

PRELIMINARY STUDY ON ADAPTIVE KNOT STRATEGIES IN CUBIC B-SPLINES FOR EFFICIENT DIFFERENTIAL EQUATION SOLVING

Wisut Kitchainukoon^{1,*}, Amornrat Sangsuwan² and Krittidej Chanthawara³

Received: September 29, 2024; Revised: February 10, 2025; Accepted: March 03, 2025

Abstract

This preliminary study explores a modest approach that combines cubic B-splines with an adaptive knot refinement algorithm for solving differential equations. The method adjusts knot placement dynamically based on local error estimates, aiming to improve solution accuracy without manual intervention. While the enhancements in accuracy may be limited, particularly in areas with rapid solution changes, the approach provides a foundational framework for further research. Although the improvement in accuracy may seem modest, it lays a solid foundation for further refinement and development, paving the way for even greater advancements in computational precision and efficiency.

Keywords: Cubic B-splines; Adaptive Node Refinement; Differential Equations; Numerical Methods; Error Estimation

Introduction

In the contemporary landscape of scientific and engineering research, the numerical solution of differential equations (DEs) stands as a cornerstone, enabling the exploration and understanding of myriad phenomena that defy straightforward analytical analysis (Ames, 2014). There are three classical numerical methods for solving partial differential equations (PDEs): finite difference (LeVeque, 1998), discretizing the domain into

a grid to approximate derivatives; finite element (Nikishkov, 2004), dividing the domain into elements using polynomial functions for flexibility in complex geometries; and spectral (Rai and Mondal, 2021), employing orthogonal functions for high accuracy in smooth solutions over regular domains. Each method is selected based on the domain geometry, solution smoothness, and boundary conditions of the specific PDE. The

¹ Department of Mathematics, Faculty of Science and Technology, Loei Rajabhat University, Loei, Thailand. E-mail: wisut.kit@lru.ac.th

² Department of Mathematics, Faculty of education, Loei Rajabhat University, Loei, Thailand. E-mail: amornrat.san@lru.ac.th

³ Program of Mathematics, Faculty of science, Ubon Ratchathani Rajabhat University, Ubon Ratchathani, Thailand. E-mail: krittidej.c@ubru.ac.th

* Corresponding author

essence of these numerical methods lies in their ability to approximate solutions to complex DEs with remarkable accuracy, thereby opening avenues for advancement in fields as diverse as fluid dynamics (Bhatti *et al.*, 2020), financial modeling (Gilli *et al.*, 2019), and materials science (Yang *et al.*, 2023).

Within this spectrum of numerical techniques, spline-based methods, particularly those utilizing cubic B-splines (Hussain and Fahdil, 2021), have emerged as the preferred choice. Cubic B-splines consist of piecewise cubic polynomials that pass through a series of control points, ensuring smoothness and continuity up to the second derivative across segments. These splines are widely used in computer graphics (Li, 2020), animation (Li, 2020), and data interpolation (Kumar *et al.*, 2020) due to their flexibility and the ease with which they control the curve's shape by adjusting the control points. The properties of cubic B-splines, such as local control and minimal curvature, make them ideal for creating smooth, natural-looking curves and surfaces in digital modeling and design tasks, including efficient numerical schemes for solving DEs (Majeed *et al.*, 2020; Tayebi *et al.*, 2022).

The advent of self-adaptive knot refinement algorithms marks a significant leap forward in the numerical analysis domain (Kaennakham *et al.*, 2010). The primary merit of such algorithms is their capability to dynamically adjust the mesh density in response to the evolving needs of the solution landscape, thereby ensuring higher accuracy with minimal manual intervention (Kaennakham and Chuathong, 2019; Park and DeSpirito, 2022). This adaptability is crucial in areas where the solution gradients are steep or exhibit complex behavior, necessitating a denser knot distribution to capture

the nuances of the solution accurately (Kissami *et al.*, 2020). The integration of a self-knot adaptation mechanism not only elevates the precision of the numerical solution but also significantly reduces the computational effort and expertise required for mesh configuration. This, in turn, democratizes access to high-fidelity numerical simulations, making them a practical tool for researchers and practitioners across various disciplines (Mungaia and Bhatta, 2015; Offermans *et al.*, 2020; Struchkov *et al.*, 2020; de Frahan *et al.*, 2023; Yang *et al.*, 2023).

