Journal of Basrah Researches (Sciences) **52**(1), 1 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.1

Applying the Fractional Cubic Spline Method to Solve Dynamic Systems Numerically

Karwan S. Mohammed[1], Faraidun K. Hamasalh[2]

[1]University of Sulaimani, Department of Mathematics, College of Education, Sulaimani, Iraq.
[2]Sulaimani Polytechnic University, Bakrajo Technical Institute, Sulaimani, Iraq.

A R T I C L E I N F O

Received 28 September 2025
Revised 04 January 2026
Accepted 04 January 2026
Published 30 June 2026

K e y w o r d s :

Spline approximation, Caputo fractional derivative, computational modelling, and convergence analysis.

Citation: K. S. Mohammed, F. K. Hamasalh, J. Basrah Res. (Sci.) **52**(1), 1 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.1

A B S T R A C T

In this work, we present a new numerical technique for solving systems of differential equations in arising dynamical systems that is based on fractional cubic spline interpolation. The spline consists of several dimensions, and one of the fractional dimensions is $\alpha = \frac{5}{3}$. In order to estimate the solution of nonlinear or linear dynamical systems, the suggested method builds a fractional cubic spline with suitable continuity and differentiability constraints and applies it iteratively. We can find the results for each step, and the error rate decreases, which in our sample makes the relative error close to 10^7 However, this method requires tracking the dynamic system at each step, using the previous step, which takes a lot of time. Therefore, we rely on MATLAB here. Better modeling of systems having memory effects or hereditary properties which are prevalent in many physical, biological, and engineering problems is made possible by the addition of fractional variables. The method's correctness and efficacy are demonstrated by numerical trials, which also show strong agreement with analytical or benchmark solutions.

1. Introduction

The behavior of real-world processes as they evolve is explained by mathematical models of dynamic systems .Since ordinary or partial differential equations that regulate these systems are frequently nonlinear and interconnected, it is either difficult or impossible to find analytical solutions. To approximate solutions, numerical techniques like Runge-Kutta, multistep schemes, and finite differences have long been used. However, stiff dynamics, long-term dependencies, and memory effects problems, which are becoming more prevalent in complex biological, engineering, and physical systems, may be difficult for traditional numerical approaches to handle [1–3]. Fractional calculus has garnered significant attention as a powerful modelling tool to address these problems because it applies the traditional ideas of differentiation and integration to non-integer orders [4–5].

Because fractional differential equations (FDEs) have memory and hereditary features, they offer a more accurate representation of many dynamical systems. Because fractional derivatives are nonlocal, fractional differential equations (FDEs) are more challenging to solve numerically, notwithstanding their benefits. Recent studies have responded by concentrating on creating hybrid numerical methods that combine the stability and smoothness of spline-based approximations with the flexibility of fractional

***Corresponding author email:** faraidun.hamasalh@spu.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

calculus. The ability of cubic splines to preserve continuity in the function and its derivatives has made them particularly useful among these. Fractional cubic spline techniques have been presented as reliable substitutes for solving systems of DEs by altering the spline architecture to meet fractional derivative requirements [6–8].

The fractional cubic spline approach has been validated in a number of recent studies using various dynamical model types. Applications include using spline-based approaches to solve coupled nonlinear systems with memory and solving differential equations for delay systems [9–12]. Both recent modelling advances [6–8] and fundamental numerical analysis methods [1–3] serve as the foundation for these breakthroughs. In order to solve differential equation systems that arise in dynamical systems, this research suggests a novel numerical framework that uses fractional cubic spline interpolation. Numerical examples will demonstrate the concept, algorithmic structure, and efficacy of the method.

In this study, we employ a fractional spline characterized by a fractional order parameter of $\alpha = \frac{5}{3}$. Its fractional part is involved in numerical analysis, and its discovery is with the presence of exact solution or cubic spline fragments in the absence of an exact solution. A numerical construction is developed for the analysis of dynamical systems and is subsequently validated through application to a seven-dimensional model. The proposed method demonstrates a reduction in error attributable to the fractional-order formulation, thereby yielding approximations that align more closely with both the reference Runge-Kutta method and known exact solutions in comparable systems. Additionally, this paper presents and discusses several foundational theorems about the properties of the fractional spline. Numerical implementation, facilitated by MATLAB, enables comprehensive system analysis, graphical visualization, and a comparative assessment between the proposed method and the classical Runge-Kutta technique.

1.1 Riemann-Liouville Fractional Derivative:

The Riemann-Liouville fractional derivative [14] of order α (where $\alpha > 0$) is defined as:

$$D^{\alpha}f(x) = \frac{1}{\Gamma(n-\alpha)}\frac{d^n}{dx^n}\int_a^x (x-t)^{n-\alpha-1}f(t)\,dt,$$

1.2 Caputo Fractional Derivative:[14]

The Caputo fractional derivative of order α is given by:

$${}^{C}D^{\alpha}f(x) = \frac{1}{\Gamma(n-\alpha)}\int_a^x (x-t)^{n-\alpha-1}f^{(n)}(t)\,dt.$$

1.3 Fractional Taylor Series:

The fractional Taylor series [18] extends the classical Taylor series to fractional orders, allowing function approximations in terms of fractional derivatives. The expansion of a function $f(x)$ around a point x_0 is given by:

$$f(x) = \sum_{n=0}^{\infty} \frac{D^{\alpha}f(x_0)}{\Gamma(n\alpha+1)}(x-x_0)^{n\alpha}.$$

1.4 Maximum Norm [19]

The set of all bounded functions $f: x \rightarrow R$ is denoted by (x). Given its numerous good qualities, including being an algebra over R and a super space for a number of other well-known spaces, this set is deserving of the name space. The uniform norm of (x), often known as the sup-norm, is defined as follows:

$$\| f \|_{\infty} = \sup_{x\in\Omega} |f(x)|.$$

2. Model Development and Evaluation

This manuscript proposes a numerical method based on fractional cubic spline interpolation. The formulation of this model leverages a key aspect of non-polynomial spline interpolation: using the derivative at a given point to construct the interpolating polynomial.

