buildings

MDPI

Article

Comparison of Old and New Stable Explicit Methods for Heat Conduction, Convection, and Radiation in an Insulated Wall with Thermal Bridging

Humam Kareem Jalghaf [1,2], Endre Kovács [2,*] and Betti Bolló [1]

[1] Department of Fluid and Heat Engineering, University of Miskolc, 3515 Miskolc, Hungary
[2] Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary
* Correspondence: kendre01@gmail.com or fizendre@uni-miskolc.hu

Abstract: Using efficient methods to calculate heat transfer in building components is an important issue. In the current work, 14 numerical methods are examined to solve the heat transfer problem inside building walls. Not only heat conduction but convection and radiation are considered as well, in addition to heat generation. Five of the used methods are recently invented explicit algorithms, which are unconditionally stable for conduction problems. First, the algorithms are verified in a 1D case by comparing the results of the methods to an analytical solution. Then they are tested on real-life cases in the case of surface area (made of brick) and cross-sectional area (two-layer brick and insulator) walls with and without thermal bridging. Equidistant and non-equidistant grids are used as well. The goal was to determine how the errors depend on the properties of the materials, the mesh type, and the time step size. The results show that the best algorithms are typically the leapfrog-hopscotch and the modified Dufort–Frankel and odd–even hopscotch algorithms since they are quite accurate for larger time step sizes, even for 100 s as well.

Keywords: heat conduction; thermal insulation; thermal bridge; explicit time-integration; unconditionally stable numerical methods

check for updates

Citation: Jalghaf, H.K.; Kovács, E.; Bolló, B. Comparison of Old and New Stable Explicit Methods for Heat Conduction, Convection, and Radiation in an Insulated Wall with Thermal Bridging. *Buildings* **2022**, *12*, 1365. https://doi.org/10.3390/buildings12091365

Academic Editor: Gianpiero Evola

Received: 7 August 2022
Accepted: 29 August 2022
Published: 3 September 2022

1. Introduction

Energy management plays an important role in both energy generation and energy consumption, as well as in the struggle against global climate change by reducing CO_2 emissions. Climate change measures that tackle the problem by reducing greenhouse gas emissions are likely to be a major influence on global socio-economic development during the 21st Century. Changes in mean temperatures will affect several energy technologies. The most important impact is on fossil and nuclear thermal energy generation, in which higher ambient temperatures reduce the efficiency of thermal conversion (Carnot's rule) and, by heating up ambient water bodies, the efficiency of cooling. Higher mean temperatures also increase the evaporation rates in hydropower reservoirs, hence reducing the resource base for power generation. Increasing temperatures in permafrost regions can destabilize energy infrastructure such as wells and pipelines [1]. The energy control in the construction of new buildings dependent on renewable energy can make a positive difference for climate change and the economy.

There are many methods to increase the energy efficiency of buildings [2]. One of the most effective ways is to improve the thermal behavior of the building envelope by consuming the heat energy for domestic use or converting it to electricity by using solar cells in the exterior envelope and also reducing the heat loss through walls [3]. The actual space has different performances according to the position in the building, which can be the walls, roof, or floor. For optimization of this, one needs to efficiently calculate heat transfer. The heat transfer in building components can be calculated by using the general heat transfer equation, which depends on many parameters: most importantly, the material properties and boundary conditions. If a material with good thermal properties is used, a high heat transfer performance can be achieved. The boundary conditions mostly depend on the environment inside and outside of the building, and they are used as input parameters to calculate the heat through the building.

Most building heat transfer problems are multi-dimensional and transient. Moreover, the material properties, such as the specific heat and heat conductivity, can widely vary in the system [4] (p. 15). These make numerical computer simulations unavoidable. To efficiently calculate the heat flow through the building components, one needs to have a good numerical method that can handle the heat transfer parameters properly.

We are aware that plenty of numerical methods are used to solve heat transfer and similar problems, such as finite difference schemes (FDM) [5–7] and finite element methods (FEM) [8]. However, they can be extremely time-consuming since the examined system must be fully discretized both in space and time. Due to material inhomogeneities, the eigenvalues of the problem can have a very wide range (several orders of magnitude). In these cases, the problem is rather stiff, and the so-called CFL (Courant–Friedrichs–Lewy) threshold for the time step size is very small. When conventional explicit finite difference methods are applied to these problems, they will be unstable when the used time step size is larger than this small limit. That is why implicit methods, which have much better stability properties, are typically used for solving these kinds of equations; see, for example, [9–15]. They solve equation systems containing the whole system matrix; thus, they can use a lot of CPU time and computer memory, especially when the number of cells is large, which is always the case in three dimensions.

It is well known that the former rapid increase in CPU clock frequencies is over, and the tendency toward increasing parallelization in high-performance computing is very strong [16,17]. Thus, we think that time is on the side of the explicit methods because they can be much more straightforwardly parallelized. That is why we started to investigate explicit algorithms with improved stability properties, along with other scholars; see [18–26] for examples. These explicit methods can also serve as a basis for implicit methods, as in the paper [27].

Our research group recently constructed many explicit algorithms that are unconditionally stable for the heat conduction equation in arbitrary space dimensions [28–32]. In those original papers, the algorithms were tested using completely discontinuous random material parameters and initial conditions. It turned out that they can supply us with quite accurate results, and they are orders of magnitude faster than the widely used MATLAB 'ode' routines. Then, in our last paper [33], we examined 13 methods for heat conduction problems in an insulated wall. This current paper can be considered as a continuation of that paper and contains novelties from several points of view. Most importantly, here we consider not only conduction but convection and radiation as well. This latter one is nonlinear; thus, now we have a much more difficult problem. We have to adapt the algorithms to the conduction–convection–radiation equation, which, however, can be performed in more than one different way. In the case of the CNe-type methods, these ways are so nontrivial that we omitted them from this publication and included other methods instead, such as the explicit Euler scheme. Note that, apart from our works, no comparative study has been performed even about the known explicit and stable methods examined in this paper, namely the UPFD, Dufort–Frankel (DF), odd–even hopscotch, and rational Runge–Kutta methods (for the definitions, see the next section). For example, Gasparin et al. [34] made a comparison of the DF scheme with the Explicit (Euler), Crank–Nicolson, and hyperbolization schemes to simulate moisture transfer in porous materials. Then they investigate the DF, the implicit and the explicit Euler, as well as the alternating direction implicit methods to simulate heat transfer in building envelopes [26], but they do not even mention that other explicit and stable schemes exist. Their results also suggest that stable explicit methods (DF in their case) can be proposed to solve these problems. The goal of this paper is to persuade the scientific community of this by the examination of how the performance of these methods changes by varying some parameters of the physical system and of the mesh and which of them can be proposed under different circumstances. In this paper, the tests are performed in the building walls containing a metallic thermal bridge, which is also a novelty compared to our previous paper.

The outline of the paper is as follows:

Section 2 Problem statement, discretization and time integration methods.
Section 2.1 The studied system with the equations, discretization for spatially uniform and non-uniform cases.
Section 2.2 The studied numerical algorithms.
Section 3 Verification in one space dimension.
Section 3.1 Weak nonlinearity with the domination of convection.
Section 3.2 Strong nonlinearity due to large temperatures.
Section 4 Preliminaries for the simulation of the wall: materials, mesh construction, initial and boundary conditions.
Section 5 Results of the wall simulation.
Section 5.1 Surface of the wall.
Section 5.2 Cross-section of a brick wall with insulation, equidistant and non-equidistant mesh.
Section 5.3 Cross-section of the insulated wall with a thermal bridge, equidistant and non-equidistant mesh.
Section 6 Discussion, summary and future plans.

2. The Studied Problem and the Methods

2.1. The Equation and Its Spatial Discretization

The following linear parabolic PDE describes the phenomenon of the simplest Fourier-type conduction of heat in a homogeneous medium with a heat source:

$$\frac{\partial u}{\partial t} = \alpha \, \nabla^2 u + q, \tag{1}$$

where $u = u(\vec{r}, t)$ is the temperature, α is the thermal diffusivity, and q is the heat source or heat generation. Based on Newton's law of cooling, convective heat transfer can be taken into account [35] by a term $K(u_a - u)$, where the ambient temperature u_a may be considered as it does not depend on u, and thus can obviously be included into the heat source term q. It means that a term $-Ku + q$ can represent heat convection. In Section 4, it will be explained how this $K = K(\vec{r})$ coefficient is connected to the usual convection heat transfer coefficient h usually used in building energetics.

On the other hand, based on the Stefan–Boltzmann law [36] (Chapter 8), the radiative heat loss of a surface can be expressed by a term $-\sigma u^4$, where the proportionality constant σ is now the product of the surface area and the Stefan–Boltzmann constant, all of which are nonnegative quantities. The incoming radiation, which may include direct sunlight, can be similarly incorporated into the source term q as the Ku_a term above. Therefore, one can extend the heat conduction Equation (1) to include the heat generation source, convection, and radiation terms as follows:

$$\frac{\partial u}{\partial t} = \alpha \, \nabla^2 u + q - K \cdot u - \sigma \cdot u^4. \tag{2}$$

If the material properties such as the heat conductivity depend on space, one has to use an even more general equation

$$\frac{\partial u}{\partial t} = \frac{1}{c\rho} \nabla(k \nabla u) + q - K \cdot u - \sigma \cdot u^4, \tag{3}$$

where $k = k(\vec{r}, t)$, $c = c(\vec{r}, t)$ and $\rho = \rho(\vec{r}, t)$ are the heat conductivity, specific heat and the mass density, respectively. Let us assume that the c and ρ functions are positive and the $\alpha = k/(c\rho)$ equation holds.

