Eurasian J. Sci. Eng., 2023, 9(2), 252-261
https://doi.org/10.23918/eajse.v9i2p20

Published by Tishk International University
Available at: https://eajse.tiu.edu.iq/

RESEARCH ARTICLE Open Access

NUMERICAL SOLUTION OF THE HEAT EQUATION BY CUBIC B-SPLINE COLLOCATION METHOD

Hoshman Q. Hamad [1*] and Younis A. Sabawi[1, 2]

[1] Department of Mathematics, Faculty of Science and Health, Koya University, Koya-IRAQ
[2] Department of Mathematics Education, Faculty of Education, Tishk International University, Erbil-IRAQ

Article History
Received: 05.01.2023
Revised: 04.05.2023
Accepted: 21.06.2023
Communicated by: Dr. Orhan Tug
**Email address:*
hoshman.qadir@koyauniversity.org
**Corresponding Author*

Abstract:
This work proposes a numerical scheme for heat parabolic problem by implementing a collocation method with a cubic B-spline for a uniform mesh. The key idea of this method is to apply forward finite difference and Crank–Nicolson methods for time and space integration, respectively. The stability of the presented scheme is proved through the Von-Neumann technique. It is shown that it is unconditionally stable. The accuracy of the suggested scheme is computed through the L_2 and L_∞-norms. Numerical experiments are also given and show that it is compatible with the exact solutions.

Keywords: *Collocation Methods; Cubic B-Spline Functions; Heat Equation.*

1. Introduction

Consider the following linear parabolic heat equation

(1) $$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad u(x,t) \in [0,1] \times [0,T],$$

with initial condition

(2) $$u(x,0) = f(x),$$

and boundary conditions

(3) $$u(0,t) = u(1,t) = 0.$$

This problem is one of the well-known second order linear partial differential equation. Numerous authors have extensively researched this issue over a period of many years. However, given that many physical phenomena can be expressed as PDEs with boundary conditions, it is still a fascinating problem. The heat equation is crucial to many different fields of science. Numerous methods have been developed to solve parabolic such as finite difference method [1-7] and by compact finite difference method [8-10]. Furthermore, some extra ordinary problems has been numerically investigated by finite element methods such as Galerkin method, least square method and collocation method with quadratic, cubic, quintic and septic B-splines [11-16]. Various techniques of both the cubic spline and cubic B-spline collocation methods and their application have been developed to obtain the numerical solution of the differential equations such as [17-18].

This paper aims to link a finite difference approach with the cubic B spline method for solving heat problem (1) subject to (2) and (3). A key idea for the proposed scheme is to use the Crank–Nicolson method to discrtize the derivative of time while, cubic B-spline is use to interpolate the solutions at time. The stability of suggested method is proved.

The rest of this work is structured like that. In Section 2, the description of the cubic B spline method is introduced. In section 3 and 4, The model of problems is presented. Stability analysis is given in section 5. Numerical experiments are shown for different types of examples in section 6, Finally, conclusions are given in section 7.

2. Description of Cubic B-spline Collocation Method

This section constructs numerical solution for the presented problem (1). Let $a = x_0 < x_1 <, \cdots, < x_{N-1} < x_N = b$ is partitioned on the space domain with $h = x_{i+1} - x_i = \frac{b-a}{N}$ for $i = 0,1,2, \ldots, N$. The typical third-degree B-spline basis functions, given as

$$U(x_i, t) = \sum_{i=-1}^{n+1} \delta_i(t)\, B_i^3(x), \quad i = 0,1, \cdots, N, \tag{4}$$

Where

$$B_i^3(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3, & [x_{i-2}, x_{i-1}] \\ -3(x - x_{i-1})^3 + 3h(x - x_{i-1})^2 + 3h^2(x - x_{i-1}) + h^3, & [x_{i-1}, x_i] \\ -3(x_{i+1} - x)^3 + 3h(x_{i+1} - x)^2 + 3h^2(x_{i+1} - x) + h^3, & [x_i, x_{i+1}] \\ (x_{i+2} - x)^3, & [x_{i+1}, x_{i+2}] \\ 0 & otherwise. \end{cases} \tag{5}$$

Where $\delta_i(t), i = -1,0, \ldots, N+1$ are unknown time-dependent quantity to be determined at each time level from boundary conditions and the initial conditions. At the knots, nodal values and its principal two derivatives are obtained using the cubic functions (5). The value of $B_i(x)$ and its derivatives $B_i'(x)$ and $B_i''(x)$ at the knots are given in Table 1.

