Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Automating numerical partial differentiation

Henry Samambgwa * and Thomas Musora

Department of Mathematics and Statistics, School of Natural Sciences and Mathematics, Chinhoyi University of Technology, 78 Magamba Way, Off Chirundu Road, Chinhoyi, Zimbabwe.

Global Journal of Engineering and Technology Advances, 2026, 26(01), 069-078

Publication history: Received on 30 November 2025; revised on 05 January 2026; accepted on 07 January 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.26.1.0001

Abstract

This study develops an algorithm and MATLAB implementation for executing numerical partial differential operations of arbitrary order. The central difference numerical estimation scheme is adapted for repetitive numerical partial differentiation. A flowchart and MATLAB code are developed to programmatically execute numerical partial differential operations. The implementation is tested for three types of functions, a polynomial, trigonometric and exponential functions. In validation, the program demonstrated high accuracy with minimal error. The program improves analytic capabilities of researchers in dealing with large data sets to produce accurate results in limited time.

Keywords: Numerical Partial Differentiation; MATLAB source code; Algorithm; Flowchart

1. Introduction

The use of computational algorithms to solve mathematical problems immensely increases the capabilities of researchers in handling repeated complex arithmetic operations and analysing large volumes of data. Automation algorithms contribute positively to developing solutions for a wide arena challenging study sectors including economics, engineering, agriculture and astronomy among many others.

Where analytical solutions to partial differential equations are very difficult or impossible to find, alternative solutions can be estimated using infinite series [1] or obtained numerically using finite difference estimations [2,3]. Various studies have developed valuable methods for executing numerical partial differentiation.

Shior et al [4] solved the heat model partial differential equations using MATLAB. The study discretised temperature and time steps and simulated the evolution of temperature over one thousand time steps. The simulation correctly represented the physical spread governed by the heat equation. Schneider [5] used Ampere's and Faraday's laws to simulate the evolution of magnetic and electric fields over 250 time steps in a C++ programing environment. The study replaced differential terms with finite difference approximation formulae in the differential equations. Oufi [6] used the finite difference time domain simulation method to simulate partial differential equations. One of the priorities was reducing computational time. This was achieved by parallelising, that is simulating independent components concurrently. Ghadge [7] solved Laplace's equation using the second order central difference formula. By first indexing and laying out each point on a two dimensional square mesh, each new value was determined as the average of its four neighbouring values.

MATLAB is a conducive environment for mathematical analysis combined with software development and coding [8,9]. It hosts versatile functions that enable the use of modifiable variables in iterative schemes [10]. Calculations with a large number of iterations can be executed in a short amount of time [11,12]. Rathour et al [13] explored the MATLAB *dsolve*

* Corresponding author: Henry Samambgwa

functions for symbolic differentiation. They were able to solve bivariate ordinary differential equations without separating variables.

Recent studies developed methods for implementing symbolic partial differentiation programmatically [14]. Partial differential equations can be solved numerically by replacing differential terms with finite difference formulae [15,16]. Finite difference methods, derived from Taylor series [17], are widely used for solving numerical partial differentiation problems [18,19]. This study develops a scheme for executing partial differentiation operations of arbitrary order in MATLAB.

Aim

To develop a numerical scheme for partial differentiation of arbitrary order.

Objectives

- Develop a flow chart for executing numerical partial differentiation,
- Develop MATLAB source code that executes numerical partial differentiation of arbitrary order.

2. Methodology

2.1. Programing environment

The algorithm and source code developed in this study is adapted to the MATLAB online [20] analysis and programing environment.

2.2. Central difference differentiation scheme

The central difference numerical differentiation scheme is formally stated in equation (1) [21].