In the case of Equation (2) in one dimension, the most standard central difference formula

$$\frac{\partial^2}{\partial x^2}u(x_i) \approx \frac{\frac{u(x_{i+1})-u(x_i)}{\Delta x}+\frac{u(x_{i-1})-u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i-1}-2u_i+u_{i+1}}{\Delta x^2},$$

is applied to the $\alpha\ \nabla^2 u$ term, which is second-order in Δx, $i = 1\ ,\ \ldots, N$, where N is the total number of nodes. With this, we obtain the following spatially discretized form of Equation (2) in one space dimension:

$$\frac{du_i}{dt} = \alpha\ \frac{u_{i-1}-2u_i+u_{i+1}}{\Delta x^2} + q - Ku_i - \sigma {u_i}^4. \tag{4}$$

Let us now present the similar discretization of Equation (3). This procedure is described in more detail in Chapter 5 of the book [37], for the case of underground reservoirs. Let us suppose that the k, c, and ρ quantities are functions of the space variables instead of being constants. Now if Equation (3) is discretized using a one-dimensional, equidistant grid, we have

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \frac{1}{c(x_i)\rho(x_i)\Delta x}\left[k\left(x_i\ +\frac{\Delta x}{2}\right)\frac{u(x_i+\Delta x)-u(x_i)}{\Delta x} + k\left(\ x_i\ -\frac{\Delta x}{2}\right)\frac{u(x_i-\Delta x)-u(x_i)}{\Delta x}\right].$$

Now we switch from node to cell variables, which means that u_i, ρ_i, and c_i is going to be the (average) temperature, density, and specific heat of cell i, respectively, while $k_{i,i+1}$ is the heat conductivity between the cell labelled by i and its right-hand side neighbor. Now the spatially discretized form of Equation (3) will have the form

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x}\left(k_{i,i+1}\frac{u_{i+1}-u_i}{\Delta x} + k_{i-1,i}\frac{u_{i-1}-u_i}{\Delta x}\right)\ + q - Ku_i - \sigma {u_i}^4$$

We generalize this for a non-equidistant grid with a non-uniform cross section, which is still one-dimensional. If the length and the (average) cross section of the cell is denoted by $\Delta\ x_i$ and A_i, then the distance between the center of the cell i and its neighbor j can be expressed as $d_{ij} = \left(\Delta x_i + \Delta x_j\right)/2$. The area of the interface between cell i and its right neighbor is approximated as $A_{i,i\pm1} \approx A_i \approx A_{i\pm1}$. Now, we have

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x_i A_i}\left(A_{i,i+1}k_{i,i+1}\frac{u_{i+1}-u_i}{d_{i,i+1}} + A_{i,i-1}k_{i,i-1}\frac{u_{i-1}-u_i}{d_{i,i-1}}\right) + q - Ku_i - \sigma {u_i}^4.$$

The volume and the heat capacity of the cell is calculated as $V_i = A_i\ \Delta x_i$ and $C_i = c_i\rho_i V_i$, respectively. We introduce the thermal resistance between these neighboring cells, which is estimated as $R_{ij} \approx d_{ij}/k_{ij}A_{ij}$. With these quantities, the spatially discretized form of Equation (3) can be written as follows:

$$\frac{du_i}{dt} = \frac{u_{i-1}-u_i}{R_{i-1,i}C_i} + \frac{u_{i+1}-u_i}{R_{i+1,i}C_i} + q - Ku_i - \sigma {u_i}^4. \tag{5}$$

This can be straightforwardly generalized further, thus we have the following system of ordinary differential equations (ODEs) for the time derivative of each cell variable with an arbitrary number of neighbors:

$$\frac{du_i}{dt} = \sum_{j\neq i}\frac{u_j-u_i}{R_{i,j}C_i} + q - Ku_i - \sigma {u_i}^4. \tag{6}$$

This ODE system can be used in the case of arbitrary (e.g., unstructured) grids, which may contain cells of various shapes and properties. Of course, irregular discretization can decrease the spatial accuracy, but in this paper, only rectangle-shaped cells are used.

2.2. The Applied Numerical Methods

We present the algorithms adapted for Equations (2) and (3), which have the spatially discretized form (4) and (6), respectively. First, their formula is given for the simplest case, i.e., one dimensional, equidistant mesh, Equation (4), which is used for verification. The more general forms, which can be applied to (6) are immediately given because these will be used to simulate the heat transfer in the wall.

Only equidistant temporal discretization is used in this paper with time step size Δt, and u_i^n denotes the temperature of cell i at the time $n\Delta t$. In the case of Equation (4), the usual expression for the mesh-ratio $r = \frac{\alpha \Delta t}{\Delta x^2}$ is used for the 1D equidistant mesh. On the other hand, for the general discretization (6), the following quantities are introduced

$$r_i = \Delta t \sum_{j \neq i} \frac{1}{C_i R_{ij}} \quad \text{and} \quad A_i = \Delta t \sum_{j \neq i} \frac{u_j^n}{C_i R_{ij}} + \Delta t \cdot q_i. \tag{7}$$

The first quantity is the generalization of r, and more precisely, for Equation (4), the $r_i = 2r$ relation holds. The second quantity is an aggregated quantity, which mediates the temperature of the neighbors of cell i.

1. The most widespread explicit algorithm to solve the heat conduction equation is the FTCS (forward time central space) scheme, in which the time integration is based on the explicit Euler method. Now this is adapted to our case in the most standard way; thus, the special and the general formulas are the following:

$$u_i^{n+1} = (1-2r)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q_i - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4,$$
$$\text{and } u_i^{n+1} = (1-r_i)u_i^n + A_i - \Delta t \cdot K_i \cdot u_i^n - \Delta t \cdot \sigma_i \cdot \left(u_i^n\right)^4.$$

2. We make an attempt to modify the Explicit Euler method using a trick, which is similar to the one used in the case of the UPFD or the pseudo-implicit method (see later). It means that during discretization, the convection and the radiation terms are taken into account fully or partly at the new time level, thus with a nonlocality in time. It is worth noting that non-standard discretization sometimes means nonlocality in space [38]. In our case, the time-discretized equation will be the following:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + q_i - K \cdot u_i^{n+1} - \sigma \cdot u_i^{n+1}\left(u_i^n\right)^3$$

   With this, we obtain the new explicit formulas as follows

$$u_i^{n+1} = \frac{(1-2r)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q_i}{1 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^n\right)^3}, \tag{8}$$

   and similarly, for the general case

$$u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i}{1 + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot \left(u_i^n\right)^3}$$

   Since now the convection and the radiation terms are present in the denominator, they can hardly yield numerical instability even for very large values of the temperature. In the current work, we call this scheme the Non-Standard Explicit Euler Method (NS-Exp).

3. Heun's method, sometimes called the explicit trapezoidal rule, is probably the most common second-order Runge–Kutta (RK) scheme for ODEs and ODE systems [39], so it is straightforward to use it as a component of the method of lines. It starts with a

predictor step, which is an explicit Euler stage. In the cases of Equations (4) and (6), it has the form:

$$u_i^{\text{pred}} = (1-2r)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4,$$
$$\text{and } u_i^{\text{pred}} = (1-r_i)u_i^n + A_i - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4.$$

Now the corrector-step follows, which uses the average of the obtained and the old values of the u variable:

$$u_i^{n+1} = u_i^n - r\left(u_i^n + u_i^{\text{pred}}\right) + r\frac{u_{i-1}^n + u_{i-1}^{\text{pred}} + u_{i+1}^n + u_{i+1}^{\text{pred}}}{2} + \frac{\Delta t}{2} \cdot \left((q + q^{new}) - K \cdot \left(u_i^n + u_i^{\text{pred}}\right) - \sigma \cdot \left(u_i^n + u_i^{\text{pred}}\right)^4\right),$$

and

$$u_i^{n+1} = u_i^n - r_i\frac{u_i^n + u_i^{\text{pred}}}{2} + \frac{\Delta t}{2}\left(A_i + A_i^{new} - K \cdot \left(u_i^n + u_i^{\text{pred}}\right) - \sigma \cdot \left(u_i^n + u_i^{\text{pred}}\right)^4\right),$$

where $A_i^{new} = \Delta t \sum\limits_{j \neq i} \frac{u_j^{\text{pred}}}{C_i R_{ij}} + \Delta t \cdot q_i$.

4. The UPFD method was constructed by Chen-Charpentier and Kojouharov [40] for the linear diffusion–advection–reaction equation. It is actually a witty and non-standard combination of the explicit and implicit Euler-discretizations, where only the actual node is treated implicitly and the neighbors explicitly as follows:

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

Recently we adapted it to the case of Equations (2) and (3); see Algorithm 2 in [32]. In the case of Equation (1), it reads as follows:

$$u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q_i}{1 + 2r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^n\right)^3}, \tag{9}$$

and the general form for Equation (3) is:

$$u_i^{n+1} = \frac{u_i^n + A_i}{1 + r_i + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot \left(u_i^n\right)^3}. \tag{10}$$

One can see that the convection and the radiation terms are treated similarly as in Equation (8).

5. The Dufort–Frankel (DF) algorithm is obtained from the so-called leapfrog explicit scheme by a modification; see p. 313 in [41]. It is a known but non-traditional explicit scheme, which is unconditionally stable for the linear heat equation. Now the formula for the case of Equations (4) and (6) are as follows:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q - 2\Delta t \cdot K \cdot u_i^n - 2\Delta t \cdot \sigma \cdot \left(u_i^n\right)^4}{1+2r},$$
$$\text{and } u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - 2 \cdot \Delta t \cdot K \cdot u_i^n - 2 \cdot \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4}{1+r_i}.$$

One can see that the formulas contain u_i^{n-1}; thus, it is a two-step but one-stage method. Since it is not self-starter, another method must be used to start the DF method by the calculation of u_i^1. For this purpose, we apply the UPFD Formulas (9) and (10).

