

Article

# Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates

Ali Habeeb Askar <sup>1,2,3</sup> , Ádám Nagy <sup>1</sup>, Imre Ferenc Barna <sup>4</sup>  and Endre Kovács <sup>1,\*</sup> 

<sup>1</sup> Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary; 20156@uotechnology.edu.iq (A.H.A.); adam.nagy0310@gmail.com (Á.N.)

<sup>2</sup> Department of Fluid and Heat Engineering, University of Miskolc, 3515 Miskolc, Hungary

<sup>3</sup> Mechanical Engineering Department, University of Technology—Iraq, Baghdad 10066, Iraq

<sup>4</sup> Wigner Research Center for Physics, 1051 Budapest, Hungary; barna.imre@wigner.hu

\* Correspondence: kendre01@gmail.com or endre.kovacs@uni-miskolc.hu

**Abstract:** We utilize the travelling-wave Ansatz to obtain novel analytical solutions to the linear diffusion–reaction equation. The reaction term is a function of time and space simultaneously, firstly in a Lorentzian form and secondly in a cosine travelling-wave form. The new solutions contain the Heun functions in the first case and the Mathieu functions for the second case, and therefore are highly nontrivial. We use these solutions to test some non-conventional explicit and stable numerical methods against the standard explicit and implicit methods, where in the latter case the algebraic equation system is solved by the preconditioned conjugate gradient and the GMRES solvers. After this verification, we also calculate the transient temperature of a 2D surface subjected to the cooling effect of the wind, which is a function of space and time again. We obtain that the explicit stable methods can reach the accuracy of the implicit solvers in orders of magnitude shorter time.

**Keywords:** diffusion–reaction equations; heat conduction; convection; analytical solution; unconditionally stable numerical methods; explicit time-integration



**Citation:** Askar, A.H.; Nagy, Á.; Barna, I.F.; Kovács, E. Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates. *Computation* **2023**, *11*, 127. <https://doi.org/10.3390/computation11070127>

Academic Editor:  
Sivasankaran Sivanandam

Received: 31 May 2023  
Revised: 20 June 2023  
Accepted: 27 June 2023  
Published: 29 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

It is well known that linear transient diffusion and heat conduction are analogous phenomena: mathematically, both of them are described by the simplest parabolic PDE (partial differential equation). Diffusion means the distribution of the particles is changing [1], while in heat conduction, energy is transported via interacting particles of the material [2].

Considering extra source terms in the regular diffusion equation, we arrive to a more general type of the reaction–diffusion equations. Numerous general mathematical aspects of such systems can be found in various monographs such as [3–5]. For example, the role of the reaction–diffusion equations in plasma physics is analyzed in [6].

According to Newton’s law of cooling, heat loss by convection is proportional to the temperature if it is measured compared to the actual ambient temperature. So, for  $x, t \in \mathbb{R}$ , the simplest PDE in one space dimension which can describe heat conduction and convection is:

$$\frac{\partial u(x, t)}{\partial t} = D \frac{\partial^2 u(x, t)}{\partial x^2} - K u(x, t), \quad u(x, t = 0) = u^0(x) \quad (1)$$

where  $u : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ ;  $(x, t) \mapsto u(x, t)$  is the unknown function (temperature in the case of heat conduction and concentration in the case of particle diffusion), and  $D \in \mathbb{R}$  is the constant diffusion coefficient. In the case of heat conduction,  $D = k/(c\rho)$  is the thermal diffusivity, while  $c$ ,  $\rho$ , and  $k$  are the specific heat, the density, and the heat conductivity of

the material, respectively. The term  $-Ku$  is a reaction term, so (1) is a regular diffusion-reaction PDE. Typically,  $u^0$  is the given initial function, while the boundary conditions will be discussed in the concrete analytical and numerical examples.

The convection cooling process has wide applications in various engineering fields, such as steel production and the manufacture of air-driven turbine blades [7]. The cooling medium is used in various industrial manufacturing procedures to maintain a suitable temperature throughout the manufacturing process [8]. In high-load electronics, cooling systems are also routinely applied. The primary goal of these is to maintain the improved operational speed of the machine and mitigate possible damage. In the most prevalent case of computers, when the temperature of the CPU (central processing unit) exceeds the safe operating limits [9], a mechanism will be activated to cool it, usually by a heat pipe [10]. Utilizing heat-pipe-cooled micro-reactors presents a promising option for decentralized, remote electricity supply and space power technology due to their exceptional inherent safety and minimal maintenance requirements [11].

It is well known that for the diffusion equation, some analytical solutions exist, which can be found in basic textbooks [1,2]. These solutions have crucial relevance to understanding the diffusion process itself. As a second point, these solutions help to test the properties and performance of old and new numerical methods.

The regular and irregular diffusion equations were investigated in detail by one of the authors in recent years [12]. For the regular diffusion equation with the travelling wave, self-similar, travelling profile, or with some generalized self-similar trial functions, numerous new types of analytic solutions were found. These solutions have a much more complicated structure than the well-known Gaussian (plus an error function in the most general case). These new solutions contain Kummer's or Whittaker functions with quadratic arguments of a new free self-similar parameter which results in larger variety of the solutions. These solutions can have a drastically different rate of decay than the fundamental Gaussians. We derived solutions which have some oscillatory behavior with a rapid power law decay both in space and time.

In this work, we convey new results for a modified version of PDE (1), which will have a reaction coefficient that is not a constant, but depends both on time and space, which is the most important novelty of this work.

Equation (1) and other equations of a similar nature are sometimes solved by analytical and semi-analytical methods such as residual power series method [13] and optimal variational iteration method [14,15]. However, the most common way of solution is numerical integration [16], and numerous techniques have been put forth for this. Most of them fall into one of two categories: explicit schemes or implicit schemes. In comparison to one another, both types have a significant benefit and a disadvantage. The widely used explicit approaches, such as those based on explicit Runge–Kutta time integration, execute a time step in a reasonably short amount of time and are easily parallelizable. They have a restricted stability region [17] though, which means that if the time step size goes over the so-called CFL (Courant–Friedrichs–Lewy) limit, the solution is anticipated to grow indefinitely.

Since the implicit methods' stability is inherently greater, many scientists consider them to be superior and frequently employ them for these and other related equations. For example, Mbroh and Munyakazi [18] solved one and two space-dimensional singularly perturbed reaction–diffusion problems by discretization of the space variable using a fitted operator finite difference method and then using the Crank–Nicolson scheme with Richardson extrapolation. Ndou et al. [19] combined the proper orthogonal decomposition method with the unconditionally positive finite difference (UPFD) algorithms to effectively solve linear and nonlinear advection–diffusion–reaction equations. Kumar et al. [20] employed the open-source package FEniCSx which uses a preconditioned Krylov subspace method to simulate heat transfer in a three-dimensional bracket. Jiang and Zhang [21] solved semilinear and fully nonlinear advection–diffusion–reaction equations by Krylov implicit integration factor methods, but the linear part of the system was exactly integrated.

Heidari et al. [22] utilized a Legendre–Gauss–Lobatto spectral collocation method to solve a general diffusion–reaction equation. Kolev et al. [23] proposed a numerical method that preserves the nonnegative property of the variable of the solutions of a diffusion–reaction equation–system. Their method also applies implicit time discretization. In the case of the implicit methods, it is necessary to solve a set of algebraic equations in each time step, which is a difficult operation to parallelize. Large amounts of RAM are involved, and the calculations can be extremely slow. This occurs frequently in multidimensional space when the matrix is an enormous and non-tridiagonal but sparse matrix, but its inverse is a dense matrix. So, in practice, not direct but iterative procedures are used for solving the equation system, about which we will write more in the appropriate section later.

Semi-explicit or semi-implicit methods, which can be considered as combinations of the explicit and the implicit approach, can be quite effective as well. Beuken et al. [24] constructed versions of these methods based on the backward differentiation formula and Adams–Bashforth schemes for ODE systems, which can be an order of magnitude faster than some traditional methods. Fedoseev et al. [25] improved the performance of semi-implicit composition integration methods by variable step size control algorithms. The paper of Ji and Xing [26] proposes a set of time-marching methods which uses the generalized Padé approximation, the Gauss–Legendre quadrature, and explicit Runge–Kutta schemes to solve systems of ODEs. All of these can be applied to PDEs such as (1) after semi-discretization. Settanni and Sgura [27] solved a Lotka–Volterra type system of 2D reaction–diffusion equations by treating the diffusion part implicitly and the reaction part by some well-known explicit schemes as well as the alternating direction implicit (ADI) methods. Yadav et al. [28] recently developed new IMEX RK-type time-integrating methods for solving stiff convection–diffusion–reaction systems, where the diffusion and the reaction terms are treated implicitly. However, we think none of these methods overcome the problems of the explicit and implicit methods mentioned above. Recently, Essoungue et al. demonstrated that explicit methods can be more than competitive compared to implicit ones even if the usage of small time step sizes are required [29].

Based on these facts, we have to agree with those scholars who invest their energy to create and examine explicit and unconditionally stable algorithms. For example, Chen–Charpentier and Kojouharov [30] proposed the so-called UPFD (unconditionally positive finite difference) scheme for the linear diffusion–advection–reaction equation. Later, Appadu [31], and recently, Savovic et al. [32] continue the examination of this algorithm. Pourghanbar et al. [33] used the alternating direction explicit (ADE) method of Saulyev type to solve a nonlinear PDE faster than the main implicit methods. Al-Bayati et al. [34] compared the performance of the ADE, the ADI, and the odd–even hopscotch methods to solve a system of diffusion–reaction equations. They found that the implicit method is more accurate but slower than the explicit methods.

Our research group has recently constructed several explicit methods for equations similar to (1). In our original publications, it was analytically shown that they are unconditionally stable for the linear diffusion or, in some cases, diffusion–reaction equation. The methods have been extensively tested in the case of the diffusion equation under circumstances where the diffusion coefficient is constant [35], depends on time [36], depends on space [37], or on both variables simultaneously [38]. In all cases, the novel algorithms proved to be very competitive when they are compared to the traditional methods, which were mostly explicit ones, and in a few cases, implicit “ode” solvers of MATLAB. One of the novelties from the numerical point of view is that now these novel methods are tested against the standard implicit method when the obtained algebraic system is solved by preconditioned conjugate gradient (PCG) as well as generalized minimal residual (GMRES) methods. The motivation of the paper from the numerical point of view is to compare the performance of these solvers and show that the traditional ones are far from being optimal to solve these problems, and the explicit and stable schemes can easily outperform them.

The structure of this paper is the following. We present the analytical solution of the equations for the different reaction function in Section 2. The discretization procedure

and the used numerical algorithms are described in Section 3. The analytical solutions are reproduced by the methods and the numerical errors are examined in Section 4. Then, in Section 5, we take an application of heat transfer on a two-dimensional surface of two adjacent materials placed vertically, and the upper part of the body is exposed to constant heat convection and the lower part to variable heat convection with time. A conclusion section summarizes our findings at the end of the paper.

## 2. Analytical Solution

Let us study the one-dimensional diffusion–reaction equation given by:

$$\frac{\partial u(x, t)}{\partial t} = D \frac{\partial^2 u(x, t)}{\partial x^2} + F(x, t) u(x, t) \quad (2)$$

where  $F$  is the known coefficient of the reaction term, which is usually non-positive. To derive physically relevant solutions, we apply the travelling wave Ansatz:

$$u(x, t) = f(x - ct) = f(\eta) \quad (3)$$

where  $\eta$  is the reduced variable and  $f(\eta)$  is called the shape function. In this sense, we think that we can find solutions which have oscillatory behavior instead of the usual self-similar type diffusive and dispersive Gaussian-like solutions, which we derived in our former studies [12]. The first and second continuous derivatives of  $f(\eta)$  are supposed to exist with respect to  $\eta$  and denoted by primes. Of course, the form of  $F(x, t)$  has to be specified first. It is clear at the first moment that  $F(x, t)$  could be almost any function from the mathematical point of view. Considering physical arguments (e.g.,  $F(x, t)$  is finite, or even goes to zero at infinite time and space coordinates, or  $F(x, t)$  has no singularities at any points), the possible forms of the  $F(x, t)$  are radically reduced. It is clear that if  $F(x, t)$  would go to infinity at large times or spatial coordinates that would mean an infinite driving term in the equation, which is far from physical reality. We tried numerous functions, e.g., Gaussian, exponential, and power law terms but found solutions only for the two terms shown in the following.

It is always interesting to study the influence of a well-localized impulse-like source; therefore, we tried the Lorentzian form and luckily found highly non-trivial solutions. As a second interesting system, the effect of periodic driving was investigated. It looked logical that for a traveling wave, Ansatz (capable of describing periodic or quasi-periodic solutions) applied to periodic driving gives reasonable periodic or quasi-periodic finite solutions. Our guess was confirmed, which is analyzed in the following.