Table 1: The value of cubic B-spline and its derivatives at the knot's points

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
B_i	0	1	4	1	0
B_i'	0	$-{}^{3}/_{h}$	0	${}^{3}/_{h}$	0
B_i''	0	${}^{6}/_{h^2}$	$-{}^{12}/_{h^2}$	${}^{6}/_{h^2}$	0

3. Implementation of the Method

Applying forward finite-difference approach with utilizing Crank–Nicolson rule in (1), gives:

(7) $$\frac{u^{n+1}-u^n}{k}-\alpha\left[\frac{(u_{xx})^{n+1}+(u_{xx})^n}{2}\right]=0,$$

Where $k=\Delta t$ is the time step.

Using approximate function (4) and cubic B-spline functions (5), the approximate values $U(x)$, and their derivatives up to second order are determined in terms of the time parameters $\delta_i(t)$, as

(8) $$A_1\delta_{i-1}^{n+1}+A_2\delta_i^{n+1}+A_1\delta_{i+1}^{n+1}=A_3\delta_{i-1}^{n}+A_4\delta_i^{n}+A_3\delta_{i+1}^{n},$$

Where $\beta=\frac{\alpha\,\Delta t}{2}$, and

$$A_1=1-\frac{6\beta}{h^2},\qquad A_2=4+\frac{12\beta}{h^2},$$

$$A_3=1+\frac{6\beta}{h^2},\qquad A_4=4-\frac{12\beta}{h^2},$$

After simplifying equations (8) which consists of $(N+1)$ linear equations with $(N+3)$ unknowns $(\delta_{-1},\delta_0,\ldots,\delta_N,\delta_{N+1})^T$. To address the challenge of not unique problem, in this work, we impose boundary condition (3) along with eliminating δ_{-1},δ_{N+1}. Therefore, the system obtained can be reduced to a matrix system of dimension $(N+1)\times(N+1)$ as

$$\mathbb{A}p^{n+1}=\mathbb{B}p^n+\mathbb{C},$$

Where

$$\mathbb{A}=\begin{bmatrix} A_2 & 2A_1 & 0 & 0 & 0 & \ldots & 0\\ A_1 & A_2 & A_1 & 0 & 0 & \ldots & 0\\ 0 & A_1 & A_2 & A_1 & 0 & \ldots & 0\\ 0 & 0 & A_1 & A_2 & A_1 & \ldots & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \ldots & \vdots\\ 0 & \ldots & 0 & 0 & A_1 & A_2 & A_1\\ 0 & 0 & \ldots & 0 & 0 & 2A_1 & A_2 \end{bmatrix}_{(N+1\times N+1)} \qquad p^{n+1}=\begin{bmatrix}\delta_0^{n+1}\\ \delta_1^{n+1}\\ \delta_2^{n+1}\\ \vdots\\ \delta_{n-2}^{n+1}\\ \delta_{n-1}^{n+1}\\ \delta_n^{n+1}\end{bmatrix}_{(N+1\times 1)},$$

$$\mathbb{B}=\begin{bmatrix} A_4 & 2A_3 & 0 & 0 & 0 & \ldots & 0\\ A_3 & A_4 & A_3 & 0 & 0 & \ldots & 0\\ 0 & A_3 & A_4 & A_3 & 0 & \ldots & 0\\ 0 & 0 & A_3 & A_4 & A_3 & \ldots & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \ldots & \vdots\\ 0 & \ldots & 0 & 0 & A_3 & A_3 & A_3\\ 0 & 0 & \ldots & 0 & 0 & 2A_3 & A_4 \end{bmatrix}_{(N+1\times N+1)} \qquad p^{n}=\begin{bmatrix}\delta_0^{n}\\ \delta_1^{0}\\ \delta_2^{n}\\ \vdots\\ \delta_{n-2}^{n}\\ \delta_{n-1}^{n}\\ \delta_n^{n}\end{bmatrix}_{(N+1\times 1)},$$

$$\mathbb{C}=\begin{bmatrix}\frac{h}{3}\left(A_1u'(x_0,t_{n+1})-A_3u'(x_0,t_n)\right)\\ 0\\ 0\\ \vdots\\ 0\\ 0\\ \frac{h}{3}\left(-A_1u'(x_n,t_{n+1})+A_3u'(x_n,t_n)\right)\end{bmatrix}_{(N+1\times 1)}$$

The above tri-diagonal system of matrix will be solved by a modified form Thomas algorithm.