This study focuses on synthesizing the cubic B-spline method with a self-adaptive knot refinement algorithm to provide an automatic and effective approach for numerically solving DEs. Leveraging the smoothness and efficiency of cubic B-splines alongside the adaptability of the knot refinement process, this research aims to enhance numerical solution accuracy while reducing computational resources and manual tuning. The paper is organized into six sections: the mathematical structure of cubic B-splines is discussed in Section 2, the knot-adaptive algorithm in Section 3, the computational process in Section 4, with numerical test cases in Section 5. Section 6 concludes with the main findings and future research directions.

The Cubic B-spline Method

The piecewise polynomial function s in the partition $a = x_0 < x_1 < \dots < x_{N-1} < x_N = b$ on interval $[a, b]$ is called a spline of degree k if $s \in C^{k-1}[a, b]$ and is a polynomial of degree at most k on each subinterval $[x_i, x_{i+1}]$. The typical cubic B-spline can be defined as follows

$$B_i(x) = \begin{cases} \frac{(x - x_{i-2})^3}{(x_{i+1} - x_{i-2})(x_i - x_{i-2})(x_{i-1} - x_{i-2})} & \text{for } x_{i-2} \leq x < x_{i-1} \\ \frac{(x - x_{i-2})^2(x_i - x)}{(x_{i+1} - x_{i-2})(x_i - x_{i-2})(x_i - x_{i-1})} + \frac{(x - x_{i-2})(x_{i+1} - x)(x - x_{i-1})}{(x_{i+1} - x_{i-2})(x_{i+1} - x_{i-1})(x_i - x_{i-1})} & \text{for } x_{i-1} \leq x < x_i \\ \frac{(x_{i+2} - x)(x - x_{i-1})^2}{(x_{i+2} - x_{i-1})(x_{i+1} - x_{i-1})(x_i - x_{i-1})} + \frac{(x - x_{i-2})(x_{i+1} - x)^2}{(x_{i+1} - x_{i-2})(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} + \frac{(x_{i+2} - x)^2(x - x_{i+2})}{(x_{i+1} - x_{i-2})(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} & \text{for } x_i \leq x < x_{i+1} \\ \frac{(x_{i+2} - x_{i-1})(x_{i+2} - x_i)(x_{i+1} - x_i)}{(x_{i+2} - x_{i-1})(x_{i+2} - x_i)(x_{i+2} - x_{i+1})} & \text{for } x_{i+1} \leq x < x_{i+2} \\ 0 & \text{for otherwise} \end{cases} \quad (1)$$

These functions can be simplified in the context of cubic splines with equally spaced knots. However, when the knots are not uniformly distributed, a piecewise polynomial function s defined on the partition $a = x_0 < x_1 < \dots < x_{N-1} < x_N = b$ is a spline of degree k if $s \in C^{k-1}[a, b]$ and is a polynomial of degree at most k on each subinterval $[x_i, x_{i+1}]$.

The uniform cubic B-spline is defined as a partition with equally spaced knots. These knots are uniformly distributed across the interval. However, not all knots are suitable for every subinterval, as some may lead to over-computation or errors. Therefore, it is essential to identify suitable knots for each subinterval to mitigate these issues. Identifying appropriate knots within specified intervals is crucial. By utilizing uniformly distributed knots, the error in each subinterval can be systematically evaluated. In instances of significant error, an additional mid-knot is introduced within that subinterval. Conversely, if the error is minimal, the knot is removed from the interval.