### 2.1. Lorentzian Coefficient of the Reaction Term

First, we consider the following form for the  $F$  function:

$$F(x, t) = \frac{a}{1 + (x - ct)^2} \quad (4)$$

where  $a \in \mathbb{R}$  is the strength parameter (amplitude) of the applied pulse. By utilizing the Ansatz (3), the reduced ODE is immediately obtained:

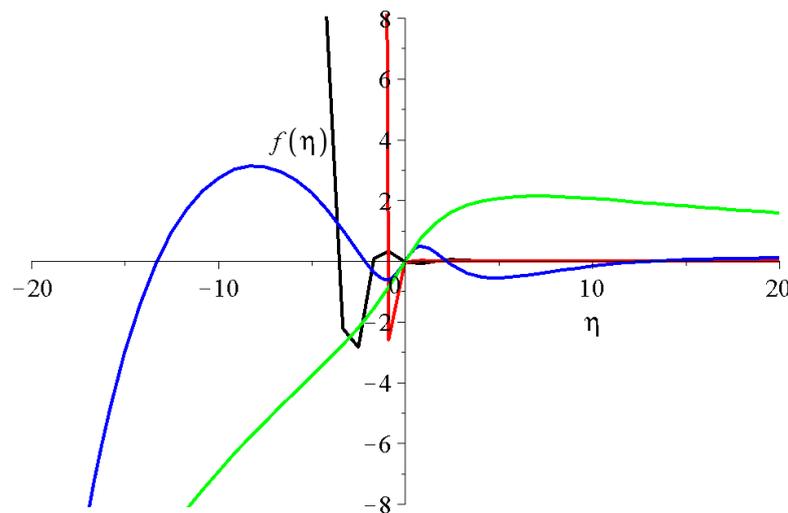
$$-c f'(\eta) = D f''(\eta) + \frac{a}{1 + \eta^2} f(\eta)$$

The solution of this ODE is given by the MAPLE 12 software:

$$f(\eta) = e^{-\frac{c\eta}{2D}} (1 + \eta^2) \left[ c_1 \text{HeunC} \left( 0, -\frac{1}{2}, 1, \frac{c^2}{16D^2}, \Psi, -\eta^2 \right) + c_2 \eta \cdot \text{HeunC} \left( 0, \frac{1}{2}, 1, \frac{c^2}{16D^2}, \Psi, -\eta^2 \right) \right], \quad (5)$$

where HeunC is the Heun function [39], and we introduced the abbreviation  $\Psi = \frac{4aD+8D^2-c^2}{16D^2}$ . More general technical details about the Heun’s function, together with other special functions, can be found in [39,40]. For additional specific information about Heun’s differential equation and on various Heun’s function, please consult [41,42]. It is almost impossible to give a complete parameter study of Equation (5) for the entire range of  $(c_1, c_2, c, a, D)$ . We just restrict us for the  $c_1 = 0, c_2 = 1, c > 0, a > 0, D > 0$  subspace. Considering negative propagation velocity ( $c < 0$ ) just means reflecting the solution to the  $y$ -axis. For negative source strength ( $a < 0$ ), we obtain non-decaying solutions at large  $\eta$ , which means either asymptotic saturation or divergence, which we consider unphysical. Diffusion processes where the concentration or the number of particles explode violate energy and matter conservation laws and it is rather counterintuitive that we obtain it typically for negative and not for positive values of  $a$ . So, although the solutions are mathematically valid for arbitrary values of the  $x$  and  $t$  variables, the solution on the whole real axis cannot always describe a real physical process. Nevertheless, the obtained functions—in principle—can describe real processes in any finite interval if the system boundaries are not closed, and energy or particles can enter from the surrounding space, which is reflected mathematically by the boundary conditions.

Numerous shape functions are presented in Figure 1 for different parameter sets. Smaller wave velocity shifts the singularity to the left and causes more oscillations (see the difference between the black and red curves in Figure 1). Increasing the diffusion coefficient  $D$  against the source strengths parameter  $a$ , but still being ( $a > D$ ) smears out the decay range of the diffusion (compare the blue curve to the black one). Finally, considering that the diffusion parameter is larger than the source strength ( $a < D$ ), the oscillations disappear, maintaining a very flat local maximum and a very slow decay (compare the blue curve to the green one).

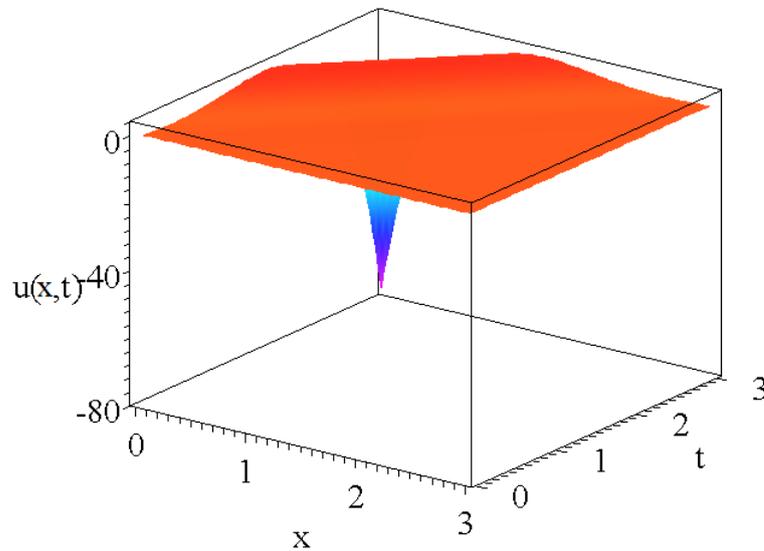


**Figure 1.** The shape function of Equation (5) for the different parameter sets  $(c_1, c_2, c, a, D)$ ; the black, red, blue, and green lines are for  $(0, 1, 0.5, 7.4, 0.3)$ ,  $(0, 1, 2, 7.4, 0.3)$ ,  $(0, 1, 0.5, 7.4, 1.8)$ , and  $(0, 1, 0.5, 2.4, 4.1)$ , respectively.

The  $x$  and  $t$  dependent solution of the reaction–diffusion PDE has the form of:

$$u(x, t) = e^{-\frac{c(x-ct)}{2D}} (1 + (x - ct)^2) \left[ \begin{array}{l} c_1 \text{HeunC} \left( 0, -\frac{1}{2}, 1, \frac{c^2}{16D^2}, \Psi, -(x - ct)^2 \right) \\ + c_2 (x - ct) \text{HeunC} \left( 0, \frac{1}{2}, 1, \frac{c^2}{16D^2}, \Psi, -(x - ct)^2 \right) \end{array} \right] \quad (6)$$

We note that since our analytical solution is valid on the whole real axis, boundary conditions need to be specified only when the analytical solution is going to be reproduced by numerical methods. The time development of a solution function  $u(x, t)$  of Equation (6) is presented in Figure 2 for the same parameter set as the shape function is shown. We checked and found that all the solutions of  $u(x, t)$  for all the other three parameter sets given in Figure 1 have qualitatively the same shape.



**Figure 2.** The solution function of Equation (6) for the parameters of  $D = 0.3$ ;  $a = 7.4$ ;  $c = 1$ ; ( $\Psi = 5.572222$ );  $c_1 = 0$ ;  $c_2 = 1$ , respectively.

2.2. Cosine Function as the Coefficient of the Reaction Term

For second case of investigation, we consider the following form for the  $F$  function:

$$F(x, t) = a \cos(x - ct), \quad a \in \mathbb{R} \tag{7}$$

As we mentioned above, periodic driving has always been a fundamental interest. By applying the travelling wave Ansatz (3) to transform the PDE (2), the following reduced ODE is immediately obtained.

$$-c f'(\eta) = D f''(\eta) + a \cos(\eta) \cdot f(\eta)$$

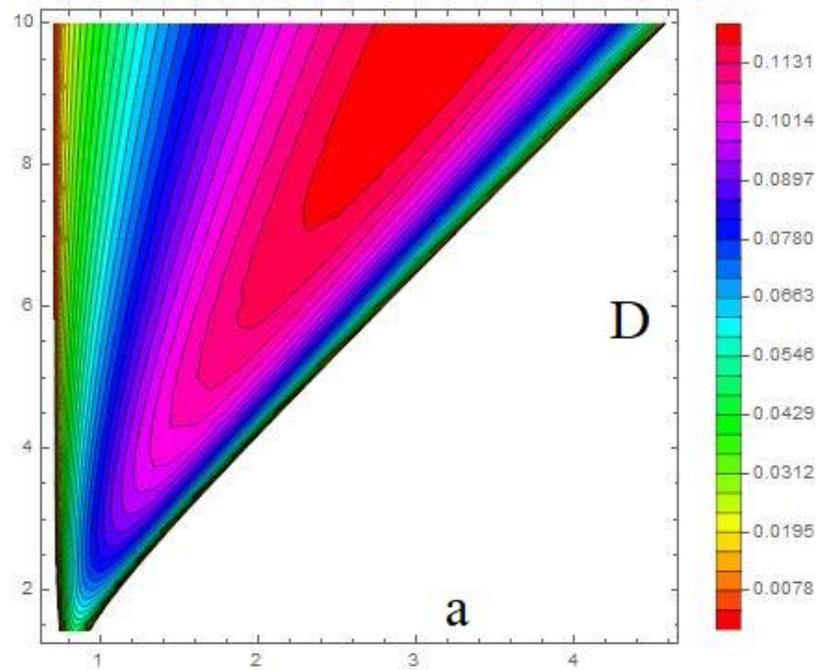
The solution of this ODE is also given by the MAPLE 12 software:

$$f(\eta) = e^{\frac{-c\eta}{2D}} \left[ c_1 \text{MathieuC} \left( \frac{-c^2}{D^2}, -\frac{2a}{D}, \frac{\eta}{2} \right) + c_2 \text{MathieuS} \left( \frac{-c^2}{D^2}, -\frac{2a}{D}, \frac{\eta}{2} \right) \right], \tag{8}$$

where MathieuC and MathieuS are the Mathieu functions. General information about the mathematical properties of these functions can be found in [39]. More detailed analysis, however, is given in the monographs of [43–46].

To investigate the parameter dependence of the Mathieu functions, we have to follow a different strategy. It is well known from the theory of Mathieu functions that depending on the ratio of the first and second parameters, different regimes exist for the solutions. In the unstable regime, the Mathieu functions are not normalizable; therefore, they can go to infinity. In the stable regime, the function remains finite and can be normalized, which means that the integral of the function has a finite value.

Figure 3 presents the stability diagram of the Mathieu S function for  $\eta = 1$  and for  $c = 1$ . The presented contour plot was evaluated with Wolfram Mathematica 12 software package. We applied the  $0 \leq a \leq 5$  and  $0 \leq D \leq 10$  parameter regime to present the stability regime of the solution which contains the Mathieu S function. In the colored regime, the function is normalizable and has no singularities. In the white regime, the function is singular. We mention that the stability chart of the Mathieu functions (without the Gaussian function) has a completely different shape: some parts are similar to the tulip flower; for figure see [39] (28.17.1.).

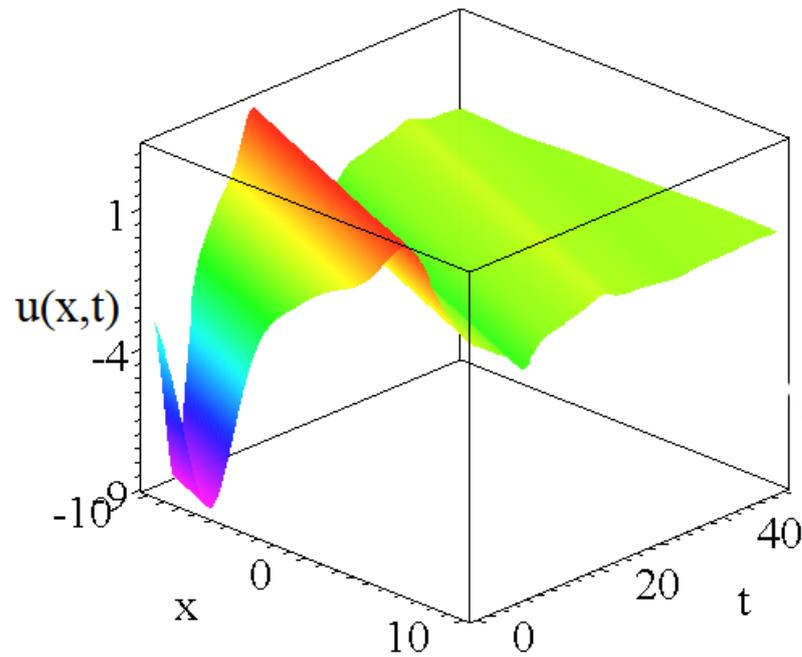


**Figure 3.** The stability diagram of the shape function  $f(\eta)$  in the form of Equation (8) for  $c = 1$  for the Mathieu S part ( $c_1 = 0, c_2 = 1$ ). The horizontal axis shows the range of the parameter  $a$ , which is responsible for the strength of the source term, while the vertical axis shows the range of the  $D$ , which is the diffusion coefficient. In the white regions, the function the integral of the function is infinite and therefore not stable. In the colored region, the integral is finite and has the value as the color shows.