4. The Initial State

To deal with the initial parameters δ_i^0, this can be done by using the initial conditions (2) and the derivatives at the boundaries in the following way:

(9) $$f(x_i) = \delta_{i-1}^0 + 4\delta_i^0 + \delta_{i+1}^0,$$

$$(U')(x_0, 0) = \frac{3}{h}(-\delta_{-1} + \delta_1) = f'(x_0)$$

$$(U'')(x_0, 0) = \frac{6}{h^2}(\delta_{-1} - 2\delta_0 + \delta_1) = f''(x_0),$$

$$(U)(x_i, 0) = \delta_{i-1} + 4\delta_i + \delta_{i+1} = f(x_i),$$

$$(U')(x_N, 0) = \frac{3}{h}(-\delta_{N-1} + \delta_{N+1}) = f'(x_N),$$

(10) $$(U'')(x_N, 0) = \frac{6}{h^2}(\delta_{N-1} - 2\delta_N + \delta_{N+1}) = f''(x_N).$$

Applying boundary and initial conditions to eliminate the unknowns from the system of Eq. (10), imply that

$$(U')(x_0, 0) = f'(x_0), \quad (U')(x_N, 0) = f'(x_N).$$

Combing above equation with Eq. (11), reads

$$\delta_{-1}^0 = \delta_1 - \frac{h}{3}f'(x_0),$$

(11) $$\delta_{N+1}^0 = \delta_{N-1} + \frac{h}{3}f'(x_N),$$

The system obtained after simplifying and eliminating the functions values of δ, can be solved by any algorithm. The numerical solution of presented method can be determined from the time evaluation of the vectors δ_j^n by using the recurrence relations.

$$U(x_i, t_n) = \delta_{i-1} + 4\delta_i + \delta_{i+1},$$

From Eqs. (10) and (12), the resulting matrix system of (N+1) linear equations with (N+1) unknowns, written as

$$\begin{bmatrix} 4 & 2 & 0 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 4 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & \dots & 0 & 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} \delta_0^0 \\ \delta_1^0 \\ \delta_2^0 \\ \vdots \\ \delta_{n-2}^0 \\ \delta_{n-1}^0 \\ \delta_n^0 \end{bmatrix} = \eta = \begin{bmatrix} f(x_0) + \frac{h}{3}f'(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \\ f(x_N) - \frac{h}{3}f'(x_n) \end{bmatrix}.$$

5. Stability Analysis of the Method

The aim of this section is to find stability condition of the presented approach through Von-Neumann stability method. Start with Eq. (8), we have

$$A_1\delta_{i-1}^{n+1} + A_2\delta_i^{n+1} + A_1\delta_{i+1}^{n+1} = A_3\delta_{i-1}^{n} + A_4\delta_i^{n} + A_3\delta_{i+1}^{n},$$

Where A_1, A_2, A_3, and A_4 are given in Eqs. (9). Now substituting $\delta_j^n = \xi^n exp(ij\phi)$ in (9), where $\phi = sh$, s is the mode number, $i = \sqrt{-1}$ and ξ is the amplification factor of the schemes, becomes

$$\xi^{n+1}\left(A_1\delta_{j-1}^{n+1} + A_2\delta_j^{n+1} + A_1\delta_{j+1}^{n+1}\right) = \xi^n\left(A\delta_{j-1}^{n} + A_4\delta_j^{n} + A_3\delta_{j+1}^{n}\right),$$

(12) $$\xi(A_1 \exp(-i\phi) + A_2 + A_1 \exp(i\phi)) = (A_3 \exp(-i\phi) + A_4 + A_3 \exp(i\phi)).$$

Simplifying (12), imply that

(13) $$\xi = \frac{X_1}{X_2},$$

Where

$$X_1 = 2A_3cos\phi + A_4,$$

$$X_2 = 2A_1cos\phi + A_2.$$

For the stability of the technique, we need to prove that $|\xi| \leq 1$, so for we only need to prove that $X_2 \geq X_1$ or $X_2 - X_1 \geq 0$,

$$X_2 - X_1 = \left[\left(2\left(1 - \frac{6\beta}{h^2}\right)cos\phi + 4 + \frac{12\beta}{h^2}\right) - \left(2\left(1 + \frac{6\beta}{h^2}\right)cos\phi + 4 - \frac{12\beta}{h^2}\right)\right],$$

Take $cos\phi = 1$, for the minimum value of $X_2 - X_1$, we obtain $X_2 - X_1 = 0$. Hence $X_2 - X_1 \geq 0$ and $X_2^2 \geq X_1^2$ so $|\xi| \leq 1$, hence the scheme is unconditional stable.