The Proposed Knot Adaptation Strategy

In regions with sparse data knots, problem-solving is often inadequate, necessitating the use of higher data-density knots. It is crucial to set appropriate conditions for selecting the right knots to address each specific problem. Some problems require high accuracy in localized areas, and such regions demand a greater number of knots compared to others.

The process of adding or removing knots from sub-intervals is known as adaptive knot refinement. This criterion-driven process adapts by marking only those elements that meet predetermined conditions for refinement. The first step in any adaptive knot refinement is to identify and mark the intervals requiring refinement based on specific criteria. If an interval exhibits high error, it is marked for refinement. Additionally, if the refinement scheme necessitates transition elements around the refined region, the locations for these must be marked and identified.

The refinement manner

This research project employs the h-refinement manner within the cubic spline method to enhance computational accuracy. This approach involves increasing the number of spline knots, resulting in a denser knot placement. By refining the intervals between knots, higher spline resolution is achieved, particularly in areas requiring greater detail. This knot refinement ensures that numerical

solutions are more accurate and reliable, optimizing the use of computational resources to capture critical features of the problem.

The proposed algorithm

To explore this research direction, this work outlines an adaptation procedure based on a simple idea, explained as follows.

Step 1: Define the initial points by starting with five equally spaced points.

Step 2: Perform interpolation using a cubic B-spline to obtain $P_3(x)$.

Step 3: Evaluate the error at the midpoint $x_c \in [x_i, x_{i+1}]$, calculated a $x_c = \frac{1}{2}(x_i + x_{i+1})$. Use the midpoint to reformulate the problem $L(f(x)) = g(x)$ and compute the error $m_i = |L(P_3(x_c)) - g(x_c)|$.

Step 4: Determine the maximum error $m_r = \max(m_i)$ within each interval and use this value for comparison.

Step 5: Compare the error in each interval with the average error $\bar{m} = \frac{1}{r} \sum_{i=1}^N m_i$. If the percentage error $\frac{1}{\bar{m}} |m_i - \bar{m}| \times 100$ is less than 30% or more than 70%, adjust accordingly: if the error is less than 30% and x_i has not been previously adjusted, remove x_i if the error is more than 70%, add a new midpoint.

Step 6: After adjustment, verify the boundaries to ensure that no points have been lost or incorrectly adjusted.

Step 7: Repeat steps 5 and 6, monitoring changes in the intervals. If changes occur, return to step 2; if no changes are detected, the process is complete.

Numerical Setups

The main calculation process

As depicted in Figure 1, to solve a PDE using a cubic spline and the proposed adaptation strategy, the process begins by initializing the necessary parameters and conditions, including boundary and initial conditions. Next, the domain is discretized into intervals for spline application. The cubic spline is then formulated and the system set up of equations derived from it and the PDE, ensuring continuity and smoothness at interval boundaries.

After solving the system to obtain spline coefficients, an adaptation strategy is applied to refine the solution. Using the spline coefficients, the PDE solution is computed at desired points and convergence checked. If not achieved, from the spline formulation step, this cycle continues until the solution converges, completing the process.

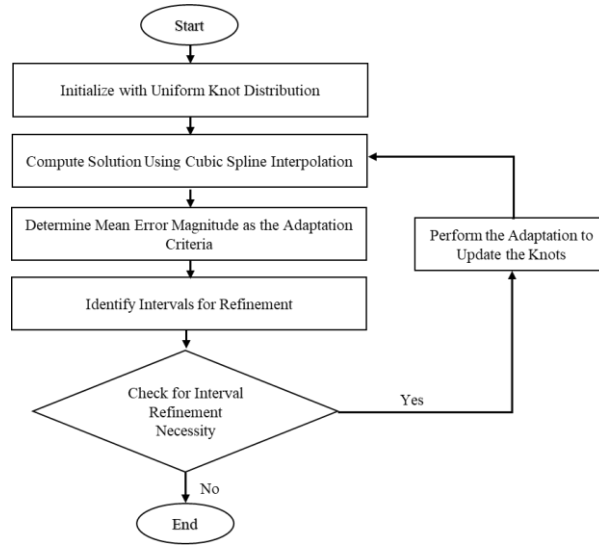


Figure 1. Iterative cubic spline process with the proposed adaptive knot adjustment