6. We make an attempt to modify this DF method using the non-standard trick as in Equation (8) to deal with convection and the radiation terms. With this, these terms pop up in the denominator as follows;

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q}{1 + 2r + 2\Delta t \cdot K + 2\Delta t \cdot \sigma \cdot \left(u_i^n\right)^3}$$

$$\text{and } u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1 + r_i + 2 \cdot \Delta t \cdot K + 2 \cdot \Delta t \cdot \sigma \cdot \left(u_i^n\right)^3}$$

7. From the family of the Rational Runge–Kutta (RRK) methods, we chose a two-stage version [42] with the following definition. In the first stage, a full step is taken by the explicit Euler (FTCS) scheme to obtain the predictor value. The increment for Equation (4) is calculated as

$$g_i^1 = r\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + \Delta t \cdot q - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4,$$

and for Equation (6):

$$g_i^1 = -ru_i^n + A_i - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot \left(u_i^n\right)^4$$

Using these g_i^1 values, the predictor values can be obtained for all grid types as

$$u_i^{\text{pred}} = u_i^n + g_i^1.$$

After this, using the predictor values obtained above, the increment of a second Euler-step is calculated:

$$g_i^2 = r\left(u_{i-1}^{\text{pred}} - 2u_i^{\text{pred}} + u_{i+1}^{\text{pred}}\right) + \Delta t \cdot q - \Delta t \cdot K \cdot u_i^{\text{pred}} - \Delta t \cdot \sigma \cdot \left(u_i^{\text{pred}}\right)^4,$$
$$\text{and } g_i^2 = -r_i u_i^{\text{pred}} + A_i^{new} - \Delta t \cdot K \cdot u_i^{\text{pred}} - \Delta t \cdot \sigma \cdot \left(u_i^{\text{pred}}\right)^4.$$

Now one needs to calculate the following scalar products

$$p_1 = \left(\vec{g}^1, \vec{g}^1\right) = \sum_{i=1}^{N} g_i^1 g_i^1, \quad p_{12} = \left(\vec{g}^1, \vec{g}^2\right) = \sum_{i=1}^{N} g_i^1 g_i^2, \quad p_2 = \left(\vec{g}^2, \vec{g}^2\right) = \sum_{i=1}^{N} g_i^2 g_i^2,$$

and with them, one obtains the final expression for the new values of the variable:

$$u_i^{n+1} = u_i^n + \frac{2p_1 g_i^1 - 2p_{12} g_i^1 + p_1 g_i^2}{4p_1 - 4p_{12} + p_2}.$$

8. The pseudo-implicit (PI) method is Algorithm 5 from [32] with parameter $\lambda = 1$. For Equation (4), the following two-stage algorithm is applied:

$$\text{Stage 1}: \ u_i^{\text{pred}} = \frac{u_i^n + {}^{r}\!/_{2}\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q}{1 + r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^n\right)^3}.$$

$$\text{Stage 2}: \ u_i^{n+1} = \frac{(1-r)u_i^n + r\left(u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}\right) + \Delta t \cdot q - \Delta t \cdot K \cdot \left(u_i^{\text{pred}} - u_i^n\right)}{1 + r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{\text{pred}}\right)^2 \cdot u_i^n}.$$

For a general grid, the two stages can be written as

$$\text{Stage 1}: \; u_i^{\text{pred}} = \frac{u_i^n + A_i}{1 + r_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^n)^3}, \text{ where } A_i^{new} = \Delta t \sum_{j \neq i} \frac{u_j^{\text{pred}}}{C_i R_{ij}} + \Delta t \, q_i$$

$$\text{Stage 2}: \; u_i^{n+1} = \frac{(1 - r_i) u_i^n + A_i^{new}}{1 + r_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{\text{pred}}\right)^2 \cdot u_i^n}, \text{ where } A_i^{new} = \Delta t \sum_{j \neq i} \frac{u_j^{\text{pred}}}{C_i R_{ij}} + \Delta t \, q_i$$

One can see that this algorithm is fully explicit, and the convection and the radiation term are treated in a quite sophisticated way at the second stage since both the u_i^n and the u_i^{pred} values are used.

9. To use an odd–even hopscotch method, a special, so-called bipartite spatial grid is necessary, where the cells are labeled as odd and even, and similarly to a checkerboard, all the nearest neighbors of the odd cells are even and vice versa. The odd–even labels are interchanged in each time step, as is shown in Figure 1A. Originally, the standard explicit Euler and implicit Euler formula were applied in the first and second stages, respectively [43]. The formulas for the special and the general cases are the following: wing two-stage algorithm is ap

$$\begin{aligned} \text{Explicit Euler}: \; & u_i^{n+1} = (1 - 2r) u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q_i - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot (u_i^n)^4 \\ & \text{and } u_i^{n+1} = (1 - r_i) u_i^n + A_i - \Delta t \cdot K_i \cdot u_i^n - \Delta t \cdot \sigma_i \cdot (u_i^n)^4 \end{aligned} \tag{11}$$

$$\text{Implicit Euler}: \; u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right) + \Delta t \cdot q_i}{1 + 2r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^n)^3} \text{ and } u_i^{n+1} = \frac{u_i^n + A_i^{new}}{1 + r_i + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot (u_i^n)^3},$$

where $A_i^{new} = \Delta t \sum_{j \neq i} \frac{u_j^{n+1}}{C_i R_{ij}} + \Delta t \cdot q_i$. Note that the implicit formula is effectively explicit since the u_j^{n+1} values have been just obtained at Stage 1. We call this version the original odd–even hopscotch (OOEH) method. However, since there is a $1 - 2r$ factor in (12), the new temperatures can be negative for large r, which can cause unstable behavior for large time step sizes due to the possibly large negative value of the term $(u_i^n)^3$ in the denominator. To avoid this, we apply a simple trick of forbidding negative values by the following simple conditional statement:

$$\text{if } u_i^{n+1} < 0 \text{ then } u_i^{n+1} = 0. \tag{12}$$

This trick will be applied in all cases in this method and the remaining methods when there is a possibility of negative temperatures.

10. We also make an attempt to modify the first stage of OOEH (Explicit Euler) as in Equation (8). With this, the convection and the radiation terms pop up in the denominator; thus, the first stage with the explicit Euler formula with condition (12) is as follows:

$$u_i^{n+1} = \frac{(1 - 2r) u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t \cdot q_i}{1 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^n)^3} \text{ and } u_i^{n+1} = \frac{(1 - r_i) u_i^n + A_i}{1 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^n)^3} \tag{13}$$

The second stage with the "implicit" Euler scheme:

$$u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right) + \Delta t \cdot q_i}{1 + 2r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^n)^3} \text{ and } u_i^{n+1} = \frac{u_i^n + A_i^{new}}{1 + r_i + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot (u_i^n)^3}.$$

11. The reversed odd–even hopscotch method (ROEH) is different from the OOEH method because it applies the formulas in the opposite order: first the implicit Euler,

then the non-standard explicit Euler formulas (13), with condition (12). However, when first-stage calculations begin with the implicit formula, the new values of the neighbors are not known. In the ROEH method, they are taken into account at the old-time level, which is the same trick as the UPFD method uses; see Formulas (9) and (10). If one changes the order of the two formulas in the code of the original NS-OEH, one immediately obtains the code of this method. It is shown that although this method is not very accurate for an equidistant mesh, it yields much smaller errors in the case of extremely stiff systems than the OOEH method [44].

12. The leapfrog-hopscotch (LH) method [31] has a structure consisting of two half and several full time steps, as one can see in Figure 1B. In the first stage (yellow box in the figure), the following special and general formulas are used:

$$u_i^{1/2} = \frac{2u_i^0 + A_i}{2 + r_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^0\right)^3} \text{ and } u_i^{1/2} = \frac{2u_i^0 + A_i}{2 + r_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^0\right)^3} \tag{14}$$

Then, for the even and odd nodes, full-time steps (denoted by green boxes in the figure) are taken strictly alternately with the formulas

$$u_i^{\mu+1} = \frac{(1-r)u_i^{\mu} + r\left(u_{i-1}^{\mu+1/2} + u_{i+1}^{\mu+1/2}\right) + \Delta t \cdot q}{1 + r + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{\mu}\right)^3} \text{ and } u_i^{\mu+1} = \frac{(1 - r_i/2)u_i^{\mu} + A_i^{\mu+1/2}}{1 + r_i/2 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{\mu}\right)^3} \tag{15}$$

with condition (12). The upper index μ is n for the even nodes and $n+1$ for the odd nodes. It is important that always the latest available values of the neighbors are used (for example, in $A_i^{\mu+1/2}$) when the new values of u are calculated, regardless of the size of the time step. This alternation goes on until the end of the last timestep (purple box in Figure 1B), where (15) is used again, but with a halved time step size, in order to reach the same final time point as the even nodes.

13. In the shifted-hopscotch (SH) method [30], the repeating block consists of five stages, two of them are half, and three of them are full-time steps. These altogether span two integer time steps for the odd and the even cells as well, as one can see in Figure 1C. The first half-sized time step is taken for the odd cells with Formula (14), which is symbolized by a yellow box with the number 1 in the figure. Then, a full-time step with Formula (15), with condition (12) for the even, the odd, and the even cells follows again, which are symbolized by green boxes with the numbers 2, 3, and 4 in the figure. Finally, a half-length time step (pink box with number 5 inside) for the odd cells closes the calculation with the formula

$$u_i^{n+2} = \frac{(2-2r)u_i^{n+1} + r\left(u_{i-1}^{n+3/2} + u_{i+1}^{n+3/2}\right) + \Delta t \cdot q}{2 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{n+1}\right)^3} \text{ and } u_i^{n+2} = \frac{(1 - r_i)u_i^{n+1} + A_i^{n+3/2}/2}{1 + \Delta t \cdot K/2 + \Delta t \cdot \sigma \cdot \left(u_i^{n+1}\right)^3/2} \tag{16}$$

with condition (12) again.

14. The ASH or Asymmetric Hopscotch Method is very similar to the SH method but contains fewer integer stages, thus using three stages instead of five (see Figure 1D). The calculation starts with a half-time step size for the odd cell with (14). Then a full-time step is coming for the even cell with formula (15) and condition (12), and finally, a half-time step size with (16), again with condition (12) for the last odd cell closes the calculation of the values.