The solution of the PDE is:

$$u(x, t) = e^{-\frac{c(x-ct)}{2D}} \left[ c_1 \text{MathieuC} \left( \frac{-c^2}{D^2}, -\frac{2a}{D}, \frac{x-ct}{2} \right) + c_2 \text{MathieuS} \left( \frac{-c^2}{D^2}, -\frac{2a}{D}, \frac{x-ct}{2} \right) \right] \tag{9}$$

In Figure 4, we present the 3D plot of this  $u(x, t)$  function for the parameter set for the parameters of  $c_1 = 0, c_2 = 1, c = 1, D = 6, a = 2$ . The Mathieu C functions look globally very similar. We present the function for positive spatial and temporal coordinates. The asymptotic decay for zero at large arguments is clear to see. A similar solution will be reproduced numerically in the latter sections. The global properties of the solution remain similar to the shape function.



**Figure 4.** The solution  $u(x, t)$  of Equation (2) with form of Equation (9) for  $c = 1, a = 2, D = 6$ . Only the Mathieu S function is presented with  $c_2 = 1$ ; the Mathieu C function looks similar.

### 3. Numerical Solution Preliminaries for Materially Homogeneous and Inhomogeneous Systems

#### 3.1. Discretization and Boundary Conditions for a 1D System

The time variable is uniformly discretized, which means  $t \in [t^0, t^{\text{fin}}]$ , and

$$t^n = t^0 + nh, \quad n = 1, \dots, T, \quad hT = t^{\text{fin}} - t^0$$

An equidistant spatial grid is constructed on the interval  $x \in [x_0, x_N = x_0 + L] \subset \mathbb{R}$ :

$$x_j = x_0 + j\Delta x, \quad j = 0, \dots, N, \quad N\Delta x = L$$

The concrete space and time domain will be given when the case studies are presented.

After the application of the most frequent central difference formula for the space derivative in Equation (2), we obtain an ODE system for the node variables:

$$\frac{du_i}{dt} = D \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + F_i^n u_i \tag{10}$$

where  $F_i^n = F(x_i, t^n)$ . This can obviously be written into a matrix form:

$$\frac{d\vec{u}}{dt} = M\vec{u} \tag{11}$$

where the matrix  $M$  of the system is  $N \times N$  dimensional and depends on time through its diagonal elements.

In the next section, where the above given analytical solutions are reproduced, prescribed Dirichlet boundary conditions (calculated using the analytical solutions) are taken into account everywhere.

### 3.2. The Used Numerical Methods

It is widely known that for the general first order ODE  $y' = f(t, y)$ , the theta method has the formula:

$$y^{n+1} = y^n + \Delta t \left[ \theta f(t^n, y^n) + (1 - \theta) f(t^{n+1}, y^{n+1}) \right] \tag{12}$$

where  $\theta \in [0, 1]$ . In the case of Equation (10), it yields:

$$u_i^{n+1} = u_i^n + \theta \left[ r(u_{i\pm 1}^n - 2u_i^n) + hF_i^n u_i^n \right] + (1 - \theta) \left[ r(u_{i\pm 1}^{n+1} - 2u_i^{n+1}) + hF_i^{n+1} u_i^{n+1} \right] \tag{13}$$

where  $u_{i\pm 1}^n = u_{i+1}^n + u_{i-1}^n$  and  $r = \frac{Dh}{\Delta x^2}$  is the well-known mesh ratio. This is clearly an implicit method if  $\theta < 1$ . By the so-called pseudo-implicit trick, we make it explicit: the neighbors  $u_{i\pm 1}^n$  in the second term at the right-hand side of (13) is taken into account at the old ( $n$ -th) time level. Using this, we obtain:

$$u_i^{n+1} = u_i^n + ru_{i\pm 1}^n + \theta[-2r + hF_i^n]u_i^n + (1 - \theta)[-2r + hF_i^{n+1}]u_i^{n+1} \tag{14}$$

which may easily be rearranged as

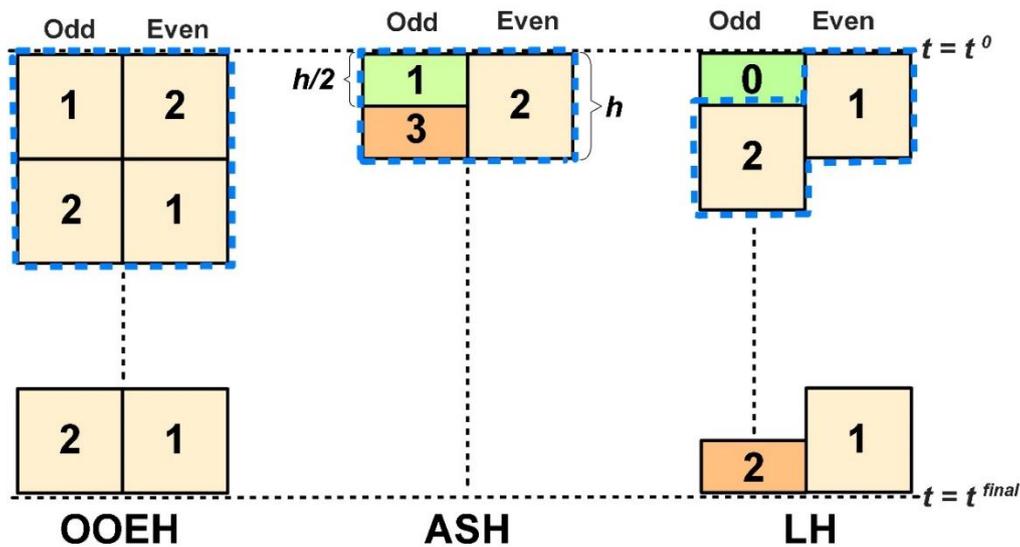
$$u_i^{n+1} = \frac{[1 - \theta(2r - hF_i^n)]u_i^n + ru_{i\pm 1}^n}{1 + (1 - \theta)[2r - hF_i^{n+1}]} \tag{15}$$

Note that the  $u_i$  variable in both the conduction and convection term is considered at the old and the new time levels as well.

1. The simplest scheme used here is the so-called unconditionally positive finite difference (UPFD) formula, which is introduced for the linear diffusion–advection–reaction equation by Chen–Charpentier and Kojouharov [30]. Now the advection term is missing; thus, the new values of the cell variables can be obtained from Equation (15) by the  $\theta = 0$  substitution.
2. The next scheme we consider is the so-called pseudo-implicit (PI) method. Its two stages apply a modified version of formula (15) with different parameters. The first stage takes a half time step with  $\theta_1 = 0$ ; then, the second stage corresponds to a full time step with  $\theta_2 = \frac{1}{2}$  which uses the results of the first stage as follows:

$$\text{Stage1 : } u_i^{\text{pred}} = \frac{u_i^n + ru_{i\pm 1}^n / 2}{1 + [r - hF_i^{n+1} / 2]}, \text{ Stage2 : } u_i^{n+1} = \frac{(1 - r)u_i^n + ru_{i\pm 1}^{\text{pred}} - h(u_i^{\text{pred}} - u_i^n)F_i^n}{1 + r - hF_i^{n+1}}$$

The application of the odd–even-type hopscotch schemes requires a special checkerboard-like space grid, in which the immediate neighbors of the so-called odd nodes are even and vice versa. In this work, three different structures are applied, see Figure 5, where only one odd and one even node is shown. The time elapses from top to bottom, and colored rectangles symbolize the stages. One exact repeating unit is surrounded by a blue dashed line for each case. In each stage, Formula (15) is employed, where the most up-to-date values of the neighbors  $u_{i\pm 1}$  is used when the new value of  $u_i$  is calculated.



**Figure 5.** Hopscotch-type space–time structures. The time elapses from the top ( $t = t^0$ ) to the bottom ( $t = t^{fin}$ ).

3. The original version of the odd–even hopscotch method ([47], denoted here by OOEH) uses only integer time steps with  $\theta = 1$  in the first and  $\theta = 0$  in the second stage.
4. In the asymmetric-hopscotch (ASH) scheme, the repeating unit consists of two half and one full-size stages. First, a half-sized time step (light green rectangle with the number ‘1’) is taken for the odd nodes with  $\theta = 0$ , and then a full-length step for the even nodes (light orange rectangle) using  $\theta = \frac{1}{2}$ , and finally a halved third stage (dark orange box) closes the calculation with  $\theta = 1$ .
5. The leapfrog-hopscotch (LH) algorithm starts with Stage 0 (not repeated, green box), which uses  $\theta = 0$ . The intermediate stages as well as the last stage (light and dark orange boxes) uses  $\theta = \frac{1}{2}$ .
6. The Dufort–Frankel (DF) algorithm [48] (p. 313) is an old but non-conventional method. In our case, it employs the following formula:

$$u_i^{n+1} = \frac{(1 - 2r)u_i^{n-1} + 2ru_{i\pm 1}^n}{1 + 2r - 2hF_i^{n+1}} \tag{16}$$

As one can see, the r. h. s. of the formula contains  $u_i^{n-1}$ , so it is a two-step method, which means it is not self-starter. The first time step has to be made by another algorithm for which we use the UPFD method. In the original DF method, the reaction term was not present. Here, we take it into account at the new time level, thus it appears in the denominator.

7. The FTCS (forward-time central-space) scheme, which is built on the explicit Euler discretization, can be obtained from (13) by the  $\theta = 1$  substitution.
8. The standard implicit scheme, which is built on the implicit Euler discretization, is obtained from (13) by the  $\theta = 0$  substitution. This yields an algebraic equation system with  $N$  unknowns, which can be solved in many ways. First, we solve it with the preconditioned conjugate gradient (PCG) method [49] which has been implemented by the built-in routine of MATLAB called pcg. The conjugate gradient method is a non-classical iterative method which can be used for solving linear equation systems with symmetric, positive definite coefficient matrix. In general, the conjugate gradient method yields high accuracy numerical solutions in the so-called A-norm. However, the convergence rate strongly depends on the spectral features of the coefficient matrix, thus it can be very slow for stiff problems. Hence, one can apply preconditioning, i.e., transforming the linear equation system into another linear equation system which is

equivalent in the sense that it has the same solution, but it has more favorable spectral features. As a consequence, one loses some accuracy but can reach more favorable convergence rates.

9. Finally, the above-mentioned algebraic system is solved by GMRES (generalized minimal residual) method [50,51]. It is a non-classical iteration method for solving linear systems of equations which are not necessarily symmetric. The essence of the method is to find an approximate solution of linear equation system, which is the most accurate approximation in the Euclidean norm if we consider a Krylov subspace with a given rank. The GMRES method has been implemented in the gmres built-in routine of MATLAB.

The PI, ASH, and LH methods were invented by our research group; thus, their description with more details can be found in our original publications. The PI was directly constructed for the diffusion–reaction equation, while the LH was optimized later [52] for this case. It is analytically proven that they are unconditionally stable if the reaction term is non-positive, and they have second-order temporal convergence. In the case of the ASH (as well as for the OOEH and DF, as far as we know), the unconditional stability (which is very rare for explicit methods) and the second-order convergence are proven only for the diffusion equation. On the other hand, the FTCS, the implicit, and the UPFD schemes have first-order convergence.

#### 4. Verification of the Numerical Methods

The main task is to investigate the numerical error, most importantly how fast it decreases with the time step size  $h$ . We begin with a simulation for all of the nine methods, choosing a quite large and fixed  $h$  and registering the error, which is calculated as follows:

$$Error = \max_{1 \leq i < N} |u_i^{\text{analytic}}(t^{\text{fin}}) - u_i^{\text{num}}(t^{\text{fin}})|$$

This formula gives the usual  $L_\infty$  error, which is the maximum absolute difference at the final time  $t^{\text{fin}}$  between the analytical and the numerical solution. Then, the time step size is divided by two subsequently while the simulation is repeated with the same parameters and settings until very small time step sizes are reached. The exception is the tolerance for the implicit methods, which always starts from 0.01 for the largest used time step size. Then, it is divided by 2.5 (3.2) in the case of the PCG (GMRES) solvers to keep the errors of the implicit discretization and that of the iterative solution comparable to optimize running time and accuracy at the same time.

