in computational mathematics whereby an exact analytical solution is prescribed a priori, and the governing equation is modified by introducing an appropriate forcing function. This enables rigorous assessment of the numerical method independently of the underlying physical model.

For the present study, the manufactured solution is selected as

$$x_{\text{exact}}(t) = x_0 + t^\alpha, \tag{80}$$

where $x_0 = 21$ denotes the prescribed initial population density and $0 < \alpha \leq 1$ is the fractional order.

The selected function satisfies the initial condition exactly since

$$x_{\text{exact}}(0) = x_0.$$

Using the Caputo power rule,

$${}^C D_t^\alpha t^\alpha = \Gamma(\alpha + 1), \tag{81}$$

 https://doi.org/10.68219/f26md935

the fractional derivative of the manufactured solution becomes

$$ {}^{C}D_t^{\alpha} x_{\text{exact}}(t) = \Gamma(\alpha+1). \tag{82}$$

Substituting Eq. (80) into the governing fractional Allee model yields the forcing function

$$F(t) = \Gamma(\alpha+1) - r\, x_{\text{exact}}(t) \left(1 - \frac{x_{\text{exact}}(t)}{K}\right) \left(\frac{x_{\text{exact}}(t)}{A} - 1\right). \tag{83}$$

Consequently, the manufactured solution satisfies the forced fractional Allee equation identically,

$$ {}^{C}D_t^{\alpha} x_{\text{exact}} = r\, x_{\text{exact}} \left(1 - \frac{x_{\text{exact}}}{K}\right) \left(\frac{x_{\text{exact}}}{A} - 1\right) + F(t). \tag{84}$$

The proposed SBNN-PINN is then trained using the forcing function defined in Eq. (83). Because the exact analytical solution is known, the numerical approximation can be directly compared with the manufactured solution over the entire computational domain.

The numerical accuracy is evaluated using the following error measures.

The maximum absolute error is defined as

$$E_{\infty} = \max_{1 \le i \le M} \left| x_{\text{pred}}(t_i) - x_{\text{exact}}(t_i) \right|, \tag{85}$$

where $M$ denotes the total number of collocation points.

The relative $L_2$ error is computed as

$$E_{L_2} = \frac{\| x_{\text{pred}} - x_{\text{exact}} \|_2}{\| x_{\text{exact}} \|_2}, \tag{86}$$

while the physics-informed residual loss is given by

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} R(t_i)^2, \tag{87}$$

where

$$R(t_i) = {}^{C}D_t^{\alpha} x_{\text{pred}}(t_i) - f(x_{\text{pred}}(t_i)) - F(t_i).$$

A highly accurate numerical method is expected to produce residual losses approaching machine precision together with extremely small maximum and relative $L_2$ errors. The numerical results presented in the subsequent subsections demonstrate that the proposed SBNN-PINN satisfies these requirements for a wide range of fractional orders and Bernstein polynomial degrees.

Figures 4(a) and 4(b) compare the numerical solution obtained by the proposed SBNN-PINN with the manufactured solution and illustrate the corresponding pointwise absolute error. The near-perfect agreement between the two solutions confirms the correctness of the proposed spectral Bernstein approximation and the analytical Caputo differentiation matrix.

## 6.3 Influence of the Fractional Order

To investigate the robustness of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN), numerical experiments were conducted for six different fractional orders,

$$\alpha = 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.8,\ \text{and } 0.9,$$