If $f(x, y)$ is a function in x and y, and h is the step size (small change in x), the partial derivative of $f(x, y)$ with respect to x, written $\frac{\partial f}{\partial x}$ or f_x is given by:

$$\frac{\partial f}{\partial x} = \frac{f(x+h, y) - f(x-h, y)}{2h}. \quad (1)$$

If k is the step size representing a small change in y, then the partial derivative of $f(x, y)$ w.r.t. y, written $\frac{\partial f}{\partial y}$ or f_y is given by:

$$\frac{\partial f}{\partial y} = \frac{f(x, y+k) - f(x, y-k)}{2k}. \quad (2)$$

The method can be extended to functions with an arbitrary number of input variables. If r is the step size representing a small change in a variable x_i where $f(x_1, x_2, \ldots, x_i, \ldots, x_n)$ is a function of n variables including x_i. Then the partial derivative of f w.r.t. x_i is given by:

$$\frac{\partial f}{\partial x_i} = \frac{f(x_1, x_2, \ldots, x_i + r, \ldots, x_n) - f(x_1, x_2, \ldots, x_i - r, \ldots, x_n)}{2r}. \quad (3)$$

2.3. Partial differential Implementation

The implementation in this study is developed to numerically estimate partial derivatives such as:

$$f_x, \quad f_y, \quad f_{xx}, \quad f_{xy}, \quad f_{yy}, \quad f_{xxx}, \quad f_{xxy}, \ldots \text{ and so on.}$$

The program developed in this research is for estimating partial derivatives of a function in two variables. The method can be extended, however, to an arbitrary number of variables.

Higher order derivatives are obtained by differentiating repeatedly, that is:

$$f_{xx} = \frac{\partial f_x}{\partial x}, \qquad f_{xy} = \frac{\partial f_x}{\partial y}, \qquad f_{xxy} = \frac{\partial f_{xx}}{\partial y}, \ldots \text{and so on.} \tag{4}$$

Using equations (1) and (2):

$$f_{xx} = \frac{\partial^2 f}{\partial x^2} = \frac{f_x(x+h, y) - f_x(x-h, y)}{2h}, \tag{5}$$

$$f_{xy} = \frac{\partial^2 f}{\partial x \partial y} = \frac{f_x(x, y+k) - f_x(x, y-k)}{2k}, \tag{6}$$

$$f_{xxy} = \frac{\partial^3 f}{\partial x \partial x \partial y} = \frac{f_{xx}(x, y+k) - f_{xx}(x, y-k)}{2k}, \tag{7}$$

. . . and so on.

2.4. Flowchart

The user must indicate the number of times to differentiate partially with respect to x (variable named dx in the program) and then the number of times to differentiate w.r.t. y (variable named dy in the program). For example, if the partial derivative to be estimated is f_{xxyyy}, the user must indicate partial differentiation to be 2 times w.r.t. x and 3 times w.r.t. y. The steps executed in the implementation are illustrated in the flowchart in **Figure 1**.

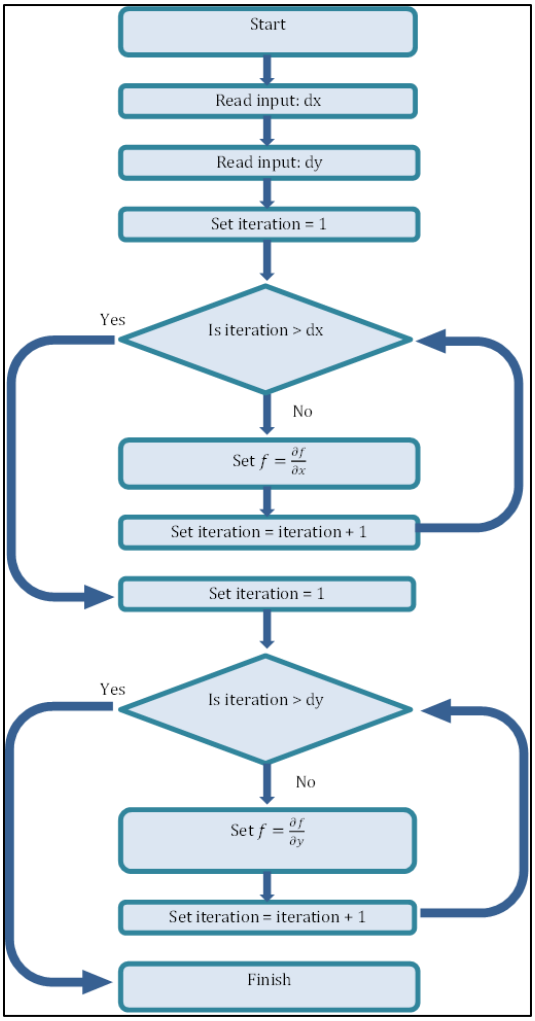

Figure 1 Flow chart for executing numerical partial differentiation.

