Multidiszciplináris Tudományok, 15. kötet, 1. sz. (2025) pp. 79–88. **https://doi.org/10.35925/j.multi.2025.1.7**

TESTING AND COMPARING THE PERFORMANCE OF NUMERICAL METHODS FOR THE HEAT CONDUCTION EQUATION PARTIAL DIFFERENTIAL EQUATION

Andicha Zain
PhD Student, University of Miskolc, Institute of Mathematics
3515 Miskolc-Egyetemváros, e-mail: andicha.zain@student.uni-miskolc.hu

Krisztián Hriczó
associate professor, University of Miskolc, Institute of Mathematics
3515 Miskolc-Egyetemváros, e-mail: krisztian.hriczo@uni-miskolc.hu

Imre Ferenc Barna
associate professor, Wigner Research Centre for Physics
1121 Budapest, Konkoly-Thege Miklós út 29–33, e-mail: barna.imre@wigner.hu

Abstract

This research collected 12 numerical algorithms that solve the transient diffusion equation with Dirichlet boundary conditions in one space dimension. Some of these methods are explicit and unconditionally stable simultaneously. A nontrivial analytical solution, recently obtained by a self-similar Ansatz, served as the reference solution. The errors were calculated and plotted as a function of the time step size and execution times. The conclusion is that there are explicit and stable methods that provide acceptable results faster than the traditional Runge-Kutta style methods.

Keywords: *Heat conduction, diffusion equation, stable numerical methods, explicit time integration, Runge-Kutta methods*

1. Introduction

The following partial differential equation (PDE) describes the movement of both thermal energy and particles:

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial u^2(x,t)}{\partial x^2}, \; u(x, t = 0) = u^0(x). \tag{1}$$

In this equation, $t, x \in \mathbb{R}$ represent variables for time and space, while $u: \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}; (x, t) \mapsto u(x, t)$ denotes the function that needs to be determined. In the context of simulations, the variable u takes on different meanings: It represents the temperature in heat conduction models (Yunus et al., 2015) and the concentration in particle diffusion studies (Agbavon et al., 2019). The positive constant α\alphaα is called the diffusion coefficient or diffusivity. The function u0u^0u0 is supposed to be given and usually called the initial condition. The significance of this equation persists, even though it was discovered more than a century ago. Notably, recent advancements in this field have yielded novel analytical solutions (Barna et

al., 2022; Barna et al., 2024; Mátyás et al., 2022; Mátyás et al., 2023; Kovács et al., 2024), further enhancing their relevance in contemporary research. This equation, along with other equations incorporating diffusion terms, is commonly addressed through numerical approaches. Among these, finite difference schemes (FDMs) stand out as a particularly popular method for solving such equations (Köroğlu et al., 2021). The categorization of these approaches encompasses both explicit and implicit methodologies, with intermediate variants such as semi-explicit and semi-implicit strategies existing within this spectrum (Beuken et al., 2022; Fedoseev et al., 2022). Explicit methods are more straightforward to implement and computationally efficient. However, they exhibit instability if the time step size exceeds the stability threshold, commonly referred to as the CFL (Courant–Friedrichs–Lewy) limit. Implicit methods, conversely, typically demonstrate unconditional stability for this linear equation. Nevertheless, their computational time is substantial and increases at a rate greater than linear when one increases the number of spatial nodes, e. g., to obtain a better resolution. Explicit algorithms often prove to be more efficient, even when constrained by the small time step size necessitated by the CFL condition (Essongue et al., 2022). Still, numerous scholars prefer implicit methods due to their reliability.