$$S_i(x) = a + b(x-x_i) + c(x-x_i)^{\frac{5}{3}} + d(x-x_i)^2 + e(x-x_i)^3 \qquad (2.1)$$

and applies it to approximate the solution of a seven-dimensional dynamical system (a typhoid model). Continuity conditions are derived, two local error estimates are stated, and numerical results are compared with a classical Runge–Kutta method on a short time interval. The fractional spline-based polynomial is intended to improve the approximation of memory-dependent and nonlinear systems. By adding a fractional-order term $c(x-x_i)^{\frac{5}{3}}$, this sophisticated formulation enhances conventional integer-order splines and provides more precise control over the curve's smoothness and long-term memory characteristics. In order to minimize residuals between the modeled function and observed data, least squares or variational approaches are used to derive the coefficients a, b, c, d, and e. When modeling complicated systems governed by fractional differential equations, where conventional numerical approaches are inadequate, such a structure is particularly helpful. The baseline slope is provided by the linear term $a+b(x-x_i)$ curvature and asymmetry are introduced by the quadratic and cubic terms $d(x-x_i)^2+e(x-x_i)^3$, which adds memory awareness and captures nonlocal behaviors. Applications of such models can be found in engineering contexts such as structural health monitoring for detecting nonlinear oscillations [13, 16] and biological systems where fractional splines are used to model population dynamics and epidemiology due to hereditary effects and time delays [15, 17]. Even with their adaptability, models with fractional or higher-degree terms need to be carefully adjusted to prevent overfitting. Such spline-based models have been used in practice, and their theoretical underpinnings have been reinforced by recent developments in fractional calculus [14, 16].

2.1 Conditions and Structure of Polynomials

These splines are constructed using polynomial coefficients and are particularly useful in approximating solutions to problems involving fractional derivatives and integrals. Numerical examples and convergence analysis are used to illustrate the accuracy and ease of use of fractional cubic splines. Key elements of the polynomial structure and conditions in fractional cubic splines are listed below.

$$\begin{cases} S_i(x_i)=y_i & S_i(x_{i+1})=y_{i+1} \\ S_i'(x_i)=y'_i & S_i''(x_{i+1})=y''_{i+1} \\ S_i^{\left(\frac{5}{3}\right)}(x_i)=y^{\left(\frac{5}{3}\right)}{}_i \ (x) & \end{cases} \tag{2.2}$$

Theorem2.1: let $s_i(x)$ be a partition of $[a,b]$ and suppose the values y_i , y'_i , y''_i and $y^{\left(\frac{5}{3}\right)}_{\boldsymbol{i}}$ are given all i. Consequently, there exists a single piecewise-defined function s of the type (2.1) on each subinterval $[x_i, x_{i+1}]$ that fulfils the criteria (2.2).

Proof:

Now find the first and second, and fractional derivative $\frac{5}{3}$ by using Caputo Fractional Derivative for $S_i(x)$ we get

$$\begin{cases} S_i(x)=a+b(x-x_i)+c(x-x_i)^{\frac{5}{3}}+d(x-x_i)^2+e(x-x_i)^3 \\ S_i'(x)=b+\frac{5}{3}c(x-x_i)^{\frac{2}{3}}+2d(x-x_i)+3e(x-x_i)^2 \\ S_i^{\frac{5}{3}}(x)=\frac{10}{9}\Gamma\left(\frac{2}{3}\right)\boldsymbol{c}+\frac{6}{\Gamma\left(\frac{1}{3}\right)}\boldsymbol{d}(\boldsymbol{x}-\boldsymbol{x_i})^{\frac{1}{3}}+\frac{27}{2\Gamma\left(\frac{1}{3}\right)}\boldsymbol{e}\,(\boldsymbol{x}-\boldsymbol{x_i})^{\frac{4}{3}} \\ S_i''(x)=2d+6e(x-x_i) \end{cases} \tag{2.3}$$

Using the polynomial structure (2.1) and the conditions (2.2), $S_i(x_i)=y_i$, we get:

$$y_i=a \tag{2.4}$$

Also from $S_i'(x_i)=y'_i$ in equations (2.4) and (2.3), we get $y'_i=b$

Now, from the fractional derivative $S_i^{\frac{5}{3}}(x)$ in equation (2.3) and $S_i^{(\frac{5}{3})}(x_i) = y^{(\frac{5}{3})}{}_i$ in equation(2.2), we get:

$$c = \frac{9\, y^{(\frac{5}{3})}{}_i}{10\,\Gamma(\frac{2}{3})}$$

To build the spline approximation to (2.1), the interval [a, b] is divided into n equal subintervals using the grid points $x_i = a + ih$; $i = 0,1,\ldots,n$ where $h = \frac{b-a}{n}$. Take a look at this restriction. S_i of S for each $[x_i, x_{i+1}]$ subinterval, $i = 0,1,\ldots,n$. Now, using equation (2.1) and $S_i(x_{i+1}) = y_{i+1}$ we obtain:

$$y_{i+1} = a + bh + ch^{\frac{5}{3}} + dh^2 + eh^3 \qquad (2.5)$$

Putting a , b , c in equation (2.5), we get

$$y_{i+1} - y_i - y'_i h - \frac{9\, y^{(\frac{5}{3})}{}_i}{10\,\Gamma(\frac{2}{3})} h^{\frac{5}{3}} = dh^2 + eh^3 \qquad (2.6)$$

Also from $S_i''(x_{i+1}) = y''_{i+1}$, and equation (2.3), we get:

$$y''_{i+1} = 2d + 6eh \qquad (2.7)$$

To find e from equations (2.6) and (2.7), we get:

$$y_{i+1} - y_i - y'_i h - \frac{9\, y^{(\frac{5}{3})}{}_i}{10\,\Gamma(\frac{2}{3})} h^{\frac{5}{3}} = dh^2 + eh^3$$

$$-\frac{1}{2}\, y''_{i+1} h^2 = -dh^2 - 3eh^3$$

$$e = \frac{1}{4h}\, y''_{i+1} - \frac{1}{2h^3} y_{i+1} + \frac{1}{2h^3} y_i + \frac{1}{2h^2} y'_i + \frac{9y^{(\frac{5}{3})}{}_i}{20\,\Gamma(\frac{2}{3}) h^{\frac{4}{3}}} \qquad (2.8)$$

