OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 05 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

SPLINE BASED NUMERICAL ALGORITHM FOR SOLVING BRATU-TYPE EQUATIONS

Saima Latif *
Faculty of Sciences, The Superior University, Lahore, Pakistan

Muhammad Amin
Faculty of Sciences, The Superior University, Lahore, Pakistan

Saima Mushtaq
Faculty of Sciences, The Superior University, Lahore, Pakistan

Nouman Iftikhar
Faculty of Sciences, The Superior University, Lahore, Pakistan

****Corresponding author: Saima Latif (**Saimalatif.31929@gmail.com**)***

Article Info

Abstract

This study explores the use of cubic B-spline collocation method for the numerical solution of well-known Bratu-Type Equations. The method's performance is evaluated against the widely used bvp4c solver. Numerical outcomes are compared to known exact solutions. Findings show that the cubic B-spline method consistently produces highly accurate results across various parameter values, maintaining minimal error even in cases involving strong nonlinearities. The numerical solutions closely align with the analytical solutions, confirming the method's robustness. In addition, the approach demonstrates excellent computational efficiency. Overall, the cubic B-spline collocation method proves to be a reliable, efficient and accurate tool for solving such nonlinear boundary value problems.

Keywords:

Cubic B-spline, Nonlinear-boundary value problems, -dependent solutions, Numerical methods, Computational efficiency, Bvp4c solver, Nonlinear equations.

Introduction

The concept of splines, particularly B-splines (basis splines), has evolved as a powerful tool in numerical analysis, computational geometry and curve fitting. Initially introduced by de Boor in the early 1970s, B-splines provided an efficient and flexible method for interpolating and approximating functions. Unlike polynomial interpolation, which suffers from the phenomenon of oscillation, B-splines offered local control over the shape of the curve, making them ideal for a wide variety of applications [1].

B-splines are piecewise polynomial functions that are defined by a set of control points and a set of knots, which partition the domain of the function. The key advantage of B-splines over traditional polynomial interpolation is that they provide greater stability, particularly in higher dimensions. The use of piecewise polynomials ensures that only a subset of control points affects the shape of the curve at any given location, allowing for better flexibility in adjusting the curve to data [2]. The term "B-spline" originates from the "basis function" that forms the building block of the spline.

The cubic B-spline, a specific type of B-spline, uses piecewise polynomials of degree 3, providing a balance between smoothness and computational efficiency. Early work on cubic B-splines can be traced back to the 1960s, when they were utilized in computer-aided design (CAD) and curve fitting. By the 1970s and 1980s, cubic B-splines became widely adopted in engineering and computer graphics, due to their desirable properties such as smoothness (C^2 continuity) and simplicity of computation [3].

In the 1980s, Farin (1986) [4] extended the use of cubic B-splines in various fields such as surface modeling and computer-aided design. Cubic B-splines were found to be particularly useful in applications involving smooth curve representation, such as in automotive body design, where smooth transitions between curve segments are critical. The development of efficient numerical methods for evaluating and constructing cubic B-splines further enhanced their popularity. Over time, the computational efficiency of cubic B-splines has been greatly improved. One significant contribution in this area was the development of collocation methods, which involve solving differential equations using B-splines as basis functions. The collocation method with cubic B-splines has been applied to a variety of problems, including boundary value problems (BVPs) and partial differential equations (PDEs), particularly for nonlinear systems. These methods have been advantageous due to their ability to handle complex geometries and nonlinearity, while maintaining high accuracy [5].

In recent years, further advancements have been made in applying cubic B-splines to more specialized fields. For instance, the method has been adapted to solve problems in computational fluid dynamics, structural analysis and image processing. In particular, cubic B-splines have been effectively used in the numerical solution of nonlinear boundary value problems, where they provide both accuracy and computational efficiency [6, 7].

A prominent area of application for cubic B-splines has been in the numerical solution of nonlinear boundary value problems (BVPs). These types of problems, often arising in fields such as physics, engineering and finance, require robust numerical methods due to their complexity. The cubic B-spline method has gained traction due to its ability to handle nonlinearities with greater stability and less computational cost compared to traditional methods like finite difference and finite element methods.