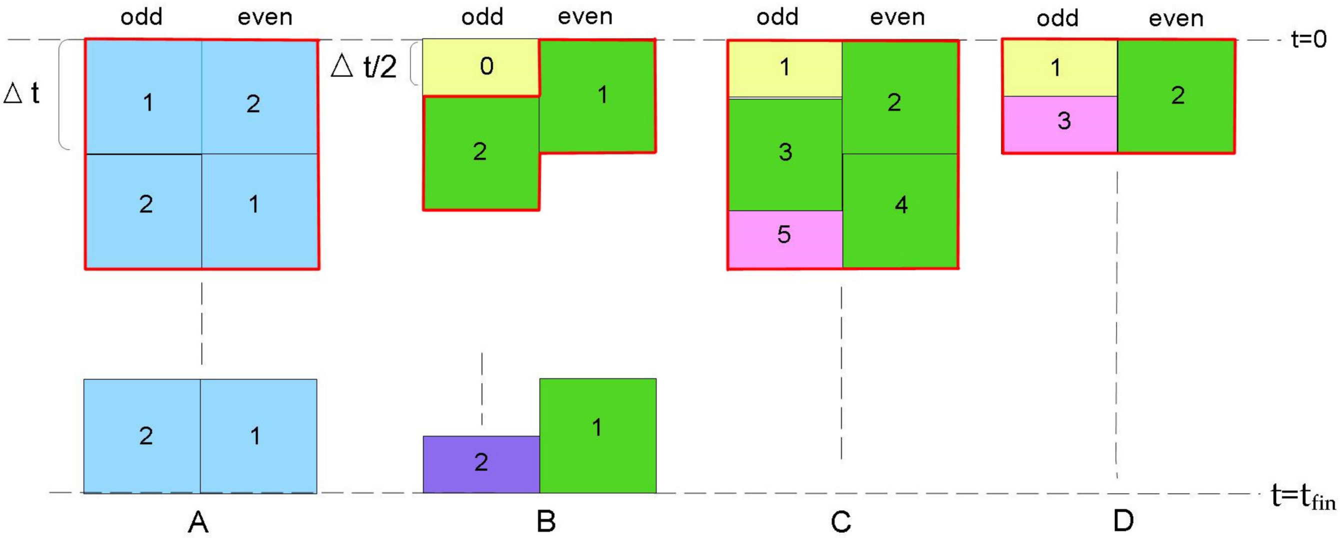

Figure 1. Space–time structure of (**A**) the original hopscotch and the reversed hopscotch methods, (**B**) the shifted-hopscotch method, (**C**) the leapfrog-hopscotch method, (**D**) the asymmetric hopscotch method.

Remark 1. In our original publications, most of these methods are tested to the heat conduction Equation (1) with or without the source term. Only the UPFD and the PI method were defined by Equation (3). With the exception of the ASH scheme, the stability and convergence properties are analytically proven for Equation (1), which are in complete agreement with the numerical results. The explicit Euler and UPFD methods are first-order in time step size, while all other methods are second-order. All of these methods are unconditionally stable for the linear heat conduction equation, except of course Euler's and Heun's methods. It means that the temperature remains finite (i.e., errors are not amplified without bounds) for an arbitrary large time step size, i.e., the previously mentioned CFL limit is not valid for them. However, this is not equivalent to them always being accurate. In fact, the price one has to pay for unconditional stability is that consistency is only conditional. This means that decreasing space step size with a constant time step size actually yields (slowly) decreasing accuracy (but not worsening stability properties as in the case of the traditional explicit methods). These are examined analytically as well as numerically in our previous papers, especially in [45], e.g., for very small time step sizes, the explicit Euler scheme is more accurate than the UPDF method, while Heun's method is often more accurate than, e.g., the PI method. For some examples, see Section 4 in this paper. It must be underlined again that only very few explicit methods have unconditional stability even for the simple heat equation, e.g., explicit Runge–Kutta methods are never A-stable [46] (p. 60). Furthermore, in our original papers, more recently in [33], several tests have been performed where the running times were measured. We always obtained that the running time of these algorithms is approximately proportional to the number of stages. We think that the new terms, e.g., the radiation term, do not significantly affect this proportionality, but the comprehensive running-time measurements are planned in our next paper.

In Table 1, we enlist the methods and their abbreviations. The last two columns summarize the information on whether the method is old or recently elaborated by our research group for the heat/diffusion equation, and whether it is unconditionally stable for Equation (1) or not.

Table 1. List of the methods and their abbreviations.

Abbrev.	**Name of the Method**	**Recent for Heat Conduction**	**Stable for Heat Conduction**
ExpE	Explicit Euler	no	no
NS-ExpE	Explicit Euler with non-standard treatment of convection and radiation	no	no
Heun	Heun, i.e., explicit trapezoidal	no	no
UPFD	Unconditionally positive finite difference	no	yes
DF	Dufort–Frankel	no	yes
NS-DF	Dufort–Frankel with non-standard treatment of convection and radiation	no	yes
RRK	Rational Runge–Kutta	no	yes
PI	Pseudo-implicit	yes	yes
OOEH	Original odd–even hopscotch	no	yes
NS-OEH	Original odd–even hopscotch with non-standard treatment of convection and radiation	no	yes
RH	Reversed hopscotch	yes	yes
LH	Leapfrog-hopscotch	yes	yes
SH	Shifted hopscotch	yes	yes
ASH	Asymmetric hopscotch	yes	yes

Remark 2. Most of the methods have never been applied to Equation (2). Here, we try to extensively test them, but the analytical investigation of them is out of the scope of this paper.

3. Verification Using a 1D Analytical Solution

In this section, our goal is to verify the methods from the mathematical point of view, thus units for the quantities are not used. The used analytical solution [32] of Equation (2) is the following:

$$u^{\text{exact}}(x,t) = te^{\,x-\,t}, \tag{17}$$

which is valid if $\alpha = 1\,,\ \ K = 2$ and heat source function is $q(x,t) = \sigma t^4 e^{4x-4t} + e^{x-t}$. The initial condition and the Dirichlet boundary conditions are obtained simply by substituting the initial t and boundary x values to the analytical solution, respectively, and we fix $\sigma = 10^{-7}$. The (global) numerical error is the absolute difference of the numerical solutions u_j^{num} of the examined method and the reference solution u_j^{ref} (which is the analytical solution here) at final time t^{fin}. We use these node- or cell-errors of the nodes or cells to calculate the maximum (also called L_∞) error:

$$\text{Error}(L_\infty) = \max_{1\le j\le N}\left|u_j^{\text{ref}}(t^{\text{fin}}) - u_j^{\text{num}}(t^{\text{fin}})\right|. \tag{18}$$

3.1. Weak Nonlinearity

Here, the analytical solution is reproduced numerically in the $(t,x) \in [1,\ \ 2] \times [0,\ \ 4]$ computational domain. The L_∞ errors as a function of the time step size Δt can be seen in Figure 2 in a log–log diagram for $\Delta x = 0.01$. The last curve, a straight dashed line is proportional to Δt^2, in order to help to determine the slope of the error curves, which gives the order of temporal convergence of the schemes. One can see most methods are second-order, as we already mentioned. The algorithms are verified since the errors tend to a small number much below 10^{-4} for decreasing time step sizes (see the left side of the figure). This residual error cannot be avoided since it is due to the discretization of the space, which is kept constant here. On the right-hand side of the figure, very large time step sizes are depicted. With this, our goal is to demonstrate that these methods have very good stability properties, i.e., their error is limited even for these large step sizes. In Figure 3, the initial function u^0, and the final analytical solution u^{exact} is plotted with two numerical results at the final time as well. One can see that the numerical solutions are as smooth as the analytical one, no unphysical oscillations appear.

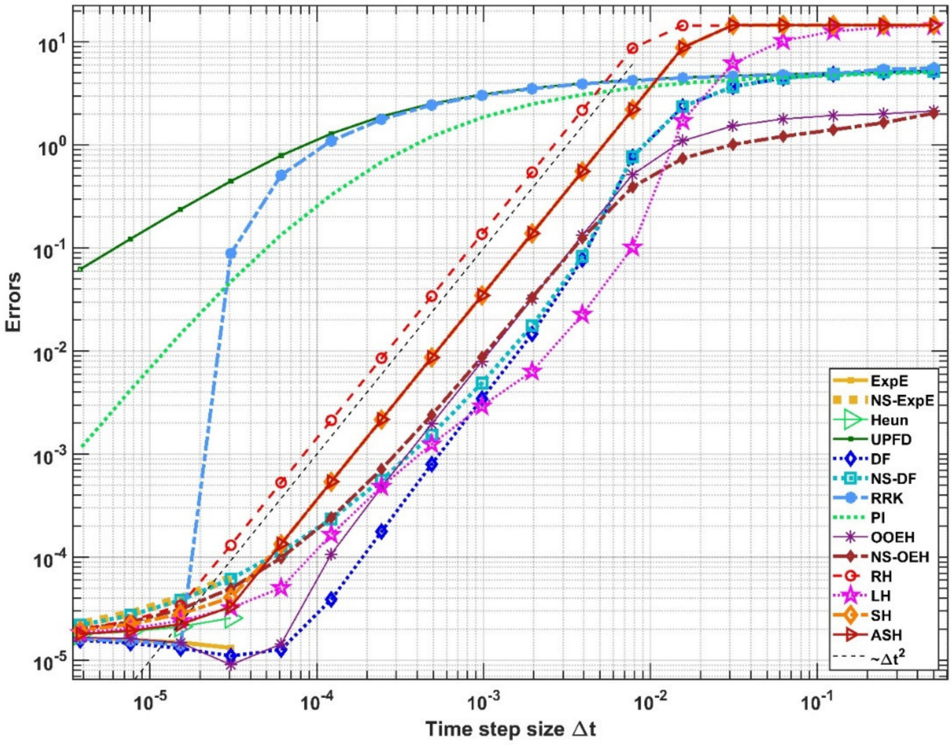

Figure 2. The L_∞ errors as a function of the time step size Δt for the 14 numerical schemes of Equation (2). The thin black dashed line is proportional to Δt^2.