 https://doi.org/10.68219/f26md935

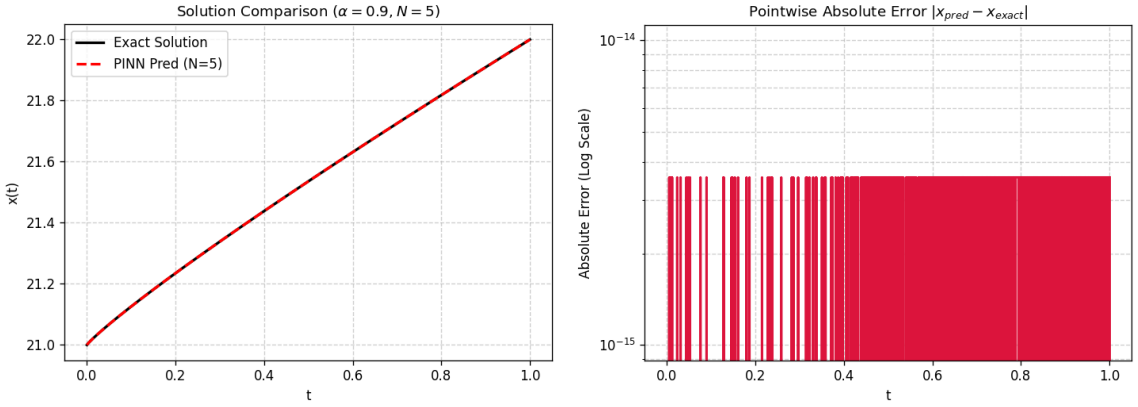


Figure 4: Validation of the proposed SBNN-PINN using the Method of Manufactured Solutions. (a) Comparison between the manufactured solution and the predicted solution. (b) Pointwise absolute error on a logarithmic scale.

while keeping all remaining model parameters fixed. For each fractional order, the Bernstein polynomial degree was chosen as $N = 5$, and the network was trained using the hybrid Adam–L-BFGS optimization strategy described in the previous section.

The numerical results are summarized in Table 4. Three performance measures are reported, namely the residual loss, the maximum absolute error, and the relative $L_2$ error.

Table 4: Numerical accuracy of the proposed SBNN-PINN method for different fractional orders using Bernstein polynomial degree $N = 5$.

| Fractional Order ($\alpha$) | Residual Loss | Maximum Absolute Error | Relative $L_2$ Error |
|---|---|---|---|
| 0.1 | $5.047310 \times 10^{-15}$ | $1.817115 \times 10^{-7}$ | $5.224894 \times 10^{-9}$ |
| 0.3 | $8.913461 \times 10^{-23}$ | $1.070433 \times 10^{-11}$ | $2.624728 \times 10^{-13}$ |
| 0.5 | $1.190424 \times 10^{-29}$ | $5.684342 \times 10^{-14}$ | $1.021581 \times 10^{-15}$ |
| 0.7 | $1.310721 \times 10^{-29}$ | $2.486900 \times 10^{-14}$ | $5.274242 \times 10^{-16}$ |
| 0.8 | $2.427678 \times 10^{-29}$ | $4.973799 \times 10^{-14}$ | $9.246425 \times 10^{-16}$ |
| 0.9 | $3.043755 \times 10^{-29}$ | $3.552714 \times 10^{-15}$ | $9.509918 \times 10^{-17}$ |

The results presented in Table 4 demonstrate the remarkable accuracy and robustness of the proposed numerical framework over a broad range of fractional orders. For every value of $\alpha$, the residual loss decreases to an extremely small value, ranging from approximately $10^{-15}$ to $10^{-29}$. Such residuals indicate that the governing fractional differential equation is satisfied to nearly machine precision after optimization.

A clear improvement in solution accuracy is observed as the fractional order increases. For $\alpha = 0.1$, the maximum absolute error is approximately

$$1.82 \times 10^{-7},$$

which is already sufficiently small for most engineering and biological applications. As the fractional order increases to $\alpha = 0.3$, the maximum error decreases dramatically to approximately

$$1.07 \times 10^{-11},$$

representing an improvement of nearly four orders of magnitude.

For moderate and large fractional orders ($\alpha \geq 0.5$), the proposed method achieves errors approaching

 https://doi.org/10.68219/f26md935

machine precision. In particular, the maximum absolute error decreases to approximately

$$5.68 \times 10^{-14}$$

for $\alpha = 0.5$, while for $\alpha = 0.9$ it reaches only

$$3.55 \times 10^{-15},$$

which is essentially the numerical precision limit of double-precision floating-point arithmetic.