Recent research has focused on improving the stability, convergence, and error estimation of cubic B-spline collocation methods, making them more suitable for large-scale, high-dimensional problems. The method's adaptability in both one-dimensional and multi-dimensional settings has contributed to its widespread use in various domains. Furthermore, with the advent of high-performance computing, cubic B-splines are increasingly being used for solving large, nonlinear BVPs that require precise numerical solutions in real-time [8].

The cubic B-spline collocation method continues to evolve as a powerful and versatile tool in numerical analysis. From its early use in computer-aided design to its current application in solving nonlinear boundary value problems, cubic B-splines have proven to be a robust method for obtaining accurate numerical solutions. Their flexibility, computational efficiency and ability to handle nonlinear systems make them a valuable tool in modern scientific and engineering computations. With further development, cubic B-splines hold the potential to address more complex, multi-dimensional problems across various domains.

2. Background of Bratu Type Boundary Value Problem

The Bratu-type Boundary Value Problem (BVP) is a classical nonlinear differential equation that has captured considerable interest in the mathematical community due to its unique characteristics and wide applicability. This problem is represented by the following equation:

$$u''(x)+\lambda\, e^{u(x)} = 0$$
$$u(0)=0, \qquad u(1)=0$$

Here, $x(t)$ is the unknown function and λ is a positive parameter. The nonlinearity in this equation, arising from the exponential term, makes it an ideal candidate for studying the existence and multiplicity of solutions, as well as their stability. The Bratu equation is significant because it exhibits different solution behaviors depending on the value of the parameter λ. For certain values of λ, the problem may not have real solutions, while for others, multiple solutions may exist [9].

The solution behavior of the Bratu BVP is highly sensitive to the value of the parameter λ. When λ exceeds a critical value λ_c, the equation does not yield any real solution. If $\lambda = \lambda_c$, there exists a unique solution, while for values less than λ_c, two distinct solutions are typically found. These variations in the solution based on the parameter demonstrate the equation's role in modeling real-world systems where thresholds or critical points define the transition between different regimes [10]. The Bratu BVP serves as a classic example of nonlinear equations with parameter-dependent solutions and has been used extensively to demonstrate the behavior of solutions in various types of boundary value problems.

The Bratu-type problem has found numerous applications in fields such as reactor physics, combustion theory and biology. In reactor physics, the equation models the temperature distribution in a nuclear reactor, where the rate of reaction depends exponentially on temperature. Similarly, in combustion theory, the Bratu BVP is used to describe the ignition and propagation of flames, where the reaction rate follows an exponential dependence on temperature [11]. In biological systems, the equation can be used to model the dynamics of population growth or disease spread, where interactions between species or individuals exhibit nonlinear behavior. The wide range of applications highlights the importance of solving the Bratu-type problem accurately.

Solving the Bratu BVP analytically is challenging due to its nonlinear nature. As a result, many numerical techniques have been developed to approximate solutions. Among the most common methods are the shooting method and finite difference methods. These techniques transform the boundary value problem into an initial value problem, which can be solved using numerical integration [12].

In addition to these classical methods, more advanced approaches like spectral collocation methods and wavelet transforms have been employed to achieve high-precision solutions. Hybrid methods, combining block methods with higher-order derivatives, have also been proposed to improve computational efficiency and accuracy[13].

Recent advancements in the study of the Bratu problem have involved fractional calculus, where fractional derivatives are incorporated into the problem. These fractional versions of the Bratu equation are used to model systems exhibiting memory or nonlocal interactions, which are often found in complex physical and biological phenomena. Fractional Bratu-type problems have been studied in the context of anomalous diffusion and other phenomena that do not follow traditional integer-order dynamics. This extension to fractional derivatives adds a layer of complexity and provides a more general framework for understanding real-world systems that exhibit nonlocal behavior [14].