2.5. MATLAB source code

The MATLAB source code in **Figure 2** executes the numerical partial differentiation steps illustrated by the flowchart in **Figure 1**.

```
%User sets values of f, dx, dy, h and k.

                                iteration=1;
while iteration<=dx
    f = @ (x, y) (f (x + h, y)-f (x-h, y))/(2*h);
    iteration=iteration+1;
end

iteration=1;
while iteration<=dy
    f = @ (x, y) (f (x, y + k)-f (x, y-k))/(2*k);
    iteration=iteration+1;
end
```

Figure 2 MATLAB source code for arbitrary order numerical partial differentiation.

3. Analysis

The numerical partial differentiation program was tested on three types of function. A polynomial, trigonometric and exponential functions.

Sample function A: $3x^2y^3 - 5x^3y + 18x - 5$,

Sample function B: $2\sin xy + x\cos y$,

Sample function C: $3e^{2y} - ye^x$.

Eight partial derivative operations were estimated for each function:

$$f_x, f_y, f_{xy}, \quad f_{xx}, \quad f_{yy}, \quad f_{xxy}, \quad f_{xyy}, \quad \text{and} \quad f_{xxyy}, \tag{8}$$

At nine sample points, $(x, y) \in \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)$, with step sizes $h = 0.05$ and $k = 0.05$.

The partial derivatives where estimated numerically and compared with values calculated from analytic formulae (**Table 1**). The results are summarised in **Tables 2-4**.

3.1. Analytic partial derivatives

Table 1 Analytic formulae for partial derivatives of sample functions A, B and C.

Partial differential operation	**Function**		
$f(x,y)$	$3x^2y^3 - 5x^3y + 18x - 5$	$2\sin xy + x\cos y$	$3e^{2y} - ye^x$
f_x	$6xy^3 - 15x^2y + 18$	$2y\cos xy + \cos y$	$-ye^x$
f_{xx}	$6y^3 - 30xy$	$-2y^2\sin xy$	$-ye^x$
f_y	$9x^2y^2 - 5x^3$	$2x\cos xy - x\sin y$	$6e^{2y} - e^x$
f_{xy}	$18xy^2 - 15x^2$	$2\cos xy - 2xy\sin xy - \sin y$	$-e^x$
f_{xxy}	$18y^2 - 30x$	$-4y\sin xy - 2xy^2\cos xy$	$-e^x$
f_{yy}	$18x^2y$	$-2x^2\sin xy - x\cos y$	$12e^{2y}$
f_{xyy}	$36xy$	$-4x\sin xy - 2x^2y\cos xy - \cos y$	0
f_{xxyy}	$36y$	$-4\sin xy\ -8xy\cos xy\ +\ 2x^2y^2\sin xy$	0

Table 2 Comparison of estimated numerical and calculated analytic values of partial derivatives for Sample function A.