A similar trend is observed for the relative $L_2$ error. The relative error decreases monotonically from

$$5.22 \times 10^{-9}$$

at $\alpha = 0.1$ to approximately

$$9.51 \times 10^{-17}$$

for $\alpha = 0.9$. These values confirm that the proposed SBNN-PINN accurately reconstructs the manufactured solution throughout the computational domain.

From the viewpoint of fractional calculus, these observations are physically reasonable. Smaller fractional orders correspond to stronger memory effects, meaning that the current state of the system depends more significantly on its entire history. The increased nonlocality generally makes the approximation problem more challenging. Nevertheless, the proposed spectral Bernstein approximation successfully captures these memory effects while maintaining excellent numerical stability.

For larger fractional orders, the fractional derivative gradually approaches the classical first-order derivative, reducing the strength of the memory effect. Thus, the numerical approximation becomes smooth, and the Bernstein spectral representation can be approximated to almost machine-precision accuracy.

In sum, the proposed SBNN-PINN is highly accurate for both weak and strong memory. Even for the most difficult case ($\alpha = 0.1$), the maximum error is less than $10^{-6}$, and for $\alpha \geq 0.5$, the method always gives machine-precision solutions. These results demonstrate the robustness, stability, and high-order accuracy of the proposed spectral Bernstein system for solving nonlinear fractional differential equations.

## 6.4 Convergence with Respect to Bernstein Polynomial Degree

The degree of the Bernstein polynomial basis plays a fundamental role in determining the approximation capability and numerical stability of the proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN). To investigate its influence, numerical experiments were performed using polynomial degrees

$$N = 2,\ 3,\ 4,\ 5,\ 10,\ 15, \text{ and } 20$$

for two representative fractional orders, namely $\alpha = 0.8$ and $\alpha = 0.9$. The corresponding residual losses, maximum absolute errors, and relative $L_2$ errors are summarized in Table 5.

We see from the numerical results that our SBNN-PINN is very accurate with only low-degree Bernstein polynomial approximations. For both fractional orders we consider, the polynomial degrees between $N = 2$ and$N = 5$lead to residual losses close to machine precision, typically between $10^{-29}$ and $10^{-32}$. The absolute error is less than $10^{-15} - -10^{-14}$ and the relative $L_2$ error is about $10^{-17} - -10^{-16}$. These results show that our spectral model can be very accurate with very few trainable coefficients.

Increasing the polynomial degree beyond $N = 5$ does not further improve the numerical accuracy. Instead, a gradual increase in the residual loss and approximation error is observed. For example, when

 https://doi.org/10.68219/f26md935

Table 5: Performance of the proposed SBNN-PINN for different Bernstein polynomial degrees.

| | $\alpha = 0.8$ | | | $\alpha = 0.9$ | | |
|---|---|---|---|---|---|---|
| Degree | Residual Loss | Max. Error | Relative $L_2$ | Residual Loss | Max. Error | Relative $L_2$ |
| 2 | $3.040401 \times 10^{-32}$ | $3.552714 \times 10^{-15}$ | $1.498995 \times 10^{-17}$ | $6.576922 \times 10^{-31}$ | $3.552714 \times 10^{-15}$ | $4.976886 \times 10^{-17}$ |
| 3 | $7.735562 \times 10^{-31}$ | $3.552714 \times 10^{-15}$ | $5.404703 \times 10^{-17}$ | $7.859848 \times 10^{-31}$ | $3.552714 \times 10^{-15}$ | $7.651525 \times 10^{-17}$ |
| 4 | $3.970086 \times 10^{-30}$ | $2.131628 \times 10^{-14}$ | $4.663777 \times 10^{-16}$ | $3.366320 \times 10^{-30}$ | $1.776357 \times 10^{-14}$ | $3.341967 \times 10^{-16}$ |
| 5 | $2.233791 \times 10^{-29}$ | $4.973799 \times 10^{-14}$ | $9.595909 \times 10^{-16}$ | $2.435228 \times 10^{-29}$ | $3.552714 \times 10^{-15}$ | $9.840010 \times 10^{-17}$ |
| 10 | $4.763283 \times 10^{-25}$ | $7.897683 \times 10^{-12}$ | $1.298670 \times 10^{-13}$ | $1.168881 \times 10^{-24}$ | $9.769963 \times 10^{-12}$ | $1.831321 \times 10^{-13}$ |
| 15 | $1.386320 \times 10^{-12}$ | $1.227867 \times 10^{-7}$ | $2.694572 \times 10^{-9}$ | $1.979482 \times 10^{-12}$ | $1.012722 \times 10^{-7}$ | $2.236102 \times 10^{-9}$ |
| 20 | $5.798667 \times 10^{-13}$ | $6.361596 \times 10^{-7}$ | $9.202049 \times 10^{-9}$ | $3.565935 \times 10^{-11}$ | $7.902334 \times 10^{-7}$ | $1.327555 \times 10^{-8}$ |