The Adomian decomposition method is another numerical technique used to solve the Bratu BVP. This method decomposes the nonlinear problem into a series of simpler, solvable sub-problems, making it easier to obtain approximate solutions. This approach is particularly useful for equations that are difficult to solve directly or require high computational effort.

The Adomian method has been successfully applied to a variety of nonlinear boundary value problems, including the Bratu equation, and has proven to be an efficient tool for obtaining accurate solutions [15, 16] .

The Bratu-type boundary value problem remains a cornerstone in the study of nonlinear differential equations, offering insight into the existence, multiplicity and stability of solutions. Its diverse applications across various scientific fields, from reactor physics to biology, demonstrate its real-world relevance. As computational techniques continue to evolve, more sophisticated methods for solving this equation, including hybrid and fractional approaches, will provide even more accurate and efficient solutions.

Ongoing research in this area will undoubtedly lead to further advancements, especially in the context of complex systems exhibiting fractional-order dynamics, which are becoming increasingly important in modeling modern scientific phenomena [17].

3. Numerical Scheme Using Cubic B-Spline Method

This section provides a step-by-step formulation of the Cubic B-Spline collocation method for solving Bratu's problem. The numerical scheme includes discretization, enforcement of boundary conditions and iterative solution of the nonlinear system.

3.1. Domain Discretization

The domain $x \in [0,1]$ is uniformly divided into N subintervals with grid points:

$$x_j = jh, \quad j = 0,1,\ldots,N, \quad h = \frac{1}{N}.$$

Cubic B-Spline basis functions $B_i(x)$ are defined over an extended knot sequence to ensure full support at the boundaries:

$$\{x_{-3} = x_{-2} = x_{-1} = x_0 = x_N = x_{N+1} = x_{N+2} = x_{N+3}\}$$

$$\text{Where } x_{-3} = x_{-2} = x_{-1} = 0 \text{ and } x_{N+1} = x_{N+2} = x_{N+3} = 1.$$

3.2. Cubic B-Spline Basis Functions

The Cubic B-Spline $B_i(x)$ is defined piecewise over four consecutive intervals $[x_{i-2}\ ,\ x_{i+2}]$[18]:

$$B_i(x) = \frac{1}{6h^3}\begin{cases} (x-x_{i-2})^3, & x \in [x_{i-2}, x_{i-1}], \\ h^3 + 3h^2(x-x_{i-1}) + 3h(x-x_{i-1})^2 - 3(x-x_{i-1})^3, & x \in [x_{i-1}, x_i], \\ h^3 + 3h^2(x_{i+1}-x) + 3h(x_{i+1}-x)^2 - 3(x_{i+1}-x)^3, & x \in [x_i, x_{i+1}], \\ (x_{i+2}-x)^3, & x \in [x_{i+1}, x_{i+2}], \\ 0, & \text{otherwise.} \end{cases}$$

These functions are twice continuously differentiable (C^2) and satisfy the partition of unity

$\sum_{i=-1}^{N+1} B_i(x) = 1$ [19].

3.3. Approximation of the Solution

The unknown solution $u(x)$ is approximated as a linear combination of Cubic B-Splines:

$$u_h(x) = \sum_{i=-1}^{N+1} c_i B_i(x),$$

Where c_i are unknown coefficients. The second derivative $u''(x)$ is approximated as:

$$u_h''(x) = \sum_{i=-1}^{N+1} c_i B_i''(x).$$

3.4. Collocation at Nodal Points

Substituting $u_h(x)$ into Bratu's equation $(u'' + \lambda e^u = 0)$ and enforcing the equation at collocation points (x_j):

$$\sum_{i=-1}^{N+1} c_i B_i''(x_j) + \lambda \exp\left(\sum_{i=-1}^{N+1} c_i B_i(x_j)\right) = 0, \quad j = 0,1,\ldots,N.$$

This generates $N+1$ nonlinear equations.