6. Numerical Experiments

The section illustrates to show the accuracy of the suggested method, based on MATLAB programming. The error norms of L_2 *and* L_∞ are used to measure the error between the numerical and exact solutions

$$E_u(x,t) = u(x,t) - U(x,t),$$

Let us introduce the three accuracy indicators, when using space step size h, as follows

- The pointwise error
$$\mathcal{E}_u(x,t) = |E_u(x_i,t)|.$$
- The L_∞ −norm of the error
$$L_\infty(E_u,h) = \max_{0\leq i\leq N}|E_u(x_i,t)|.$$
- The L_2 − norm of the errors
$$L_2(E_u,h) = \sqrt{h\sum_{i=0}^{N}|E_u(x_i,t)|^2}.$$

Problem 1: Consider the heat equation (1) when $\alpha = \frac{1}{\pi^2}$,

$$\frac{\partial u}{\partial t} = \frac{1}{\pi^2}\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \qquad t > 0$$

With boundary and initial conditions

$$u(x,0) = \sin(\pi x), \qquad u(0,t) = u(1,t) = 0 \qquad t \geq 0,$$

The exact solution of this problem is $u(x,t) = e^{(-t)}\sin(\pi x)$.

Table 2: Pointwise error norm for problem 1

x	$u(x,t)$	$\mathrm{U}(x,t)$	$\mathcal{E}_u(x,t)$
0	0.00000000	-3.19398e-03	3.19398e-03
0.1	1.13681e-01	1.10387e-01	3.29353e-03
0.2	2.16234e-01	2.12703e-01	3.53099e-03
0.3	2.97621e-01	2.93824e-01	3.79685e-03
0.4	3.49874e-01	3.45876e-01	3.99806e-03
0.5	3.67879e-01	3.63807e-01	4.07241e-03
0.6	3.49874e-01	3.45876e-01	3.99806e-03
0.7	2.97621e-01	2.93823e-01	3.79685e-03
0.8	2.16234e-01	2.12703e-01	3.53099e-03
0.9	1.13681e-01	1.10387e-01	3.29353e-03
1	4.50522e-17	-3.19398e-03	3.19398e-03

Table 3: L_2 and L_∞ error norm obtained from problem 1

h	L_2 $error$	L_∞ $error$
1/8	5.99598e-03	6.35888e-03
1/16	1.46332e-03	1.59188e-03
1/32	3.61537e-04	3.98094e-04
1/64	8.98544e-05	9.95313e-05
1/128	2.23977e-05	2.48833e-05

Table 4: Maximum absolute error obtained for problem 1

h	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$
1/8	2.12022e-03	3.56016e-03	4.64353e-03	5.54634e-03
1/16	5.29112e-04	8.90835e-04	1.16286e-03	1.38891e-03
1/32	1.32213e-04	2.22740e-04	2.90821e-04	3.47360e-04
1/64	3.30490e-05	5.56869e-05	7.27116e-05	8.68480e-05
1/128	8.26199e-06	1.39219e-05	1.81783e-05	2.17125e-05

Problem 2: we consider the heat equation (1) when $\alpha = 1$,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \qquad t > 0$$

With boundary and initial conditions

$$u(x,0) = \sin(\pi x), \qquad u(0,t) = u(1,t) = 0 \qquad t \geq 0,$$

The exact solution of this problem is $u(x,t) = e^{(-\pi^2 t)}\sin(\pi x)$.