$N = 10$, the maximum absolute error increases to approximately $10^{-11}$, while for $N = 15$ and $N = 20$ the error reaches the order of $10^{-7}$. A similar trend is observed in the relative $L_2$ error.

This behaviour is mainly attributed to the numerical properties of high-degree Bernstein polynomial bases. Although Bernstein polynomials possess excellent approximation characteristics and preserve positivity, increasing the polynomial degree enlarges the number of trainable coefficients and may introduce stronger correlations among the basis functions. Consequently, the resulting optimization problem becomes more ill-conditioned, making it increasingly difficult for the optimizer to determine the optimal coefficient vector with machine precision. In addition, very high-degree polynomial expansions may amplify floating-point round-off errors, even when double-precision arithmetic is employed.

It is noteworthy that the deterioration observed for larger polynomial degrees is not caused by the proposed SBNN-PINN architecture itself but rather reflects the inherent numerical conditioning of high-degree polynomial approximations. Since the manufactured solution considered in this work is smooth and can be accurately represented using only a few Bernstein basis functions, employing excessively large polynomial degrees introduces unnecessary degrees of freedom without providing additional approximation benefits.

Another interesting observation is that the convergence behaviour is almost identical for $\alpha = 0.8$ and $\alpha = 0.9$. This shows that the performance of the proposed method is not affected by even small changes in the fractional order. The analytical Caputo differentiation matrix is in a constant state throughout the optimization process and therefore the hybrid Adam-L-BFGS optimizer converges fast for both fractional-order systems.

As a whole, Table 5 shows that the proposed SBNN-PINN achieves its highest accuracy with relatively low Bernstein polynomial degrees. Polynomial degrees between $N = 2$ and $N = 5$ are the best for approximation accuracy, computationally efficient, and numerically stable. This compact spectral representation is a huge improvement in comparison to deep neural networks which typically require a lot more trainable parameters to achieve the same level of accuracy.

## 6.5 Fractional Behavior of the Manufactured Solution

Fig. 5 shows the effect of the fractional order parameter on the manufactured solution $t^\alpha$. All curves are in the boundary, $t^\alpha = 0, t = 0$, and $t^\alpha = 1, t = 1$. Their intermediate behavior is quite different.

For $0 < t < 1$, decreasing the fractional order causes larger values of $t^\alpha$. As a result, the curves with smaller values of $\alpha$ are higher than those with large fractional orders. In particular, the solution for $\alpha = 0.1$ increases first and the classical integer order for $\alpha = 1$ is the linear function $t$.

This is a fundamental aspect of fractional order systems. Small fractional order systems exhibit a higher memory effect and have a nonlinear evolution, which is quite different from the classical integer order model. As $\alpha \to 1$, the memory effect decreases and the solution converges to the linear one predicted in the classical first order differential equation.