Put the equation (2.8) in equation (2.7) to find d, we get :

$$y''_{i+1} = 2d + 6\left(\frac{1}{4h}\, y''_{i+1} - \frac{1}{2h^3} y_{i+1} + \frac{1}{2h^3} y_i + \frac{1}{2h^2} y'_i + \frac{9y^{(\frac{5}{3})}{}_i}{20\,\Gamma(\frac{2}{3}) h^{\frac{4}{3}}}\right) h$$

$$d = -\frac{1}{4}\, y''_{i+1} + \frac{3}{2h^2} y_{i+1} - \frac{3}{2h^2} y_i - \frac{3}{2h} y'_i - \frac{27y^{(\frac{5}{3})}{}_i}{20\,\Gamma(\frac{2}{3}) h^{\frac{1}{3}}} \qquad (2.9)$$

The coefficients in equation (2.1) have the following form:

$$\begin{cases} a_i = y_i \\ b_i = y'_i \\ c_i = \frac{9\, y^{(\frac{5}{3})}{}_i}{10\,\Gamma(\frac{2}{3})} \\ d_i = -\frac{1}{4}\, y''_{i+1} + \frac{3}{2h^2} y_{i+1} - \frac{3}{2h^2} y_i - \frac{3}{2h} y'_i - \frac{27y^{(\frac{5}{3})}{}_i}{20\,\Gamma(\frac{2}{3}) h^{\frac{1}{3}}} \\ e_i = \frac{1}{4h}\, y''_{i+1} - \frac{1}{2h^3} y_{i+1} + \frac{1}{2h^3} y_i + \frac{1}{2h^2} y'_i + \frac{9y^{(\frac{5}{3})}{}_i}{20\,\Gamma(\frac{2}{3}) h^{\frac{4}{3}}} \end{cases}$$

3. Continuity Condition for Spline Functions

Applying the first derivative continuity at the knots yields the following consistency relations: $S_i^{(m)}(x_i) = S_{i-1}^{(m)}(x_i)$, where m=1.

$$S_i'(x_i) = b_i + \frac{5}{3}c_i(x_i - x_i)^{\frac{2}{3}} + 2d_i(x_i - x_i) + 3e_i(x_i - x_i)^2$$

$$S_i'(x_i) = b_i \ , \quad S_i'(x_i) = y'_i \tag{3.1}$$

Also

$$S'_{i-1}(x_i) = b_{i-1} + \frac{5}{3}c_{i-1}(x_i - x_{i-1})^{\frac{2}{3}} + 2d_{i-1}(x_i - x_{i-1}) + 3e_{i-1}(x_i - x_{i-1})^2$$

$$S'_{i-1}(x_i) = b_{i-1} + \frac{5}{3}c_{i-1}h^{\frac{2}{3}} + 2d_{i-1}h + 3e_{i-1}h^2$$

$$S'_{i-1}(x_i) = y'_{i-1} + \frac{5}{3}\left(\frac{9\,y^{\left(\frac{5}{3}\right)}_{i-1}}{10\,\Gamma\left(\frac{2}{3}\right)}\right)h^{\frac{2}{3}} + 2\left(-\frac{1}{4}\,y''_i + \frac{3}{2h^2}y_i - \frac{3}{2h^2}y_{i-1} - \frac{3}{2h}y'_{i-1} - \frac{27y^{\left(\frac{5}{3}\right)}_{i-1}}{20\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\right)h +$$

$$3\left(\frac{1}{4h}\,y''_i - \frac{1}{2h^3}y_i + \frac{1}{2h^3}y_{i-1} + \frac{1}{2h^2}y'_{i-1} + \frac{9y^{\left(\frac{5}{3}\right)}_{i-1}}{20\,\Gamma\left(\frac{2}{3}\right)h^{\frac{4}{3}}}\right)h^2$$

$$S'_{i-1}(x_i) = \frac{3}{2h}y_i - \frac{3}{2h}y_{i-1} - \frac{1}{2}y'_{i-1} + \frac{3\,y^{\left(\frac{5}{3}\right)}_{i-1}}{20\,\Gamma\left(\frac{2}{3}\right)}h^{\frac{2}{3}} + \frac{h}{4}\,y''_i \tag{3.2}$$

At this point, using the continuity condition and equations (3.1) and (3.2), we obtain:

$$S_i'(x_i) = S'_{i-1}(x_i)$$

$$y'_i = \frac{3}{2h}y_i - \frac{3}{2h}y_{i-1} - \frac{1}{2}y'_{i-1} + \frac{3\,y^{\left(\frac{5}{3}\right)}_{i-1}}{20\,\Gamma\left(\frac{2}{3}\right)}h^{\frac{2}{3}} + \frac{h}{4}\,y''_i$$

$$\frac{3}{2h}y_i - y'_i + \frac{h}{4}\,y''_i = \frac{3}{2h}y_{i-1} + \frac{1}{2}y'_{i-1} - \frac{3\,y^{\left(\frac{5}{3}\right)}_{i-1}}{20\,\Gamma\left(\frac{2}{3}\right)}h^{\frac{2}{3}}$$

$$y_i - \frac{2h}{3}y'_i + \frac{h^2}{6}\,y''_i = y_{i-1} + \frac{h}{3}y'_{i-1} - \frac{y^{\left(\frac{5}{3}\right)}_{i-1}}{10\,\Gamma\left(\frac{2}{3}\right)}h^{\frac{5}{3}} \tag{3.3}$$

Based on the continuity requirements of the spline function employed for approximating dynamic system models, this rule delivers a high-accuracy estimate for y_i. It achieves this by incorporating the function's first derivative, second derivative, and a fractional derivative of order $\frac{5}{3}$. This formulation enables significantly more precise numerical solutions for differential systems, particularly in representing complex dynamic behavior.

4. Convergence Analysis

This section discusses some important theorems, as well as the problem of fractional cubic spline convergence.