Table 5: Pointwise error for problem 2

x	$u(x,t)$	$U(x,t)$	$\mathcal{E}_u(x,t)$
0	0.00000000	-6.06765e-03	6.06765e-03
0.1	4.29259e-02	3.68093e-02	6.11655e-03
0.2	8.16499e-02	7.54147e-02	6.23519e-03
0.3	1.12381e-01	1.06013e-01	6.36777e-03
0.4	1.32112e-01	1.25644e-01	6.46823e-03
0.5	1.38911e-01	1.32405e-01	6.50554e-03
0.6	1.32112e-01	1.25644e-01	6.46823e-03
0.7	1.12381e-01	1.06013e-01	6.36777e-03
0.8	8.16499e-02	7.54147e-02	6.23519e-03
0.9	4.29259e-02	3.68093e-02	6.11655e-03
1	1.70117e-17	-6.06765e-03	6.06765e-03

Table 6: L_2 and L_∞ error norm obtained for problem 2

h	L_2 *error*	L_∞ *error*
1/10	6.58262e-03	6.50555e-03
1/20	1.60905e-03	1.62641e-03
1/40	3.97711e-04	4.06605e-04
1/80	9.88593e-05	1.01652e-04
1/160	2.46438e-05	2.54129e-05

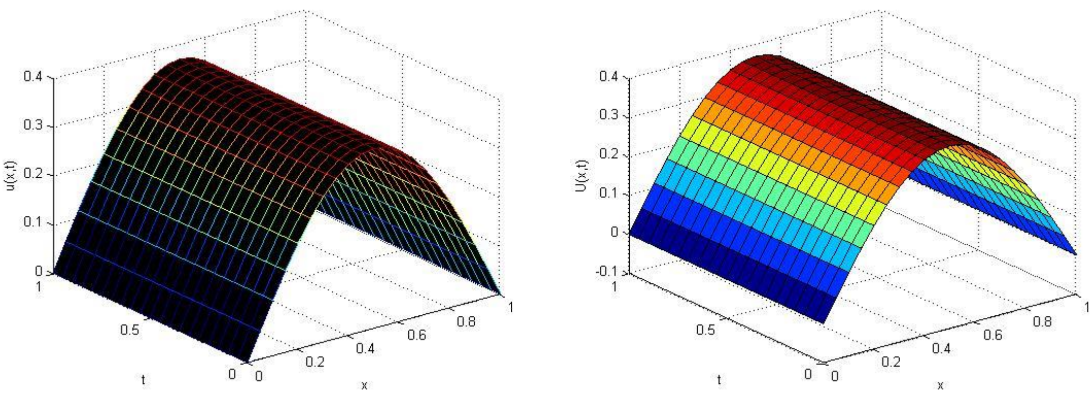

Figure 1: Exact and approximate solution of problem 1 in the domains 0≤x≤1, 0≤t≤1,

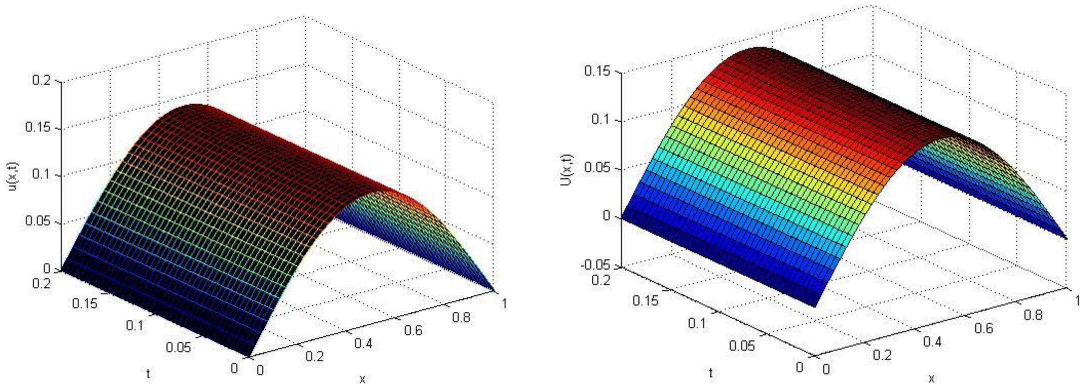

Figure 2: Exact and approximate solution of problem 2 in the domains 0≤x≤1, 0≤t≤0.2