Partial derivative		**Value**									**MSE**
Point		(-1,-1)	(-1,0)	(-1,1)	(0,-1)	(0,0)	(0,1)	(1,-1)	(1,0)	(1,1)	
f_x	Numeric	39.013	18.000	-3.013	18.013	18.000	17.988	27.013	18.000	8.988	
	Analytic	39.000	18.000	-3.000	18.000	18.000	18.000	27.000	18.000	9.000	
	Error	-0.013	0.000	0.013	-0.013	0.000	0.012	-0.013	0.000	0.012	0.000
f_y	Numeric	14.008	5.008	14.008	0.000	0.000	0.000	4.008	-4.993	4.008	
	Analytic	14.000	5.000	14.000	0.000	0.000	0.000	4.000	-5.000	4.000	
	Error	-0.008	-0.008	-0.008	0.000	0.000	0.000	-0.008	-0.007	-0.008	0.000
f_{xy}	Numeric	-33.028	-15.028	-33.028	-0.013	-0.013	-0.013	3.003	-14.998	3.003	
	Analytic	-33.000	-15.000	-33.000	0.000	0.000	0.000	3.000	-15.000	3.000	
	Error	0.028	0.028	0.028	0.013	0.013	0.013	-0.003	-0.002	-0.003	0.000
f_{xx}	Numeric	-36.000	0.000	36.000	-6.000	0.000	6.000	24.000	0.000	-24.000	
	Analytic	-36.000	0.000	36.000	-6.000	0.000	6.000	24.000	0.000	-24.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{yy}	Numeric	-18.000	0.000	18.000	0.000	0.000	0.000	-18.000	0.000	18.000	
	Analytic	-18.000	0.000	18.000	0.000	0.000	0.000	-18.000	0.000	18.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{xxy}	Numeric	48.015	30.015	48.015	18.015	0.015	18.015	-11.985	-29.985	-11.985	
	Analytic	48.000	30.000	48.000	18.000	0.000	18.000	-12.000	-30.000	-12.000	
	Error	-0.015	-0.015	-0.015	-0.015	-0.015	-0.015	-0.015	-0.015	-0.015	0.000
f_{xyy}	Numeric	36.000	0.000	-36.000	0.000	0.000	0.000	-36.000	0.000	36.000	
	Analytic	36.000	0.000	-36.000	0.000	0.000	0.000	-36.000	0.000	36.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{xxyy}	Numeric	-36.000	0.000	36.000	-36.000	0.000	36.000	-36.000	0.000	36.000	
	Analytic	-36.000	0.000	36.000	-36.000	0.000	36.000	-36.000	0.000	36.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 3 Comparison of estimated numerical and calculated analytic values of partial derivatives for Sample function B.

Partial derivative		**Value**									**MSE**
Point		(-1,-1)	(-1,0)	(-1,1)	(0,-1)	(0,0)	(0,1)	(1,-1)	(1,0)	(1,1)	
f_x	Numeric	-0.540	1.000	1.621	-1.459	1.000	2.540	-0.540	1.000	1.621	
	Analytic	-0.540	1.000	1.621	-1.460	1.000	2.540	-0.540	1.000	1.621	
	Error	-0.001	0.000	0.001	-0.001	0.000	0.001	-0.001	0.000	0.001	0.000
f_y	Numeric	-1.921	-1.999	-0.239	0.000	0.000	0.000	1.921	1.999	0.239	
	Analytic	-1.922	-2.000	-0.239	0.000	0.000	0.000	1.922	2.000	0.239	
	Error	-0.001	-0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000
f_{xy}	Numeric	0.238	1.998	-1.445	2.839	2.000	1.156	0.238	1.998	-1.445	
	Analytic	0.239	2.000	-1.444	2.842	2.000	1.159	0.239	2.000	-1.444	
	Error	0.002	0.003	0.001	0.003	0.000	0.002	0.002	0.003	0.001	0.000
f_{xx}	Numeric	-1.682	0.000	1.682	0.000	0.000	0.000	1.682	0.000	-1.682	
	Analytic	-1.683	0.000	1.683	0.000	0.000	0.000	1.683	0.000	-1.683	
	Error	-0.001	0.000	0.001	0.000	0.000	0.000	0.001	0.000	-0.001	0.000
f_{yy}	Numeric	-1.142	0.999	2.221	0.000	0.000	0.000	1.142	-0.999	-2.221	
	Analytic	-1.143	1.000	2.223	0.000	0.000	0.000	1.143	-1.000	-2.223	
	Error	-0.001	0.001	0.002	0.000	0.000	0.000	0.001	-0.001	-0.002	0.000
f_{xxy}	Numeric	4.438	0.005	4.438	0.000	0.000	0.000	-4.438	-0.005	-4.438	
	Analytic	4.447	0.000	4.447	0.000	0.000	0.000	-4.447	0.000	-4.447	
	Error	0.009	-0.005	0.009	0.000	0.000	0.000	-0.009	0.005	-0.009	0.000
f_{xyy}	Numeric	3.898	-0.999	-4.978	-0.535	-0.999	-0.545	3.898	-0.999	-4.978	
	Analytic	3.906	-1.000	-4.987	-0.540	-1.000	-0.540	3.906	-1.000	-4.987	
	Error	0.008	-0.001	-0.009	-0.005	-0.001	0.005	0.008	-0.001	-0.009	0.000
f_{xxyy}	Numeric	-5.960	0.000	5.960	0.000	0.000	0.000	5.960	0.000	-5.960	
	Analytic	-6.005	0.000	6.005	0.000	0.000	0.000	6.005	0.000	-6.005	
	Error	-0.045	0.000	0.045	0.000	0.000	0.000	0.045	0.000	-0.045	0.001

Table 4 Comparison of estimated numerical and calculated analytic values of partial derivatives for Sample function C.

Partial derivative		**Value**									**MSE**
Point		(-1,-1)	(-1,0)	(-1,1)	(0,-1)	(0,0)	(0,1)	(1,-1)	(1,0)	(1,1)	
f_x	Numeric	0.368	0.000	-0.368	1.000	0.000	-1.000	2.719	0.000	-2.719	
	Analytic	0.368	0.000	-0.368	1.000	0.000	-1.000	2.718	0.000	-2.718	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	-0.001	0.000	0.001	0.000
f_y	Numeric	0.446	5.642	44.040	-0.187	5.010	43.408	-1.905	3.292	41.690	
	Analytic	0.444	5.632	43.967	-0.188	5.000	43.334	-1.906	3.282	41.616	
	Error	-0.001	-0.010	-0.074	-0.001	-0.010	-0.074	-0.001	-0.010	-0.074	0.002
f_{xy}	Numeric	-0.368	-0.368	-0.368	-1.000	-1.000	-1.000	-2.719	-2.719	-2.719	
	Analytic	-0.368	-0.368	-0.368	-1.000	-1.000	-1.000	-2.718	-2.718	-2.718	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.000
f_{xx}	Numeric	0.368	0.000	-0.368	1.001	0.000	-1.001	2.721	0.000	-2.721	
	Analytic	0.368	0.000	-0.368	1.000	0.000	-1.000	2.718	0.000	-2.718	
	Error	0.000	0.000	0.000	-0.001	0.000	0.001	-0.002	0.000	0.002	0.000
f_{yy}	Numeric	1.629	12.040	88.965	1.629	12.040	88.965	1.629	12.040	88.965	
	Analytic	1.624	12.000	88.669	1.624	12.000	88.669	1.624	12.000	88.669	
	Error	-0.005	-0.040	-0.296	-0.005	-0.040	-0.296	-0.005	-0.040	-0.296	0.030
f_{xxy}	Numeric	-0.368	-0.368	-0.368	-1.001	-1.001	-1.001	-2.721	-2.721	-2.721	
	Analytic	-0.368	-0.368	-0.368	-1.000	-1.000	-1.000	-2.718	-2.718	-2.718	
	Error	0.000	0.000	0.000	0.001	0.001	0.001	0.002	0.002	0.002	0.000
f_{xyy}	Numeric	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
	Analytic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
f_{xxyy}	Numeric	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
	Analytic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
	Error	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

4. Discussion

The MATLAB source code in Figure 2 was used to implement the selected numerical partial differentiation operations on sample functions A, B and C at nine sample points. The analytic formulae in Table 1 were used to calculate the exact values of the same partial derivatives at the same sample points. The results in Tables 2-4 show that the mean square errors were negligible for all the selected partial differential operations on all the tested sample functions at all the selected sample points. This indicates close alignment between the numerically estimated partial derivatives and the calculated exact partial derivative values. This demonstrates that the numerical partial differential estimation method developed in this study is highly accurate and can thus be used and trusted to produce accurate results.

5. Conclusion

This study successfully developed and validated an implementation for executing numerical partial differential operations or arbitrary order. The method demonstrated its reliability in producing highly accurate numerical estimates of partial derivatives. This improves the capabilities for researchers to analyse large data sets in little time and yield

highly accurate results. The study has laid a sturdy foundation for further research in partial differential analysis involving a large number of partial differential operations of arbitrary order.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declared no competing interests.

References

[1] Agbata, B. C., Ani, B. N., Shior, M. M., Ezugorie, I. G., Paul, R. V., Meseda, P. K. (2022). Analysis of Adomian Decomposition Method and Its Application for Solving Linear and Nonlinear Differential Equations. Asian Research Journal of Mathematics.

[2] Alsidrani, F., Kilicman, A., Senu, N. (2023). On the Modified Numerical Methods for Partial Differential Equations Involving Fractional Derivatives. Axioms. MDPI.

[3] Lee, R., & Zhang, T. (2023). Comparative study of numerical methods for one dimensional heat and wave equations using MATLAB. International Journal of Numerical Analysis and Modeling.

[4] Shior, M. M., Agbata, B. C., Obeng-Denteh, W., Kwabi, P. A., Ezugorie, I. G., Marcos, S., Asante-Mensa, F., Abah, E. (2024). Numerical Solution Of Partial Differential Equations Using Matlab: Applications To One-Dimensional Heat And Wave Equations. Scientia Africana. University of Portharcourt.

[5] Schneider, J. B. (2025). Understanding the Finite-Difference Time-Domain Method. Washington State University.

[6] Oufi, A. A. (2021). Computer Modeling Using The Finite-Difference Time-Domain (FDTD) Method for Electromagnetic Wave Propagation(FDTD) Method for Electromagnetic Wave Propagation. University of Maine.

[7] Ghadge, R. B. (2025). Analytical and numerical methods for solving partial differential equations. International Journal of Engineering, Science and Mathematics.

[8] Samambgwa, H., Musora, T., Kamusha, J. (2025). Deriving mathematical models from neural networks: A method for deducing individual effects of factors on a response variable. World Journal for Advanced Engineering Technology and Sciences.

[9] Shior, M. M., Agbata, B. C., Acheneje, G. O., Gbor, G. D., Musa, S. (2023). Solution of system and higher-order differential equations by fourth-order Runge-Kutta with MATLAB. East African Scientific Journal of Engineering and Computer Science.

[10] Samambgwa, H., Musora, T. (2025). Automating upper triangular matrix neutralisation for minimal error. World Journal of Advanced Engineering Sciences and Technology.

[11] Samambgwa, H., Musora, T. (2025). Automating Symbolic Partial Differentiation on Multivariate Polynomials. World Journal of Advanced Engineering Sciences and Technology.

[12] Musora, T. (2023). Application of Panel Data Analysis and Modelling to Economic Data: A Case of Determinants of Economic Growth for SADC and Zimbabwe. Chinhoyi University of Technology.

[13] Rathour, L., Obradovic, D., Mishra, L. N., Khatri, K., Mishra, V. N. (2023). Solving partial differential equations in MATLAB. Journal of Advances in Physics.

[14] Samambgwa, H. Musora, T. (2025). Automating symbolic partial differentiation on multivariate polynomials. Global Journal of Engineering and Technology Advances.

[15] Behera, S., Behera, D. K. (2024). Use of various finite difference methods for solving partial differential equations.

[16] Adak, M. (2020). Comparison of Explicit and Implicit Finite Difference Schemes on Diffusion Equation. Mathematical Modeling and Computational Tools. Springer Proceedings in Mathematics & Statistics.

[17] Mojumder, S. H., Haque, N., Alam. (2023). Efficient Finite Difference Methods for the Numerical Analysis of One-Dimensional Heat Equation. Journal of Mathematics and Physics. Scientific Research Publishing.

[18] Shearer, M., & Levy, R. (2020). Introduction to PDEs and waves for the atmosphere and ocean. University of North Carolina. Chapel Hill.

[19] Rathod, C. (2024). Paper on numerical solutions of partial differential equations. Shodkosh: Journal of Visual and Performing Arts.

[20] MATLAB Online. (2025). Accessed at: matlab.mathworks.com. Mathworks.

[21] Sambu, S. (2025). A Study on the Accuracy of Central Difference and Finite Element Techniques for Solving Elliptic Partial Differential Equations. International Journal of Integrated Sciences and Mathematics.