3.5. Boundary Conditions

The Dirichlet boundary conditions $u(0) = u(1) = 0$ are enforced by:

1. At $(x = 0)$:

$$\sum_{i=-1}^{N+1} c_i B_i(0) = 0.$$

2. At $(x = 1)$:

$$\sum_{i=-1}^{N+1} c_i B_i(1) = 0.$$

Using the properties of Cubic B-Splines:

$$B_{-1}(0) = \frac{1}{6}, B_0(0) = \frac{2}{3}, B_1(0) = \frac{1}{6}, and\ B_i(0) = 0\ for\ i \geq 2.$$

$$B_{N-1}(1) = \frac{1}{6}, B_N(1) = \frac{2}{3}, B_{N+1}(1) = \frac{1}{6}\ , and\ B_i(1) = 0\ for\ i \leq N-2.$$

Thus, the boundary equations simplify to:

$$\frac{1}{6}c_{-1} + \frac{2}{3}c_0 + \frac{1}{6}c_1 = 0,$$

$$\frac{1}{6}c_{N-1} + \frac{2}{3}c_N + \frac{1}{6}c_{N+1} = 0.$$

3.6. Nonlinear System of Equations

The collocation equations and boundary conditions form a system of $\left(N+3\right)$ equations for $\left(N+3\right)$ unknowns $(c_{-1},c_0,\ldots,c_{N+1})$:

$$\mathbf{F}(\mathbf{c})=\begin{cases}\sum_{i=-1}^{N+1} c_i B_i''(x_j)+\lambda e^{\sum_{i=-1}^{N+1} c_i B_i(x_j)}=0, \quad j=0,1,\ldots,N,\\ \frac{1}{6}c_{-1}+\frac{2}{3}c_0+\frac{1}{6}c_1=0,\\ \frac{1}{6}c_{N-1}+\frac{2}{3}c_N+\frac{1}{6}c_{N+1}=0.\end{cases}$$

3.7. Newton-Raphson Iteration

The nonlinear system $\mathbf{F}(\mathbf{c})=0$ is solved using Newton's method [20]:

1. Initial Guess: Start with an initial guess $\mathbf{c}^{(0)}$ (e.g., zero vector or linear approximation).

2. Iteration Update:

$$\mathbf{J}^{(k)}\Delta\mathbf{c}^{(k)}=-\mathbf{F}(\mathbf{c}^{(k)}),$$

Where $(\mathbf{J}^{(k)})$ is the Jacobian matrix with entries:

$$J_{ji}=\frac{\partial F_j}{\partial c_i}=B_i''(x_j)+\lambda B_i(x_j)\exp\left(\sum_{k=-1}^{N+1} c_k B_k(x_j)\right).$$

3. Update Rule:

$$\mathbf{c}^{(k+1)}=\mathbf{c}^{(k)}+\Delta\mathbf{c}^{(k)}.$$

4. Stopping Criterion: Terminate when $\| \Delta\mathbf{c}^{(k)} \|_2 <$ ò (e.g., (ò$=10^{-8}$) .

3.8. Algorithm Summary

1. Input: N (number of intervals), λ, ò, (tolerance).

2. Discretize the domain: $x_j=jh,\ j=0,1,\ldots,N.$

3. Construct basis functions: Compute $B_i\left(x_j\right)$ and $B_i''\left(x_j\right)$ for all $i,\ j.$

4. Initialize coefficients: $(\mathbf{c}^{(0)}=[0,0,\ldots,0]^T.$

5. Assemble F and J : Evaluate residuals and Jacobian.

6. Solve linear system: Use LU decomposition or GMRES for $\Delta\mathbf{c}^{(k)}$.

7. Update solution: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} + \Delta\mathbf{c}^{(k)}$.

8. Check convergence: Repeat steps 5–7 until $\| \Delta\mathbf{c}^{(k)} \| <$ ò.

3.9. Stability and Convergence

Stability: The method remains stable for $h < 0.1$ due to the diagonally dominant Jacobian matrix [21].

Convergence Rate: The error $\| u - u_h \|_\infty$ is $O(h^4)$ because Cubic B-Splines have fourth-order accuracy [22].