Theorem 4.1: Let $y_i(x) = C^m([a,b])$ be a function with integer smoothness order $m \geq 0$, and let $S_i(x)$ denotes the fractional cubic spline constructed from values $y_i(x)$ and integer-order derivatives $y^m(x_i)$ at nodes $\{x_i\}_{i=0}^n$. Then the error estimation $\left|S_i(x_{i+1}) - y_i(x)\right| = \left|h y_i^{(1)}(\varepsilon_1)\right|$, where ε_1 belong to $[x_{i+1}, x_i]$

Proof:

Now first find $S_i(x)$ from equation (2.3), we get

$S_i(x) = a + b(x - x_i) + c(x - x_i)^{\frac{5}{3}} + d(x - x_i)^2 + e(x - x_i)^3$

Since $x = x_{i+1}$ and $h = x_{i+1} - x_i$,

$$S_i(x) = a + bh + ch^{\frac{5}{3}} + dh^2 + eh^3 \quad (4.1)$$

Putting the coefficients a, b, c, d, and e in equation (4.1), we get

$$S_i(x_{i+1}) = y_i + y'_i h + \frac{5}{3}\left(\frac{9\,y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)}\right)h^{\frac{2}{3}} + 2\left(-\frac{1}{4}\,y''_{i+1} + \frac{3}{2h^2}y_{i+1} - \frac{3}{2h^2}y_i - \frac{3}{2h}y'_i - \frac{27y^{\left(\frac{5}{3}\right)}_i}{20\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\right)h +$$

$$3\left(\frac{1}{4h}\,y''_{i+1} - \frac{1}{2h^3}y_{i+1} + \frac{1}{2h^3}y_i + \frac{1}{2h^2}y'_i + \frac{9y^{\left(\frac{5}{3}\right)}_i}{20\,\Gamma\left(\frac{2}{3}\right)h^{\frac{4}{3}}}\right)h^2$$

$$S_i(x_{i+1}) = y_{i+1} \quad (4.2)$$

By expanding y_{i+1} using the Taylor series, we get:

$$y_{i+1} = y_i + hy_i^{(1)}(\varepsilon) \quad (4.3)$$

From (4.2) and (4.3) we get:

$$S_i(x_{i+1}) = y_i + hy_i^{(1)}(\varepsilon) \quad (4.4)$$

Now from (4.4) and $\left|S_i(x_{i+1}) - y_i(x)\right|$ and, we get:

$$\left|S_i(x_{i+1}) - y_i(x)\right| = \left|y_i + hy_i^{(1)}(\varepsilon) - y_i(x)\right| .$$

Since $y_i = y_i(x)$

$\left|S_i(x_{i+1}) - y_i(x)\right| = \left|hy_i^{(1)}(\varepsilon)\right|$, where ε_1 belong to $[x_{i+1}, x_i]$

Theorem 4.2: Let s(x) be fractional cubic spline and $x = x_i$, then error estimation

$$\left|S_i''(x_i) - y''_i(x)\right| \le \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\left\|y^{\left(\frac{5}{3}\right)}_i\right\| + \frac{h}{2}\,w(x), \quad \text{where} \quad w(x) = \left\|y_i^{(3)}(\varepsilon_1) - y_i^{(3)}(\varepsilon_2)\right\| \quad , \quad \text{and}$$

ε_1 , ε_1 belong to $[x_{i+1}, x_i]$.

Proof:

Now first find $S_i''(x)$ from equation (2.3), we get

$S_i''(x_i) = 2d + 6e(x_{i+1} - x_i)$

$S_i''(x_i) = 2d + 6eh$

$$S_i''(x_i) = 2\left(-\frac{1}{4}\,y''_{i+1} + \frac{3}{2h^2}y_{i+1} - \frac{3}{2h^2}y_i - \frac{3}{2h}y'_i - \frac{27y^{\left(\frac{5}{3}\right)}_i}{20\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\right)$$

$$S_i''(x_i) = -\frac{1}{2}\,y''_{i+1} + \frac{3}{h^2}y_{i+1} - \frac{3}{h^2}y_i - \frac{3}{h}y'_i - \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}} \quad (4.5)$$

By expanding y_{i+1} and y''_{i+1} using the Taylor series, we get:

$$y''_{i+1} = y''_i + h\,y_i^{(3)}(\varepsilon_1) \quad (4.6)$$

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y_i^{(3)}(\varepsilon_2) \quad (4.7)$$

$$S_i''(x_i) = -\frac{1}{2}\left(y''_i + h\,y_i^{(3)}(\varepsilon_1)\right) + \frac{3}{h^2}\left(y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y_i^{(3)}(\varepsilon_2)\right) - \frac{3}{h^2}y_i - \frac{3}{h}y'_i - \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}$$

$$S_i''(x_i) = y''_i - \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}} - \frac{h}{2}\,y_i^{(3)}(\varepsilon_1) + \frac{h}{2}y_i^{(3)}(\varepsilon_2) \quad , \tag{4.8}$$

from (4.8) and $\left|S_i''(x) - y''_i(x)\right|$, we get

$$\left|S_i''(x_i) - y''_i(x)\right| = \left|y''_i(x) - \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}} - \frac{h}{2}\,y_i^{(3)}(\varepsilon_1) + \frac{h}{2}y_i^{(3)}(\varepsilon_2) - y''_i(x)\right|$$

$$\left|S_i''(x_i) - y''_i(x)\right| = \left|-\frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}} - \frac{h}{2}\,y_i^{(3)}(\varepsilon_1) + \frac{h}{2}y_i^{(3)}(\varepsilon_2)\right|$$

$$\left|-\frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}} - \frac{h}{2}\,y_i^{(3)}(\varepsilon_1) + \frac{h}{2}y_i^{(3)}(\varepsilon_2)\right| \le \left|\frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\right| + \left|-\frac{h}{2}\,y_i^{(3)}(\varepsilon_1) + \frac{h}{2}y_i^{(3)}(\varepsilon_2)\right|$$

$$\le \left|\frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\right| + \frac{h}{2}\left|-y_i^{(3)}(\varepsilon_1) + y_i^{(3)}(\varepsilon_2)\right| \ ,$$

$$\le \frac{27y^{\left(\frac{5}{3}\right)}_i}{10\,\Gamma\left(\frac{2}{3}\right)h^{\frac{1}{3}}}\left\|y^{\left(\frac{5}{3}\right)}_i\right\| + \frac{h}{2}\,w(x)$$

5. Numerical Discussion

The effectiveness of the suggested construction equation (3.3) for solving differential equation systems in dynamic system modelling using a fractional cubic spline approach is validated by the numerical results (Tables 5.1 and 5.3**)** which show the results of the method. This method uses fractional derivatives in spline-based approximations, which makes it possible to accurately show complex and nonlinear behaviours. We use MATLAB to do the method, which is the way we need to hunt the system seven by seven in each step, because MATLAB and the fractional part will be part of the system in numbers that we find in different steps, because of the spline pieces. The obtained approximations are smooth and stable, and the numerical errors are significantly smaller than those produced by the classical method. In general, this combination of fractional cubic splines and MATLAB makes it easy and powerful to look at and model dynamic systems. We compare the spline results with those obtained by the Runge–Kutta method (Table 5.2) (Table 5.4), and compute the relative absolute errors between them (Table 5.3) (Table 5.6). (Figure 5.1) (Figure 5.1) also shows an analytical comparison of the two methods**.**