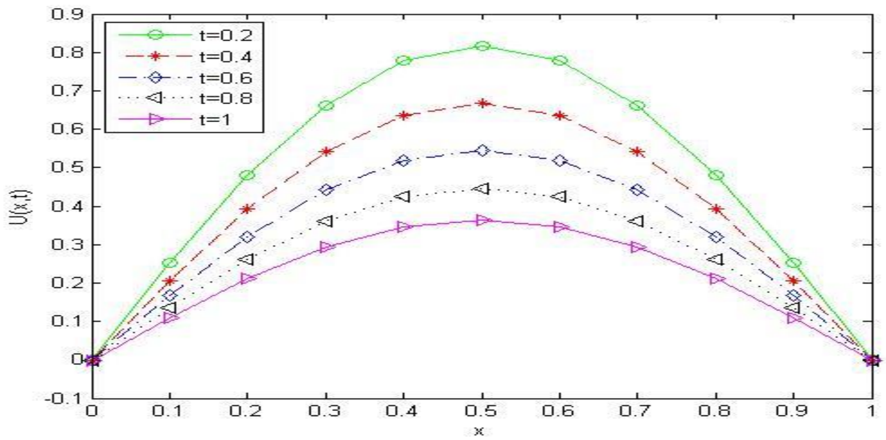

Figure 3: Approximate solution for problem 1 with different time levels

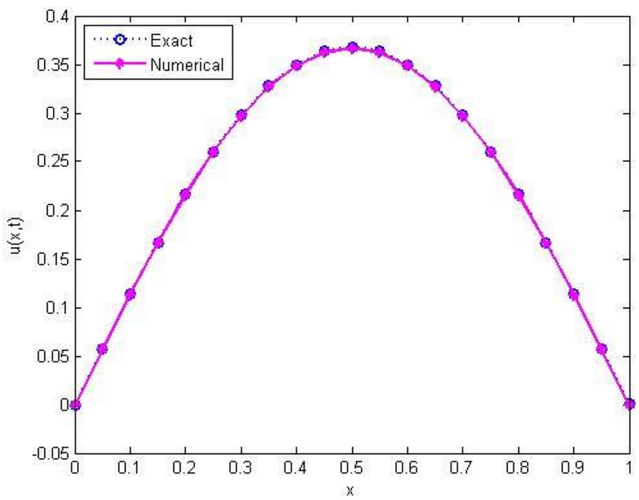

Figure 4: Comparison between the exact solution u(x,t) and numerical solution U(x,t) for problem 1

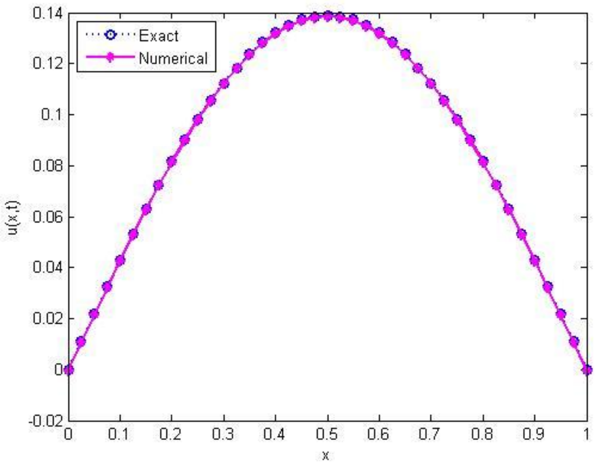

Figure 5: Comparison between the exact solution u(x,t) and numerical solution U(x,t) for problem 2

he numerical solutions are obtained for domains $[0,1]$ at different time levels $t = 0.2, 0.4, 0.6, 0.8$ with different levels of N. In Table 3 and 6, the L_2 and L_∞ errors norm are calculated for different space levels. Furthermore, the pointwise error is measured in domain $0 \leq x \leq 1$. These results show that it is close form the exact solution. The comparison of the numerical with the exact solutions is shown graphically in Figs.1- 5. These figures show that there is a good agreement between exact and numerical solutions.

7. Conclusion

This paper aims to investigate a numerical solution for the heat equation. The proposed approach is based on a finite difference method with the cubic B-splines function. More specifically, the cubic B spline method used the space variable and the finite difference method for the time variable for the partial differential equation case. The stability analysis of the method is shown to be unconditionally stable. The precision of the scheme has been measured by considering two test problems and calculating and error norms for different time levels. Numerical experiments demonstrated that the results that are obtained from the proposed method is efficient, reliable, fruitful, and powerful.

8. Acknowledgment

Authors would like to thank of the financial support from Koya and Tishk international Universities.

References

[1] Yang X, Ge Y, Zhang L. A class of high-order compact difference schemes for solving the Burgers' equations. Applied mathematics and computation. 2019; (358): 394-417. https://doi.org/10.1016/j.amc.2019.04.023

[2] Manaa SA, Moheemmeed MA, Hussien YA. A numerical solution for sine-gordon type system. Tikrit Journal of Pure Science. 2010; 15(3): 106-13.