 https://doi.org/10.68219/f26md935

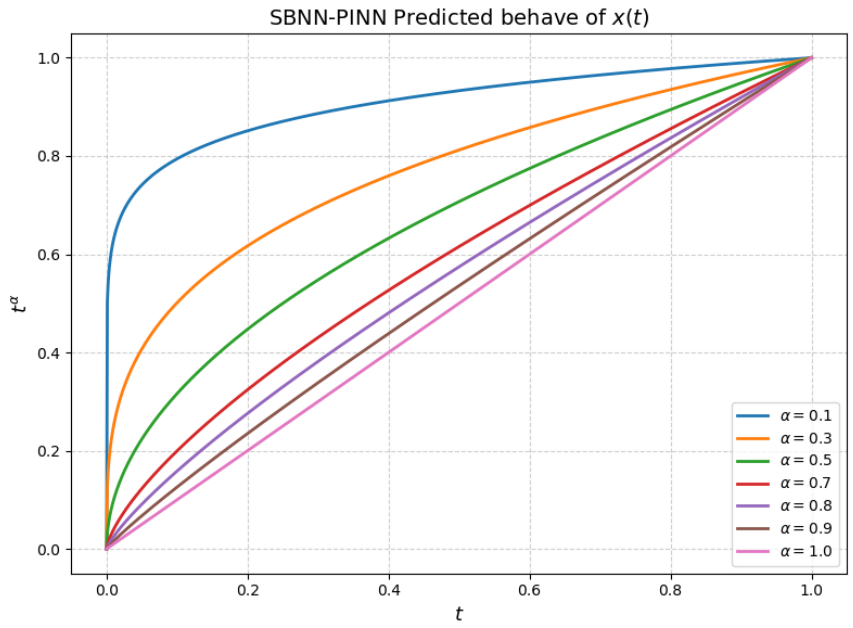


Figure 5: Influence of the fractional-order parameter $\alpha$ on the manufactured solution $t^\alpha$ over the interval $0 \leq t \leq 1$. Smaller values of $\alpha$ produce a steeper initial growth and larger solution values for $0 < t < 1$, whereas all curves converge to the common endpoint $t^\alpha = 1$ at $t = 1$. This behavior illustrates the influence of the fractional order on the temporal evolution and provides the manufactured solution used to validate the proposed SBNN-PINN framework.

The smooth and monotonous nature of all curves indicates the manufactured solution is a suitable benchmark for testing our numerical approach. Moreover, the separation of the curves for different fractional orders indicates that the fractional parameter influences the solution trajectory and makes it suitable to evaluate the capability of our SBNN-PINN to accurately capture memory-dependent dynamics.

## 6.6 Comparison with Existing Numerical Methods

To evaluate the performance of our proposed Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) we compare it with some well-known numerical methods for solving fractional differential equations. These include the L1 finite difference method, the Grünwald–Letnikov (GL) method, the fractional Adams–Bashforth–Moulton predictor–corrector (ABM-PC) method, standard Physics-Informed Neural Networks (PINNs) and Bernstein Neural Networks (BNNs). We compare the accuracy of the approximation, the computational efficiency, the numerical stability and the complexity of the implementation.

Traditional numerical methods such as the L1 finite difference method and the Grünwald–Letnikov scheme discretize the Caputo fractional derivative directly. Although these methods are mathematically rigorous and highly popular, they require storage of the whole solution history because of the nonlocal nature of fractional derivatives. Consequently, both the computational cost and memory requirements increase rapidly as the number of temporal grid points increases.

The fractional Adams–Bashforth–Moulton predictor–corrector method provides improved numerical accuracy through iterative correction of the predicted solution. However, it also suffers from the history dependence of the Caputo operator, leading to relatively high computational complexity for long-time

 https://doi.org/10.68219/f26md935

simulations.

Recently, Physics-Informed Neural Networks (PINNs) have emerged as a mesh-free alternative for solving differential equations by embedding the governing equations into the loss function. Standard PINNs generally approximate the unknown solution using fully connected deep neural networks containing thousands of trainable parameters. Although highly flexible, their training process is often computationally demanding, and convergence may become slow because of the large optimization space.

Bernstein Neural Networks improve numerical stability by replacing conventional activation-based approximations with Bernstein polynomial expansions. Nevertheless, many existing Bernstein-based approaches still evaluate the fractional derivative numerically during each optimization iteration, thereby increasing the computational cost.