It is not widely recognized that the choice between the disadvantages of the explicit and implicit methods is, in fact, unnecessary, as there exist explicit methods that are simultaneously unconditionally stable. A substantial number of these methods have been evaluated in the literature (Saleh et al., 2022; Saleh et al., 2023). Two examples which have been known for decades are the odd-even Hopscotch (Gourlay, 1970) and the Dufort–Frankel schemes. Both of them are quite accurate and reliable. Recently, new members of this family were proposed (Nagy et al., 2021; Nagy et al., 2021), which can outperform these old schemes. It was demonstrated that explicit and stable methods can efficiently solve not only the diffusion equation but nonlinear reaction–diffusion equations such as the Fisher equation (Khayrullaev et al., 2024), the Huxley equation, the Nagumo equation (Kovács et al., 2024), the sine–Gordon equation (Liang et al., 2014), the Frank–Kamenetskii equation (Harley, 2010), and the Kardar–Parisi–Zhang equation (Sayfidinov et al., 2022; Kovács et al., 2024) as well. Furthermore, they were utilized to simulate heat transfer in a photovoltaic solar panel (Nagy et al., 2024), moisture transfer in porous media (Gasparin et al., 2018), the heat storage in phase-change materials (Jalghaf et al., 2024), and the effect of thermal bridges in insulators on building walls (Omle et al., 2023).

This work attempts to compare the effectiveness of a couple of stable and explicit techniques with commonly used methods, including the Explicit and Implicit Euler, Crank–Nicolson, and fourth-order Runge–Kutta approaches. The analysis involves the plotting of the errors for each method not only as a function of the time step size but also of the execution time.

2. The discretization and the numerical algorithms

2.1. The spatial and temporal discretization

The time variable is uniformly discretized, i.e. $t \in \left[t^0, t^{\text{fin}}\right]$, and $t^n = t^0 + nh,\ n = 1, \ldots, T,\ hT = t^{\text{fin}} - t^0$. Equidistant spatial mesh $x_j = x_0 + j\Delta x,\ j = 0, \ldots, N,\ N\Delta x = L$ covers the examined interval $[x_0, x_N = x_0 + L] \subset \mathbb{R}$. The mesh ratio can be defined as $r = \frac{ah}{\Delta x^2}$, where a is defined by *Equation (1)*. We use the following analytical solution to compare of the results of the numerical time integration methods:

$$u(x,t)=\frac{e^{-\frac{x^2}{4Dt}}}{t^{11/2}}\left(-\frac{945\sqrt{D}}{16}+\frac{4725\left(\frac{x}{t^{1/2}}\right)^2}{32\sqrt{D}}-\frac{1575\left(\frac{x}{t^{1/2}}\right)^4}{32D^{\frac{3}{2}}}+\frac{315\left(\frac{x}{t^{1/2}}\right)^6}{64D^{\frac{5}{2}}}-\frac{45\left(\frac{x}{t^{1/2}}\right)^8}{256D^{\frac{7}{2}}}-\frac{\left(\frac{x}{t^{1/2}}\right)^{10}}{512D^{\frac{9}{2}}}\right) \tag{2}$$

The normalized form of this equation will be utilized to establish Dirichlet boundary conditions.

2.2. The description of the 12 used numerical methods

1. Among these methodologies, the constant neighbor (CNe) scheme represents the fundamental and longstanding approach. The application of a specific formula to each node is essential when employing of *Equation (1).*

$$u_i^{n+1}=u_i^n\cdot e^{-2r}+\frac{u_{i-1}^n+u_{i+1}^n}{2}\left(1-e^{-2r}\right) \tag{3}$$

2. The CpC methodology comprises two distinct phases. Initially, the predictor stage employs a fractional time-step of ½h duration, utilising the CNe approach. Subsequently, the corrector stage implements a full-time step CNe, where in the neighbours are re-evaluate based on the predictor values obtained in the first phase.
3. The linear neighbour (LNe) approach consists of distinct stages. Initially, the predictor phase employs a complete time step CNe. Following this, the method calculates the aggregated "slopes" of neighbours are calculate as

$$s_i=u_{i-1}^{\text{pred}}+u_{i+1}^{\text{pred}}-u_{i-1}^{\text{n}}-u_{i+1}^{\text{n}} \tag{4}$$

For the uniform scenario, the two-stage LNe method's corresponding corrector values were later furnished.

$$u_i^{n+1}=u_i^n e^{-2r}+\frac{u_{i-1}^n+u_{i+1}^n}{2}\left(1-e^{-2r}\right)+\frac{s_i}{2}\left(1-\frac{1-e^{-2r}}{2r}\right) \tag{5}$$

4. Corrector stage (5) of the LNe scheme undergoes a subsequent iteration by utilizing the previously obtained corrector values. The incorporation of an additional stage, bringing the total to three, results in the formulation of the LNe3 scheme.
5. The CLL method (Kovács et al., 2022) exhibits significant similarity to LNe3, with the distinction of employing fractional time steps of ⅔h at the first and second stages, there by achieving third-order temporal convergence.
6. The leapfrog-hopscotch (LH) methodology begins with a half-time step, which serves as a critical preparatory stage in numerical integration. This initial phase establishes a gradual framework, setting the alignment of variable to ensure temporal precision and computational robustness. Starting with this offset, the approach effectively aligns its predictive and corrective components, facilitating fluid transition and preserving the intrinsic structure of the modelled physical system. It uses a so-called θ formula, which in our case has the form:

$$u_i^{n+1} = \frac{(1-2r\theta)u_i^n + r\left(u_{i-1}^{latest} + u_{i+1}^{latest}\right)}{1+2r(1-\theta)} \tag{6}$$

Following (Nagy, Omle, et al. 2021) we implemented the zeroth stage ½h, $\theta = 0$. In the middle stages: $\theta = \frac{1}{2}$, and in the final stage: ½h, $\theta = \frac{1}{2}$.

7. The leapfrog-hopscotch-CNe (LH-CNe) approach maintains the fundamental structure of LH while consistently applying the CNe formula with an appropriately selected time-step magnitude.
8. DF is called Dufort–Frankel method as an example of explicit and unconditional stable methods (Hirsch, 1988). It is a one-stage algorithm using *Equation (7).*

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+2r} \tag{7}$$

In our comparative analysis, we utilized a selection of widely recognized numerical methods: the explicit Euler scheme (alternatively known as forward time cantered space [FTCS]), the implicit Euler approach (also termed backward time cantered space [BTCS]), the Runge–Kutta of fourth-order algorithm (RK4), and the Crank–Nicolson technique. These methods represent some of the most frequently employed finite difference method (FDM) solvers in computational mathematics.

3. Verification of the Number Methods

The MATLAB environment was used for all simulation. We computed the error of each approach for a certain time step size, which entails comparing the analytical and numerical solution node by node and choosing of difference to the highest positive of value. We use the L_2 Norm of the Error; the formula is as shown in *Equation (8).*

$$\|Err\|_2 = \sqrt{\sum_{j=1}^{N-1} \left| u_j^{analytical}\left(t^{fin}\right) - u_j^{num}\left(t^{fin}\right)\right|^2} \tag{8}$$

3.1. This case study 1 with $N = 1000+1$

In case study 1 we use *Equation (2)* we started the initial condition as u^0, the parameters are used $D = 1, t \in [0.01, 0.004], x \in [-1, 1]$ and N=1000+1, thus $\Delta x = 0.005$. During the simulation the elapsed time started from 0.0021 to 0.1662.

We plot the analytical *u* function in *Figure 1*. The concentration of u in term of x in the case of the initial function u^0, for t^{fin} exact analytical solution, CLL and LH methods. *Figure 2* illustrates the error rates of the 12 investigated methods as the time-step sizes decreased. The Crank–Nicolson (CN) method demonstrated superior accuracy with the lowest error for large and medium time-step sizes. However, it requires very long execution times, similar to the Implicit Euler method.

Figure 3 presents the L_2 error function, highlighting the Runge–Kutta 4th method's rapid error reduction to minimal values with is 10^{-4} relatively efficient execution time. The Crank–Nicolson method exhibits high accuracy and stable performance over time, despite slightly longer execution periods.

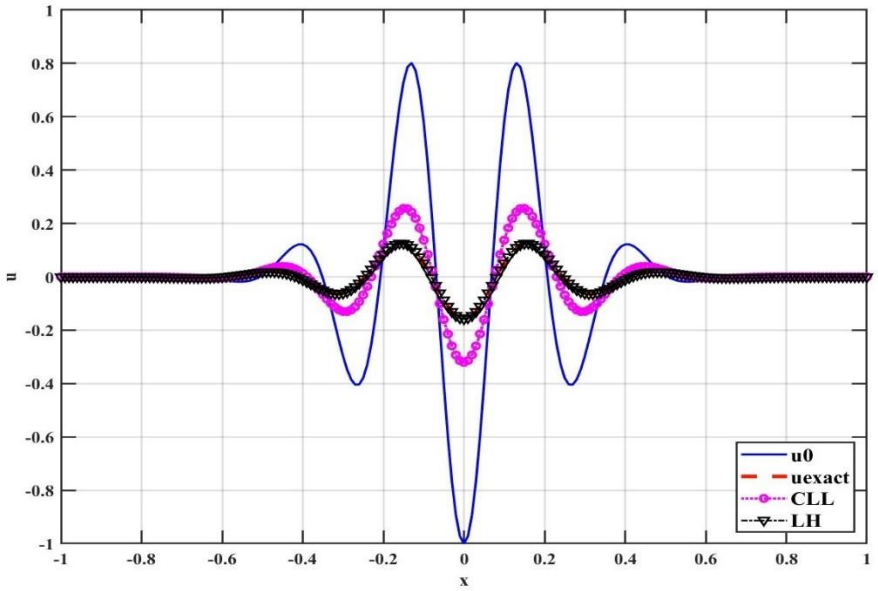

Figure 1. *The concentration of u in term of x in the case of the initial function u^0, for t^{fin} exact analytical solution, CLL and LH methods*

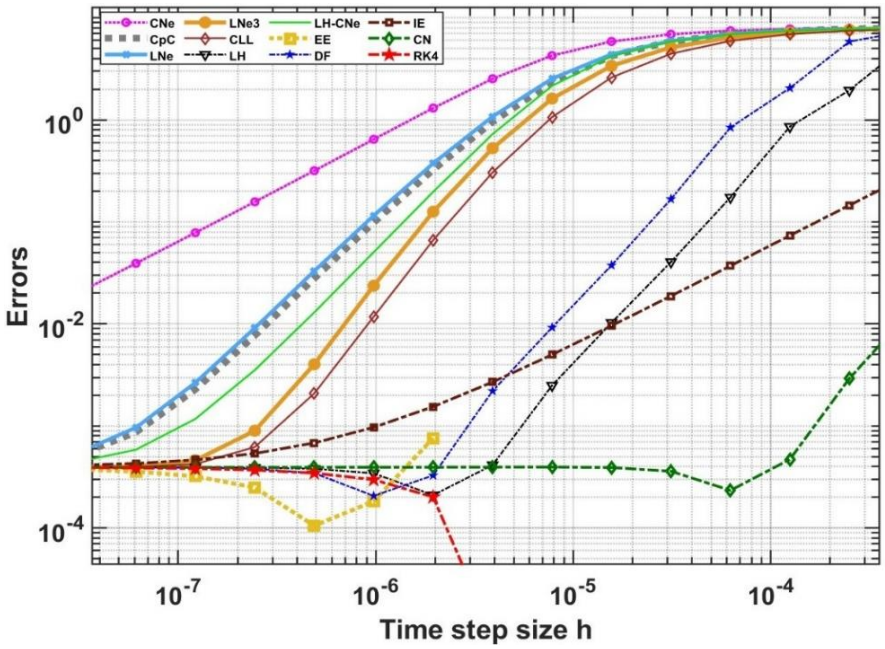

Figure 2. *Case 1 for errors in term of temporal step-size*

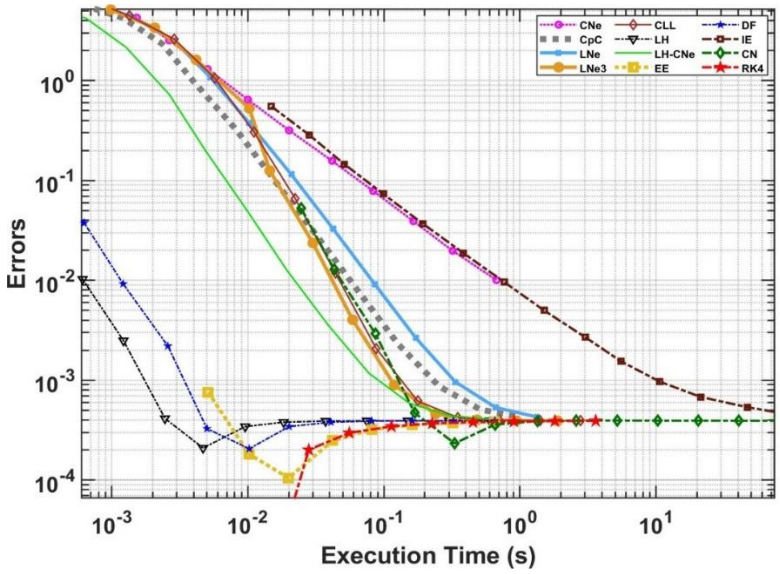

Figure 3. *The L_2 error in term of the execution time (s)*

3.2. This case study 1 with $N = 500+1$

In case study 2 we use also *Equation (2)* the parameters are used $D = 1, t\ \epsilon[0.01, 0.004], x\ \epsilon[-0.2, 0.2]$ and N=500+1, thus $\Delta x = 0.005$. During the simulation the elapsed time started from 0.0039 to 0.1660, hMax 3.20. 10^{-6} and StiffRatio 3.1210^{6}.

Figure 4 illustrates the error rates of the 12 investigated methods as time step-sizes decrease.

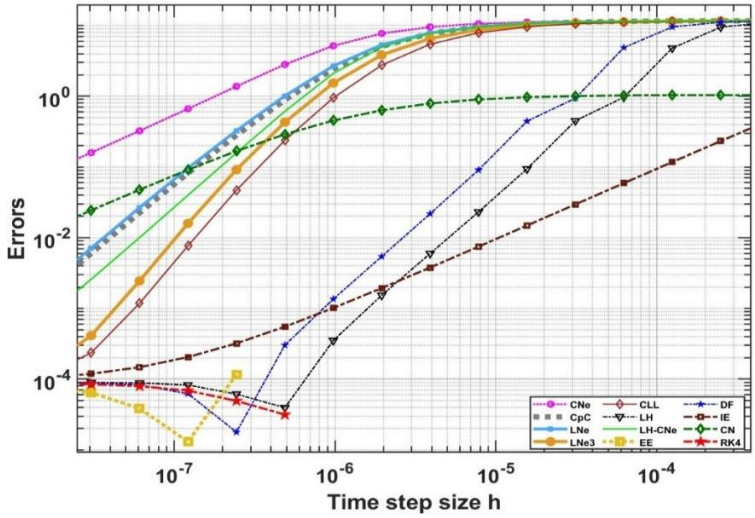

Figure 4. *Case 2 for errors in term of temporal step-size*

The Runge–Kutta 4th order method shows the smallest error for small time-step sizes, performing good. The Crank–Nicolson (CN) have very stable error levels across various at h, indicating

robustness regardless of the time step size. The CNe is a lower-order method with a little error decrease with decreasing time step sizes. In *Figure 5*, we plot the L_2 error function using *Equation (8)*, the LNe3, DF and LH are the best-performing methods in terms of execution time and error. The CN and IE perform worst.

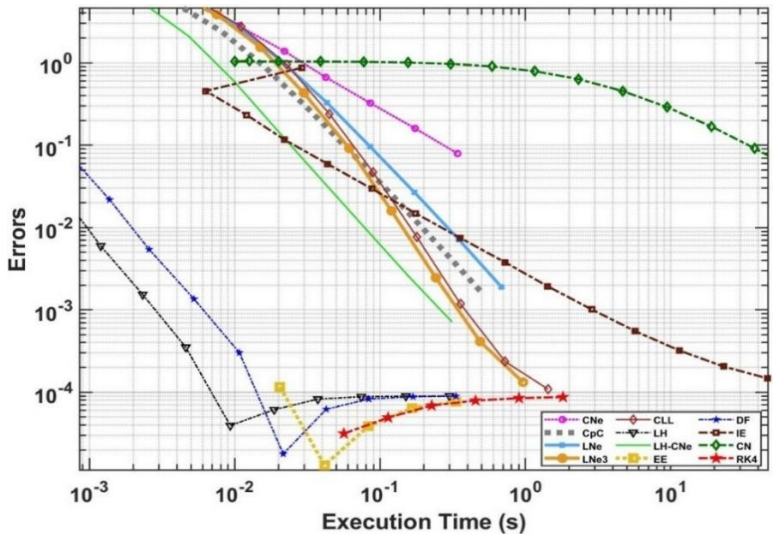

Figure 5. *The L_2 error in term of the execution time (s)*

3.3. This case study 3 with larger final time

This case study we also use the same of *Equation (2)* the parameters are used $D = 1, t\,\epsilon[0.01, 0.2], x\,\epsilon[-0.1, 1]$ and N=1000+1, thus $\Delta x = 0.005$. During the simulation the elapsed time started from 0.0021 to 0.6649, hMax $2.00.10^{-6}$ and StiffRatio $5.00.10^{5}$.

Figure 6 illustrates the error rates of the 12 investigated methods as time step sizes decrease.

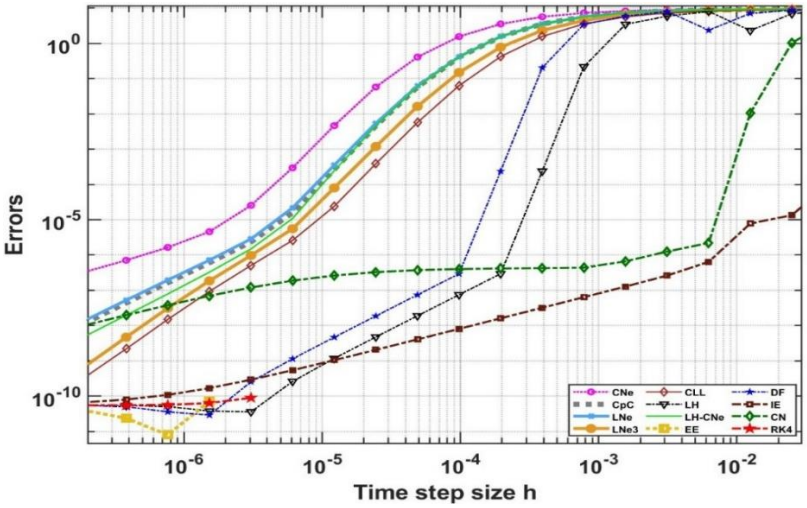

Figure 6. *Case 3 for errors in term of temporal step-size*

The Runge–Kutta 4th, DF and EE achieved the best accuracy for small time step sizes. For the increases of error denoted from the methods such as CNe, IE and CN.

In *Figure 7* we plot the L_2 error function using *Equation (8)*, The LH achieves low errors with minimal execution time, and DF also demonstrates outstanding performance with slightly larger execution times.

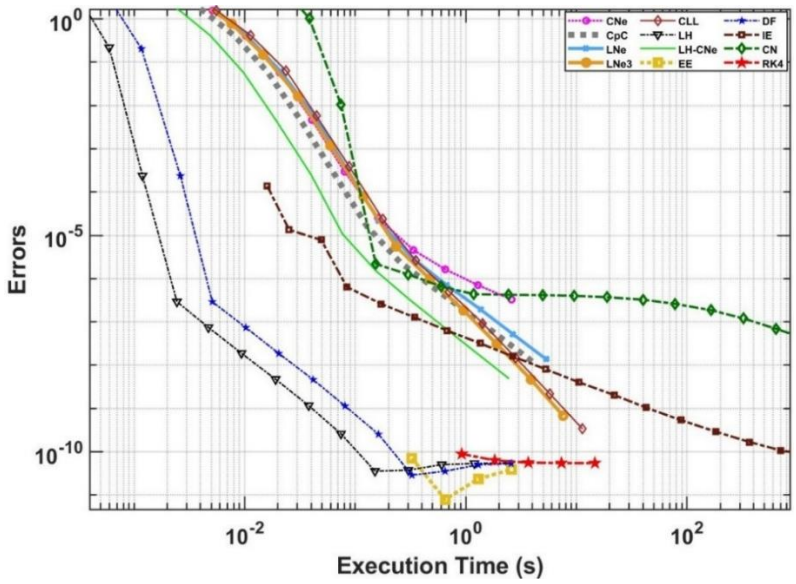

Figure 7. *The L_2 error in term of the execution time (s)*

4. Summary

Twelve numerical methods were applied to solve one-dimensional heat conduction or diffusion equations. Eight of these 12 methods are explicit schemes that are unconditionally stable for the examined equation. The remaining four methods are widespread and standard.

It is observed that the explicit and stable` methods, particularly the leapfrog-hopscotch scheme, are much more efficient than the standard methods if the execution times are considered. The explicit Euler and Runge–Kutta schemes are only conditionally stable; thus, they can provide good results only for very small-time step sizes in all three 3 case studies. In our first and second case studies, the simulated time (the time between the initial and final times) was short, whereas it was larger in the three-case study. However, the parameter *N* is larger in case studies 1 and 3, but smaller in case study 2. We observed that the relative performances of the methods were the same, with the following exceptions. In case study 1, the Crank–Nicolson (CN) method became accurate, but in case studies 2 and 3, it became less accurate. However, the LNe3 method was accurate in both cases. However, the performance of the Implicit Euler scheme was improved. This is probably due to the L-stability of this method, since in the second and third case studies, the norm of the solution in the final time was much less because of the longer decay time.

References

[1] Cengel, Y. A., and Ghajar, A. J. (2015). *Heat and mass transfer, fundamentals & application*. Fifth Edition in SI Units. McGraw-Hill, 5th Edition New York: McGraw-Hill Science/ Engineering/Math.

[2] Agbavon, K. M., Appadu, A. R., and Khumalo, M. (2019). On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term. *Advances in Difference Equations*, 146, 1–33. **https://doi.org/10.1186/s13662-019-2080-x**

[3] Askar, A. H., Nagy, Á., Barna, I. F., and Kovács, E. (2023). Analytical and numerical results for the diffusion-reaction equation when the reaction coefficient depends on simultaneously the space and time coordinates. *Computation*, 11 (7), 127. **https://doi.org/10.3390/computation11070127**

[4] Barna, I. F., and Mátyás, L. (2022). Advanced analytic self-similar solutions of regular and irregular diffusion equations. *Mathematics,* 10 (18), 3281. **https://doi.org/10.3390/math10183281**

[5] Barna, I. F., and Mátyás, L. (2024). Diffusion cascades and mutually coupled diffusion processes. *Mathematics*, 12 (20), 3298. **https://doi.org/10.3390/math12203298**

[6] Beuken, L., Cheffert, O., Tutueva, A., Butusov, D., and Legat, V. (2022). Numerical stability and performance of semi-explicit and semi-implicit predictor–corrector methods. *Mathematics*, 10 (12). **https://doi.org/10.3390/math10122015**

[7] Essongue, S., Ledoux, Y., and Ballu, A. (2022). Speeding up mesoscale thermal simulations of powder bed additive manufacturing thanks to the forward Euler Time-Integration Scheme: A critical assessment. *Finite Elements in Analysis and Design*, 211, 103825. **https://doi.org/10.1016/j.finel.2022.103825**

[8] Fedoseev, P., Pesterev, D., Karimov, A., and Butusov, D. (2022). New step size control algorithm for semi-implicit composition ODE solvers. *Algorithms*, 15 (8), 275. **https://doi.org/10.3390/a1580275**

[9] Gasparin, S., Berger, J., Dutykh, D., and Mendes, N. (2018). Stable explicit schemes for simulation of nonlinear moisture transfer in porous materials. *Journal of Building Performance Simulation*, 11 (2), 129–144. **https://doi.org/10.1080/19401493.2017.1298669**

[10] Gourlay, A. R. (1970). Hopscotch: A fast second-order partial differential equation solver. *IMA Journal of Applied Mathematics*, 6 (4), 375–390. **https://doi.org/10.1093/imamat/6.4.375**

[11] Harley, C. (2010). Hopscotch Method: The numerical solution of the Frank-Kamenetskii partial differential equation. *Applied Mathematics and Computation*, 217, 4065–4075. **https://doi.org/10.1016/j.amc.2010.10.020**

[12] Jalghaf, H. K., and Kovács, E. (2024). Simulation of phase change materials in building walls using effective heat capacity model by recent numerical methods. *Journal of Energy Storage*, 83, 110669. **https://doi.org/10.1016/j.est.2024.110669**

[13] Khayrullaev, H., Omle, I., and Kovács, E. (2024). Systematic investigation of the explicit, dynamically consistent methods for Fisher's equation. *Computation*, 12 (3), 49. **https://doi.org/10.3390/computation12030049**

[14] Köroğlu, C., and Aydın, A. (2021). Exact and nonstandard finite difference schemes for the Burgers equation B(2,2). *Turkish Journal of Mathematics*, 45 (2), 647–660. **https://doi.org/10.3906/mat-2009-51**

[15] Kovács, E., Barna, I. F., Bognár, G., Mátyás, L., and Hriczó, K. (2024). Analytical and numerical study of diffusion propelled surface growth phenomena. *Partial Differential Equations in Applied Mathematics*, 11, 100798. **https://doi.org/10.1016/j.padiff.2024.100798**

[16] Kovács, E., Majár, J., and Saleh, M. (2024). Unconditionally positive, explicit, fourth order method for the diffusion- and Nagumo-type diffusion–reaction equations. *Journal of Scientific Computing*, 98 (2), 1–39. **https://doi.org/10.1007/s10915-023-02426-9**

[17] Kovács, E., Nagy, Á., and Saleh, M. (2022). A new stable, explicit, third-order method for diffusion-type problems. *Advanced Theory and Simulations*, 5 (6), 2100600. **https://doi.org/10.1002/adts.202100600**

[18] Liang, Z., Yan, Y., and Cai, G. (2014). A Dufort-Frankel difference scheme for two-dimensional Sine-Gordon equation. *Discrete Dynamics in Nature and Society*, 2014 (1), 784387. **https://doi.org/10.1155/2014/784387**

[19] Mátyás, L., and Barna, I. F. (2022). General self-similar solutions of diffusion equation and related constructions. *Romanian Journal of Physics*, 67, 101.

[20] Mátyás, L., and Barna, I. F. (2023). Even and odd self-similar solutions of the diffusion equation for infinite horizon. *Universe*, 9 (6), 264. **https://doi.org/10.3390/universe9060264**

[21] Nagy, Á., Bodnár, I., and Kovács, E. (2024). Simulation of the thermal behavior of a photovoltaic solar panel using recent explicit numerical methods. *Advanced Theory and Simulations*, 7 (7), 2400089. **https://doi.org/10.1002/adts.202400089**

[22] Nagy, Á., Omle, I., Kareem, H., Kovács, E., Barna, I. F., and Bognár, G. (2021). Stable, explicit, Leapfrog-Hopscotch algorithms for the diffusion equation. *Computation*, 9 (8), 92. **https://doi.org/10.3390/computation9080092**

[23] Nagy, Á., Saleh, M., Omle, I., Kareem, H., and Kovács, E. (2021). New stable, explicit, Shifted-Hopscotch algorithms for the heat equation. *Math. Comput. Appl*., 26 (61). https://www.mdpi.com/2297-8747/26/3/61/htm **https://doi.org/10.3390/mca26030061**

[24] Omle, I., Askar, A. H., Kovács, E., and Bolló, B. (2023). Comparison of the performance of new and traditional numerical methods for long-term simulations of heat transfer in walls with thermal bridges. *Energies*, 16 (12), 4604. **https://doi.org/10.3390/en16124604**

[25] Saleh, M., Kovács, E., and Barna, I. F. (2023). Analytical and numerical results for the transient diffusion equation with diffusion coefficient depending on both space and time. *Algorithms*, 16 (4), 184. **https://doi.org/10.3390/a16040184**

[26] Saleh, M., Kovács, E., Barna, I. F., and Mátyás, L. (2022). New analytical results and comparison of 14 numerical schemes for the diffusion equation with space-dependent diffusion coefficient. *Mathematics*, 10 (15), 2813. **https://doi.org/10.3390/math10152813**

[27] Sayfidinov, O., Bognár, G., and Kovács, E. (2022). Solution of the 1D KPZ equation by explicit methods. *Symmetry*, 14 (4), 699. **https://doi.org/10.3390/sym14040699**

[28] Hirsch, C. (1988). *Numerical computation of internal and external flows.* Volume 1: Fundamentals of Numerical Discretization; Wiley: Hoboken, NJ, USA.