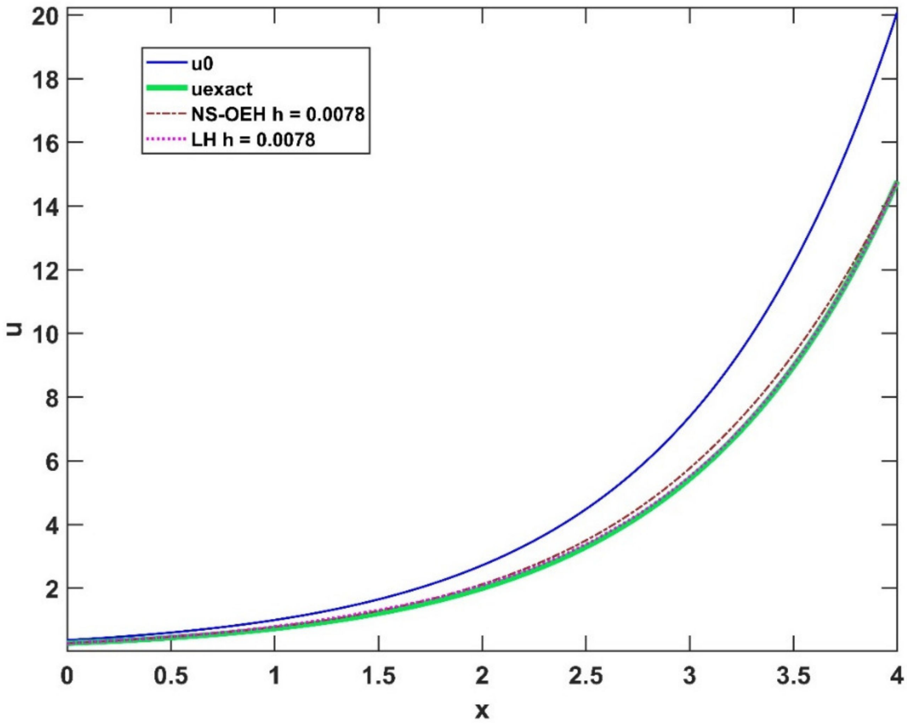

Figure 3. The temperature values as a function of the x variable in the case of the initial function u^0, the analytical solution at $t^{\text{fin}} = 2$, the NS-OEH method and the LH method for $\Delta t = \ 7.8 \cdot 10^{-3}$.

3.2. Strong Nonlinearity

Here the $(t, x) \in [5,\ 6] \times [6,\ 10]$ computational domain is considered with the same parameters as in the previous subsection. The initial function has a very similar exponential curve as in Figure 3. However, now the largest initial temperature is 742 (instead of 20), thus the nonlinear term σu^4 can have values above $3 \cdot 10^4$, which can be considered as a very strong nonlinearity. The L_∞ errors as a function of the time step size Δt can be seen in Figure 4 for $\Delta x = 0.01$. The OOEH algorithm now yields a very large error for large time step size, namely $1.8 \cdot 10^5$ in spite of condition (12). The (original) DF method (but not the NS-DF method) also has some accuracy problems but at medium time step sizes. For larger values of the temperature (i.e., for stronger nonlinearity), they would become unstable.

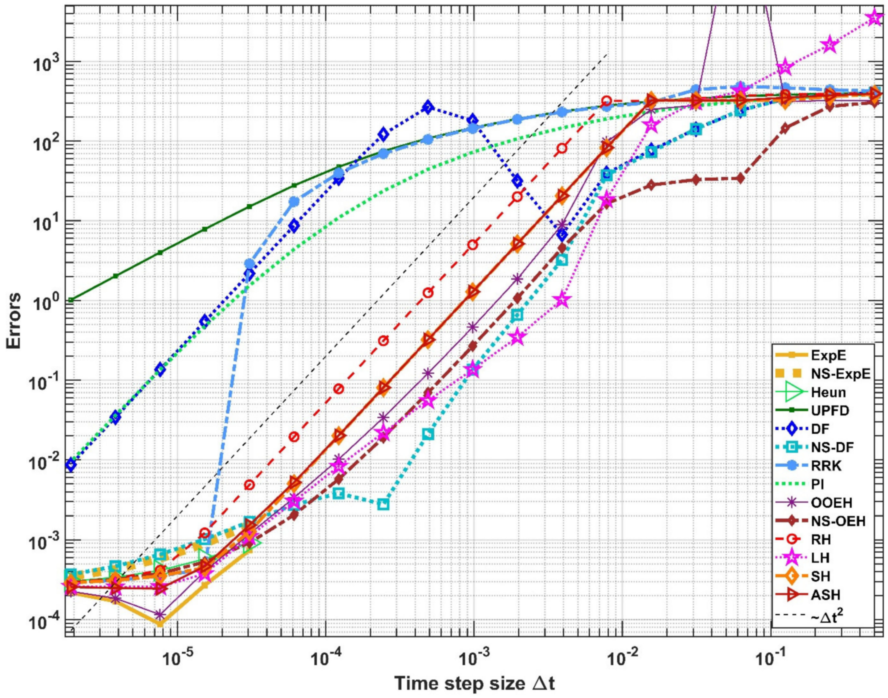

Figure 4. The L_∞ errors as a function of the time step size Δt for the numerical solutions of Equation (2).

4. Preliminaries for the Simulation of the Wall

4.1. Materials and Structures

In the present work, real material properties are taken into account. For the conduction term, they are listed in Table 2. Note that inside a material, these coefficients are constants, i.e., they do not depend on time, space or temperature, but at the boundary of the materials, they have a discontinuity.

Table 2. The properties of the used materials [36].

	ρ (kg·m^{-3})	k(W·m^{-1}·K^{-1})	c(J·kg^{-1}·K^{-1})
Brick	1600	0.73	800
Glass wool	200	0.03	800
Steel structure	7800	16.2	840

As one can see in Figure 5, the following cases are considered:

(A) Surface of the wall made of brick only.
(B) Two-layer cross-section of a wall consisting of brick and glass wool insulator.
(C) The same two-layer cross-section with a steel structure thermal bridge.

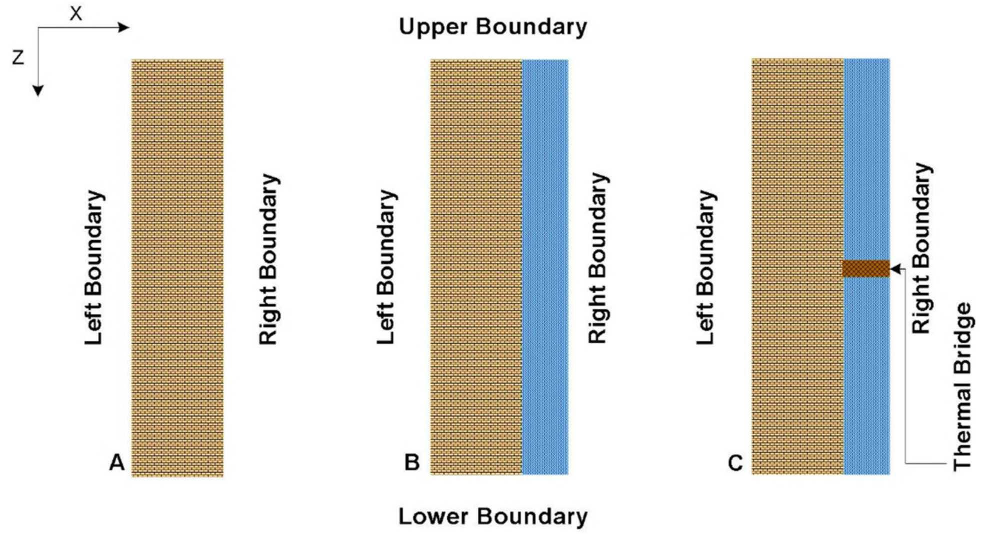

Figure 5. (**A**) One-layer wall, (**B**) wall with insulator, and (**C**) wall with insulator and thermal bridge.

4.2. Mesh Construction

In this subsection, a wall segment with a volume $1 \times 1 \times 1$ m is considered. The y direction is perpendicular to the surface of Figures 5 and 6. In the current approximation, physical quantities are not changing in the y-direction, thus that dimension is irrelevant. From the mathematical point of view, they are two-dimensional problems, and thus, we use $\Delta y_i = 1$ m for simplicity. For the other two coordinates, $(x, z) \in [0,\ 1] \times [0,\ 1]$; thus, the total area of the meshes is 1 m^2. Two types of meshes have been constructed, namely an equidistant mesh with square-shaped cells and a non-equidistant mesh with rectangular cells. The number of the cells along axis x and z axis are set to N_{x} = 100 and N_{z} = 100, thus, we have a grid with a total cell number $N = N_x N_z = 10,000$. In the non-equidistant case, there are wide cells on the left side and narrow ones on the right side of the wall. The width decreases gradually following a geometric series. As it is well known, for $r \neq 1$ the sum of the first n + 1 terms $(n = N_x - 1)$ of a geometric series, up to the term r^n, is

$$a + ar + ar^2 + ar^3 + \ldots\ldots + ar^n = \sum_{k=0}^{n} ar^k = a\left(\frac{1 - r^{n+1}}{1 - r}\right).$$

We used r = 0.98, and a = 0.0234, which gives $\Delta x_1 = 0.0234$ on the left side and $\Delta x_{N_x} = 0.98^{99} \cdot \Delta x_1 = 0.00317$ on the right side. In the z-direction, we used the constant grid element height (Δz = 0.01). The obtained meshes are presented in Figure 6.

We apply an equidistant grid in the case of the surface of the wall and equidistant and non-equidistant grids to the cross-section of the wall with an insulator. In the cross-section case, the left 50% of the cells are always brick, and the right 50% are insulators for programming simplicity. It means that the volume of the brick and the insulator is the same in the equidistant case, but, if there is a gradual change in the x-direction, the thickness of the insulator is smaller (0.269 m). The thermal bridge has the same thickness as the insulator in the x direction; thus, the horizontal position of the bridge is from x = 0.5 m to x = 1 m for equidistant and from x = 0.731m to x = 1 m for the non-equidistant mesh. The height of the bridge is one cell (1 cm) in the z direction, i.e., 0.01 m, while it is positioned in row number 50 from z = 0.49 m to z = 0.50 m.

We repeat that the temperature in the middle of the cell is considered the cell temperature. The heat capacity of the cells is calculated as $C_i = c_i \rho_i \Delta x_i \Delta z$, while for the thermal resistance in the x-direction, the approximate formula $R_{i,i+1} \approx \frac{\Delta x_i}{k_{i,i+1} Ax}$ is used. Here, Ax is the area of the cell-surface perpendicular to x, which now can be given as $Ax = \Delta y \Delta z = \Delta z$.