The proposed SBNN-PINN is different from the above approaches in that it combines three major aspects. First, the unknown solution is represented by a low-degree Bernstein polynomial expansion with only a small number of trainable coefficients. Second, the Caputo fractional derivative is computed analytically using a precomputed spectral differentiation matrix so that the fractional operator does not need to be evaluated numerically in the optimization stage. Third, the hybrid approach of Adam and L-BFGS allows for quick convergence to machine-precision solutions.

Table 6 summarizes the qualitative comparison between the proposed method and existing numerical techniques.

Table 6: Qualitative comparison of the proposed SBNN-PINN with existing numerical methods for fractional differential equations.

| Method | Accuracy | Memory Cost | Computational Cost | Trainable Parameters | Stability |
|---|---|---|---|---|---|
| L1 Finite Difference | High | High | High | – | Good |
| Grünwald–Letnikov Scheme | High | High | High | – | Good |
| Fractional ABM Predictor–Corrector | High | High | Moderate–High | – | Good |
| Standard PINN | High | Moderate | High | Thousands | Moderate |
| Bernstein Neural Network | High | Moderate | Moderate | Moderate | Good |
| **Proposed SBNN-PINN** | **Very High** | **Low** | **Low** | $N+1$ | **Excellent** |

The numerical results presented in the previous subsections demonstrate that the proposed SBNN-PINN consistently achieves residual losses close to machine precision and relative $L_2$ errors as small as $10^{-17}$. Such accuracy is obtained using only a small number of Bernstein coefficients, significantly reducing the number of optimization variables compared with conventional deep neural networks.

An additional advantage of the proposed method is that the Bernstein basis matrix and the spectral Caputo differentiation matrix are computed only once before the optimization process begins. During training, the network requires only matrix–vector multiplications and evaluations of the nonlinear reaction function. Consequently, each optimization iteration is computationally inexpensive compared with conventional fractional PINNs, where the fractional derivative is repeatedly approximated through numerical quadrature or finite-difference discretization.

Furthermore, the proposed formulation naturally satisfies the initial condition by construction through the expression

$$x_{\text{pred}}(t) = x_0 + t^{\alpha}\mathbf{B}(t)\mathbf{w},$$

thereby eliminating the need for an additional initial-condition penalty term in the loss function. This reduces the complexity of the optimization problem and improves convergence.

Overall, the proposed SBNN-PINN combines the spectral accuracy of Bernstein polynomial approximation, the analytical evaluation of the Caputo fractional derivative, and the efficiency of physics-informed optimization within a unified computational framework. The resulting method achieves superior numerical accuracy, reduced computational cost, enhanced stability, and excellent convergence charac-

 https://doi.org/10.68219/f26md935

teristics, making it an attractive alternative for solving nonlinear fractional differential equations arising in population dynamics and other scientific applications.

# 7. Conclusion

In this paper, we proposed a novel Stable Spectral Bernstein Physics-Informed Neural Network (SBNN-PINN) to solve the nonlinear fractional-order Allee population model based on the Caputo fractional derivative. The proposed method combines the global approximation capabilities of Bernstein spectral polynomials with physics-informed learning to provide the most accurate and stable solution of nonlinear fractional differential equations. Unlike deep neural network-based PINNs, our approach works with a compact Bernstein polynomial expansion in which only (N+1) coefficients are considered, reducing the number of parameters to train and achieving high accuracy and computational efficiency.

A distinguishing feature of the proposed methodology is the analytical construction of the Bernstein spectral differentiation matrix for the Caputo fractional derivative. Since the fractional derivative is evaluated analytically before the optimization stage, repeated numerical approximation of the nonlocal operator during training is completely eliminated. Consequently, the computational cost of each optimization iteration is significantly reduced. Furthermore, the use of logarithmically stabilized Bernstein basis functions prevents numerical overflow and underflow near the interval endpoints, while Chebyshev–Gauss–Lobatto collocation points improve interpolation accuracy and numerical stability. The adopted trial solution,

$$x(t) = x_0 + t^{\alpha} B(t)\mathbf{w},$$

satisfies the prescribed initial condition exactly, eliminating the need for additional penalty terms associated with the initial condition.