[3] Sabawi YA, Pirdawood MA, Khalah AD. Semi-Implicit and Explicit Runge Kutta Methods for Stiff Ordinary Differential Equations. Journal of Physics: Conference Series. 2021; 1999(1). https://doi.org/10.1088/1742-6596/1999/1/012100

[4] Pirdawood MA, Rasool HM, sabawi YA, Azeez BF. Mathematical Modeling and Analysis for COVID-19 Model by Using Implicit-Explicit Rung-Kutta Methods. Academic Journal of Nawroz University. 2022; 8(11): 65-73. https://doi.org/10.25007/ajnu.v11n3a1244

[5] Sadeeq MI, Omar FM, Pirdawood MA. Numerical Solution of Hirota-satsuma Coupled Kdv System by Rbf-ps Method. The Journal of Duhok University. 2022; 25(2): 164-175. https://doi.org/10.26682/sjuod.2022.25.2.15

[6] Pirdawood MA, Sabawi YA. High-order solution of Generalized Burgers–Fisher Equation using compact finite difference and DIRK methods. Journal of Physics: Conference Series. 2021; 1999(1). https://doi.org/10.1088/1742-6596/1999/1/012088

[7] Sabawi A, Ahmed SB, Hamad HQ. Numerical Treatment of Allen's Equation Using Semi Implicit Finite Difference Methods. Eurasian Journal of Science and Engineering. 2022; 8(1): 90-100. https://doi.org/10.23918/eajse.v8i1p90

[8] Sun H, Zhang J. A high-order compact boundary value method for solving one-dimensional heat equations. Numerical Methods for Partial Differential Equations: An International Journal. 2003; 16(9): 846-857. https://doi.org/10.1002/num.10076

[9] Mohebbi A, Dehghan M. High-order compact solution of the one-dimensional heat and advection–diffusion equations. Applied mathematical modelling. 2010; 34(10): 3071-3084. https://doi.org/10.1016/j.apm.2010.01.013

[10] Biazar J, Mehrlatifan B. A compact finite difference scheme for reaction-convection-diffusion equation. hiang Mai Journal of Science. 2017; 3: 1559-1568.

[11] Sabawi YA. A posteriori error analysis in finite element approximation for fully discrete semilinear parabolic problems. In Finite Element Methods and Their Applications. IntechOpen. 2020.

[12] Ibrahim S. Numerical Approximation Method for Solving Differential Equations. Eurasian Journal of Science & Engineering. 2020; 6(2): 157-168. https://doi.org/10.23918/eajse.v6i2p157

[13] Sabawi YA. A Posteriori $L_\infty(H^1)$ Error Bound in Finite Element Approximation of Semdiscrete Semilinear Parabolic Problems. In 2019 first international conference of computer and applied sciences (cas), IEEE; 2019; Baghdad, Iraq. https://doi.org/10.1109/CAS47993.2019.9075699

[14] Saka B, Dağ İ. Quartic B-spline collocation method to the numerical solutions of the Burgers' equation. Chaos, Solitons & Fractals. 2007; 32(3): 1125-1137. https://doi.org/10.1016/j.chaos.2005.11.037

[15] Dağ İ, Saka B, Boz A. B-spline Galerkin methods for numerical solutions of the Burgers' equation. Applied Mathematics and Computation. 2005; 166(3): 506-522. https://doi.org/10.1016/j.amc.2004.06.078

[16] Ramadan A, El-Danaf aS, Abd Alaal FE. A numerical solution of the Burgers' equation using septic B-splines. Chaos, Solitons & Fractals. 2005; 26(4): 1249-1258. https://doi.org/10.1016/j.chaos.2005.02.019

[17] EL-Danaf TS, Raslan KR, Ali KK. Collocation method with cubic B-splines for solving the generalized regularized long wave equation. Collocation method with cubic B-splines for solving the generalized regularized long wave equation. 2016; 15(1): 39-59. http://dx.doi.org/10.17654/NM015010039

[18] Mittal RC, Tripathi A. Numerical solutions of generalized Burgers–Fisher and generalized Burgers–Huxley equations using collocation of cubic B-splines. International Journal of Computer Mathematics. 2015; 92(5): 1053-1077. https://doi.org/10.1080/00207160.2014.920834