5.1 Dynamic Model: This system is taken from [20], and its parameters are taken from the same source. We analyze the system in two cases, the first case h=0.1 and the second case h=0.05

$$\frac{dv}{dt} = 33 + 0.26s + 0.22r - 0.8841v$$
$$\frac{ds}{dt} = 67 + 0.13v + 0.7204r - 4.7266s$$
$$\frac{de}{dt} = 2.900625s - 0.8221e$$
$$\frac{dn}{dt} = 1.561875s + 0.142e - 0.8573n$$
$$\frac{dr}{dt} = 0.851n + 0.676e - 0.9445r$$
$$\frac{dk}{dt} = 0.75v - 0.0041k$$
$$\frac{db}{dt} = 0.0818e + 0.0712n - 0.0645b$$

Where $t = (0,1)$, $v(0) = 300, s(0) = 1000,\ e(0) = 50$,
$n(0) = 25,\quad r(0) = 23 \quad k(0) = 120 \quad b(0) = 500$

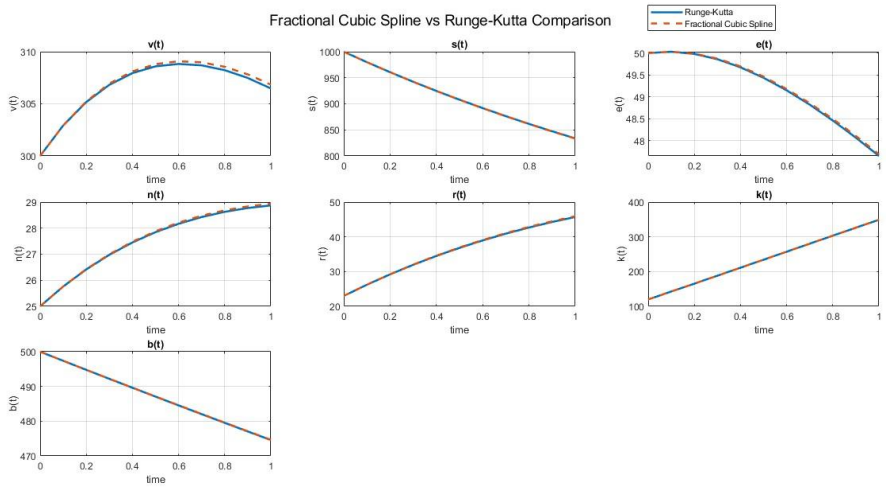

Figure 5.1 The approximate solution of each fractional cubic spline and Runge-Kutta. The figure shows the approximate solutions from both methods plotted together.

Case 1 / where h=0.1. This system uses construction (3.3) and uses Runge-Kutta to find the approximate solution. Each of the solutions is shown in Tables 5.1, Table 5.2, and Figure 5.1 shows both methods. Table 5.3 then shows the absolute relative error between the two analyses.

Table 5.1: The approximate solution of the system using a fractional cubic spline.

t	v(t)	s(t)	e(t)	n(t)	r(t)	k(t)	b(t)
0	300	1000	50	25	23	120	500
0.1	302.921277	979.896034	50.031248	25.760900	26.216094	142.560086	497.373441
0.2	305.224754	960.631450	49.987492	26.425140	29.218048	165.290904	494.775689
0.3	306.933631	942.207973	49.872418	26.994540	31.998431	188.162302	492.199329
0.4	308.108643	924.581743	49.694274	27.476927	34.563865	211.133929	489.642322
0.5	308.807228	907.708535	49.460831	27.879918	36.922310	234.168826	487.103132
0.6	309.082147	891.545686	49.179207	28.210670	39.082381	257.233744	484.580490
0.7	308.981408	876.052579	48.855863	28.475838	41.053064	280.298988	482.073316
0.8	308.548500	861.190788	48.496635	28.681581	42.843543	303.338157	479.580676
0.9	307.822720	846.924075	48.106776	28.833576	44.463073	326.327874	477.101761
1	306.839508	833.218344	47.691002	28.937051	45.920883	349.247501	474.635865

By using the reconstruction formula (3.3), we were able to successfully obtain approximate solutions in Table 5.1, enabling us to numerically approximate the system's behaviour. The corresponding dynamical systems' exact analytical solution, which offers a precise and closed-form expression for the actual evolution of the variables, has also been found.

Table 5.2: The approximate solution of the system using Runge-Kutta.

t	v(t)	s(t)	e(t)	n(t)	r(t)	k(t)	b(t)
0	300	1000	50	25	23	120	500
0.1	302.920838	979.877011	50.031181	25.760860	26.216105	142.560160	497.373444
0.2	305.170520	960.656849	49.980893	26.416667	29.199024	165.304440	494.768709
0.3	306.822793	942.285992	49.859084	26.976900	31.956716	188.185251	492.184323
0.4	307.944046	924.714128	49.674707	27.450276	34.497769	211.160273	489.618960
0.5	308.594026	907.894003	49.435816	27.844808	36.831221	234.191930	487.071435
0.6	308.826477	891.781267	49.149640	28.167854	38.966412	257.246910	484.540684
0.7	308.689728	876.334324	48.822669	28.426168	40.912855	280.295741	482.025760
0.8	308.227216	861.514185	48.460719	28.625947	42.680127	303.312397	479.525813
0.9	307.477965	847.284327	48.068993	28.772872	44.277777	326.273955	477.040090
1	306.477020	833.610552	47.652143	28.872148	45.715252	349.160273	474.567920

In Table 5.2, the classical Runge–Kutta method has been employed in order to obtain an approximate numerical analysis of the given system. This method enabled us to compute a sequence of approximate values that effectively illustrate the system's dynamic behaviour over time.