Computation configuration

The computations were carried out on a system equipped with an Intel(R) Core(TM) i5-10210U CPU running at 1.60 GHz (2.11 GHz) and 16 GB of installed RAM. The machine runs a 64-bit Windows 11 Home Single Language operating system, version 23H2. The numerical experiments were performed using Maple 2021.0 for all symbolic and numerical computations.

Numerical Experiments

This section presents four numerical examples to demonstrate the feasibility and effectiveness of combining the cubic spline method with the proposed adaptive knot strategy in enhancing the performance of differential equation solution approximations. Each example results in a different number of final knots, and the final results are verified against both the exact solutions and relevant documented literature (where applicable).

Example 1

The investigation starts by considering the following linear boundary value problem:

$$y'' + y' - 6y = x \quad (2)$$

for $0 \leq x \leq 1$, with the boundary conditions being expressed as follows:

$$y(0) = 0, y(1) = 1 \quad (3)$$

The exact solution to this problem is given by the following equation:

$$y(x) = \frac{(43 - e^2)e^{-3x} - (43 - e^{-3})e^{2x}}{36(e^{-3} - e^2)} - \frac{1}{6}x - \frac{1}{36} \quad (4)$$

The outcomes of the method using 10 cubic polynomials are as follows:

$$y(x) = \begin{cases} 0.5649477649x - 0.2824738823x^2 + 0.7257335982x^3 & ; 0 \leq x < 0.2 \\ 0.0006336622 + 0.5554428313x - 0.2349492128x^2 + 0.6465258144x^3 & ; 0.2 \leq x < 0.4 \\ -0.0021903387 + 0.5766228364x - 0.2878992217x^2 + 0.6906508204x^3 & ; 0.4 \leq x < 0.5 \\ -0.0118718679 + 0.6347120173x - 0.4040775942x^2 + 0.7681030752x^3 & ; 0.5 \leq x < 0.6 \\ -0.0365648262 + 0.7581768113x - 0.6098522548x^2 + 0.8824223333x^3 & ; 0.6 \leq x < 0.7 \\ -0.0894702649 + 0.9849143933x - 0.9337630696x^2 + 1.036665571x^3 & ; 0.7 \leq x < 0.8 \\ -0.163368993 + 1.262034630x - 1.280163368x^2 + 1.180999029x^3 & ; 0.8 \leq x < 0.85 \\ -0.231872979 + 1.503813439x - 1.56460906x^2 + 1.29254637x^3 & ; 0.85 \leq x < 0.9 \\ -0.323090917 + 1.807873202x - 1.902453211x^2 + 1.417673823x^3 & ; 0.9 \leq x < 0.95 \\ -0.303223842 + 1.745135069x - 1.836413072x^2 + 1.394501845x^3 & ; 0.95 \leq x < 1 \end{cases} \quad (5)$$

Table 1. Comparison of Cubic B-spline results with and without adaptive knot refinement

x_i	Cubic B-spline (N=5)	Adaptive Cubic B-spline (N=10)	Munguia and Bhatta, 2015 (N=20)	Exact
0	0.	0.	0.	0.
0.05	0.0275166610	0.0276319202	0.0275351538	0.0275370030
0.15	0.0805006430	0.0808358533	0.0806989046	0.0807133503
0.2	0.1070530049	0.1074964665	0.1074050277	0.1074285618
0.25	0.1343591179	0.1349120101	0.1348698493	0.1349034522
0.3	0.1629122200	0.1635772794	0.1635814257	0.1636255434
0.35	0.1931956921	0.1939771689	0.1940222368	0.1940768511
0.45	0.2608975153	0.2619259012	0.2620373739	0.2621112966
0.5	0.2993440925	0.3004776266	0.3006133878	0.3006953692
0.55	0.3415774923	0.3427794185	0.3429355142	0.3430239995
0.7	0.4966780473	0.4980021971	0.4981637819	0.4982584989
0.75	0.5600557426	0.5613170912	0.5614627582	0.5615536310
0.8	0.6304641696	0.6316256573	0.6317367884	0.6318199784
0.85	0.7086719223	0.7097234368	0.7097979583	0.7098689953
0.95	0.8924186384	0.8929026955	0.8929120525	0.8929423789