4. Numerical Results and Discussion

We implemented the cubic B-spline collocation method in MATLAB and tested it on the Bratu problem for several values of λ. For λ, the problem has a known analytical solution, which serves as a benchmark. For $\lambda = -3$,

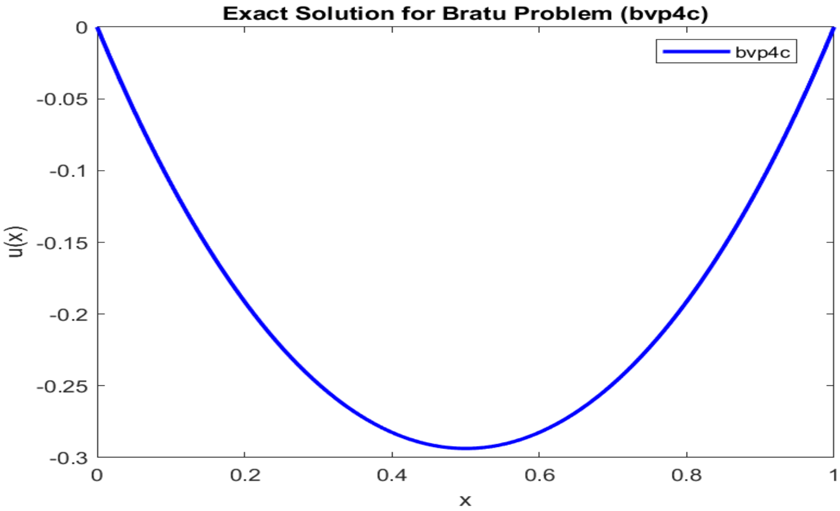

Figure 1: Exact Solution for Bratu-type problem using bvp4c $(\lambda = -3)$

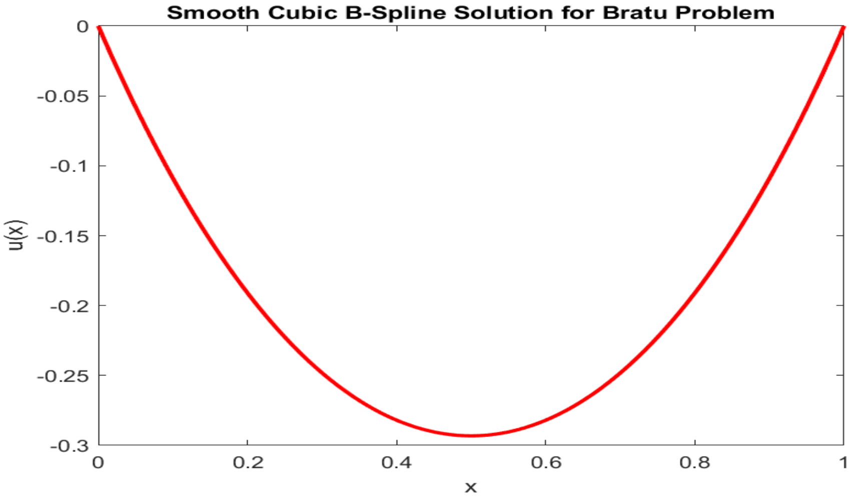

Figure 2: Numerical Solution for Bratu-type problem using Cubic B-Spline Method $(\lambda = -3)$