Table 5.3: Relative local truncation error between the fractional cubic spline solution and the Runge-Kutta method.

t	Error (v(t))	Error (s(t))	Error (e(t))	Error (n(t))	Error (r(t)))	Error (k(t))	Error (b(t))
0	0	0	0	0	0	0	0
0.1	1.45×10^{-6}	1.94×10^{-5}	1.34×10^{-4}	1.55×10^{-6}	4.2×10^{-7}	5.19×10^{-7}	6.03×10^{-9}
0.2	1.78×10^{-4}	2.64×10^{-5}	1.32×10^{-4}	3.21×10^{-4}	6.52×10^{-4}	8.19×10^{-5}	1.41×10^{-5}
0.3	3.61×10^{-4}	8.28×10^{-5}	2.67×10^{-4}	6.54×10^{-4}	1.31×10^{-3}	1.22×10^{-4}	3.05×10^{-5}
0.4	5.35×10^{-4}	1.43×10^{-4}	3.94×10^{-4}	9.71×10^{-4}	1.92×10^{-3}	1.25×10^{-4}	4.77×10^{-5}
0.5	6.91×10^{-4}	2.04×10^{-4}	5.06×10^{-4}	1.26×10^{-3}	2.47×10^{-3}	9.87×10^{-5}	6.51×10^{-5}
0.6	8.28×10^{-4}	2.64×10^{-4}	6.02×10^{-4}	1.52×10^{-3}	2.98×10^{-3}	5.12×10^{-5}	8.22×10^{-5}
0.7	9.45×10^{-4}	3.22×10^{-4}	6.8×10^{-4}	1.75×10^{-3}	3.43×10^{-3}	1.16×10^{-5}	9.87×10^{-5}
0.8	1.04×10^{-3}	3.75×10^{-4}	7.41×10^{-4}	1.94×10^{-3}	3.83×10^{-3}	8.49×10^{-5}	1.14×10^{-4}
0.9	1.12×10^{-3}	4.25×10^{-4}	7.86×10^{-4}	2.11×10^{-3}	4.18×10^{-3}	1.65×10^{-4}	1.29×10^{-4}
1	1.18×10^{-3}	4.7×10^{-4}	8.15×10^{-4}	2.25×10^{-3}	4.5×10^{-3}	2.5×10^{-4}	1.43×10^{-4}

In (Table 5.3), we present the relative local truncation errors obtained when comparing the numerical results of our fractional cubic spline method with those generated by the classical Runge–Kutta method. The relative local truncation error estimator (REE) is defined as

$\text{REE} = \left|\frac{y_{n+1}-y^*_{n+1}}{y_{n+1}}\right|$ You can find the REE for a typhoid model by comparing the Runge-Kutta solutions y^* the model with the fractional cubic spline. The values in the table reflect the degree of accuracy of the spline-based technique at each step, showing how closely it approximates the more established Runge–Kutta solution. Small error values indicate that the spline method successfully captures the system's behaviour with high precision, while slightly larger values highlight intervals where the approximation deviates more significantly. Overall, the table demonstrates the effectiveness of the fractional cubic spline approach in minimizing local truncation errors throughout the interval of integration.

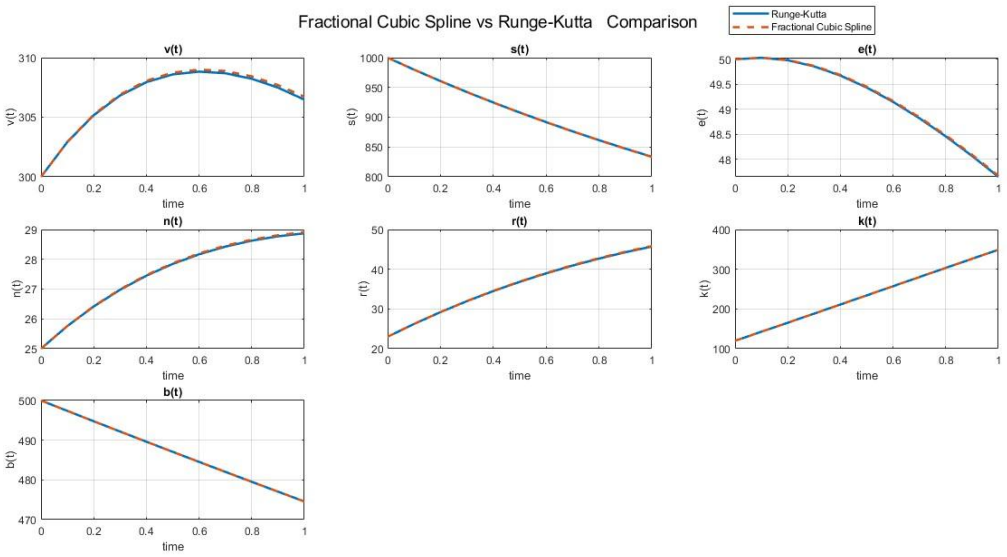

Figure 5.2: The approximate solution of each fractional cubic spline and Runge-Kutta. The figure shows the approximate solutions from both methods plotted together.

Case 2 / where h=0.05. This system uses construction (3.3) and uses Runge-Kutta to find the approximate solution. Each of the solutions is shown in Tables 5.4 Table 5.5, and Figure 5.2 shows both solutions. Table 5.6 then shows the absolute relative error between the two analyses.

Table 5.4: The approximate solution of the system using a fractional cubic spline.

t	v(t)	s(t)	e(t)	n(t)	r(t)	k(t)	b(t)
0	300	1000	50	25	23	120	500
0.1	302.935352	979.869853	50.032846	25.763004	26.220965	142.556241	497.372982
0.2	305.219222	960.617754	49.986413	26.423984	29.216502	165.293192	494.767102
0.3	306.906373	942.211023	49.868437	26.989665	31.988710	188.169543	492.181457
0.4	308.060289	924.602738	49.687533	27.468323	34.545122	211.143911	489.614809
0.5	308.739497	907.747312	49.451616	27.867763	36.894165	234.178990	487.066011
0.6	308.997232	891.601339	49.167865	28.195242	39.044765	257.241450	484.534018
0.7	308.881683	876.123774	48.842763	28.457465	41.006130	280.301649	482.017891
0.8	308.436353	861.275939	48.482142	28.660611	42.787610	303.333312	479.516784
0.9	307.700459	847.021466	48.091240	28.810363	44.398587	326.313222	477.029943
1	306.709307	833.326203	47.674749	28.911942	45.848384	349.220937	474.556692

By using the reconstruction formula (3.3), we were able to successfully obtain approximate solutions in Table 5.4, enabling us to numerically approximate the system's behaviour. The corresponding dynamical systems' exact analytical solution, which offers a precise and closed-form expression for the actual evolution of the variables, has also been found.