Table 1 compares the cubic B-spline results using five knots, adaptive cubic B-spline results using 10 knots, solutions from Munguia and Bhatta, 2015, and exact solutions for various values. The adaptive cubic B-spline method consistently yields results closer to the exact values compared to the non-adaptive method, demonstrating the benefits of the proposed knot refinement strategy. For instance, at $x = 0.2$, the adaptive method achieves an approximation of 0.1074964665, which is closer to the exact value of 0.1074285618 than the non-adaptive method's 0.1070530049. Similarly, at $x = 0.5$, the adaptive method's result is 0.3004776266 compared to the exact value of 0.3006953692, showing improved accuracy over the non-adaptive method's 0.2993440925. Although the improvements are not drastic, the adaptive strategy marks a promising enhancement in accuracy, indicating its potential for further refinement and application in solving Des

Example 2

The second test case involves the following linear boundary value problem:

$$y'' + 2y' + 5y = 6 \cos(2x) - 7 \sin(2x) \quad (6)$$

for $0 \leq x \leq \frac{\pi}{4}$, with the boundary conditions being expressed as:

$$y(0) = 4, y(\frac{\pi}{4}) = 1 \quad (7)$$

The exact solution of the problem is defined as follows:

$$y(x) = 2(1 - e^{-x}) \cos(2x) + 2 \sin(2x) \quad (8)$$

The resulting polynomials of the method using 11 cubic polynomials are as follows:

$$y(x) = \begin{cases} 4.000000000 - 0.005917421x - 6.994081481x^2 + 2.390177329x^3 & 0 \leq x < 0.039269908 \\ 3.999993632 - 0.005430777x - 7.006472316x^2 + 2.495348561x^3 & 0.039269908 \leq x < 0.078539816 \\ 3.999928720 - 0.002951405x - 7.038040856x^2 + 2.629328770x^3 & 0.078539816 \leq x < 0.157079633 \\ 3.999149454 + 0.011931426x - 7.132788380x^2 + 2.830389390x^3 & 0.157079633 \leq x < 0.314159265 \\ 3.993805254 + 0.062964769x - 7.295232515x^2 + 3.002747943x^3 & 0.314159265 \leq x < 0.471238898 \\ 3.987256098 + 0.104658997x - 7.383711365x^2 + 3.065333376x^3 & 0.471238898 \leq x < 0.549778714 \\ 3.986203685 + 0.110400014x - 7.394154025x^2 + 3.071665629x^3 & 0.549778714 \leq x < 0.628318531 \\ 3.989158869 + 0.096290433x - 7.3716990963x^2 + 3.059752938x^3 & 0.628318531 \leq x < 0.667588439 \\ 3.994142986 + 0.073888974x - 7.338140173x^2 + 3.042997870x^3 & 0.667588439 \leq x < 0.706858347 \\ 4.002596343 + 0.038018401x - 7.287395134x^2 + 3.019066385x^3 & 0.706858347 \leq x < 0.746128255 \\ 3.181185202 + 3.340713997x - 11.713832071x^2 + 4.996581569x^3 & 0.746128255 \leq x < 0.785398164 \end{cases} \quad (9)$$