The system matrix  $M$  has two parts: the reaction term gives only diagonal elements, while the diffusion term gives off diagonal ones as well. If the reaction is non-positive,  $M$  has only negative eigenvalues. If one denotes the eigenvalues with smallest (largest) absolute value by  $\lambda_{\text{MIN}}$  ( $\lambda_{\text{MAX}}$ ), then the CFL limit for the FTCS scheme can be exactly calculated as  $h_{\text{CFL}}^{\text{EE}} = |2/\lambda_{\text{MAX}}|$ . Furthermore, the ratio  $\lambda_{\text{MAX}}/\lambda_{\text{MIN}}$  can be called the stiffness ratio of the problem.

The numerical calculations in this section are performed using the MATLAB R2018a software on a desktop computer with a DualCore Intel Pentium G3220, 3000 MHz CPU, 8 GB DDR3-1333 RAM. The analytical solutions and the boundary conditions containing the Mathieu or Heun functions are calculated by the Maple software and then imported to MATLAB. However, the values of the boundary conditions are supposed to be calculated for all time steps, which would consume much more time than performing the numerical time–integration algorithms for all the nodes of the mesh. We avoided these huge running times using a trick: the boundary values are calculated only in  $N_{tb}$  time points, placed uniformly on the whole time interval. Then, linear interpolation using the pre-calculated boundary values at the two appropriate times in the vicinity of the actual time of the simulation is used to gain the values of the boundary conditions. It is always checked that this approximation yields a smaller error than the errors of the numerical methods around the middle of the space interval.

4.1. Case Study 1 with Small Value of Parameter  $a$

In this numerical experiment, we use Equation (4) and the following parameters:

$$D = 1, \quad a = -0.1, \quad c = 1.4, \quad c_1 = 0.8763, \quad c_2 = 0.6572, \quad (17)$$

$$N_x = 2000, \quad x_0 = -2, \quad \Delta x = 0.005, \quad N_{tb} = 2000, \quad t^0 = 1, \quad t^{\text{fin}} = 3$$

Since the contribution of the time-dependent term to the matrix elements are small, the CFL limit is only slightly changing and its value is approximately  $h_{CFL}^{EE} = 1.25 \cdot 10^{-5}$ . The stiffness ratio is about  $2 \cdot 10^7$ . The errors of the nine examined methods for decreasing time step sizes are shown in a log-log diagram in Figure 6. The minimum possible errors are determined by the (fixed) space discretization, see on the bottom left of the figure. We plot the analytical  $u$  function in Figure 7 at the initial and final time, as well as the numerical solution for the PCG and the LH solvers. We chose a time step size 0.002, for which the accuracy is more than acceptable in many engineering or biological applications.

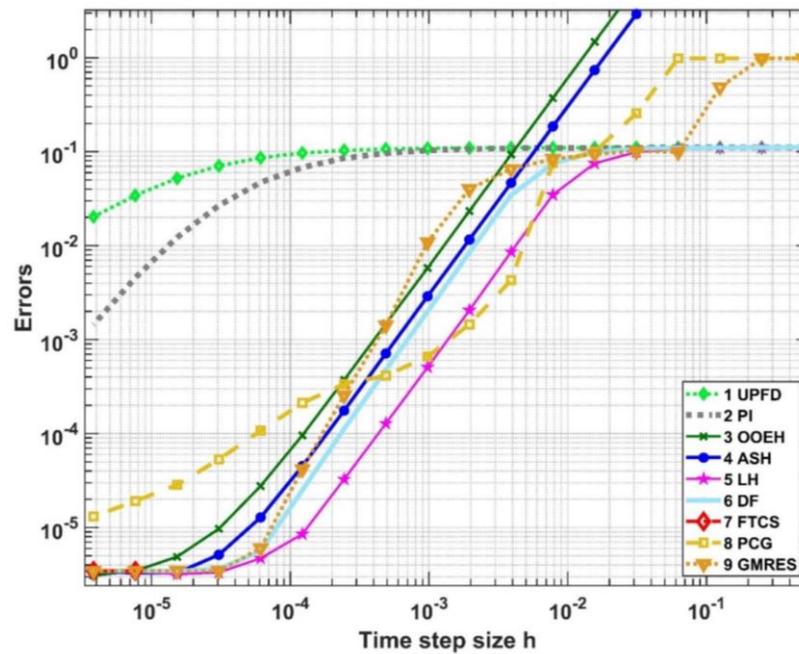
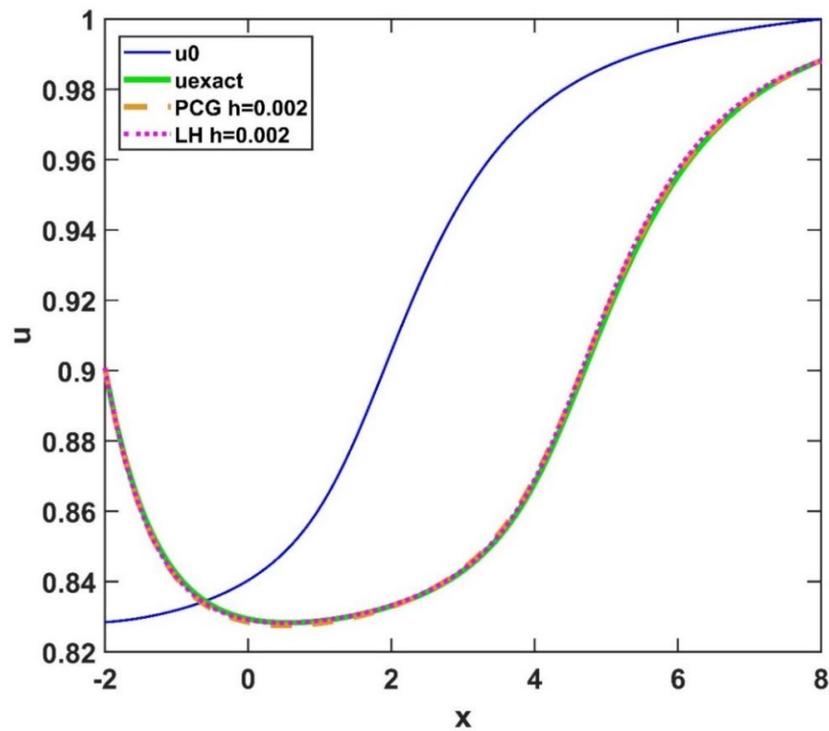


Figure 6. Errors as a function of the temporal step size for Case study 1.

One can see that the implicit methods roughly give the same error as the best explicit methods. However, their running time is usually more than three orders of magnitude larger than those of the explicit methods. The LH and DF methods can give a reasonable result in 0.1 s, while the implicit solvers need around 100 s. The explicit FTCS scheme is stable only below the CFL limit, and therefore only two points are presented at the bottom left side of the figure. To give a usable (actually very accurate) result, it needs about 20 s, which is still less than for the implicit methods. One can also see that the UPFD and the PI methods converge rather slowly. This is due to the fact [35] that their truncation errors contain more extra terms than those of the other methods.

The running times here are just examples. They will be measured more systematically in the next section, where realistic simulation of heat transfer in 2D will be conducted.



**Figure 7.** The concentration  $u$  as a function of  $x$  in the case of the initial function  $u^0$ , the exact analytical solution at  $t^{\text{fin}}$ , the implicit-PCG algorithm (with Tolerance  $10^{-5}$ ), and the LH scheme for  $h = 0.002$  in the case of small value of  $a$  (Case study 1). The maximum errors (and running times) are 0.0015 (98.7 s) and 0.0022 (0.107 s) for the PCG and LH methods, respectively.

4.2. Case Study 2 with Large Value of Parameter  $a$

In this numerical experiment, we use Equation (4) and the following parameters:

$$D = 1, \quad a = -5, \quad c = 1.9, \quad c_1 = 0.07987, \quad c_2 = 0.18153, \quad (18)$$

$$N_x = 1000, \quad x_0 = 1, \quad \Delta x = 0.004, \quad N_{tb} = 2000, \quad t^0 = 1, \quad t^{\text{fin}} = 1.6$$

The CFL limit is slightly fluctuating around  $h_{CFL}^{EE} = 8 \cdot 10^{-6}$ , while the stiffness ratio is increasing from  $1.25 \cdot 10^5$  to  $2.42 \cdot 10^5$ . The errors of the nine examined methods for decreasing time step sizes are shown in Figure 8. The running times of the implicit solvers are 200–1100 times longer than for the explicit schemes for the same time step sizes.

We plot the analytical  $u$  function in Figure 9 at the initial and final time, as well as the numerical solution for the LH and the implicit-GMRES solvers. In case of the former, the time step size is  $2.3 \cdot 10^{-3}$ , the maximum error is 0.029, while the running time is 0.011 s. In case of the GMRES solver, the time step size is  $5.86 \cdot 10^{-4}$ , the tolerance is  $2.9 \cdot 10^{-6}$ , the error is 0.048, while the running time is 21.2 s. The first non-divergent run for FTCS scheme took 4.2 s, but it produced a very accurate solution.

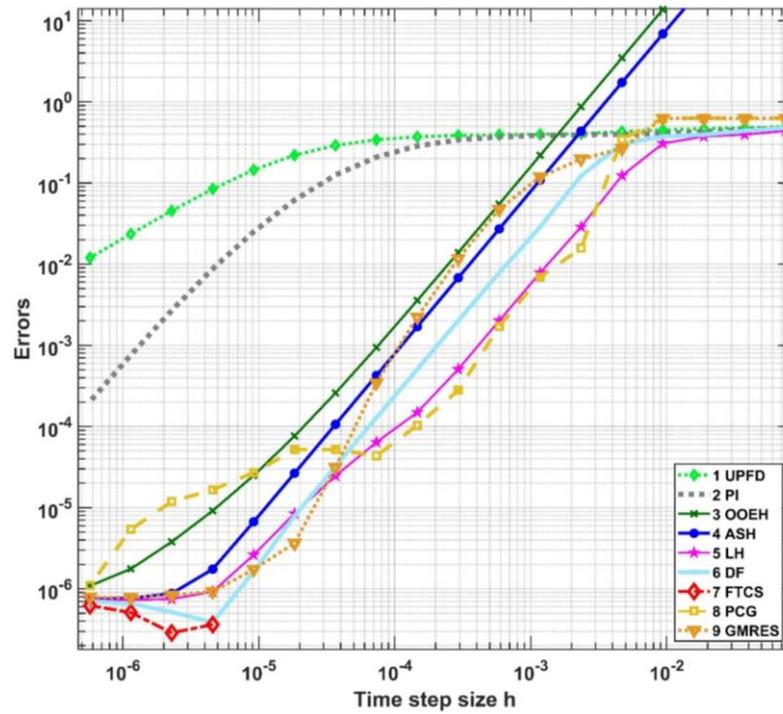


Figure 8. Errors as a function of the temporal step size for Case study 2.

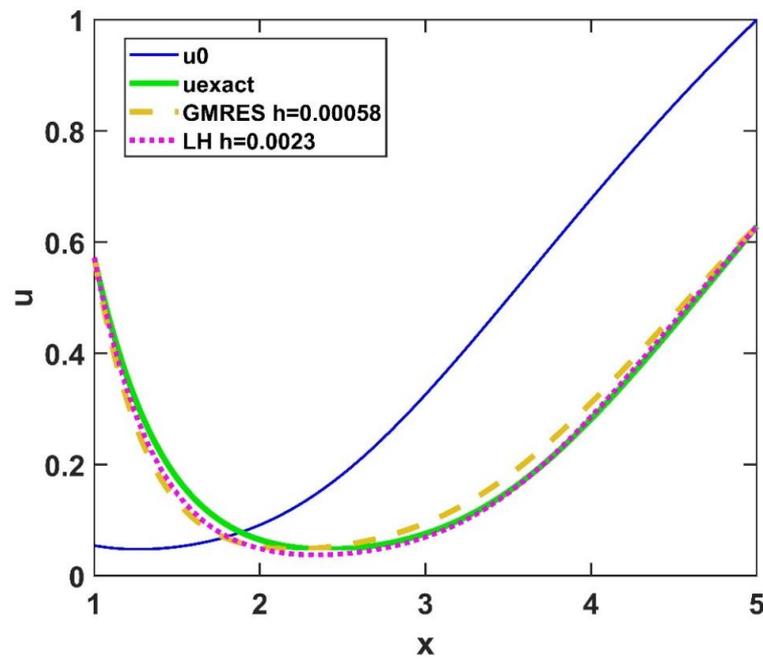


Figure 9. The concentration  $u$  as a function of  $x$  in the case of the initial function  $u^0$ , the exact analytical solution at  $t^{\text{fin}}$ , the implicit-GMRES algorithm, and the LH scheme. For more data, please see the main text.

#### 4.3. Case Study 3 with the Cosine Reaction-Term

In this numerical experiment, we use Equation (7) and the following parameters:

$$\begin{aligned}
 D = 1, \quad a = 0.8, \quad c = 0.5, \quad c_1 = 0.98697, \quad c_2 = 0.78958, \\
 N_x = 500, \quad x_0 = 0, \quad \Delta x = 0.02, \quad N_{tb} = 200, \quad t^0 = 1, \quad t^{\text{fin}} = 2
 \end{aligned}
 \tag{19}$$

The CFL limit is  $h_{CFL}^{EE} = 2 \cdot 10^{-4}$ , while the stiffness ratio cannot be calculated in the usual way since there are positive eigenvalues due to the reaction term. The errors of the nine examined methods for decreasing time step sizes are shown in Figure 10. The running times of the implicit solvers are usually 200–300 times longer than for the explicit schemes.

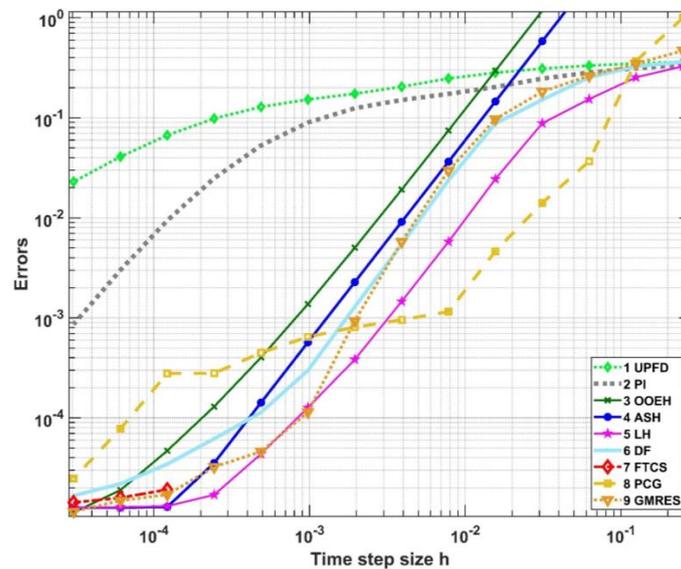


Figure 10. Errors as a function of the temporal step size for Case study 3.

We plot the analytical  $u$  function in Figure 11 at the initial and final time, as well as the numerical solution for the LH and the implicit-GMRES solvers for time step size 0.02. In the case of the former, the maximum error is 0.0450, while the running time is 0.01 s. In the case of the PCG solver, the error is 0.0406, while the running time is 0.2 s. This running time is only 20 times larger than that of the LH scheme due to the crude tolerance (0.01), which was chosen to reach roughly the same accuracy. The first non-divergent run for the FTCS scheme took 0.1 s. One can see that the explicit and stable methods, especially the LH, are significantly faster than the traditional methods.

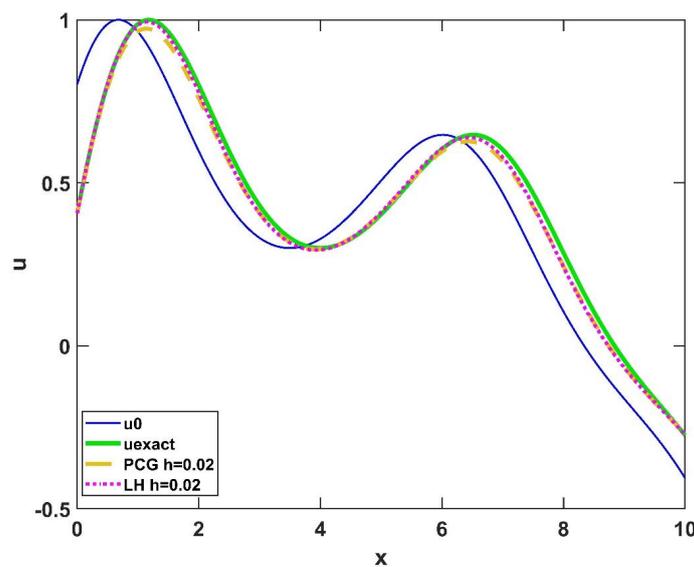


Figure 11. The concentration  $u$  as a function of  $x$  in the case of the initial function  $u^0$ , the exact analytical solution at  $t^{fin}$ , the implicit-PCM algorithm, and the LH scheme.

We have to note that this reaction function is positive at half of the evaluation points. The problem is that, as it was mentioned, the stability of the numerical schemes is proven only for non-positive reaction terms. One can see that the methods behave very well in the presented case. However, for increasing amplitude  $a$  of the reaction term, they gradually become unstable. We present this case study only as a demonstration that the proposed schemes can be very effective even in this case, but the thorough investigation of the stability of the schemes in the case of reaction terms which are not always negative is out of the scope of this paper.

### 5. Numerical Simulation of Surface Subjected to Wind

In this section, all the running times are measured on a desktop computer with an Intel Core i7-324 11700F (16 CPUs) and 64 GB RAM is used, while the program we used is MATLAB R2020b.

#### 5.1. The Structure and the Materials of the Surface

In this section, we model a wall surface with dimensions of 1 m on the  $x$  and  $y$  axes and 0.1 m on the  $z$  axis. In Figure 12, a wall's surface is examined. Half of the surface ( $0 < x < 0.5$  m) is made of brick and the other half ( $0.5 \text{ m} < x < 1$  m) is made of insulation.

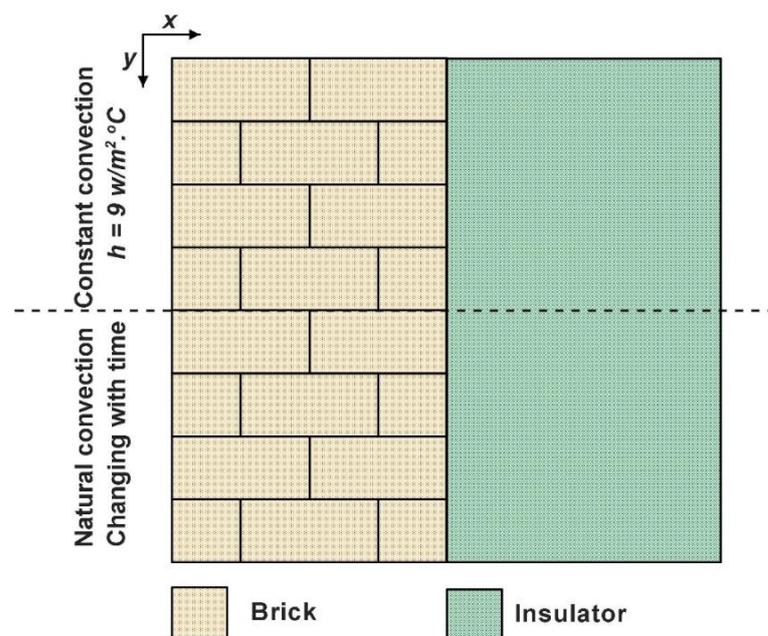


Figure 12. The surface of the modelled wall with two layers.

In Table 1, the material attributes are considered. These coefficients exhibit a sharp discontinuity when one material passes into the other. On the other hand, these coefficients are constant inside a material.

Table 1. Properties of the applied materials [53].

	$\rho$ ( $\text{kg}\cdot\text{m}^{-3}$ )	$c$ ( $\text{J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$ )	$k$ ( $\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$ )
Brick	1900	840	0.73
Rigid Polyurethane Foam	320	1400	0.023

#### 5.2. Mesh Construction

We assume that in the  $z$  direction there is no heat transfer and the physical properties do not change, thus we fix  $\Delta z = 0.1\text{m}$ . From the computational point of view, a two-dimensional issue is studied. The  $x$  and  $y$  coordinates fall within the unit interval, and

uniform square cells are used. We created meshes with four different resolutions with  $N_x = (50, 70, 90, \text{ and } 100)$  and  $N_y = (50, 70, 90, \text{ and } 100)$  for the number of cells in the  $x$ -axes and in the  $z$ -axes; therefore,  $N = N_x N_y = (2500, 4900, 8100, \text{ and } 10,000)$  is the global number of cells.

### 5.3. Discretization and Boundary Conditions in Inhomogeneous Media

Our goal now is to present a step-by-step derivation of the general discretization of the above problem starting from a simple one-dimensional problem. If the material properties depend on space, but the number of space dimensions is still one, the following equation can be the starting point for heat conduction:

$$c(x)\rho(x)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( k(x)\frac{\partial u}{\partial x} \right) \tag{20}$$

We discretize the function  $k$ , and at the same time the space derivatives in Equation (20) by the standard central difference formula to obtain:

$$c(x_i)\rho(x_i)\frac{\partial u}{\partial t} \Big|_{x_i} = \frac{1}{\Delta x} \left[ k\left(x_i + \frac{\Delta x}{2}\right) \frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} + k\left(x_i - \frac{\Delta x}{2}\right) \frac{u(x_i - \Delta x) - u(x_i)}{\Delta x} \right]. \tag{21}$$

Equations (20) and (21) are based on the node picture, typically used by mathematicians. Instead of node variables, let us introduce cell variables to arrive at a resistance-capacitance-type model of heat conduction. It means that  $u_i$ ,  $c_i$ , and  $\rho_i$  is the approximation of the average temperature, specific heat, and density of cell  $i$ , by their value at the cell center. Furthermore,  $k_{i,i+1}$  is the heat conductivity between cell  $i$  and its (right) neighbor, estimated by its value at the border of the cells. Now, the previous formula will have the form:

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x} \left( k_{i,i+1} \frac{u_{i+1} - u_i}{\Delta x} + k_{i-1,i} \frac{u_{i-1} - u_i}{\Delta x} \right) \tag{22}$$

Let us now consider heat conduction in a thin rod with cross-section  $S$ , divided into cells only along its length. The volume and the heat capacity of the cell can be given as  $V = S\Delta x$  and  $C_i = c_i\rho_iV$ , respectively. On the other hand, the thermal resistance between two neighboring cell is estimated as  $R_{i, i+1} \approx \Delta x / (Sk_{i, i+1})$ . Now, the equation for the time derivative of the temperature of each cell in the rod is as follows:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i}C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i}C_i} \tag{23}$$

At this point, we generalize this treatment into the case of a thin plate, laying in the  $x$ - $y$  plane with thickness  $\Delta z$ , which computationally is the two-dimensional problem we actually study. The heat capacity of the cells are  $C_i = c_i\rho_i\Delta x\Delta z\Delta y$ , while for thermal resistance in the  $x$ -axes, we use the formula  $R_{i,i+1} \approx \frac{\Delta x}{k_{i,i+1}S_x}$ , and  $S_x = \Delta y\Delta z$  is the surface area perpendicular to  $x$ . Therefore, the  $x$  and  $y$  direction resistances in the surface simulation with non-homogenous material and uniform mesh can be expressed as

$$R_{i,i+1} \approx \frac{\Delta x}{k_i\Delta z\Delta y} \text{ and } R_{i,i+N_x} \approx \frac{\Delta y}{k_i\Delta x\Delta z} \tag{24}$$

respectively, where the cell labelled  $i + N_x$  appears directly underneath the cell labelled  $i$ . The horizontal and vertical resistance between cells  $i$  and  $i + 1$  will vary if the materials on the two sides of the cell border are different [54]:

$$R_{i,i+1} \approx \frac{\Delta x}{2k_i\Delta y\Delta z} + \frac{\Delta x}{2k_{i+1}\Delta y\Delta z} \tag{25}$$

and

$$R_{i,i+N_x} \approx \frac{\Delta y}{2k_i \Delta x \Delta z} + \frac{\Delta y}{2k_{i,i+N_x} \Delta x \Delta z} \tag{26}$$

As a generalization of Equation (23), one may construct the ODE system for the time derivative of the cell variables for a generic grid by using the above approximations, as follows:

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} + q_i - K \cdot u_i \tag{27}$$

The term  $q$  arises because the temperature is not taken related to the ambient temperature  $u_a$ , since the latter is changing. In that case, heat conduction can be taken into account as a term  $K(u_a - u)$ , where the ambient temperature  $u_a$  does not depend on  $u$ , and thus  $Ku_a$  is considered as a heat source term  $q$ . The source term potentially includes other heat sources such as radiation or the Joule-heat due to electric currents.

#### 5.4. The Numerical Algorithms Used

The following two quantities will be used in the definition of the methods:

$$A_i^n = h \sum_{j \neq i} \frac{u_j^n}{R_{i,j} C_i} + h \cdot q_i, \text{ and } r_i = h \sum_{j \neq i} \frac{1}{R_{i,j} C_i}, \quad i = 1, \dots, N, \quad n = 0, \dots, T$$

The quantity  $r$  is similar to the mesh-ratio  $r$ , introduced after Equation (13). The quantity  $A_i^n$  will transmit data to cell  $i$  from its surrounding ones. The two explicit methods we use here to solve Equation (27) are generalized via the following formulas.