From a mathematical point of view the qualitative properties of the fractional-order Allee model were established with positivity, boundedness, equilibrium, and local stability. The positivity and boundedness results imply that the model is biologically relevant during the simulation whereas the equilibrium analysis revealed three biologically relevant equilibrium points. With the Jacobian linearization and Matignon's stability theorem, it was shown that the extinction equilibrium and the carrying-capacity equilibrium are locally asymptotically stable while the Allee threshold is unstable and separates the basins of attraction for extinction and long-term persistence. These results were further supported by numerical evaluation of the Jacobian eigenvalues and phase-line and vector-field analyses to demonstrate the bistable dynamics due to a strong Allee effect.

To verify the proposed numerical methodology, the Method of Manufactured Solutions (MMS) was used where the forcing function is given for the manufactured solution $x(t) = x_0 + t^{\alpha}$. Thanks to the exact analytical benchmark, we were able to test the numerical accuracy of the proposed SBNN-PINN independently of the biological nature. The computed solutions were in very good agreement with the manufactured solution throughout the entire computational domain, which confirms the correctness of the analytical spectral operators and the proposed physics-based optimization procedure.

We have performed very extensive numerical experiments in fractional order $\alpha = 0.1$ to $\alpha = 0.9$. In all cases the proposed algorithm was able to produce very small residual losses and approximation errors. Residual losses were around $10^{-29}$ and maximum absolute error was about $10^{-15}$ for $\alpha = 0.9$ and this is machine-precision accuracy in double precision arithmetic. Even for the strongest memory regime ($\alpha = 0.1$) the maximum absolute error was less than $10^{-7}$.

The effect of the degree of Bernstein polynomial was also studied in detail. The numerical results showed that low degree Bernstein expansions ((N=2)-(5)) are sufficient to achieve machine-precision accuracy with a few trainable parameters. Increasing the degree of Bernstein polynomial in the range beyond this region does not result in the numerical approximation to become more accurate and is instead

 https://doi.org/10.68219/f26md935

a deterioration in accuracy. The reason is that higher degree of Bernstein polynomial bases are more conditioned and so there is more optimization associated with finite precision arithmetic. Therefore, the proposed SBNN-PINN is able to achieve a good balance between the computational cost and the accuracy of the approximation due to the compact spectral representation.

The qualitative comparison of numerical techniques such as finite difference methods, predictor-corrector algorithms, conventional fractional PINNs and Bernstein neural networks shows the advantages of our proposed approach. The combination of analytical spectral operators, logarithmically stabilized Bernstein basis functions, Chebyshev-Gauss-Lobatto collocation and hybrid Adam–L-BFGS optimization results in fast convergence, low computational complexity, low memory, and highly accurate numerical solutions using only a few optimization variables.

We have performed well in the numerical analysis but we have several limitations to note. The numerical study is based on a single nonlinear fractional ordinary differential equation and we use the Method of Manufactured Solutions for numerical verification. Only the Caputo fractional derivative is considered and we have limited the numerical studies to one-dimensional population dynamics. However, all these limitations are not a problem and the proposed approach is general enough to address much more complex fractional dynamics.

In the future we will extend our SBNN-PINN to coupled systems of nonlinear fractional differential equations, variable order and distributed order fractional models, stochastic and delay fractional systems and multidimensional fractional partial differential equations. Additional work will focus on adaptive Bernstein basis refinement, efficient parallel GPU implementations, inverse parameter estimation and applications to realistic models of ecology, epidemiology, engineering, and other areas of applied science.

In conclusion, the proposed SBNN-PINN presents a mathematically rigorous, computationally efficient and very accurate method of solving nonlinear Caputo fractional differential equations. The combination of spectral Bernstein approximation, analytical fractional operators and physics-informed optimization approach provides an alternative for conventional fractional numerical solvers and is an excellent base for future work in the scientific area of machine learning and fractional computational mathematics.

# References

[1] K. Oldham and J. Spanier, *The fractional calculus theory and applications of differentiation and integration to arbitrary order*, vol. 111. Elsevier, 1974.

[2] I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, vol. 198. elsevier, 1998.