Therefore, in the case of the surface simulation (homogeneous material and uniform mesh), the horizontal and vertical resistances can be written as

$$R_{i,i+1} \approx \frac{\Delta x_i}{k_i \Delta z} \text{ and } R_{i,i+N_x} \approx \frac{\Delta z}{k_i \Delta x_i},$$

respectively, where the cell i + N_x is below cell i. In the case of the insulated wall, the material properties or the sizes of the two neighboring cells are different; thus, the resistance between cells i and i + 1 is

$$R_{i,i+1} \approx \frac{\Delta x_i}{2k_i \Delta z} + \frac{\Delta x_{i+1}}{2k_{i+1} \Delta z}.$$

The vertical resistances can be approximated as

$$R_{i,i+N_x} \approx \frac{\Delta x_i}{2k_i \Delta z} + \frac{\Delta x_{i,i+N_x}}{2k_{i,i+N_x} \Delta z}.$$

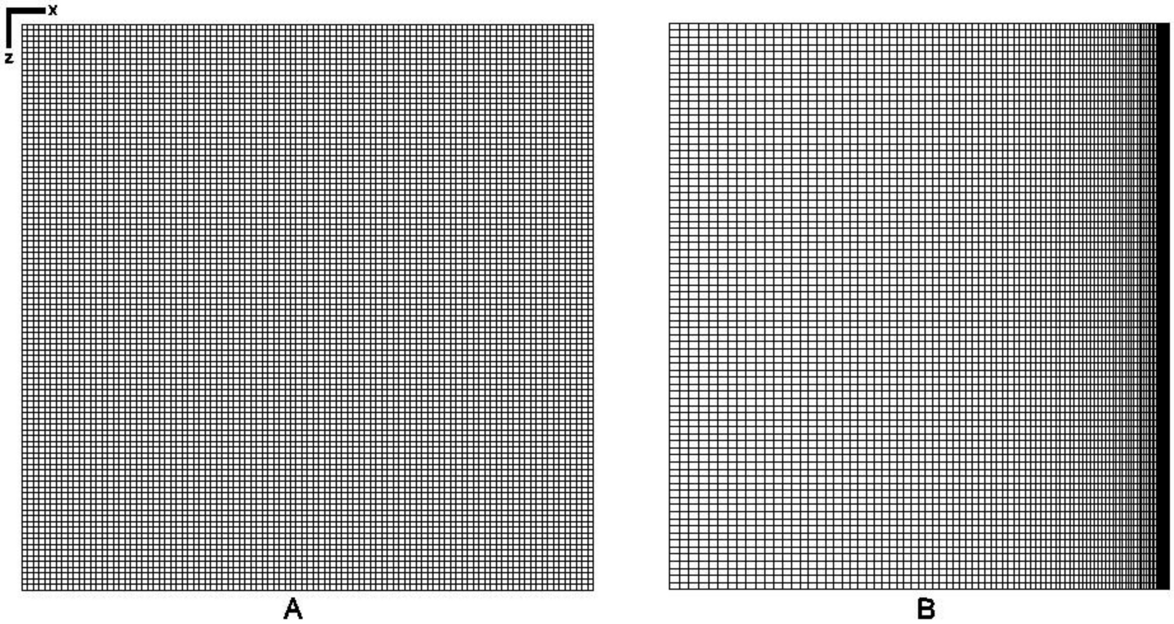

Figure 6. (**A**) Equidistant mesh. (**B**) Gradual change in the x direction.

4.3. The Initial and the Boundary Conditions

Zero Neumann boundary conditions are used in all cases for all boundaries, which forbids conductive heat transfer at the boundaries:

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

This is implemented by setting zero for the matrix elements describing heat conduction through the boundary via the setting of the appropriate resistances to infinity.

I Surface area. In this case, the radiation and convection transfer heat to the y direction, i.e., perpendicular to the plane of Figure 6.

The initial condition is a linear function of the z variable:

$$u(x, z, \ t = 0) = 303 - 293z. \tag{19}$$

We know that this vertical change of initial temperatures may be rare in reality, but with this, we can avoid the case when nothing is changing along the z direction, which would be a 1D problem mathematically.

For the heat convection, we have used values from the literature [36] for the convection heat transfer coefficient h, as shown in Table 3. The universal Stefan–Boltzmann constant

$5.67 \cdot 10^{-8} \frac{\mathrm{W}}{\mathrm{m}^2 \cdot \mathrm{K}^4}$ is multiplied by the appropriate emissivity constant since the surface is not a black body. With this, we obtain realistic values for σ^*. The heat generation contains a fraction of the solar radiation, with which we obtain the value of q^*, as shown below. The ambient temperature of the air is taken to be 30 °C ≈ 303 K.

Table 3. The heat source, convection and radiation parameters of the wall in the case of surface area [36].

	$h\left(\frac{W}{m^2 \cdot K}\right)$	$\sigma^*\left(\frac{W}{m^2 \cdot K^4} \times 10^{-8}\right)$	$q^*_{sunny}\left(\frac{W}{m^2}\right)$	$q^*_{shadow}\left(\frac{W}{m^2}\right)$
All elements	4	4	800	300

The term q also contains the convective heat gain due to the nonzero temperature u_a of the air (in Kelvin), with which we obtain the value of q as follows. The convective and radiative energy transfer is perpendicular to the surface, and it is happening in the y direction. Therefore, these are proportional to the free surface area of the element, which is $\Delta x \Delta z$ here. Using this, the values of the coefficients in our Equations (2) and (3), and we obtain:

$$K = \frac{h}{c\rho\ \Delta y},\ \ \sigma = \frac{\sigma^*}{c\rho\ \Delta y},\ \ q = \frac{q^*}{c\rho\ \Delta y} + \frac{h}{c\rho\ \Delta y} \cdot u_{\mathrm{a}},$$

where, as it was mentioned, $\Delta y = 1$ m.

We supposed that the right half of the surface is in the shadow; thus, the incoming heat is much less there. More precisely, we have

- for the first half of N (sunny part): $q = \frac{1}{c\rho} \times 800 \frac{W}{m^2} + \frac{h}{c\rho} \times 303\mathrm{K}$;
- for the second half of N (shadow part): $q = \frac{1}{c\rho} \times 300 \frac{W}{m^2} + \frac{h}{c\rho} \times 303\mathrm{K}$.

II Cross Sectional Area: In this case, the interior elements cannot gain or lose heat by the heat source, heat convection, or radiation. Elements on the right and left sides can transfer heat by radiation and convection to the x direction with the values shown in Table 4.

Table 4. The heat source, convection, and radiation parameters on both sides of wall elements in the case of cross-sectional area.

	$h\left(\frac{W}{m^2 \cdot K}\right)$	$\sigma^*\left(\frac{W}{m^2 \cdot K^4} \times 10^{-8}\right)$	$q^*(W)$
Right Elements	2	5	500
Left Elements	4	4	500

We obtain the values of the coefficients in our equations, as follows

$$K = \frac{h}{c\rho\ \Delta x},\ \ \sigma = \frac{\sigma^*}{c\rho\ \Delta x},\ \ q = \frac{q^*}{c\rho\ \Delta x} + \frac{h}{c\rho \cdot \Delta x} \cdot u_{\mathrm{a}}$$

We supposed that the right elements and left elements have the following heat source convection and radiation, as follows:

For the left elements (interior side): $q = \frac{1}{c\rho} \times 500 \frac{W}{m^2} + \frac{h}{c\rho \cdot \Delta x} \times 293\mathrm{K}$

For the right elements (external side): $q = \frac{1}{c\rho} \times 500 \frac{W}{m^2} + \frac{h}{c\rho \cdot \Delta x} \times 303\mathrm{K}$

The initial condition is again a linear function of the z variable:

$$u(x, z,\ t = 0) = 303 - 288z.$$

The time is measured in seconds, so the time step size will also be given in seconds in Section 5.

5. Simulation Results

In this section, the final time (the end of the examined time interval) is $t_{\text{fin}} = 10{,}000$ s. The reference solution is obtained by the ode15 s solver of MATLAB, which is a variable-step, variable-order solver based on the (implicit) numerical differentiation formulas of orders 1 to 5, where the letter s means that the solver has been developed for stiff problems. It is applied here with a very small tolerance (10^{-10}) to obtain a very accurate reference solution, which is used in Equation (18) to calculate the maximum error. We note that since this ODE routine solves the same spatially discretized system as our numerical methods, the residual error is missing; thus, as we will see, the errors do not tend to a constant as Δt is decreased, unlike in Figures 2 and 4.

In the case of a non-uniform grid, it makes sense to calculate the energy error:

$$\text{Error}(Energy) = \sum_{1 \le j \le N} C_j \left| u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}}) \right|, \tag{20}$$

where C is the heat capacity of the cells; see Section 2.1. This quantity conveys information on how much energy in the system is not placed in the proper cell. Since energy is measured in Joules and the heat capacities are large numbers in SI units, the energy errors are much larger numbers than the maximum (temperature) errors due to the summation in definition (20) as well.

5.1. Surface of the Wall

Here, a one-layer brick wall (see Figure 5A) is simulated. As it is written in point I above, we applied linear initial and zero Neumann boundary conditions. We have performed the simulations with the equidistant mesh. In Figure 7, the maximum errors as a function of the time step sizes are presented for all methods. Note that the hopscotch-type algorithms, especially the original OOEH and the NS-OEH, are more accurate than the other algorithms. Heun's method is very accurate only below the CFL limit, but above this limit, it cannot give any meaningful results. In Figure 8, we present the initial and the final temperature distribution, where both the effect of the initial condition and the shadow on the right side of the wall can be observed.

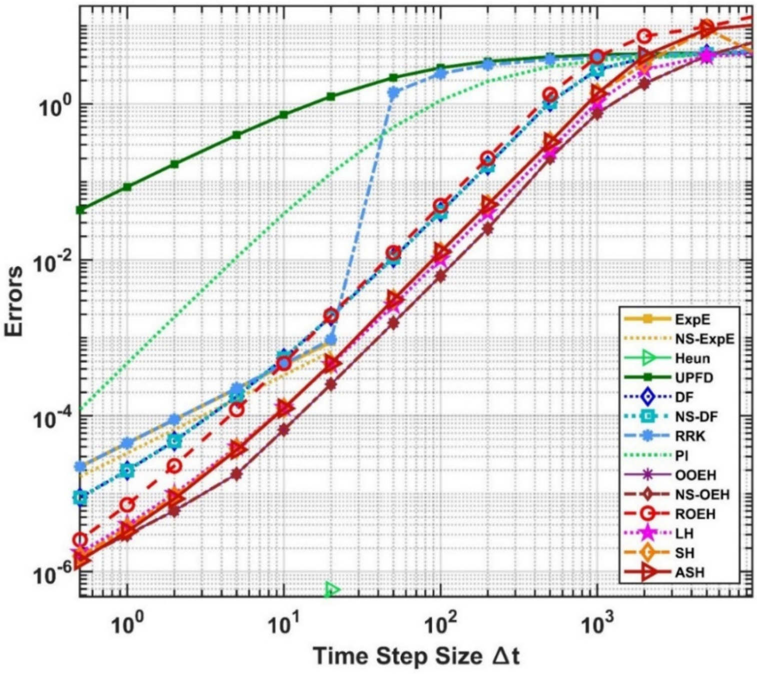

Figure 7. The maximum errors as a function of the time step size Δt for the 14 examined methods in the case of a surface area.

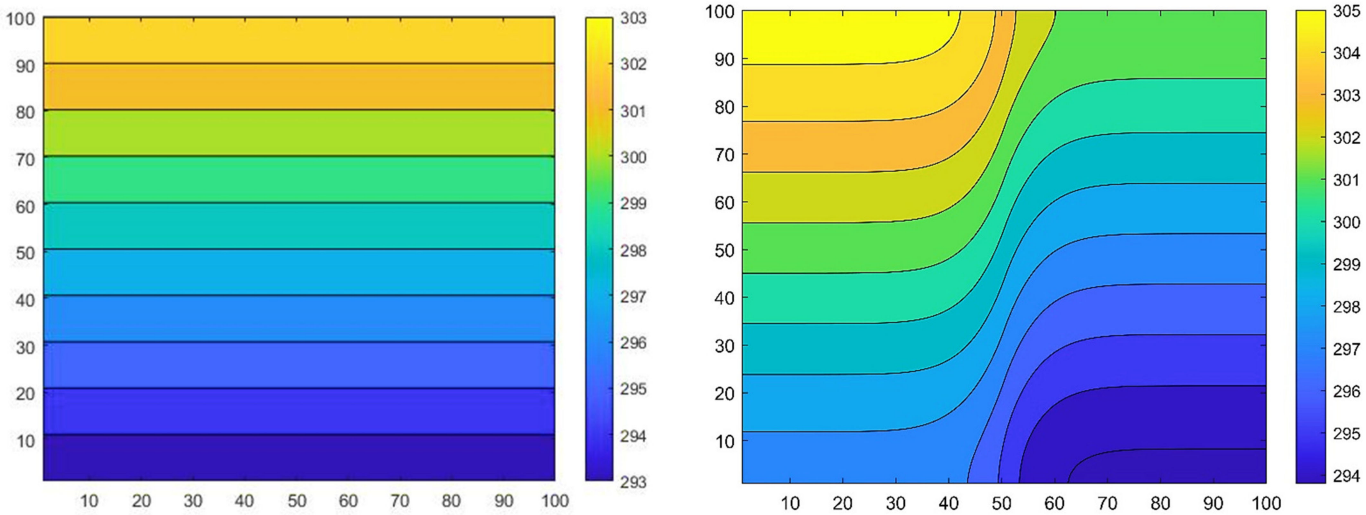

Figure 8. The temperature distribution contour in Kelvin for the equidistant mesh at initial (**left**) and final time (**right**), in the case of multilayer cross-sectional area. The numbers on the vertical and horizontal axes of the contours are the indices of the cells, which are the same as the coordinates in cm units.

5.2. Cross-Section of a Brick Wall with Insulation

Here, the linear initial and Neumann boundary conditions of point II are applied to the multilayer wall. The maximum errors are plotted for equidistant and non-equidistant meshes in Figures 9 and 10, while the energy errors for the non-equidistant mesh can be seen in Figure 11. The temperature distribution contours for the initial and final time moments are shown in Figure 12. One can see that the temperature of the right-hand side of the wall is increasing due to the larger temperature outside, but the insulator lets this heat penetrate the wall only very slowly.

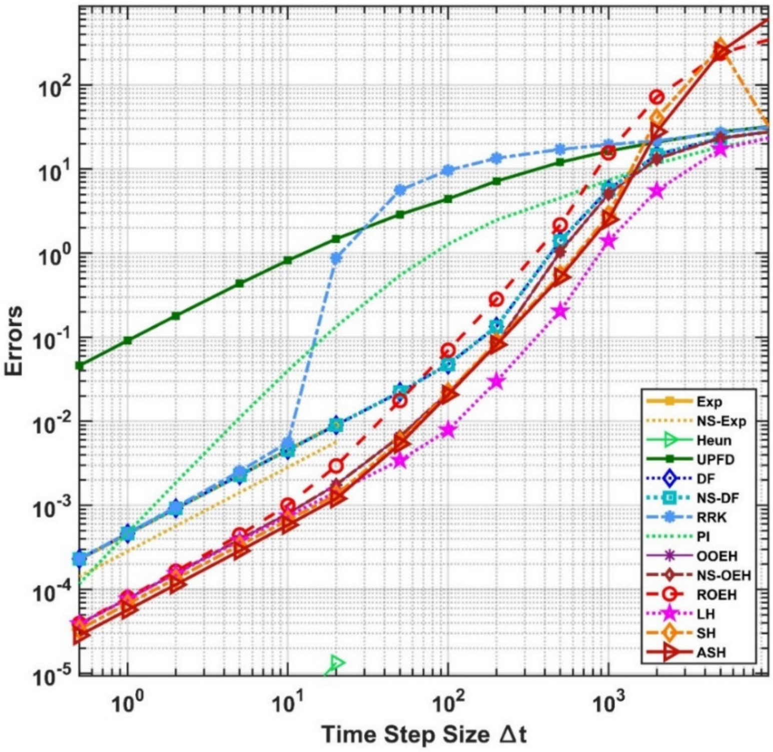

Figure 9. The maximum errors as a function of the time step size Δ*t* for the equidistant mesh.

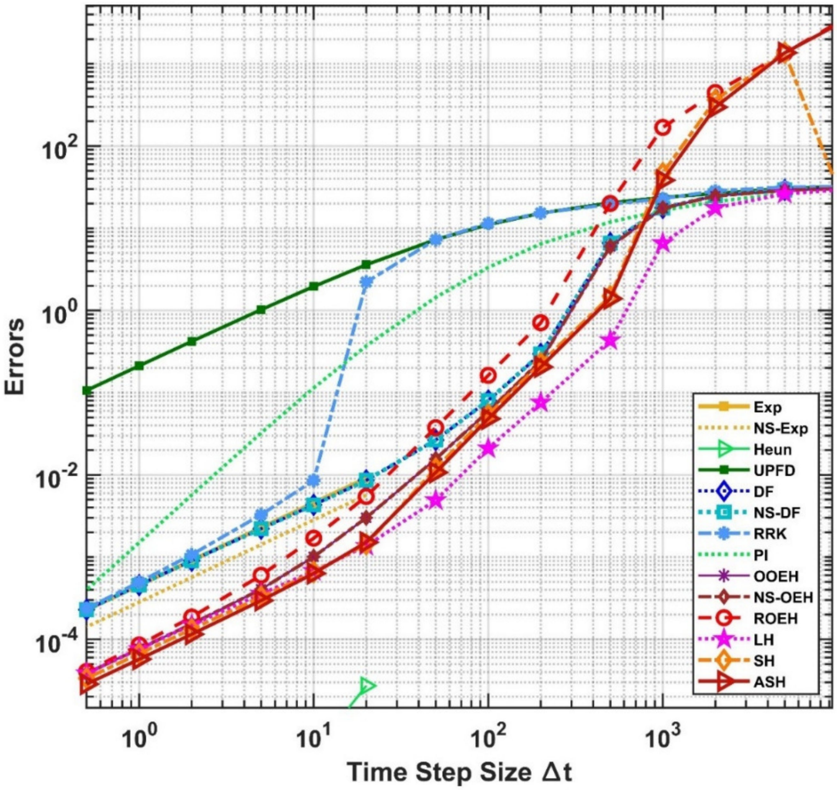

Figure 10. The maximum errors as a function of the time step size for the non-equidistant mesh.

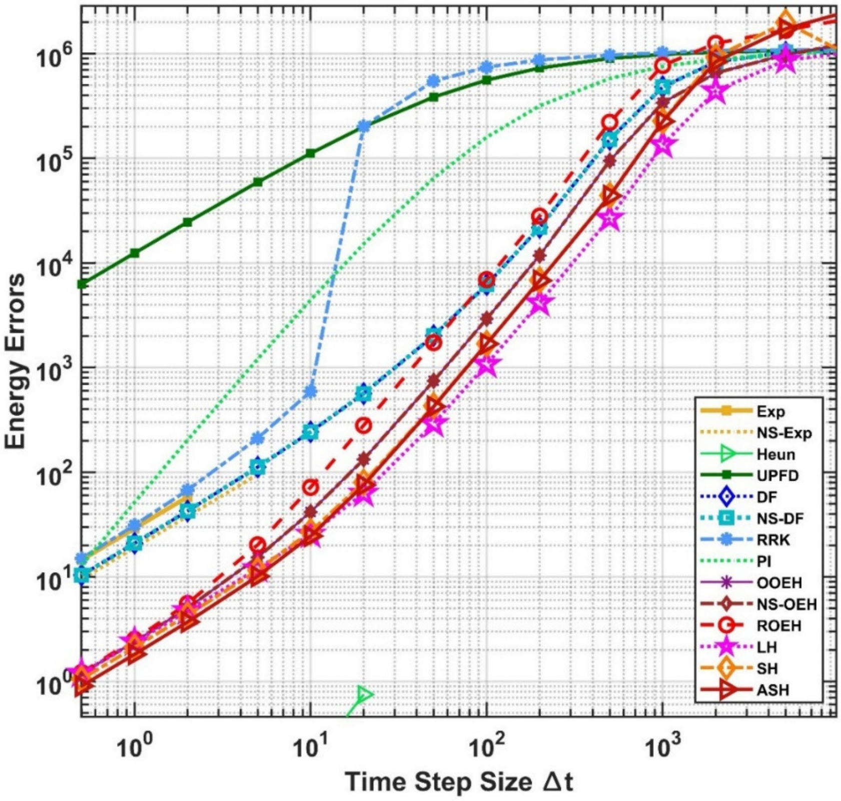

Figure 11. The energy errors as a function of the time step size Δ*t* for the non-equidistant mesh.