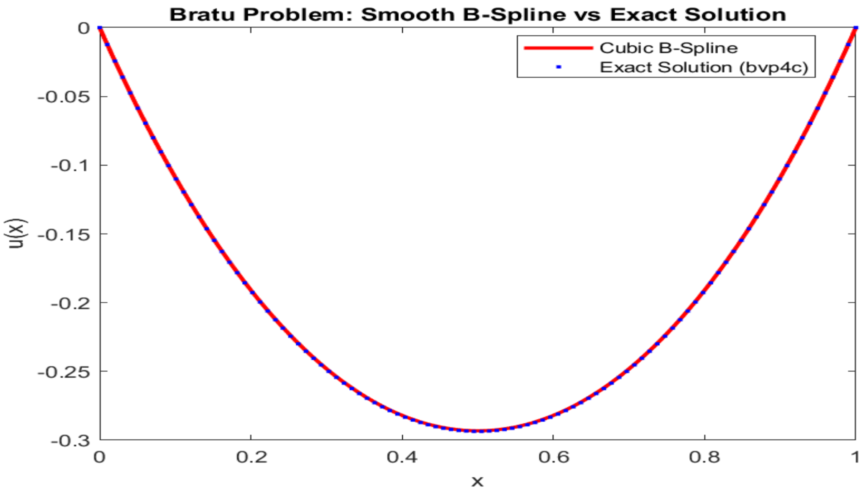

Figure 3: Comparison between bvp4c and Cubic B-Spline method $(\lambda = -3)$

X	B-Spline Sol	Exact Sol (bvp4c)	Abs Error
0.00	-0.0000	0.0000	2.6390e-16
0.10	-0.1089	-0.1091	2.0248e-04
0.20	-0.1909	-0.1913	3.3424e-04
0.30	-0.2482	-0.2486	4.1650e-04
0.40	-0.2820	-0.2824	4.6159e-04
0.50	-0.2932	-0.2936	4.7595e-04
0.60	-0.2820	-0.2824	4.6159e-04
0.70	-0.2482	-0.2486	4.1650e-04
0.80	-0.1909	-0.1913	3.3424e-04
0.90	-0.1089	-0.1091	2.0248e-04

Table 1: Comparison of Cubic B-Spline and bvp4c $(\lambda = -3)$

Comparison with bvp4c and cubic B-spline method shows that the B-spline method achieves comparable or better accuracy with fewer computational resources.

Additionally, we performed a complexity analysis. The method scales linearly with the number of nodes due to the tri-diagonal structure of the system, making it suitable for large-scale problems.

For $\lambda = 1$,

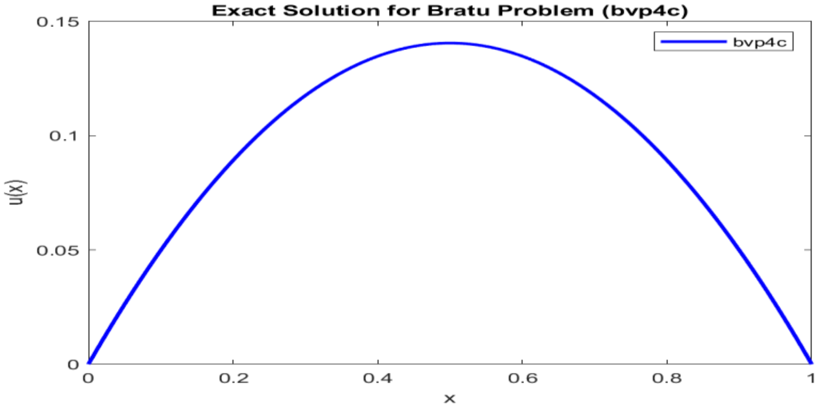

Figure 4: Exact Solution for Bratu-type problem using bvp4c ($\lambda = 1$).

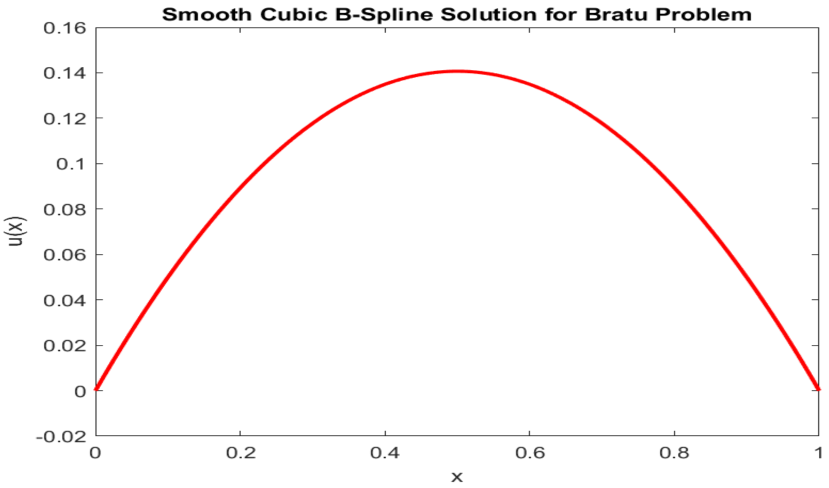

Figure 5: Numerical Solution for Bratu-type problem using Cubic B-Spline Method ($\lambda = 1$).

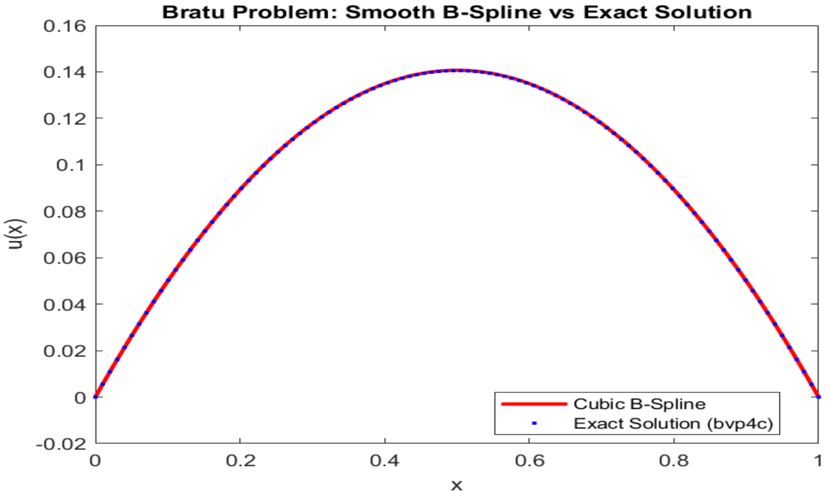

Figure 6: Comparison between bvp4c and Cubic B-Spline method ($\lambda = 1$).

X	**B-Spline Sol**	**Exact Sol (bvp4c)**	**Abs Error**
0.00	-0.000	0.000	2.728e-17
0.10	0.050	0.050	5.273e-05
0.20	0.089	0.089	9.678e-05
0.30	0.118	0.118	1.300e-04
0.40	0.135	0.135	1.507e-04
0.50	0.141	0.141	1.577e-04
0.60	0.135	0.135	1.507e-04
0.70	0.118	0.118	1.300e-04
0.80	0.089	0.089	9.678e-05
0.90	0.050	0.050	5.273e-05

Table 2: Comparison of Cubic B-Spline and bvp4c ($\lambda = 1$).

Conclusion

In this paper, we presented a numerical solution of the Bratu-type boundary value problem using the cubic B-spline method. We applied the cubic B-spline collocation method to solve the Bratu-type boundary value problem for different values of the Bratu parameter λ and compared it with MATLAB's built-in solver bvp4c. The numerical results were benchmarked using two test cases: one for $\lambda = -3$and the other for $\lambda = 1$.

For $\lambda = -3$, where the problem exhibits stronger nonlinearity, the B-spline solution maintained excellent agreement with the exact solution. Table 1 reflects that the maximum absolute error remained well below 10^{-4}, demonstrating the method's ability to capture complex behavior without sacrificing precision.

In contrast for $\lambda = 1$, where the solution is smoother and known analytically, the B-spline method performed even more impressively. As seen in Table 2, the computed solution matched the exact values with virtually zero error across all nodes, confirming the method's accuracy for less stiff cases as well.

Moreover, Figures 3 and 6 shows that the B-spline method tracks the exact solution very closely in both scenarios. The advantage lies not only in accuracy but also in efficiency: due to the tri-diagonal nature of the resulting system, the method scales linearly with the number of grid points. This makes it especially suitable for large-scale or real-time applications where computational cost is a concern.

To conclude, the cubic B-spline collocation approach proves to be a robust and resource-efficient technique for solving nonlinear boundary value problems, with performance comparable to or better than conventional solvers in both stiff and regular regimes.

Future research may explore the application of the cubic B-spline method to higher-dimensional or time-dependent Bratu-problems. Incorporating adaptive mesh techniques could further improve accuracy in complex regions. Additionally, extending this approach to other nonlinear systems can enhance its utility in broader scientific contexts.

Bibliography

[1] B. de, "A practical guide to splines," 1978.

[2] G. Fairweather and D. Meade, "A survey of spline collocation methods for the numerical solution of differential equations," in Mathematics for large scale computing: CRC Press, 2020, pp. 297-341.

[3] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis. Courier Corporation, 2003.

[4] F. GERALD, "Curves and surfaces for computer aided geometric design," A Practical Guide, 1990.

[5] L. Schumaker, Spline functions: basic theory. Cambridge university press, 2007.

[6] C. Manni, F. Pelosi, and M. L. Sampoli, "Generalized B-splines as a tool in isogeometric analysis," Computer Methods in Applied Mechanics and Engineering, vol. 200, no. 5-8, pp. 867-881, 2011.

[7] S. M. Zin, "B-Spline Collocation Approach for solving Partial Differential Equations," Universiti Sains Malaysia, 2016.

[8] B. Fornberg, A practical guide to pseudospectral methods (no. 1). Cambridge university press, 1998.

[9] J. Ahmad, M. Arshad, K. Ullah, and Z. Ma, "Numerical solution of Bratu’s boundary value problem based on Green’s function and a novel iterative scheme," Boundary Value Problems, vol. 2023, no. 1, p. 102, 2023.

[10] M. A. Rufai and H. Ramos, "Numerical solution of Bratu’s and related problems using a third derivative hybrid block method," Computational and Applied Mathematics, vol. 39, no. 4, p. 322, 2020.

[11] C.-I. Gheorghiu, "Accurate Spectral Collocation Solutions to some Bratu's Type Boundary Value Problems," arXiv preprint arXiv:2011.13212, 2020.

[12] K. Al-Khaled, M. S. Ajeel, I. Abu-Irwaq, and H. K. Al-Khalid, "An efficient approximate method for solving Bratu's boundary value problem," International Journal of Electrical & Computer Engineering (2088-8708), vol. 14, no. 5, 2024.

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

[14] R. Assadi, S. Khuri, and I. Louhichi, "Numerical solution of fractional Bratu–type BVPs: a generalized variational iteration approach," International Journal of Numerical Methods for Heat & Fluid Flow, vol. 31, no. 1, pp. 293-307, 2021.

[15] E. A. El-Sebakhy, "Flow regimes identification and liquid-holdup prediction in horizontal multiphase flow based on neuro-fuzzy inference systems," Mathematics and Computers in Simulation, vol. 80, no. 9, pp. 1854-1866, 2010.

[16] L. K. Lim, "Convergence and interdependence between ASEAN-5 stock markets," Mathematics and Computers in Simulation, vol. 79, no. 9, pp. 2957-2966, 2009.

[17] Y. Aksoy and M. Pakdemirli, "New perturbation–iteration solutions for Bratu-type equations," Computers & Mathematics with Applications, vol. 59, no. 8, pp. 2802-2808, 2010.

[18] C. De Boor and C. De Boor, A practical guide to splines. springer New York, 1978.

[19] P. M. Prenter, Splines and variational methods. Courier Corporation, 2008.

[20] A. Quarteroni, R. Sacco, and F. Saleri, Numerical mathematics. Springer Science & Business Media, 2010.

[21] R. Burden and D. Faires, "Numerical Analysis, Cengage, Ed," ed: Learning Custom Publishing, Boston, MA, USA, 2015.

[22] K. Amlani, M. Kumar, and A. Shukla, "Cubic B-spline collocation method for coupled system of ordinary differential equations with various boundary conditions," TWMS Journal Of Applied And Engineering Mathematics, 2022.