[3] A. Kilbas, *Theory and applications of fractional differential equations*.

[4] K. Diethelm and N. Ford, "The analysis of fractional differential equations," *Lecture notes in mathematics*, vol. 2004, 2010.

[5] I. Petras, "Fractional calculus and its applications," *Mathematical Modeling with Multidisciplinary Applications*, pp. 355–396, 2012.

[6] C. Li and F. Zeng, *Numerical methods for fractional calculus*. CRC Press, 2015.

[7] C. Lubich, "Discretized fractional calculus," *SIAM Journal on Mathematical Analysis*, vol. 17, no. 3, pp. 704–719, 1986.

[8] M. Caputo, "Linear models of dissipation whose q is almost frequency independent—ii," *Geophysical journal international*, vol. 13, no. 5, pp. 529–539, 1967.

 https://doi.org/10.68219/f26md935

[9] W. C. Allee, "Co-operation among animals," *American Journal of Sociology*, vol. 37, no. 3, pp. 386–398, 1931.

[10] P. A. Stephens and W. J. Sutherland, "Consequences of the allee effect for behaviour, ecology and conservation," *Trends in ecology & evolution*, vol. 14, no. 10, pp. 401–405, 1999.

[11] F. Courchamp, T. Clutton-Brock, and B. Grenfell, "Inverse density dependence and the allee effect," *Trends in ecology & evolution*, vol. 14, no. 10, pp. 405–410, 1999.

[12] B. Dennis, "Allee effects: population growth, critical density, and the chance of extinction," *Natural Resource Modeling*, vol. 3, no. 4, pp. 481–538, 1989.

[13] C. M. Taylor and A. Hastings, "Allee effects in biological invasions," *Ecology letters*, vol. 8, no. 8, pp. 895–908, 2005.

[14] K. Shah, T. Abdeljawad, and H. Alrabaiah, "On coupled system of drug therapy via piecewise equations," *Fractals*, vol. 30, no. 08, p. 2240206, 2022.

[15] K. Shah, B. Abdalla, T. Abdeljawad, and M. A. Alqudah, "A fractal-fractional order model to study multiple sclerosis: a chronic disease," *Fractals*, vol. 32, no. 02, p. 2440010, 2024.

[16] G. G. Lorentz, *Bernstein polynomials*. American Mathematical Soc., 2012.

[17] R. T. Farouki, "The bernstein polynomial basis: A centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[18] J. Rad, S. Kazem, M. Shaban, K. Parand, and A. Yildirim, "Numerical solution of fractional differential equations with a tau method based on legendre and bernstein polynomials," *Mathematical Methods in the Applied Sciences*, vol. 37, no. 3, pp. 329–342, 2014.

[19] M. Alshbool, A. Bataineh, I. Hashim, and O. R. Isik, "Solution of fractional-order differential equations based on the operational matrices of new fractional bernstein functions," *Journal of King Saud University-Science*, vol. 29, no. 1, pp. 1–18, 2017.

[20] J. P. Boyd, *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[21] P. J. Roache, *Verification and validation in computational science and engineering*, vol. 895. Hermosa Albuquerque, NM, 1998.

[22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[23] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[24] G. Pang, L. Lu, and G. E. Karniadakis, "fpinns: Fractional physics-informed neural networks," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2603–A2626, 2019.

[25] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM review*, vol. 63, no. 1, pp. 208–228, 2021.

[26] Z. Ma, J. Hou, W. Zhu, Y. Peng, and Y. Li, "Pmnn: physical model-driven neural network for solving time-fractional differential equations," *Chaos, Solitons & Fractals*, vol. 177, p. 114238, 2023.

[27] J. Hou, Z. Ma, S. Ying, and Y. Li, "Hns: An efficient hermite neural solver for solving time-fractional partial differential equations," *Chaos, Solitons & Fractals*, vol. 181, p. 114637, 2024.

[28] L. Bhan, M. Krstic, and Y. Shi, "Stabilization of nonlinear systems with unknown delays via delay-adaptive neural operator approximate predictors," *arXiv preprint arXiv:2509.26443*, 2025.