Table 2. Comparison of cubic B-spline results with and without adaptive knot refinement

x_i	Cubic B-spline (N=5)	Adaptive Cubic B-spline (N=11)	Munguia and Bhatta, 2015 (N=20)	Exact
0	4.00006	4.	4.	4.0000
0.0392699082	3.9890051191	3.9891266209	3.9893275364	3.989348171
0.0785398164	3.9573677474	3.9575565816	3.9579417589	3.957978245
0.1178097245	3.9060062914	3.9061983299	3.9067485765	3.906796706
0.1570796327	3.8358391594	3.8359991188	3.8366881995	3.836744298
0.1963495409	3.7478030736	3.7479265208	3.7487313426	3.748792238
0.2356194491	3.6429080417	3.6429967884	3.6438757255	3.643938704
0.2748893573	3.5221823899	3.5222383570	3.5231428361	3.523205615
0.3141592654	3.3866544159	3.3866796634	3.3875749246	3.387635620
0.3534291736	3.2373629517	3.2373595788	3.2382321907	3.238289289
0.3926990818	3.0753885609	3.0753587303	3.0761901337	3.076242463

The adaptive method consistently yields closer approximations to the exact values, shown in Table 2, demonstrating its benefit. For instance, at $x = 0.1963495409$, the adaptive method's result is 3.7479265208, closer to the exact value of 3.748792238 than the non-adaptive result of 3.7478030736. Similarly, at $x = 0.3534291736$, the adaptive method's result is 3.2373595788, closer to the exact value of 3.238289289 than the non-adaptive result of 3.2373629517.

Example 3

As the final example in this work, the following equation is numerically investigated:

$$xy'' + y' = x \text{ for } 1 < x < 2. \quad (10)$$

Together with the boundary conditions,

$$y(1) = 1, y(2) = 1, \quad (11)$$

the exact solution to the problem can be formulated as follows:

$$y(x) = \frac{x^2}{4} - \frac{3 \ln(x)}{4 \ln(x)} + \frac{3}{4} \quad (12)$$

The results using six cubic polynomials are as follows:

$$y(x) = \begin{cases} 2.691620454 - 3.112882750x + 1.735695463x^2 - 0.314433167x^3 & 1 \leq x < 1.1 \\ 2.591176364 - 2.838944365x + 1.486660607x^2 - 0.238968071x^3 & 1.1 \leq x < 1.2 \\ 2.467296746 - 2.529245290x + 1.228578018x^2 - .1672784553x^3 & 1.2 \leq x < 1.4 \\ 2.305291916 - 2.182092094x + 0.980611458x^2 - 0.108238801x^3 & 1.4 \leq x < 1.6 \\ 2.165287475 - 1.919583761x + 0.816543747x^2 - 0.074058027x^3 & 1.6 \leq x < 1.8 \\ 2.041885880 - 1.713914436x + 0.702283010x^2 - 0.052898631x^3 & 1.8 \leq x < 2 \end{cases} \quad (13)$$

Table 3 shows that the adaptive cubic B-spline method exhibits superior accuracy, particularly evident in the lower absolute errors when compared to the exact solutions. For instance, at $x = 1.15$, the adaptive cubic B-spline provides a value of 0.9290584314, closely approximating the exact value of 0.9293996041, showing a reduced error compared to the cubic B-spline result of 0.9288694594. This pattern of improved accuracy is consistent throughout the interval, as seen at $x = 1.95$, where the adaptive cubic B-spline results in a value of 0.9779471390, aligning more closely with the exact value of 0.978019407 than the non-adaptive result of 0.9779382839. The reduction in absolute error across the dataset underscores the advantage of employing the adaptive strategy, demonstrating its capability to enhance the precision of numerical solutions in differential equation problems. These findings suggest that the adaptive cubic B-spline method is particularly valuable for achieving higher accuracy in complex computational scenarios.