Table 5.5: The approximate solution of the system using Runge-Kutta.

t	v(t)	s(t)	e(t)	n(t)	r(t)	k(t)	b(t)
0	300.0	1000.0	50.0	25.0	23.0	120.0	500.0
0.1	302.920844	979.877009	50.031182	25.760861	26.21610	142.560155	497.373443
0.2	305.170533	960.656847	49.980894	26.416668	29.199021	165.304431	494.768709
0.3	306.82281	942.285988	49.859086	26.976901	31.956712	188.185238	492.184322
0.4	307.944067	924.714123	49.674710	27.450278	34.497764	211.160258	489.618960
0.5	308.59405	907.893997	49.435819	27.844810	36.831215	234.191913	487.071434
0.6	308.826503	891.78126	49.149643	28.167856	38.966407	257.246892	484.540684
0.7	308.689755	876.33431	48.822673	28.426170	40.912850	280.295721	482.025759
0.8	308.227243	861.514176	48.460723	28.625950	42.680122	303.312377	479.525813
0.9	307.477993	847.284317	48.068997	28.772875	44.277772	326.273934	477.040090
1	306.477049	833.610541	47.652146	28.872151	45.715248	349.160252	474.567919

In Table 5.5: the classical Runge–Kutta method has been employed in order to obtain an approximate numerical analysis of the given system. This method enabled us to compute a sequence of approximate values that effectively illustrate the system's dynamic behavior over time.

Table 5.6: Relative local truncation error between the fractional cubic spline solution and the Runge-Kutta method.

t	Error (v(t))	Error (s(t))	Error (e(t))	Error (n(t))	Error (r(t)))	Error (k(t))	Error (b(t))
0	0	0	0	0	0	0	0
0.1	4.79×10^{-5}	7.30×10^{-6}	3.33×10^{-5}	8.31×10^{-5}	1.86×10^{-4}	2.75×10^{-5}	9.27×10^{-7}
0.2	1.59×10^{-4}	4.07×10^{-5}	1.10×10^{-4}	2.77×10^{-4}	5.99×10^{-4}	6.80×10^{-5}	3.25×10^{-6}
0.3	2.72×10^{-4}	7.95×10^{-5}	1.88×10^{-4}	4.73×10^{-4}	1.01×10^{-3}	8.34×10^{-5}	5.82×10^{-6}
0.4	3.77×10^{-4}	1.21×10^{-4}	2.58×10^{-4}	6.57×10^{-4}	1.37×10^{-3}	7.74×10^{-5}	8.48×10^{-6}
0.5	4.71×10^{-4}	1.62×10^{-4}	3.20×10^{-4}	8.24×10^{-4}	1.71×10^{-3}	5.52×10^{-5}	1.11×10^{-5}
0.6	5.52×10^{-4}	2.02×10^{-4}	3.71×10^{-4}	9.72×10^{-4}	2.01×10^{-3}	2.12×10^{-5}	1.38×10^{-5}
0.7	6.22×10^{-4}	2.40×10^{-4}	4.11×10^{-4}	1.10×10^{-3}	2.28×10^{-3}	2.11×10^{-5}	1.63×10^{-5}
0.8	6.78×10^{-4}	2.76×10^{-4}	4.41×10^{-4}	1.21×10^{-3}	2.52×10^{-3}	6.90×10^{-5}	1.88×10^{-5}
0.9	7.23×10^{-4}	3.10×10^{-4}	4.63×10^{-4}	1.30×10^{-3}	2.73×10^{-3}	1.20×10^{-4}	2.13×10^{-5}
1	7.58×10^{-4}	3.40×10^{-4}	4.74×10^{-4}	1.38×10^{-3}	2.91×10^{-3}	1.74×10^{-4}	2.37×10^{-5}

In Table 5.6, we present the relative local truncation errors obtained when comparing the numerical results of our fractional cubic spline method with those generated by the classical Runge–Kutta method.

We have taken the steps h = 0.1, 0.2, 0.3,... 1 while h = 0.05 and the absolute relative error in Table 5.6. Compared to Table 5.3, the errors are generally smaller by reducing h, which means that the smaller h the error decreases significantly .

6. Conclusion

One potent and successful method for addressing modelling difficulties in dynamic systems is the fractional cubic spline methodology. This approach creates a strong foundation for capturing intricate dynamic behaviours, especially those with memory and hereditary effects, by combining the long-range dependency aspects of fractional calculus with the smooth-fitting characteristics of cubic splines. The fractional cubic spline technique offers greater numerical stability, improved accuracy, and greater versatility than conventional numerical methods. With the increasing complexity of models in disciplines like biology, physics, and engineering, this approach has the potential to greatly enhance theoretical frameworks and real-world applications. This paper demonstrates that the maximum error between the Runge-Kutta method and our proposed numerical approach diminishes as the step size h is refined, as evidenced in Tables (5.3-5.6). This convergence confirms the method's sufficient precision and highlights its innovative contribution. Furthermore, the practical efficacy of the fractional cubic spline approach is examined through two distinct numerical cases. Accompanying graphs visually illustrate both the accuracy and the practicality of the method. Ultimately, the numerical outcomes obtained via the fractional cubic spline interpolation provide robust validation for the theoretical findings presented in this study. The method's usefulness in contemporary scientific computation may be further supported by future research that focuses on choosing the best fractional orders, guaranteeing stability in a variety of scenarios, and applying it to real-time control systems.

References

[1] J. Sunday, A. Shokri, J. A. Kwanamu, and K. Nonlaopon, "Numerical integration of stiff differential systems using non-fixed step-size strategy," *Symmetry*, vol. 14, no. 8, p. 1575, 2022, doi: 10.3390/sym14081575.

[2] M. Fakharany, M. M. El-Borai, and M. S. Abu Ibrahim, "Numerical analysis of finite difference schemes arising from time-memory partial integro-differential equations," *Frontiers in Applied Mathematics and Statistics*, vol. 8, p. 1055071, 2022, doi: 10.3389/fams.2022.1055071.

[3] K. Diethelm, R. Garrappa, and M. Stynes, "Good (and not so good) practices in computational methods for fractional calculus," *Mathematics*, vol. 8, no. 3, p. 324, 2020, doi: 10.3390/math8030324.

[4] J. E. Macías-Díaz, "Fractional calculus—theory and applications," *Axioms*, vol. 11, 2022, doi: 10.3390/axioms11020043.