1. The leapfrog-hopsotch (LH) uses the generalized Theta-formula [55], which reads as follows for a full time step size:

$$u_i^{n+1} = \frac{(1 - \theta(r_i - hK_i^n))u_i^n + A_i}{1 + (1 - \theta)(r_i - hK_i^{n+1})}, \quad \theta \in [0, 1] \tag{28}$$

The length of the stages and the values of the parameter  $\theta$  are the same as in the special case, see point 5 in Section 3.

2. Dufort and Frankel (DF)

The initial stage applies the UPFD scheme, which can be obtained by the  $\theta = 0$  substitution from Equation (28). In all subsequent time steps, we employ the formula:

$$u_i^{n+1} = \frac{(1 - r_i)u_i^{n-1} + 2A_i^n}{1 + r_i + 2hK} \tag{29}$$

The two implicit methods are also tested for the simulation of the wall. The standard implicit Euler discretization is applied to Equation (27) and then the obtained algebraic equation system is solved by the PCG and GMRES solvers.

#### 5.5. The Initial and the Boundary Conditions

In this section,  $t_{\text{fin}} = 22,500$  s serves as the final time (the end of the examined time period). The duration of each time step is also expressed in seconds, first with  $h = 900$  s, and it is decreased gradually to a small number ( $h = 0.01$  s). A spatially constant initial temperature  $u(x, y, t = 0) = 270$  K is set. Neumann boundary conditions with a zero temperature-flux are applied to boundaries, preventing any conductive heat transfer:

$$\frac{\partial u}{\partial x}(x, y = 0, t) = \frac{\partial u}{\partial x}(x, y = 1, t) = \frac{\partial u}{\partial y}(x, y = 0, t) = \frac{\partial u}{\partial y}(x, y = 1, t) = 0$$

The desired insulating outcome is attained by setting the necessary resistances to infinity and the value of the matrix components indicating heat conduction over the boundary to zero.

Convective heat transfer occurs in the z-direction, which is perpendicular to the plane depicted in Figure 12. For the upper part (0 < y < 0.5 m), the ambient temperature is taken to be constant, and for the lower part (0.5 m < y < 1 m), the convection circumstances are changing with time according to changing weather conditions.

Table 2 shows that the z-direction convection coefficients ( $h_c$ ) and the temperature in the upper part of the plane in Figure 12 has constant values, whereas the lower part of the elements has changing values depending on the environmental conditions. The ambient air temperature and velocity is taken to be 17 °C ≈ 290 K and  $v = 0$  m/s for the upper part, and changing according to real weather conditions in Miskolc city from 5 a.m. to 11 p.m. on the second day of January [56] for the lower part, as shown in Table 2.

**Table 2.** The temperature and heat source in case of wall surface area [56].

	Upper Elements	Lower Elements
$h_c \left( \frac{W}{m^2 \cdot K} \right)$	0.6	0.6–5.55
T (K)	290	275–280

We obtain the values of the coefficients in our Equation (27) as follows [53]:

$$K = \frac{h_c}{c\rho\Delta z}, \quad q = \frac{h_c}{c\rho \cdot \Delta z} \cdot u_a$$

We also supposed that the upper elements and lower elements have heat sources as follows:

For the upper elements:  $q_{up} = \frac{h_{cup}}{c\rho \cdot \Delta z} \times u_a$

For the lower elements:  $q_{lowl}(t) = \frac{h_{clow}(t)}{c\rho \cdot \Delta z} \times u_{alowl}(t),$

where  $K(t) = \frac{h_{clow}(t)}{c\rho\Delta z}, \quad q_i(t) = \frac{h_{clow}(t)}{c\rho \cdot \Delta z} \cdot u_{lowl}(t).$

The convection heat transfer coefficient for outside elements as a function of air velocity is estimated as follows [57]:

$$h_{clow}(t) = 0.6 + 6.64\sqrt{v}(t)$$

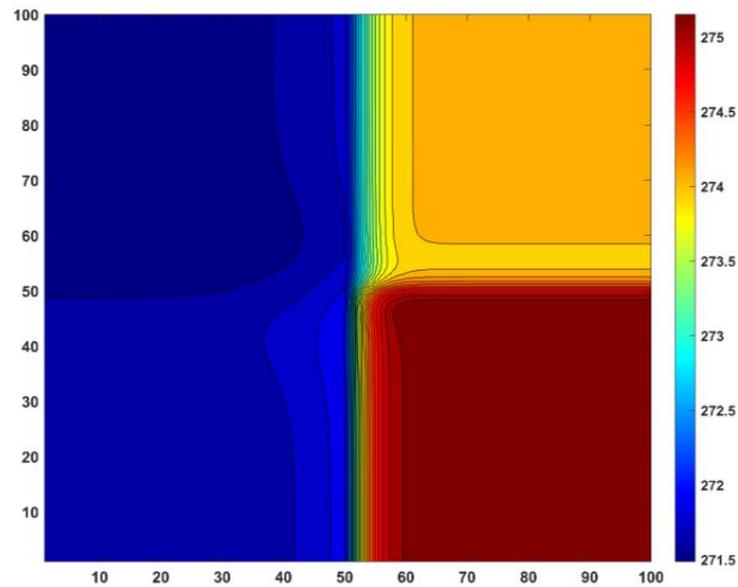
$v(t)$ : the air velocity is taken for each 900 s in [m/s].

$u_{up}$ : the upper-side air temperature 295 [°K].

$u_{lowl}(t)$ : the lower air temperature for each 900 s in [°K].

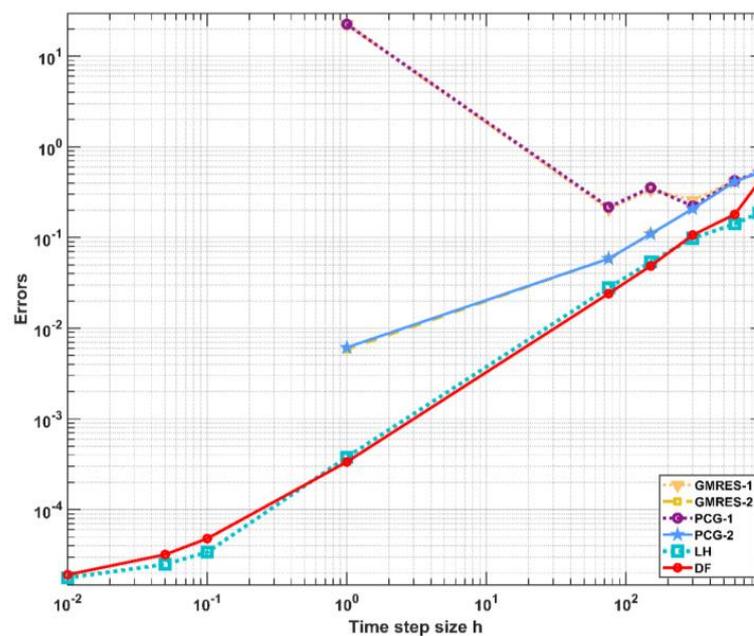
### 5.6. Results for the Surface of the Wall

Figure 13 shows the temperature distribution contour in Kelvin units for the surface area. The figure shows that in the case of the insulator (right-hand side of the figure), heat can hardly flow from the top of the figure to the bottom, so there are large temperature gradients. Moreover, because the heat capacity of the insulation layer is smaller than that of the brick layer, its temperature increases faster from the original 270 K.

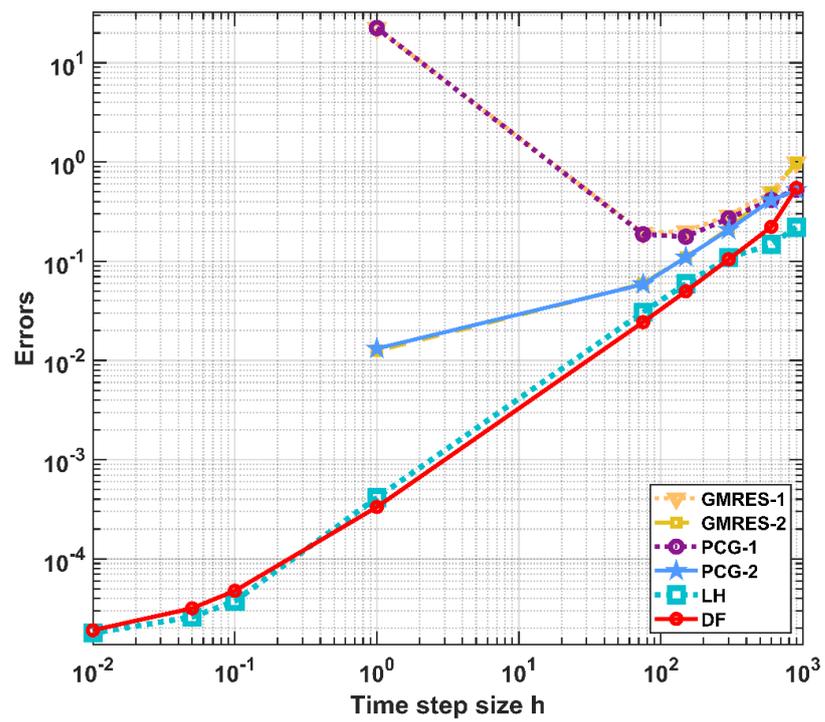


**Figure 13.** The temperature distribution contour in Kelvin units for the surface area (**upper half**) constant convection and (**lower half**) the convection changes with time depending on weather data.

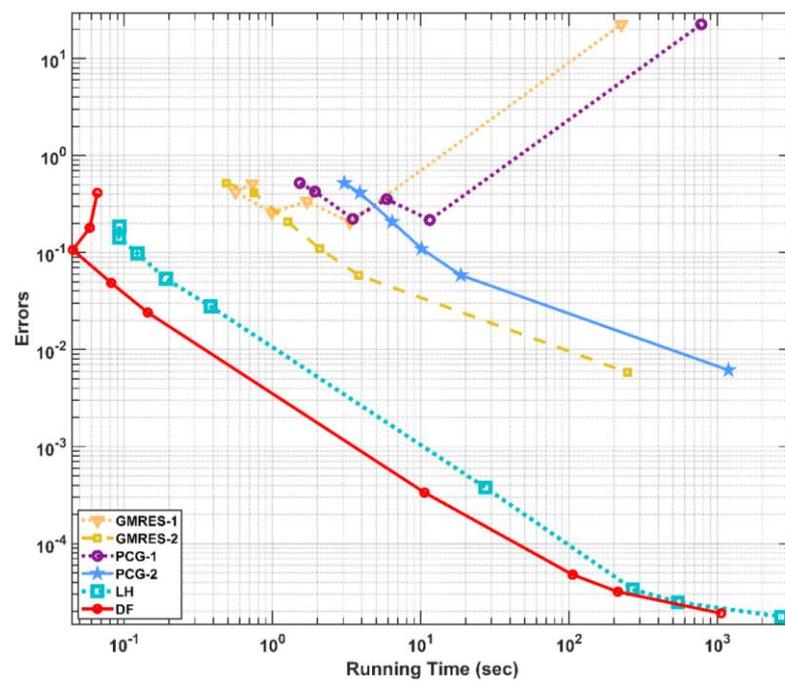
The maximum errors of the cell temperatures at the final time as a function of time step size are shown in Figures 14 and 15 for the 50 by 50 and 100 by 100 systems, respectively. One can see the LH is the most accurate scheme, followed by the DF, compared with the implicit methods with two different tolerances. Figures 16 and 17 for the 50 by 50 and 100 by 100 systems, respectively, show the running time with the maximum errors. We use many time steps for the explicit methods (LH and DF) and less for the implicit methods because they are much slower, and we see that LH and DF are faster and more accurate.



**Figure 14.** The maximum errors as a function of the time step size  $h$  for the two implicit methods with two different tolerances and the two explicit methods in the case of the 50 by 50 system.



**Figure 15.** The maximum errors as a function of the time step size  $h$  for the examined methods for the 100 by 100 system.



**Figure 16.** The maximum errors as a function of the running time for the tested methods for the 50 by 50 system.

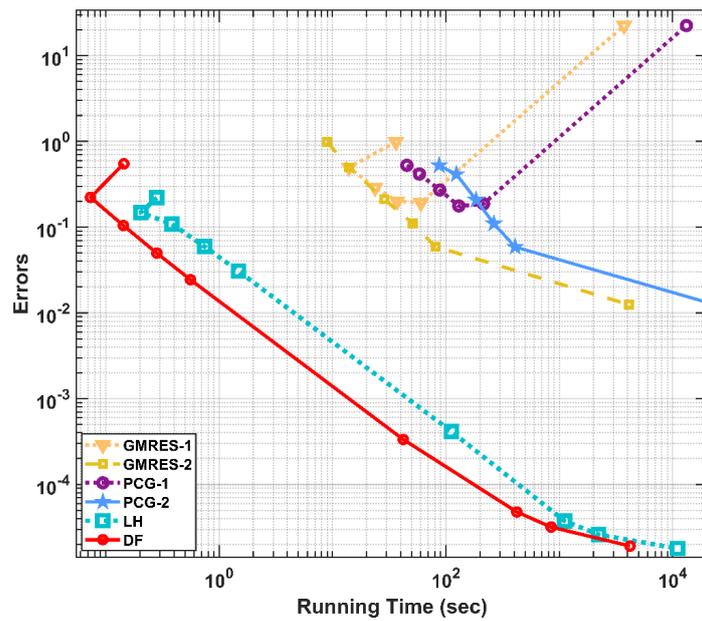


Figure 17. The maximum errors as a function of the running time for the tested methods for the 100 by 100 system.

Figure 18 shows the running time for time step size  $h = 1$  s with the total number of cells on the horizontal axis. On the right-side axis, it is shown for the LH and DF methods and on the left-side axis for implicit methods. One can see that the running time is a linear function in the case of the explicit methods, but not for the implicit methods. If using a logarithmic scale in both axis (Figure 19), it becomes visible that the slope for the explicit methods is one, while it is around two for the implicit solvers. This is expected, since the explicit methods apply loops over the cells, so the running time is directly proportional to  $N$ , while the implicit solvers work with the system matrix with size  $N \times N$ , thus the running time is proportional to  $N^2$ . It means that for even larger system sizes, e.g., in 3D, they become even less competitive with the explicit and stable methods.

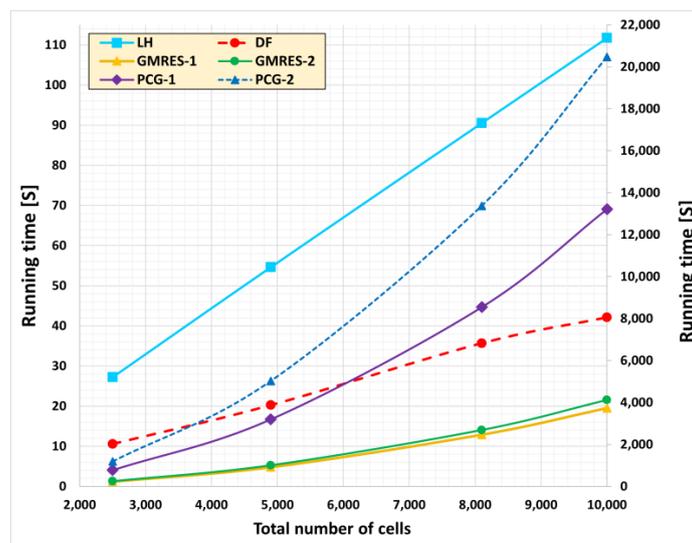
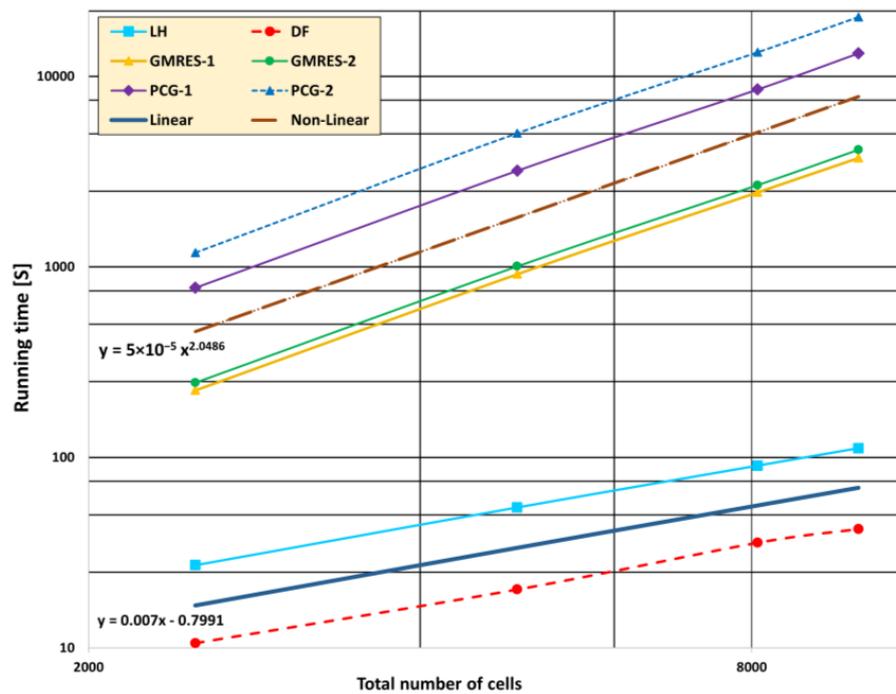


Figure 18. The running time with total number of cells for  $h = 1$  s. The left-hand side vertical axis refers to the LH and DF methods, while it is on the right-hand side axis for the implicit methods.

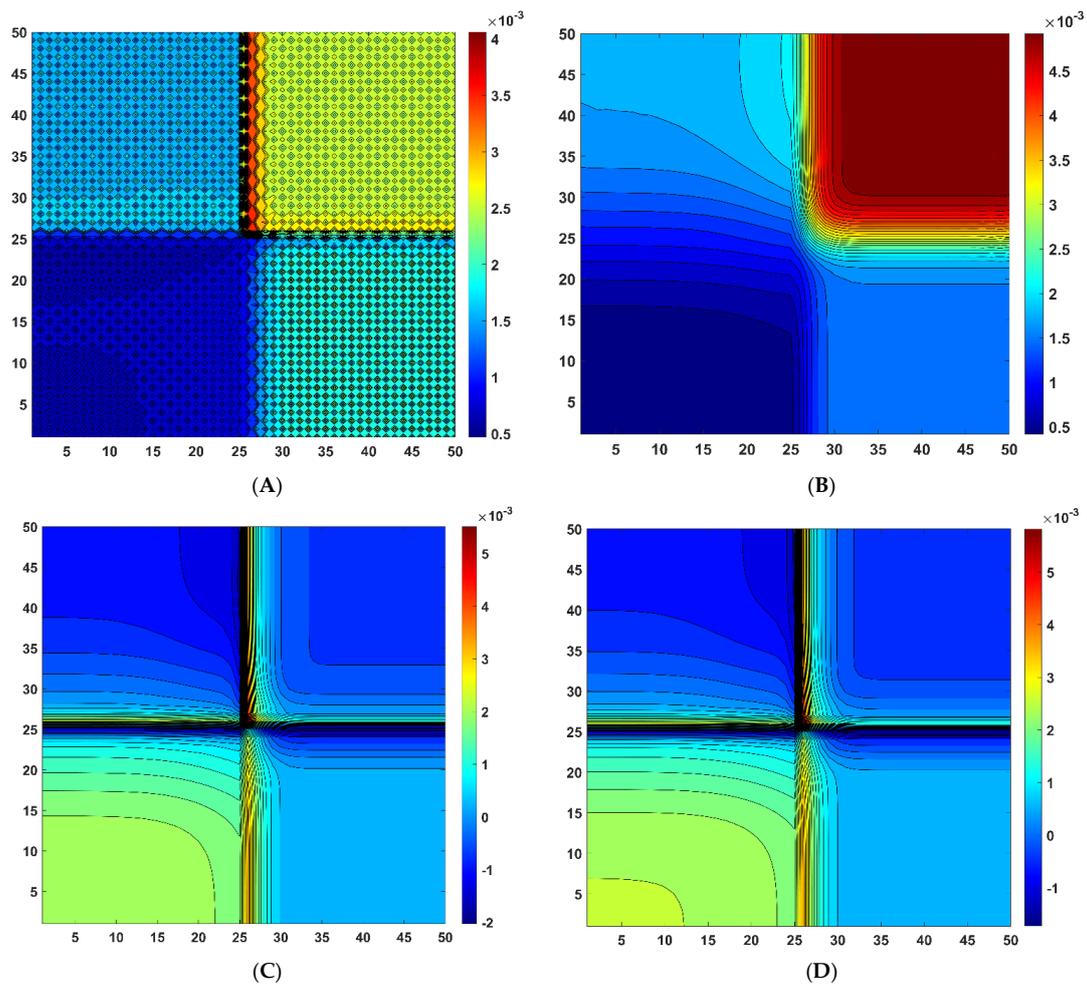


**Figure 19.** The running time in the same logarithmic scale for all the used solvers as a function of the total number of cells.

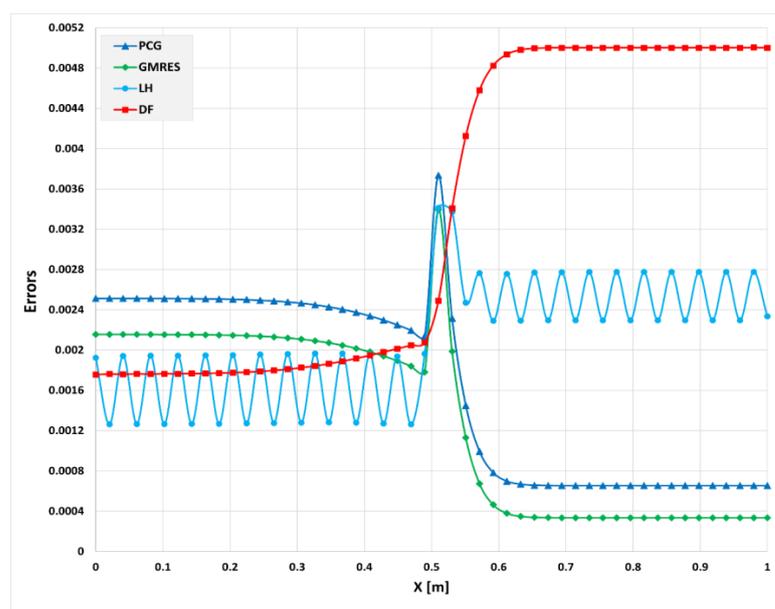
For the coarse mesh (50 by 50), we present the spatial distribution of errors in Figure 20 as 2D contour plots and in Figure 21 as curves for a specific z coordinate for the four methods. The time step sizes are chosen to reach very similar error levels. One can see that with the exception of the DF method, the errors are the largest, where the temperature gradients are largest, i.e., at the boundary of different materials and circumstances. The pattern of odd–even structure is also observable for the LH method, but even with this artificial oscillation, this algorithm is the most accurate. Table 3 shows the time step sizes, the errors, and the running times for the four methods in the case exemplified in Figures 20 and 21.

**Table 3.** The performance for the numerical methods.

Numerical Method	Time Step Size $h$ (Tolerance)	Maximum Error	Running Time (s)
leapfrog-hopscotch (LH)	10	0.0042	1.32
Dufort–Frankel (DF)	10	0.0051	0.97
generalized minimal residual (GMRES)	$1 \cdot (10^{-7})$	0.0058	246.965
preconditioned conjugate gradient (PCG)	$1 \cdot (10^{-7})$	0.0062	1187.404



**Figure 20.** The error of temperature distribution contour in Kelvin units (A) LH, (B) DF, (C) GMRES, and (D) PCG methods.



**Figure 21.** The error of temperature distribution at  $z = 80$  [cm] in Kelvin units for LH, DF, GMRES, and PCG methods along  $x$ -axis.

## 6. Conclusions and Summary

The diffusion–reaction PDE was studied where the reaction term is linear in the unknown variable, but its coefficient depends on both space and time in two different nonlinear ways. New analytical solutions were conveyed using the travelling-wave trial function. The solution for Lorentzian and cosine coefficient-function contains the Heun function and the Mathieu functions, respectively, so they are very nontrivial. These solutions were then reproduced with high accuracy by nine numerical algorithms. Six of them are explicit schemes with excellent stability properties. The rest are the standard explicit and implicit methods, which were severely outperformed by especially the Dufort–Frankel and the recently invented leapfrog-hopschotch methods. These two methods are also tested against the implicit ones in a realistic 2D case, when the temperature development of the surface of a wall was simulated. The time- and space-dependent reaction term was implemented as forced convection due to wind, while the material properties of the surface changed abruptly in space. Again, the explicit and stable methods proved to be more efficient than the implicit ones, and this advantage is expected to increase with the system size.

**Author Contributions:** Conceptualization and methodology, E.K. and I.F.B.; supervision and resources, E.K.; analytical investigation and the related visualization, I.F.B.; software, Á.N. and A.H.A.; verification and the related visualization, E.K.; numerical investigation in 2D and the related visualization, A.H.A.; writing—original draft preparation, A.H.A. and I.F.B.; writing—review and editing, E.K., A.H.A. and Á.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data available upon request from the first or the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jacobs, M.H. *Diffusion Processes*; Springer: Berlin/Heidelberg, Germany, 1935; ISBN 978-3-642-86414-8.
2. Lienhard, J.H. *A Heat Transfer Textbook*, 4th ed.; Phlogiston Press: Cambridge, MA, USA, 2017; ISBN 9780971383524.
3. Rothe, F. *Global Solutions of Reaction-Diffusion Systems*; Springer: Berlin/Heidelberg, Germany, 1984; Volume 1072, ISBN 978-3-540-13365-0.
4. Smoller, J. *Shock Waves and Reaction—Diffusion Equations*; Springer: New York, NY, USA, 1994; Volume 258, ISBN 978-1-4612-6929-8.
5. Gilding, B.H.; Kersner, R. *Travelling Waves in Nonlinear Diffusion-Convection Reaction*; Birkhäuser: Basel, Switzerland, 2004. [[CrossRef](#)]
6. Wilhelmsson, H.; Lazzaro, E. *Reaction-Diffusion Problems in the Physics of Hot Plasmas*; CRC Press: Boca Raton, FL, USA, 2000. Available online: <https://www.taylorfrancis.com/books/mono/10.1201/9781420033588/reaction-diffusion-problems-physics-hot-plasmas-wilhelmsson-lazzaro> (accessed on 13 May 2023).
7. Li, T.; Xiong, J.; Zhang, T.; Chai, X.; Liu, X. Multi-physics coupled simulation on steady-state and transients of heat pipe cooled reactor system. *Ann. Nucl. Energy* **2023**, *187*, 109774. [[CrossRef](#)]
8. Xie, Q.; Wang, Y.; Li, X.; Yang, Z.; Li, J.; Xie, Z.; Wang, X.; Cai, J.; Xu, Q. Two-dimensional transient heat transfer model of moving quenching jet based on machine learning. *Int. J. Heat Mass Transf.* **2022**, *191*, 122765. [[CrossRef](#)]
9. Kim, K.-S.; Won, M.-H.; Kim, J.-W.; Back, B.-J. Heat pipe cooling technology for desktop PC CPU. *Appl. Therm. Eng.* **2003**, *23*, 1137–1144. [[CrossRef](#)]
10. Wang, Y.; Wang, B.; Zhu, K.; Li, H.; He, W.; Liu, S. Energy saving potential of using heat pipes for CPU cooling. *Appl. Therm. Eng.* **2018**, *143*, 630–638. [[CrossRef](#)]
11. Du, H.; Ekkad, S.V.; Han, J.-C.; Lee, C.P. Detailed Film Cooling Measurements over a Gas Turbine Blade Using a Transient Liquid Crystal Image Technique. *Int. J. Rotating Mach.* **2001**, *7*, 415–424. [[CrossRef](#)]
12. Mátyás, L.; Barna, I.F. General Self-Similar Solutions of Diffusion Equation and Related Constructions. *Rom. J. Phys.* **2022**, *67*, 101.
13. Abdulrahman, A.; Gamaoun, F.; Kumar, R.V.; Khan, U.; Gill, H.S.; Nagaraja, K.; Eldin, S.M.; Galal, A.M. Study of thermal variation in a longitudinal exponential porous fin wetted with TiO<sub>2</sub>–SiO<sub>2</sub>/ hexanol hybrid nanofluid using hybrid residual power series method. *Case Stud. Therm. Eng.* **2023**, *43*, 102777. [[CrossRef](#)]
14. Turkyilmazoglu, M. Convergent optimal variational iteration method and applications to heat and fluid flow problems. *Int. J. Numer. Methods Heat Fluid Flow* **2016**, *26*, 790–804. [[CrossRef](#)]
15. Ain, Q.T.; Nadeem, M.; Karim, S.; Akgül, A.; Jarad, F. Optimal variational iteration method for parametric boundary value problem. *AIMS Math.* **2022**, *7*, 16649–16656. [[CrossRef](#)]

16. Savović, S.; Djordjevich, A. Numerical solution of the diffusion equation describing the flow of radon through concrete SEQ CHAPTER. *Appl. Radiat. Isot.* **2008**, *66*, 552–555. [[CrossRef](#)]
17. Jejenywa, O.A.; Gidey, H.H.; Appadu, A.R. Numerical Modeling of Pollutant Transport: Results and Optimal Parameters. *Symmetry* **2022**, *14*, 2616. [[CrossRef](#)]
18. Mbroh, N.A.; Munyakazi, J.B. A robust numerical scheme for singularly perturbed parabolic reaction-diffusion problems via the method of lines. *Int. J. Comput. Math.* **2021**, *99*, 1139–1158. [[CrossRef](#)]
19. Ndou, N.; Dlamini, P.; Jacobs, B.A. Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations. *Mathematics* **2022**, *10*, 2639. [[CrossRef](#)]
20. Kumar, V.; Chandan, K.; Nagaraja, K.V.; Reddy, M.V. Heat Conduction with Krylov Subspace Method Using FEniCSx. *Energies* **2022**, *15*, 8077. [[CrossRef](#)]
21. Jiang, T.; Zhang, Y.-T. Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection–diffusion–reaction equations. *J. Comput. Phys.* **2013**, *253*, 368–388. [[CrossRef](#)]
22. Heidari, M.; Ghovatmand, M.; Skandari, M.H.N.; Baleanu, D. Numerical Solution of Reaction–Diffusion Equations with Convergence Analysis. *J. Nonlinear Math. Phys.* **2022**, *30*, 384–399. [[CrossRef](#)]
23. Kolev, M.K.; Koleva, M.N.; Vulkov, L.G. An Unconditional Positivity-Preserving Difference Scheme for Models of Cancer Migration and Invasion. *Mathematics* **2022**, *10*, 131. [[CrossRef](#)]
24. Beuken, L.; Cheffert, O.; Tutueva, A.; Butusov, D.; Legat, V. Numerical Stability and Performance of Semi-Explicit and Semi-Implicit Predictor–Corrector Methods. *Mathematics* **2022**, *10*, 2015. [[CrossRef](#)]
25. Fedoseev, P.; Pesterev, D.; Karimov, A.; Butusov, D. New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers. *Algorithms* **2022**, *15*, 275. [[CrossRef](#)]
26. Ji, Y.; Xing, Y. Highly Accurate and Efficient Time Integration Methods with Unconditional Stability and Flexible Numerical Dissipation. *Mathematics* **2023**, *11*, 593. [[CrossRef](#)]
27. Settanni, G.; Sgura, I. Devising efficient numerical methods for oscillating patterns in reaction–diffusion systems. *J. Comput. Appl. Math.* **2016**, *292*, 674–693. [[CrossRef](#)]
28. Yadav, V.S.; Singh, A.; Maurya, V.; Rajpoot, M.K. New RK type time-integration methods for stiff convection–diffusion–reaction systems. *Comput. Fluids* **2023**, *257*, 105865. [[CrossRef](#)]
29. Essongue, S.; Ledoux, Y.; Ballu, A. Speeding up mesoscale thermal simulations of powder bed additive manufacturing thanks to the forward Euler time-integration scheme: A critical assessment. *Finite Elements Anal. Des.* **2022**, *211*, 103825. [[CrossRef](#)]
30. Chen-Charpentier, B.M.; Kojouharov, H.V. An unconditionally positivity preserving scheme for advection–diffusion reaction equations. *Math. Comput. Model.* **2013**, *57*, 2177–2185. [[CrossRef](#)]
31. Appadu, A.R. Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *Int. J. Numer. Methods Heat Fluid Flow* **2017**, *27*, 1412–1429. [[CrossRef](#)]
32. Savović, S.; Drljača, B.; Djordjevich, A. A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes. *Ric. di Mat.* **2022**, *71*, 245–252. [[CrossRef](#)]
33. Pourghanbar, S.; Manafian, J.; Ranjbar, M.; Aliyeva, A.; Gasimov, Y.S. An Efficient Alternating Direction Explicit Method for Solving a Nonlinear Partial Differential Equation. *Math. Probl. Eng.* **2020**, *2020*, 9647416. [[CrossRef](#)]
34. Al-Bayati, A.Y.; Manaa, S.A.; Al-Rozbayani, A.M. Comparison of Finite Difference Solution Methods for Reaction Diffusion System in Two Dimensions. *AL-Rafidain J. Comput. Sci. Math.* **2011**, *8*, 21–36. [[CrossRef](#)]
35. Nagy, Á.; Majár, J.; Kovács, E. Consistency and Convergence Properties of 20 Recent and Old Numerical Schemes for the Diffusion Equation. *Algorithms* **2022**, *15*, 425. [[CrossRef](#)]
36. Nagy, Á.; Omle, I.; Kareem, H.; Kovács, E.; Barna, I.F.; Bogнар, G. Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation. *Computation* **2021**, *9*, 92. [[CrossRef](#)]
37. Saleh, M.; Kovács, E.; Barna, I.F.; Mátyás, L. New Analytical Results and Comparison of 14 Numerical Schemes for the Diffusion Equation with Space-Dependent Diffusion Coefficient. *Mathematics* **2022**, *10*, 2813. [[CrossRef](#)]
38. Saleh, M.; Kovács, E.; Barna, I.F. Analytical and Numerical Results for the Transient Diffusion Equation with Diffusion Coefficient Depending on Both Space and Time. *Algorithms* **2023**, *16*, 184. [[CrossRef](#)]
39. Olver, F.W.J.; Lozier, D.W.; Boisvert, R.F.; Clark, C.W. *NIST Handbook of Mathematical Functions*; Cambridge University Press: New York, NY, USA, 2011; Volume 66, ISBN 978-0-521-14063-8.
40. Slavyanov, S.; Lay, W. *Special Functions: Unified Theory Based on Singularities*; Oxford University Press Inc.: Oxford, UK, 2000; ISBN 9780198505730.
41. Maier, R.S. The 192 solutions of the Heun equation. *Math. Comput.* **2006**, *76*, 811–843. [[CrossRef](#)]
42. Ronveaux, A. *Heun's Differential Equations*; Clarendon Press: Oxford, UK, 1995; ISBN 13978-0521687935.
43. Strutt, M.J.O. *Lamesche—Mathieausche—Und Verwandte Funktionen in Physik und Technik*; Chelsea Publishing Company: New York, NY, USA, 1967.
44. Meixner, J.; Schäfke, F.W. *Mathieusche Funktionen und Sphäroidfunktionen*; Springer: Berlin/Heidelberg, Germany, 1954. [[CrossRef](#)]
45. Arscott, F.M. *Periodic Differential Equations: An Introduction to Mathieu, Lamé, and Allied Functions*; Pergamon: London, UK, 2013.
46. McLachlan, N.W. *Theory and Applications of Mathieu Functions*; Oxford University Press Inc.: Oxford, UK, 1964.

47. Gourlay, A.R.; Mcguire, G.R. General Hopscotch Algorithm for the Numerical Solution of Partial Differential Equations. *IMA J. Appl. Math.* **1971**, *7*, 216–227. [[CrossRef](#)]
48. Hirsch, C. *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization*; Wiley: Hoboken, NJ, USA, 1988; Volume 1.
49. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **1934**, *49*, 1952. [[CrossRef](#)]
50. Barrett, R.; Berry, M.; Chan, T.F.; Demmel, J.; Donato, J.; Dongarra, J.; Eijkhout, V.; Pozo, R.; Romine, C.; Van Der Vorst, H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1994. [[CrossRef](#)]
51. Saad, Y.; Schultz, M.H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **1986**, *7*, 856–869. [[CrossRef](#)]
52. Askar, A.H.; Omle, I.; Kovács, E.; Majár, J. Testing Some Different Implementations of Heat Convection and Radiation in the Leapfrog-Hopscotch Algorithm. *Algorithms* **2022**, *15*, 400. [[CrossRef](#)]
53. Holman, J.P. *Heat Transfer*, 10th ed.; McGraw-Hill Educ.: New York, NY, USA, 2010; ISBN 0073529362.
54. Jalghaf, H.K.; Omle, I.; Kovács, E. A Comparative Study of Explicit and Stable Time Integration Schemes for Heat Conduction in an Insulated Wall. *Buildings* **2022**, *12*, 824. [[CrossRef](#)]
55. Omle, I.; Askar, A.H.; Kovács, E.; Bolló, B. Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges. *Energies* **2023**, *16*, 4604. [[CrossRef](#)]
56. Weather Online. Hungary Holiday Weather. 2023. Available online: <https://www.worldweatheronline.com/miskolc-weatherhistory/miskolc/hu.aspx> (accessed on 1 February 2023).
57. Duffie, J.A.; Beckman, W.A. *Solar Engineering of Thermal Processes*; Wiley: New York, NY, USA, 1980.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.