Table 3. Comparison of cubic B-spline results with and without adaptive knot refinement

x_i	Cubic B-spline (N=5)	Adaptive Cubic B-spline (N=6)	Munguia and Bhatta, 2015 (N=20)	Exact
1	1.000000002	0.999999999	1.000000000	1.00000000
1.0500	0.9726011410	0.9727021194	0.9728193862	0.9728330041
1.1000	0.9489606373	0.9491303939	0.9493484897	0.9493723572
1.1500	0.9288694594	0.9290584314	0.9293682141	0.9293996041
1.2000	0.9121185779	0.9122975726	0.9126875054	0.9127241956
1.2500	0.8985128982	0.8986775530	0.8991387600	0.8991789288
1.3000	0.8879130674	0.8880639525	0.8885741354	0.8886162825
1.5000	0.8731228984	0.8732236034	0.8737385485	0.8737781244
1.5500	0.8758112180	0.8759004345	0.8763867258	0.8764238383
1.6000	0.8808856746	0.8809637707	0.8814119218	0.8814460712
1.6500	0.8882693865	0.8883367057	0.8887447136	0.8887754817
1.7000	0.8979025569	0.8979594232	0.8983219031	0.8983489402
1.7500	0.9097296604	0.9097763797	0.9100857939	0.9101088085
1.8000	0.9236951713	0.9237320318	0.9239835710	0.9240023200
1.95	0.9779382839	0.9779471390	0.9780145272	0.978019407

Conclusions

The combination of cubic B-splines with a self-adjusting knot refinement shows promising results in enhancing the accuracy of numerical solutions to DEs. The adaptive technique improves knot placement based on local error feedback, resulting in lower errors than non-adaptive approaches without requiring manual adjustment. This automatic refinement helps minimize human error and boosts computational efficiency. For example, at $x=1.95$, the adaptive method yields a value of 0.1096133077, which is 0.08% closer than the non-adaptive method's result of 0.1096341238. Furthermore, the adaptive method achieves this improved accuracy while using fewer knots compared to the standard cubic B-spline method, thereby reducing the computational resources required. The overall error reduction across the experimental results shows improvements ranging from 0.01% to 0.10%. These findings underscore the advantage of employing adaptive strategies in numerical methods, demonstrating their ability to enhance precision in solving DEs, particularly in complex computational scenarios. Moreover, the reduction in the number of knots when applying these methods in real-world scenarios translates to decreased data and resource usage, making computations more efficient. The automatic knot adjustment allows for precise identification of optimal placement conditions, significantly improving operational efficiency and ensuring that computational efforts are focused on the most critical regions of the solution domain. Future research will focus on further refining the adaptation algorithm and exploring its application to more diverse and complex systems, with an emphasis on maximizing the benefits of fully automated solution processes.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Mathematics, Faculty of Science and Technology, Loei Rajabhat University, Loei, and the Program of Mathematics, Faculty of Science, Ubon Ratchathani Rajabhat University, Ubon Ratchathani Thailand, for their kind support.

References

- Ames, W.F. (2014). Numerical methods for partial differential equations. Academic Press.
- Bhatti, M.M., Marin, M., Zeeshan, A., and Abdelsalam, S.I. (2020). Recent trends in computational fluid dynamics. *Frontiers in Physics*, 8. <https://doi.org/10.3389/fphy.2020.593111>
- De Frahan, M.T.H., Rood, J.S., Day, M.S., Sitaraman, H., Yellapantula, S., Perry, B.A., Grout, W., Almgren, A., Zhang, W., Bell, J.B., and Chen, J.H. (2023). PeleC: An adaptive mesh refinement solver for compressible reacting flows. *International Journal of High Performance Computing Applications*, 37(2):115-131. <https://doi.org/10.1177/10943420221121151>
- Gilli, M., Maringer, D., and Schumann, E. (2019). Chapter 10 - Optimization problems in finance. *Numerical methods and optimization in finance* (2nd ed.), 219-228. <https://doi.org/10.1016/B978-0-12-815065-8.00022-4>
- Hussain, A.K., and Fahdil, A. (2021). A systematic review and numerical analysis of the B-spline curve. *Journal of Education*, 3:17-30.
- Kaennakham, S., and Chuathong, N. (2019). An automatic node-adaptive scheme applied with a RBF-collocation meshless method. *Applied Mathematics and Computation*, 348:102-125. <https://doi.org/10.1016/j.amc.2018.11.066>
- Kaennakham, S., Holdø, A., and Lambert, C. (2010). A new simple h-mesh adaptation algorithm for standard Smagorinsky LES: A first step of Taylor scale as a refinement variable. *International Journal of Multiphysics*, 4(1):33-50.
- Kissami, I., Seaid, M., and Benkhaldoun, F. (2020). Numerical assessment of criteria for mesh adaptation in the finite volume solution of shallow water equations. *Advances in*