[5] S. Raubitzek, K. Mallinger, and T. Neubauer, "Combining fractional derivatives and machine learning: A review," *Entropy*, vol. 25, no. 1, p. 35, 2022, doi: 10.3390/e25010035.

[6] H. M. Srivastava, P. O. Mohammed, J. L. Guirao, and Y. S. Hamed, "Some higher-degree Lacunary fractional splines in the approximation of fractional differential equations," *Symmetry*, vol. 13, no. 3, p. 422, 2021, doi: 10.3390/sym13030422.

[7] O. P. Agrawal, O. Defterli, and D. Baleanu, "Fractional optimal control problems with several state and control variables," *Journal of Vibration and Control*, vol. 16, no. 13, pp. 1967-1976, 2010, doi: 10.1177/1077546309353361.

[8] F. K. Hamasalh and M. A. Headayat, "The applications of non-polynomial spline to the numerical solution for fractional differential equations," in *AIP Conference Proceedings*, vol. 2334, no. 1, p. 060014, AIP Publishing LLC, 2021, doi: 10.1063/5.0042319.

[9] M. A. Yousif and F. K. Hamasalh, "A hybrid non-polynomial spline method and conformable fractional continuity equation," *Mathematics*, vol. 11, no. 17, p. 3799, 2023, doi: 10.3390/math11173799.

[10] A. Atangana and D. Baleanu, "New fractional derivatives with nonlocal and non-singular kernel: theory and application to heat transfer model," arXiv preprint, arXiv:1602.03408, 2016, doi: 10.48550/arXiv.1602.03408.

[11] M. A. Yousif, F. K. Hamasalh, A. Zeeshan, and M. Abdelwahed, "Efficient simulation of Time-Fractional Korteweg-de Vries equation via conformable-Caputo non-Polynomial spline method," *PLoS ONE*, vol. 19, no. 6, p. e0303760, 2024, doi: 10.1371/journal.pone.0303760.

[12] A. Pedas and E. Tamme, "On the convergence of spline collocation methods for solving fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 12, pp. 3502-3514, 2011, doi: 10.1016/j.cam.2010.10.054.

[13] Z. M. Alaofi, "Numerical Treatment of Some Types of Nonlinear Partial Differential Equations," Ph.D. dissertation, Victoria University, 2023. [Online]. Available: https://vuir.vu.edu.au/id/eprint/47233

[14] D. Baleanu, K. Diethelm, E. Scalas, and J. J. Trujillo, *Fractional Calculus: Models and Numerical Methods*. World Scientific, 2012, doi: 10.1142/10044.

[15] K. S. Nisar, M. Farman, M. Abdel-Aty, and C. Ravichandran, "A review of fractional-order models for plant epidemiology," *Prog. Fract. Differ. Appl.*, vol. 10, no. 3, pp. 489-521, 2024, doi: 10.18576/pfda/100313.

[16] M. D. Ortigueira and J. T. Machado, "What is a fractional derivative?," *Journal of Computational Physics*, vol. 293, pp. 4-13, 2015, doi: 10.1016/j.jcp.2014.07.019.

[17] K. Diethelm, *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*, Lecture Notes in Mathematics, vol. 2004. Springer, 2010, doi: 10.1007/978-3-642-14574-2.

[18] D. Caratelli, P. Natalini, and P. E. Ricci, "Fractional differential equations and expansions in fractional powers," *Symmetry*, vol. 15, no. 10, p. 1842, 2023, doi: 10.3390/sym15101842.

[19] L. D. Abreu and H. G. Feichtinger, "Function spaces of polyanalytic functions," in *Harmonic and Complex Analysis and its Applications*, Springer, 2013, pp. 1-38, doi: 10.1007/978-3-319-01806-5_1.

[20] A. Kailan Suhuyini and B. Seidu, "A mathematical model on the transmission dynamics of typhoid fever with treatment and booster vaccination," *Frontiers in Applied Mathematics and Statistics*, vol. 9, 2023, doi: 10.3389/fams.2023.1151270.

Journal of Basrah Researches (Sciences) **52**(1), 1 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.1

استخدام طريقة المنحنى المكعب الكسري لحل الأنظمة الديناميكية عدديًا

كاروان سليم محمد[1]، فريدون قادر حمه صالح[2]

[1]جامعة السليمانية، قسم الرياضيات، كلية التربية، السليمانية، العراق

[2]جامعة السليمانية التقنيه، معهد بكرجو التقني، السليمانية، العراق

معلومات البحث

الاستلام 28 ايلول 2025
المراجعة 04 كانون الثاني 2026
القبول 04 كانون الثاني 2026
النشر 30 حزيران 2026

الكلمات المفتاحية

سبلاين التقريب، مشتق كابوتو الكسري، النمذجة الحسابية، تحليل الاستقرار.

الملخص

في هذا البحث ، نقدم تقنية عددية جديدة لحل أنظمة المعادلات التفاضلية في الأنظمة الديناميكية الناشئة، تعتمد على استيفاء الدوال التكعيبية الكسرية. تتكون الدالة التكعيبية من عدة أبعاد، أحدها البعد الكسري $\alpha = \frac{5}{3}$. ولتقدير حل الأنظمة الديناميكية الخطية أو غير الخطية، تقوم الطريقة المقترحة بإنشاء دالة تكعيبية كسرية مع قيود مناسبة للاستمرارية والتفاضل، ثم تطبقها بشكل تكراري. يمكننا إيجاد النتائج لكل خطوة، ويتناقص معدل الخطأ، مما يجعل الخطأ النسبي في عيّنتنا قريبًا من 10^{7}. مع ذلك، تتطلب هذه الطريقة تتبع النظام الديناميكي في كل خطوة، باستخدام الخطوة السابقة، وهو ما يستغرق وقتًا طويلًا. لذلك، نعتمد هنا على برنامج ماتلاب. يُتيح إضافة المتغيرات الكسرية نمذجة أفضل للأنظمة التي تتضمن تأثيرات الذاكرة أو الخصائص الوراثية، الشائعة في العديد من المشكلات الفيزيائية والبيولوجية والهندسية. تم إثبات صحة وفعالية الطريقة من خلال التجارب العددية، والتي أظهرت أيضًا توافقًا قويًا مع الحلول التحليلية أو المعيارية.

Citation: K. S. Mohammed, F. K. Hamasalh, J. Basrah Res. (Sci.) **52**(1), 1 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.1

***Corresponding author email:** faraidun.hamasalh@spu.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq