Effect of the Forcing Function values on the Numerical Simulation of General Second Order Linear Partial Differential Equation

Liaquat Ali Zardari[1,*], Shakeel Ahmed Kamboh[1], Abbas Ali Ghoto[1] Nawab Khan Chand[2]

[1]Department of Mathematics and Statistics, QUEST, Nawabshah, Pakistan
[2]Department of Natural Sciences, Begum Nusrat Bhutto Women University, Sukkur, Pakistan
[*]Corresponding author: laz19msm@gmail.com

Abstract

The second-order PDEs are used to model a wide variety of application problems and due to their indispensable usage the computational analysis of various aspects have worked out by researchers in the field. In this study, a computational analysis of the finite difference numerical solution of the general linear second-order PDE with constant coefficients is presented. The study's objective was to examine the simultaneous effect of the varied size of the computational domain and the forcing function values on the numerical simulation. From the results, it is revealed that the forcing function affects the simulation patterns as long as the size of the domain is increased from 1×1 square unit to 2×2, 3×3, and 4×4. Also, the local node-wise solution values change considerably with the varied values of the forcing function. The effect of forcing function G values on the numerical solution is observed higher when $G < 0$ and is lower when $G > 0$. The outcomes of this research study are expected to provide ways to predict the simulations obtained by the general second-order PDE based on the varied domain size and the forcing function with constant coefficients of the PDE.

Keywords—Second-order partial differential equation, Finite difference method, Forcing function, computational analysis

1 Introduction

The general second-order partial differential equation has immense applications in a vast majority of fields of science and engineering. The various classical equations have been derived from the general second-order PDE; such as Laplace's equation, Poisson's Equations, Heat equation, Wave equation, Helmholtz equation, Convection-Diffusion equation, and many others. The solution of some classes of second-order PDEs is obtained by using different analytical and numerical methods. However, numerical methods are often considered to solve practical problems due to their easy implementation on modern computers. Numerical methods usually are based on the discretization of the continuous problem that consequently leads to the system of an algebraic equation. Then be using any iterative method and starting from the initial guess the solution is updated in each step and the process is truncated until the desired accuracy is achieved. The consistency, stability, wellposedness and convergence properties of numerical methods are often considered important research directions [3]. Though, the convergence of iterative methods has achieved much attention in the last few decades. In fact, the general second-order PDE is used to describe one of the many physical phenomena involving an unknown function (dependent variable) under the influence of any given forcing function. The forcing function may affect the simulation patterns of the dependent variable in the computational domain. It has to yet investigate the effect of the forcing function involved in the general second-order PDE on the convergence and behavior of the numerical simulation. Therefore, this study is aimed at investigating the effect of the forcing function values using the finite difference method. To achieve this goal, a 2D general second-order PDE with Dirichlet boundary conditions on the square domain of different sizes is defined to obtain the numerical simulations of the dependent variable.

Mathematically, the general form of the second-order linear PDE in 2D Cartesian coordinates is given as follows [1]:

ISSN: 2523-0379 (Online), ISSN: 1605-8607 (Print)
DOI: https://doi.org/10.52584/QRJ.2002.05.

$$G(x,y) = \begin{array}{l} A(x,y)\frac{\partial^2 u(x,y)}{\partial^2 x} + B(x,y)\frac{\partial^2 u(x,y)}{\partial x \partial y} + \\ C(x,y)\frac{\partial^2 u(x,y)}{\partial^2 y} + D(x,y)\frac{\partial u(x,y)}{\partial x} + \\ E(x,y)\frac{\partial u(x,y)}{\partial y} + F(x,y)u(x,y) \end{array} \quad (1)$$

where $u(x,y)$ is the dependent variable, x and y be the independent variables, A, B, C, D, E, and F are the coefficients depending upon x and y; and G is the forcing function. In the above Eq. (1), if the independent variable y is replaced by time t then the PDE will be considered a time-dependent general second-order PDE.

In literature; few related but not almost the same general second-order PDEs have been discussed by using the various solution strategies. For instance; a new technique for the approximate solution of a second-order PDE was proposed by separating it into three parts with the introduction of two auxiliary unknown functions [2]. The collocation method was applied by [3] to obtain the solution of linear second-order PDE; the method was based on the Bessel function of the first kind.

The first attempt to apply the meshless GFDM (Generalized Finite Difference Method) for the numerical solution of inverse heat source problems associated with steady-state heat conduction was described by [4]. Liebmann's finite difference method based on Gauss-Seidel iterations is useful for the solution of second-order linear elliptic PDE with precise boundary conditions [5]. The Boundary Value Methods (BVM) has also been used to approximate second-order PDEs, whereas the Lanczos-Chebyshev reduction procedure has been used to transform the PDEs into an equivalent second-order system [6]. Some practical numerical methods have been examined to solve a class of initial-boundary value variable coefficient fractional PDEs over the finite domain [7]. In conjunction with spectral methods, the implicit-explicit finite difference schemes have mainly been used for spatially discretized diffusion-convection type PDEs. On average, an implicit and explicit scheme is for the dispersal term and for the convection term respectively [8]. A two-dimensional time-dependent initial–boundary value problem with distributed-order space-fractional PDE was considered by [9]. The finite difference method with irregular arguments of nodes was applied to solve the second-order PDEs with any boundary conditions (Dirichlet, Neumann, and Mixed) values of derivatives of the nodes are obtained through the application of the differences formulae [10]. In many cases when the problems are defined over irregular clouds of points the mesh-free GFDM is suitable for solving the elliptic and parabolic PDE. [11] have solved different examples of PDEs over irregular clouds of nodes. The role of integer and non-integer order PDEs is essential in applied sciences and engineering. Exact solutions of these equations are sometimes difficult to find [12]. Based on the concept of two-grid a fast and efficient algorithm was derived from the finite difference method to solve the Poisson equation [13]. For solving an elliptic problem using a finite difference scheme, they arrived at a large sparse linear algebraic system. Similarly, many others [14-20] have attempted to solve the second-order linear PDE by a variety of analytical and numerical methods. In finding the numerical solution of Eq. (1) using the finite difference method the effect of the forcing function may change the solution distribution in the domain in relation to the size of the computational domain. For a particular case, the numerical solution of Eq. (1) on the unit square of varied size is to investigate the change in the solution values with respect to different values of the forcing function. The choice of the values of the forcing function may reveal the convergence of the numerical solution and may also be useful to understand the numerical simulation profiles.

2 Materials and Methods

This study is aimed at the computational analysis of the 2D general second-order linear partial differential equation. Thus, the first step in the methodology is to construct, define the initial and boundary conditions and discretize the Eq. (1) by using the finite difference method.

A 2D Cartesian domain $(x,y) \in \Omega = [0,L] \times [0,W]$ is considered and the Eq. (1) is defined on this domain. The schematic of the computation domain is exhibited by the following Figure 1.

The boundary conditions are of Dirichlet type and are defined as follows:

$u(x,0) = f_b(x,0), where 0 \le x \le L$
$u(x,W) = f_t(x,W), where 0 \le x \le L$
$u(0,y) = g_l(0,y), where 0 \le y \le W$
$u(L,y) = g_r(L,y), where 0 \le y \le W$

where the f_b, f_t, g_l, and g_r are the bottom, top, left, and right boundary conditions respectively.

After assigning the boundary conditions, Eq. (1) is discretized using the central finite difference schemes. Eq. 2 shows the discrete version of Eq. (1). Moreover, Eq. (2) can be solved explicitly by the Leibmaan iterative algorithm and the solution on the mesh nodes can be obtained as in Eq. 3.

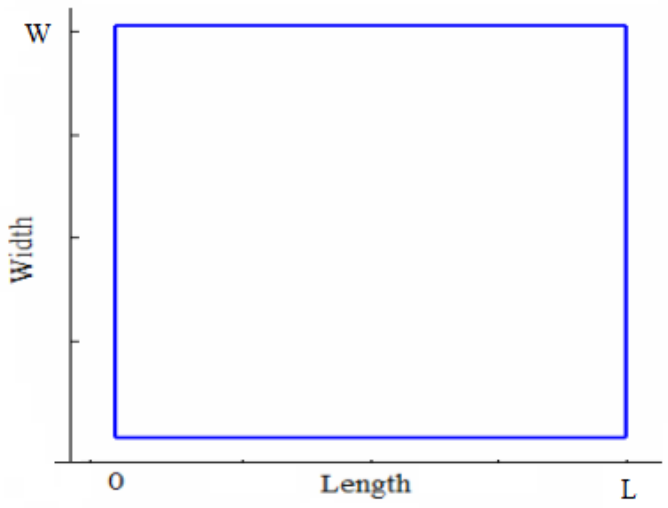

Fig. 1: Schematic of the computational domain for the solution of Eq. (1).

$$G(i,j) = \begin{array}{l} A(i,j)\frac{u(i-1,j)-2u(i,j)+u(i+1,j)}{h_1^2}+ \\ B(i,j)\frac{u(i+1,j+1)u(i+1,j-1)-u(i-1,j+1)+u(i-1,j-1)}{4h_1h_2}+ \\ C(i,j)\frac{u(i,j-1)-2u(i,j)+u(i,j+1)}{h_2^2}+ \\ D(i,j)\frac{u(i+1,j)+u(i-1,j)}{2h_1}+ \\ E(i,j)\frac{u(i,j+1)-u(i,j-1)}{2h_2}+ \\ F(i,j)u(i,j) \end{array} \tag{2}$$

$$u(i,j) = -\frac{1}{4}\frac{1}{4(2A(i,j)h_2^2+2C(i,j)h_1^2))-F(i,j)h_1^2h_2^2)}\times \begin{pmatrix} -4A(i,j)h_2^2u(i,j) - B(i,j)h_1h_2u(i+1,j+1) \\ +B(i,j)h_1h_2u(i+1,j-1) + B(i,j)h_1h_2u(i-1,j+1) \\ -B(i,j)h_1h_2u(i-1,j-1) - 4C(i,j)h_1^2u(i,j-1) \\ -4C(i,j)h_1^2u(i,j+1) - 2D(i,j)h_1h_2^2u(i+1,j) \\ +2D(i,j)h_1h_2^2u(i-1,j) + 2E(i,j)h_1^2h_2u(i,j+1) \\ +2E(i,j)h_1^2h_2u(i,j-1) + 4G(i,j)h_1^2h_2^2 \end{pmatrix}. \tag{3}$$

The associated schematic of the discretized domain is shown as a mesh of nodes in Figure 2.

For the consistent and converged solution it must follow that $2A(i,j)h_1^2 + 2C(i,j)h_1^2 - F(i,j)h_1^2h_2^2 \neq 0$ and $h_1 \neq 0, h_2 \neq 0$.

After specifying the numerical solution algorithm a user-defined MATLAB code is written to implement the methodology. The varied values of the forcing function have been tested for the convergence of the numerical solution in relation to the domain size. The data is collected for the computational analysis of the convergence and simulation profiles.

3 Results and Discussions

The simulation profiles of Eq (1) for about twenty different combinations were obtained. The computational time and the number of iterations required to converge for each combination of $(L, W, h_1, h_2 and G)$ are listed in Table 1. It is observed from Table 1. that the number of iterations increases when either the domain size or the step size is increased. While for each specific domain size or step size the iterations do not vary much in relation to the varied values of the forcing function. The numerical simulation profile is obtained for each of the combinations from Table 1. and is analyzed for the simultaneous effect of the domain size and forcing function values. Figures (3-7) exhibit the numerical

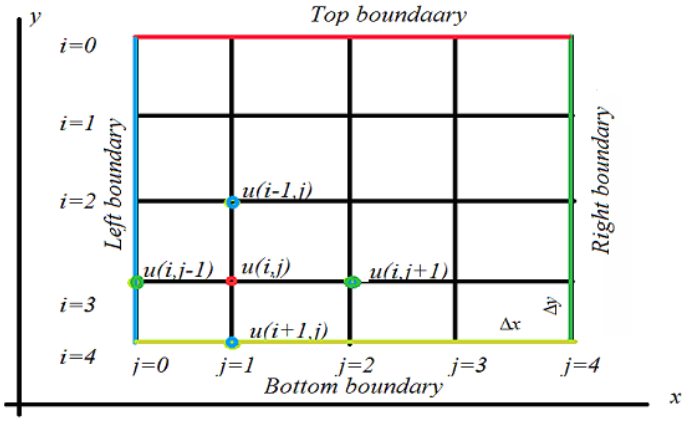

Fig. 2: Schematic of the discretized domain for the solution of Eq. (1).

simulation profiles of Eq. (1) for different values of the forcing function when the domain is of size 1 × 1 square unit. It can be seen that there is not much difference in the simulation patterns as the forcing function values do not affect more when the domain is of 1 × 1 square unit. In order to expose the local changes in the solution values the node-wise solution distribution is demonstrated in Figure 8. From the figure, it appears that for each trial the numerical solution at specific nodes remains near to each other. Interestingly, it is revealed that the forcing function affects the simulation patterns to some extent when the size is increased from 1 × 1 square unit to 2 × 2 square unit as shown in Figures (9-13). And the local node-wise solution distribution shown in Figure 14 exposes the increase in the node-wise solution values when G<0 and the decrease in the node-wise solution values when G>0. Similarly, when the domain size is set to 3 × 3 square unit the simulation profiles significantly change as compared to 1 × 1 or 2 × 2 sized domain. The local solution values also change considerably as can be seen in Figures (15-19). This time the impact of the forcing function for G<0 is even higher than the higher boundary condition applied at the top of the domain and a cavity-like pattern appears along the upper region of the domain. Finally, when the domain size is taken 4 × 4 the clear and abrupt changes in the simulation profiles of Eq. (1) are observed in Figures (20-25), not only this but there exists a cavity with respect to each value of the forcing function. The intensities of the local solution values at each node are increased significantly. In fact, the negative values of G affect the solution more and more as compared to the positive values of the forcing function G. It can be deduced that increasing the scaling factor for the domain size increases the local solution values for G<0 and decreases the local solution values for G>0.

It is still worthwhile to further generalize the results of this study to more levels but the computational complexity may confine the analysis unexpectedly. Such type of simulation patterns could be useful in the understanding of heat conduction, heat transfer, electrostatics, or fluid flow problems. There is plenty of space to further investigate the extent of simultaneous ranges of the domain size and forcing function and other coefficients of Eq. (1).

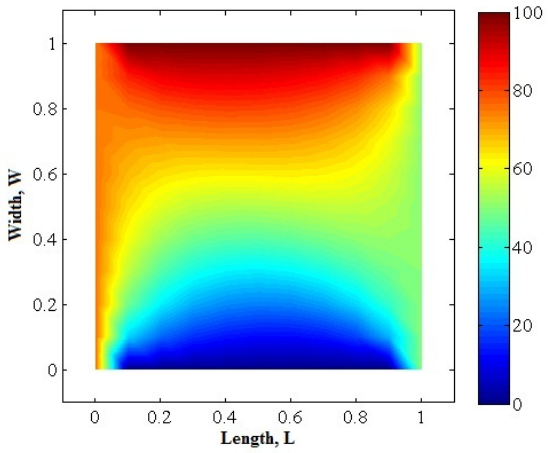

Fig. 3: Numerical simulation of Eq. (1) when L=1, W=1, h_1=0.1, h_2=0.1, G=-10

4 Conclusions

In this study a computational analysis of the general linear second-order PDE in relation to the varied domain size and the forcing function with fixed coefficients was carried out. From the analysis of numerical simulation profiles, it was revealed that in the case of the unit square domain the simulation, patterns do not affect more by changing the forcing function values.

TABLE 1: Computational time and convergence analysis at various combinations of domain size and the forcing function when A=B=C=D=E=F=1; f_b=0, f_t=100, g_l=75, and g_r=75.

S. No	L	W	h_1	h_2	G	Time (sec)	Iterations
1	1	1	0.1	0.1	-10	0.0195	231
2	1	1	0.1	0.1	-5	0.0127	230
3	1	1	0.1	0.1	0	0.0133	230
4	1	1	0.1	0.1	5	0.0136	230
5	1	1	0.1	0.1	10	0.0126	230
6	2	2	0.2	0.2	-10	0.023	260
7	2	2	0.2	0.2	-5	0.0172	258
8	2	2	0.2	0.2	0	0.0172	258
9	2	2	0.2	0.2	5	0.0158	258
10	2	2	0.2	0.2	10	0.0145	257
11	3	3	0.3	0.3	-10	0.0229	330
12	3	3	0.3	0.3	-5	0.0233	329
13	3	3	0.3	0.3	0	0.0251	327
14	3	3	0.3	0.3	5	0.0232	325
15	3	3	0.3	0.3	10	0.023	323
16	4	4	0.4	0.4	-10	0.0386	534
17	4	4	0.4	0.4	-5	0.0412	530
18	4	4	0.4	0.4	0	0.0375	526
19	4	4	0.4	0.4	5	0.0359	521
20	4	4	0.4	0.4	10	0.0359	515

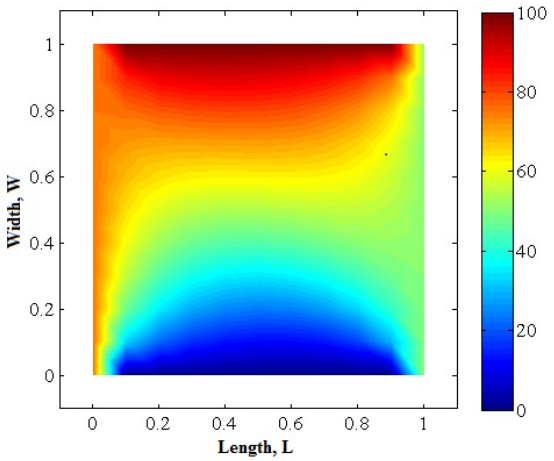

Fig. 4: Numerical simulation of Eq. (1) when L=1, W=1, h_1=0.1, h_2=0.1, G=-5

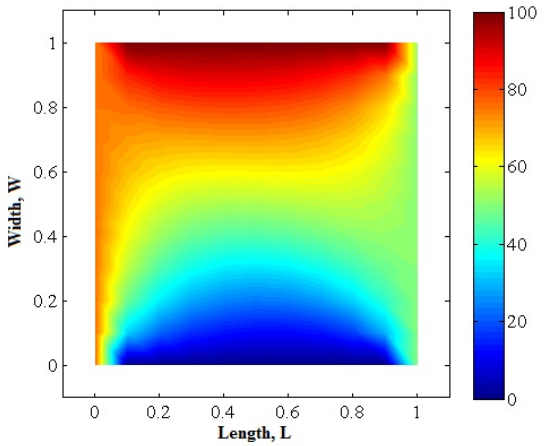

Fig. 5: Numerical simulation of Eq. (1) when L=1, W=1, h_1=0.1, h_2=0.1, G=0

But in the case of the non-unit square domain, both the simulation patterns and the node-wise solution values do significantly vary by the simultaneous change in the domain size and the forcing function values. The negative values of the forcing function affect the solution more and more as compared to the positive values of the forcing function. Further research could be conducted to generalize the results of this study by

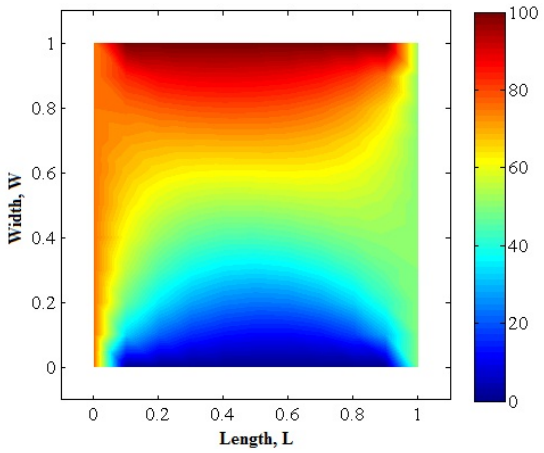

Fig. 6: Numerical simulation of Eq. (1) when L=1, W=1, h_1=0.1, h_2=0.1, G=5.

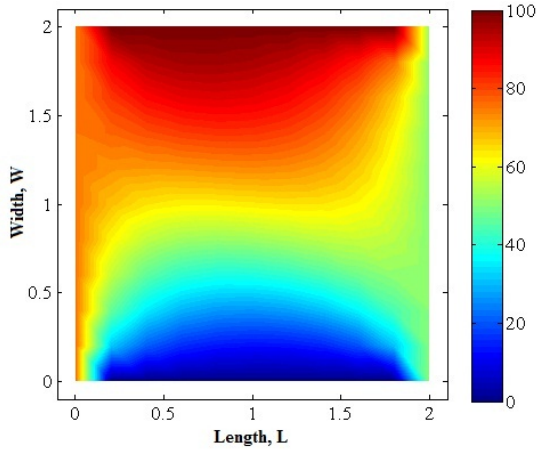

Fig. 9: Numerical simulation of Eq. (1) when L=2, W=2, h_1=0.2, h_2=0.2, G=-10.

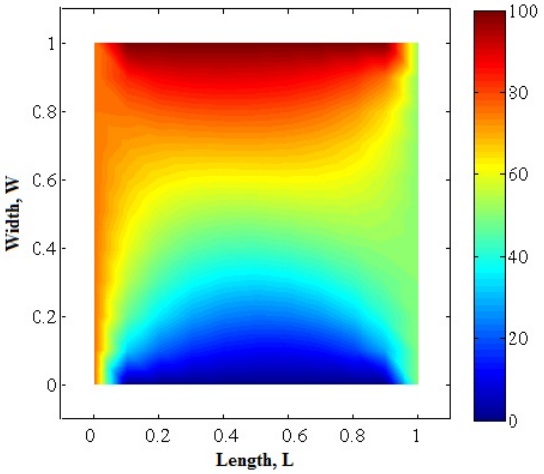

Fig. 7: Numerical simulation of Eq. (1) when L=1, W=1, h_1=0.1, h_2=0.1, G=10.

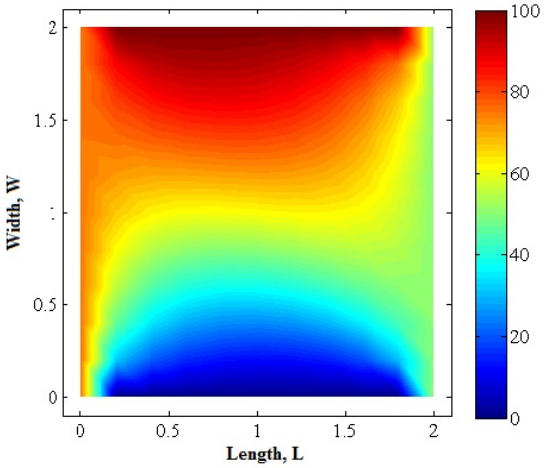

Fig. 10: Numerical simulation of Eq. (1) when L=2, W=2, h_1=0.2, h_2=0.2, G=-5.

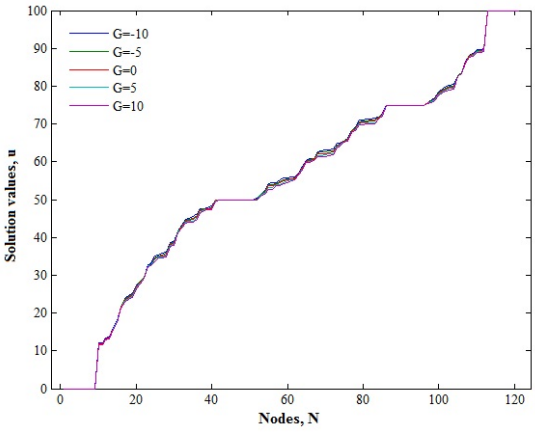

Fig. 8: Node-wise effect of forcing function G on the numerical solution when the domain size is 1 × 1.

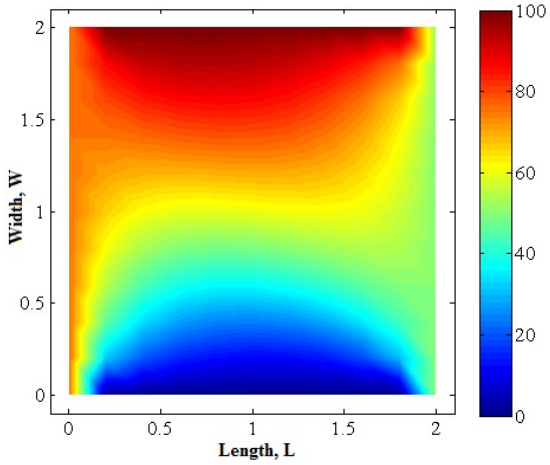

Fig. 11: Numerical simulation of Eq. (1) when L=2, W=2, h_1=0.2, h_2=0.2, G=0.

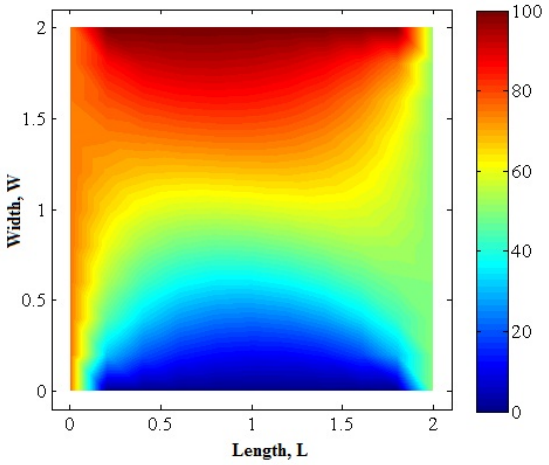

Fig. 12: Numerical simulation of Eq. (1) when L=2, W=2, h_1=0.2, h_2=0.2, G=5.

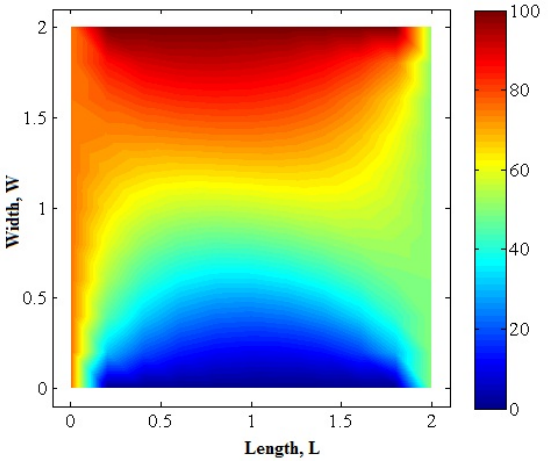

Fig. 13: Numerical simulation of Eq. (1) when L=2, W=2, h_1=0.2, h_2=0.2, G=10.

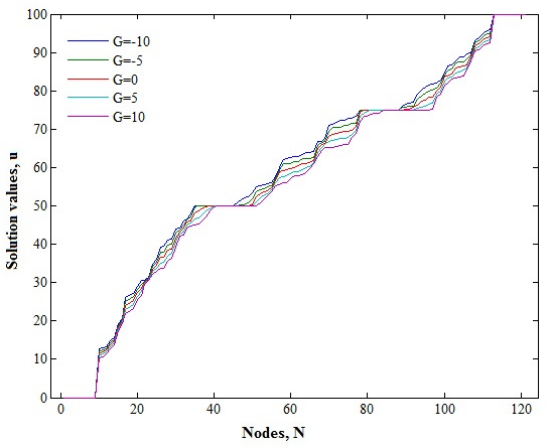

Fig. 14: Node-wise effect of forcing function G on the numerical solution when the domain size is 2 × 2.

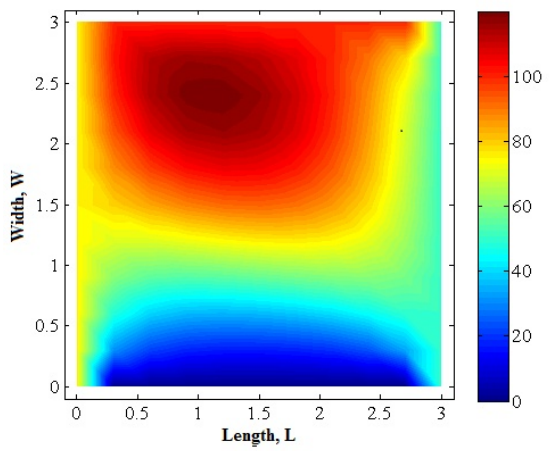

Fig. 15: Numerical simulation of Eq. (1) when L=3, W=3, h_1=0.2, h_2=0.2, G=-10.

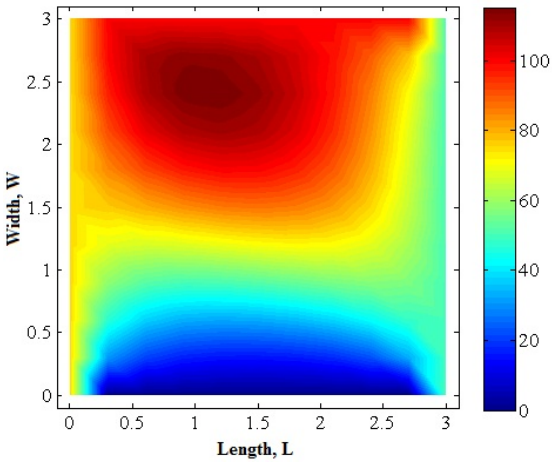

Fig. 16: Numerical simulation of Eq. (1) when L=3, W=3, h_1=0.2, h_2=0.2, G=-5.

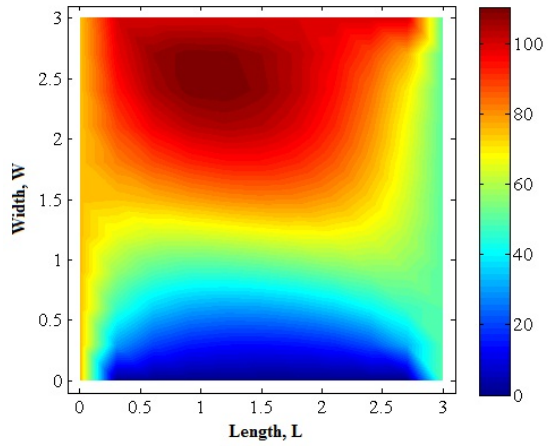

Fig. 17: Numerical simulation of Eq. (1) when L=3, W=3, h_1=0.2, h_2=0.2, G=0.

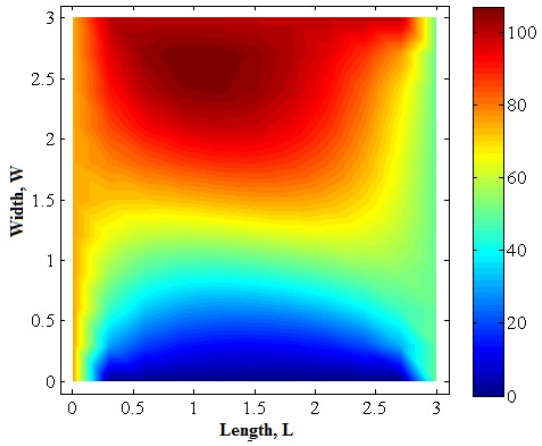

Fig. 18: Numerical simulation of Eq. (1) when L=3, W=3, h_1=0.2, h_2=0.2, G=5.

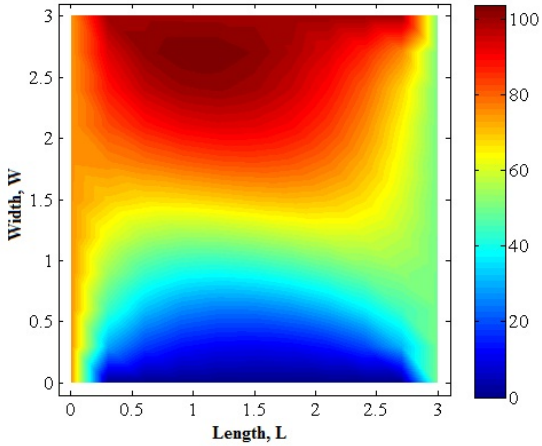

Fig. 19: Numerical simulation of Eq. (1) when L=3, W=3, h_1=0.2, h_2=0.2, G=10.

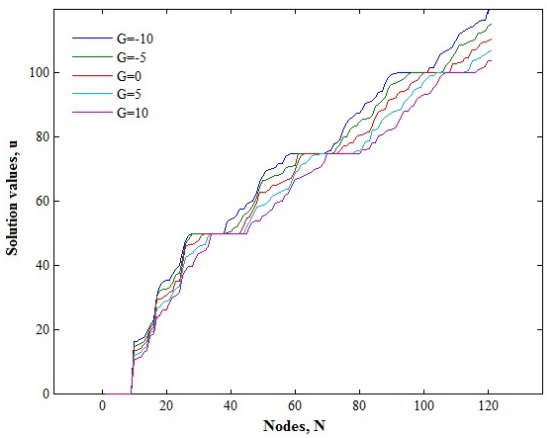

Fig. 20: Node-wise effect of forcing function G on the numerical solution when the domain size is 3 × 3.

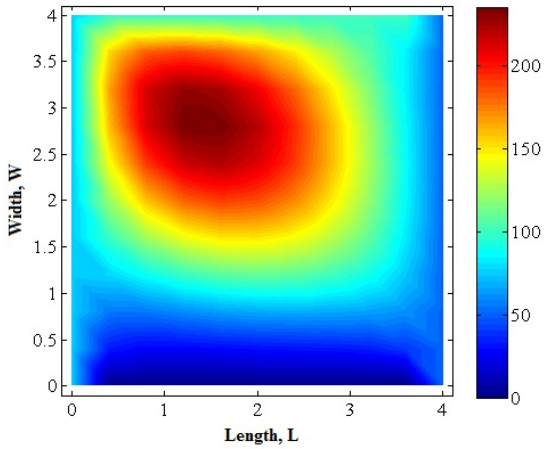

Fig. 21: Numerical simulation of Eq. (1) when L=4, W=4, h_1=0.2, h_2=0.2, G=-10.

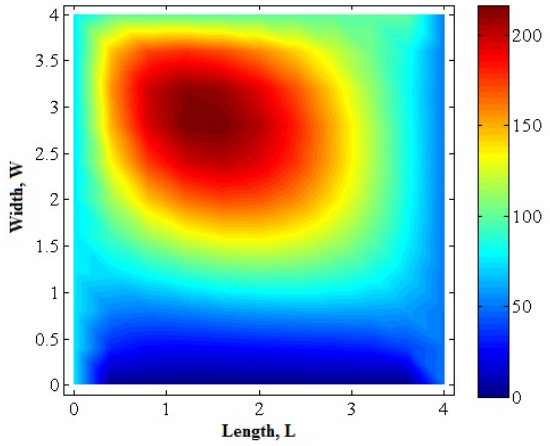

Fig. 22: Numerical simulation of Eq. (1) when L=4, W=4, h_1=0.2, h_2=0.2, G=-5.

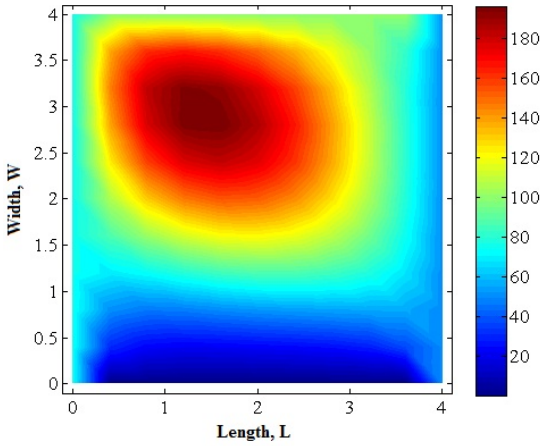

Fig. 23: Numerical simulation of Eq. (1) when L=4, W=4, h_1=0.2, h_2=0.2, G=0.

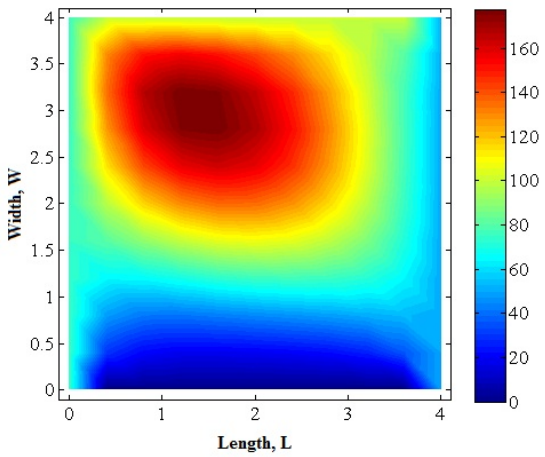

Fig. 24: Numerical simulation of Eq. (1) when L=4, W=4, h_1=0.2, h_2=0.2, G=5.

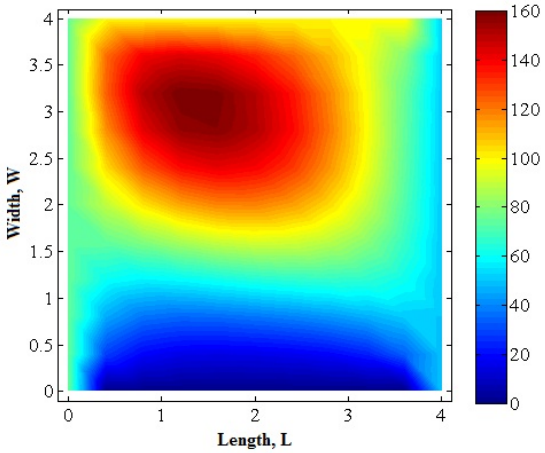

Fig. 25: Numerical simulation of Eq. (1) when L=4, W=4, h_1=0.2, h_2=0.2, G=10.

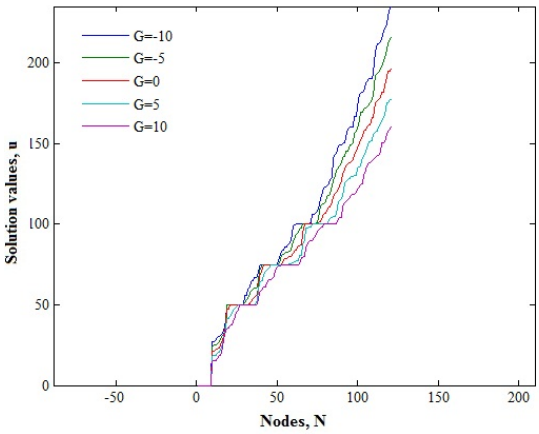

Fig. 26: Node-wise effect of forcing function G on the numerical solution when the domain size is 4 × 4.

varying the other coefficients involved in the general second-order PDE.

5 Acknowledgements

The authors are highly grateful to the Quaid-e-Awam University of Engineering, Science and Technology for providing facilities to conduct this research.

References

[1] Boas, M. L. "Mathematical methods in the physical sciences". John Wiley & Sons, (2006).

[2] Niakas, N. T., V. C. Loukopoulos, and Christos Douskos. "FDM on second-order partial differential equations in 3D." Mathematical and Computer Modelling 52, no. 1-2 (2010): 278-283.

[3] Yüzbaşı, Şuayip. "A numerical method for solving second-order linear partial differential equations under Dirichlet, Neumann and Robin boundary conditions." International Journal of Computational Methods 14, no. 02 (2017): 1750015.

[4] Gu, Yan, Lei Wang, Wen Chen, Chuanzeng Zhang, and Xiaoqiao He. "Application of the meshless generalized finite difference method to inverse heat source problems." International Journal of Heat and Mass Transfer 108 (2017): 721-729.

[5] Ndiyo, E. E., and U. A. Abasiekwere. "Comparative Analysis of Finite Difference Methods for Solving Second Order Linear Partial Differential Equations." International Journal of Mathematics Trends and Technology (IJMTT), 57(4), (2018): 277-283.

[6] Biala, T. A., S. N. Jator, and R. B. Adeniyi. "Boundary Value Methods for Second-Order PDEs via the Lanczos-Chebyshev Reduction Technique." Mathematical Problems in Engineering. Volume 2017 (2017).

[7] Meerschaert, Mark M., and Charles Tadjeran. "Finite difference approximations for two-sided space-fractional partial differential equations." Applied numerical mathematics 56, no. 1 (2006): 80-90.

[8] Fu, Hongfei, and Hong Wang. "A preconditioned fast finite difference method for space-time fractional partial differential equations." Fractional Calculus and Applied Analysis 20, no. 1 (2017): 88-116.

[9] Jia, Jinhong, and Hong Wang. "A fast finite difference method for distributed-order space-fractional partial differential equations on convex domains." Computers mathematics with applications 75, no. 6 (2018): 2031-2043.

[10] Benito, J. J., F. Urena, and L. Gavete. "Influence of several factors in the generalized finite difference method." Applied Mathematical Modelling 25, no. 12 (2001): 1039-1053.

[11] Gavete, L., J. J. Benito, and F. Ureña. "Generalized finite differences for solving 3D elliptic and parabolic equations." Applied Mathematical Modelling 40, no. 2 (2016): 955-965.

[12] Ahmad, Hijaz, Ali Akgül, Tufail A. Khan, Predrag S. Stanimirović, and Yu-Ming Chu. "New perspective on the conventional solutions of the nonlinear time-fractional partial differential equations." Complexity Volume 2020 (2020): 1-10.

[13] Moghaderi, Hamid, Mehdi Dehghan, and Masoud Hajarian. "A fast and efficient two-grid method for solving d-dimensional poisson equations." Numerical Algorithms 72 (2016): 483-537.

[14] Salsa, Sandro. Partial differential equations in action: from modelling to theory. Vol. 99. Springer, 2016.
[15] Simos, T. E., and Ch Tsitouras. "Evolutionary generation of high-order, explicit, two-step methods for second-order linear IVPs." Mathematical Methods in the Applied Sciences 40, no. 18 (2017): 6276-6284.
[16] Rudy, Samuel H., Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. "Data-driven discovery of partial differential equations." Science advances 3, no. 4 (2017): e1602614.
[17] Nakao, Mitsuhiro T., Michael Plum, and Yoshitaka Watanabe. Numerical verification methods and computer-assisted proofs for partial differential equations. Berlin: Springer, 2019.
[18] Rudy, Samuel, Alessandro Alla, Steven L. Brunton, and J. Nathan Kutz. "Data-driven identification of parametric partial differential equations." SIAM Journal on Applied Dynamical Systems 18, no. 2 (2019): 643-660.
[19] Robertsson, Johan OA, and Joakim O. Blanch. "Numerical methods, finite difference." Encyclopedia of solid earth geophysics (2020): 1-9.
[20] Ureña, F., L. Gavete, J. J. Benito, A. García, and A. M. Vargas. "Solving the telegraph equation in 2-D and 3-D using generalized finite difference method (GFDM)." Engineering Analysis with Boundary Elements 112 (2020): 13-24.