- Applied Mathematics and Mechanics, 12(2):503-526. <https://doi.org/10.4208/aamm.OA-2019-0011>
- Kumar, R., Choudhary, A., and Baskar, S. (2020). Modified cubic B-spline quasi-interpolation numerical scheme for hyperbolic conservation laws. *Applicable Analysis*, 99(1):158-179. <https://doi.org/10.1080/00036811.2018.1489961>
- LeVeque, R.J. (1998). Finite difference methods for differential equations. Draft version for AMath 585. <https://faculty.washington.edu/rjl/fdmbook/>
- Li, L. (2020). Application of cubic B-spline curve in computer-aided animation design. *Computer-Aided Design and Applications*, 18(S1):43-52. <https://doi.org/10.14733/cadaps.2021.S1.43-52>
- Majeed, A., Abbas, M., Qayyum, F., Miura, K.T., Misro, M.Y., and Nazir, T. (2020). Geometric modeling using new cubic trigonometric B-spline functions with shape parameter. *Mathematics*, 8(12):2102. <https://doi.org/10.3390/math8122102>
- Munguia, M., and Bhatta, D. (2015). Use of cubic B-spline in approximating solutions of boundary value problems. *Applications and Applied Mathematics: An International Journal*, 10(2):750-771.
- Nikishkov, G.P. (2004). Introduction to the finite element method (Lecture notes). University of Aizu, 1-45
- Offermans, N., Peplinski, A., Marin, O., and Schlatter, P. (2020). Adaptive mesh refinement for steady flows in Nek5000. *Computers and Fluids*, 197:104352. <https://doi.org/10.1016/j.compfluid.2019.104352>
- Park, M.A., and DeSpirito, J. (2022). The influence of adaptive mesh refinement on the prediction of vortex interactions about a generic missile airframe. In *AIAA SCITECH 2022 Forum*. <https://doi.org/10.2514/6.2022-1177>
- Rai, N., and Mondal, S. (2021). Spectral methods to solve nonlinear problems: A review. *Partial Differential Equations in Applied Mathematics*, 4:100043. <https://doi.org/10.1016/j.padiff.2021.100043>
- Struchkov, A.V., Kozelkov, A.S., Volkov, K.N., Kurkin, A.A., Zhuckov, R.N., and Sarazov, A.V. (2020). Numerical simulation of aerodynamic problems based on adaptive mesh refinement method. *Acta Astronautica*, 172:7-15. <https://doi.org/10.1016/j.actaastro.2020.03.019>
- Tayebi, S., Momani, S., and Arqub, O.A. (2022). The cubic B-spline interpolation method for numerical point solutions of conformable boundary value problems. *Alexandria Engineering Journal*, 61(2):1,519-1,528. <https://doi.org/10.1016/j.aej.2021.06.057>
- Yang, J., Dzanic, T., Petersen, B., Kudo, J., Mittal, K., Tomov, V., Camier, J. S., Zhao, T., Zha, H., Kolev, T., Anderson, R., and Faissol, D. (2023). Reinforcement learning for adaptive mesh refinement. In *International Conference on Artificial Intelligence and Statistics*, 1-18. <https://doi.org/10.48550/arXiv.2103.01342>