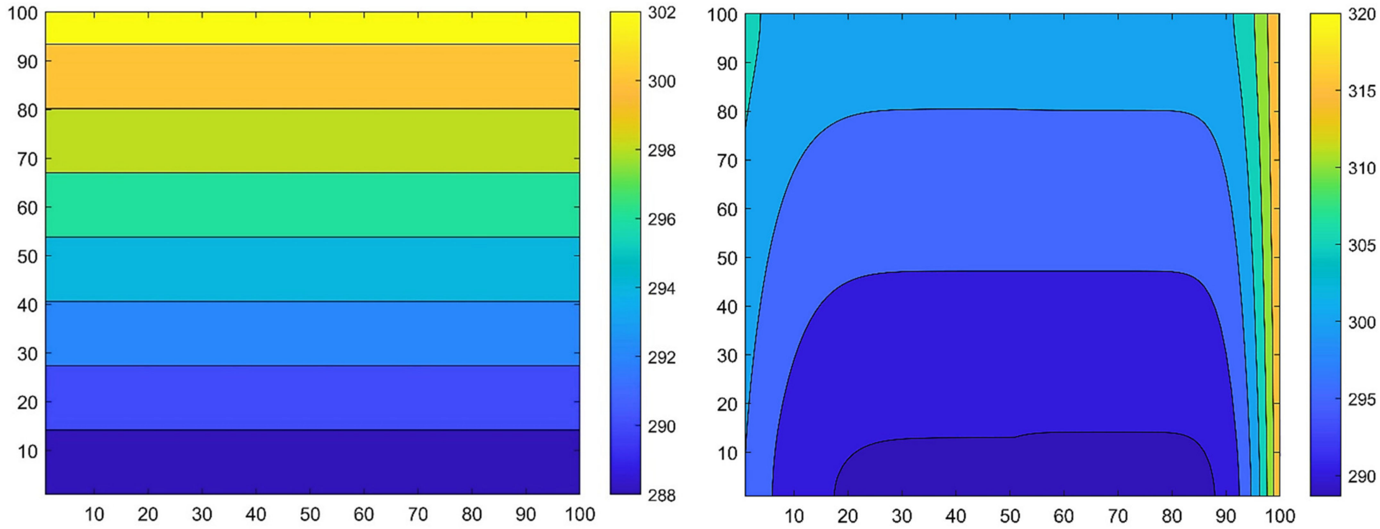

Figure 12. The temperature distribution contour in Kelvin for the equidistant mesh at initial (**left**) and final time (**right**) in the case of the multilayer cross-sectional area. The numbers on the vertical and horizontal axes of the contours are the indices of the cells.

5.3. Brick Wall with Insulation and Thermal Bridging

The conditions enlisted in point II are applied again for the multilayer wall with thermal bridging. The maximum errors are plotted for equidistant and non-equidistant meshes in Figures 13 and 14, respectively, while the energy errors for the non-equidistant mesh can be seen in Figure 15. The maximum and the energy error curves are very similar, and the most noticeable difference is that the SH and the ASH methods have larger maximum errors but smaller energy errors than the DF and the NS-DF methods.

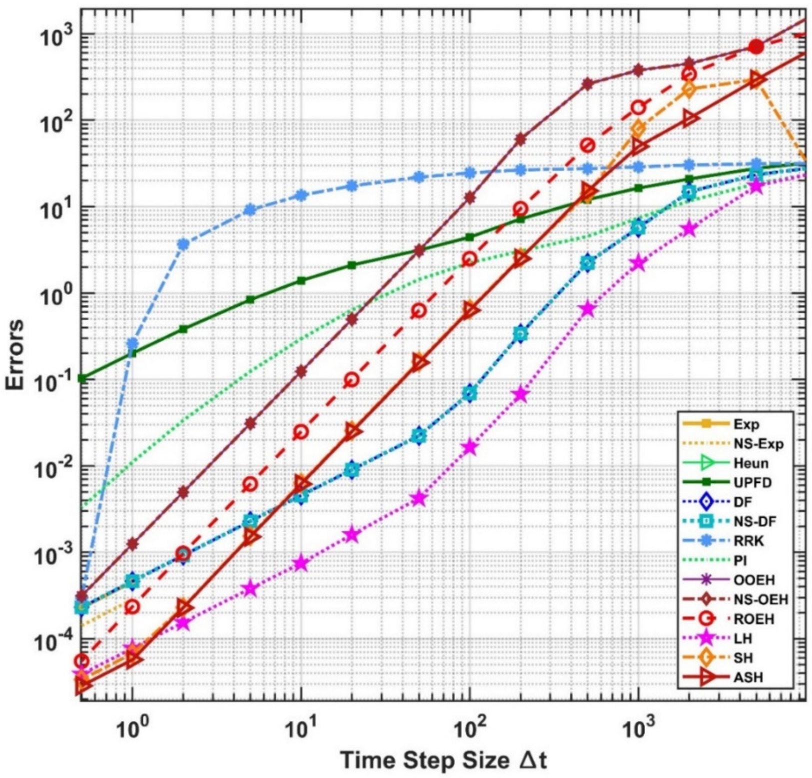

Figure 13. The maximum errors as a function of the time step size Δt for the equidistant mesh and thermal bridging.

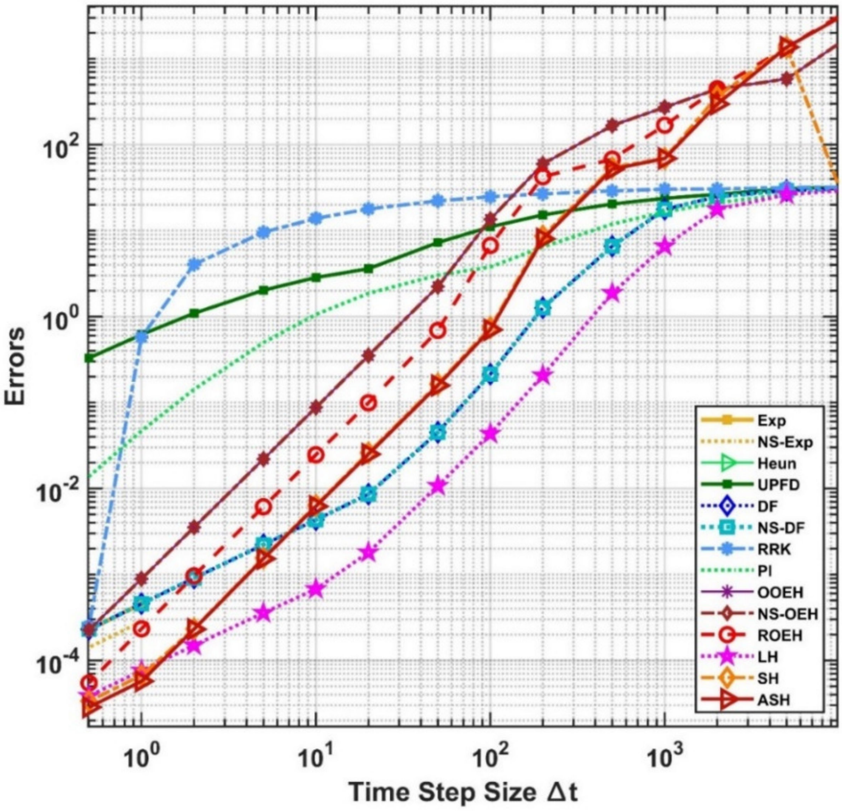

Figure 14. The maximum errors as a function of the time step size for the non-equidistant mesh.

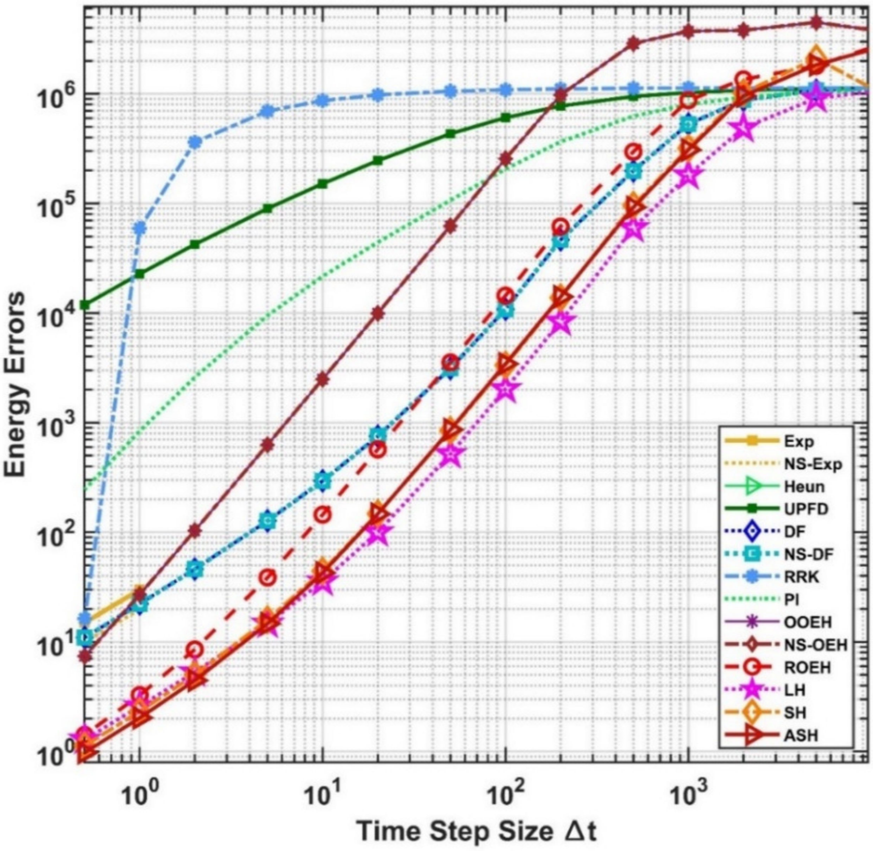

Figure 15. The energy errors as a function of the time step size for the non-equidistant mesh.

In Figure 16, the temperature contour is presented for the initial and the final time moment, respectively, for the equidistant mesh. To let the readers see the effect of the thermal bridge more accurately, we constructed Figure 17, where the final temperature for z = 0.495 as a function of the *x* variable is shown with and without the thermal bridge.

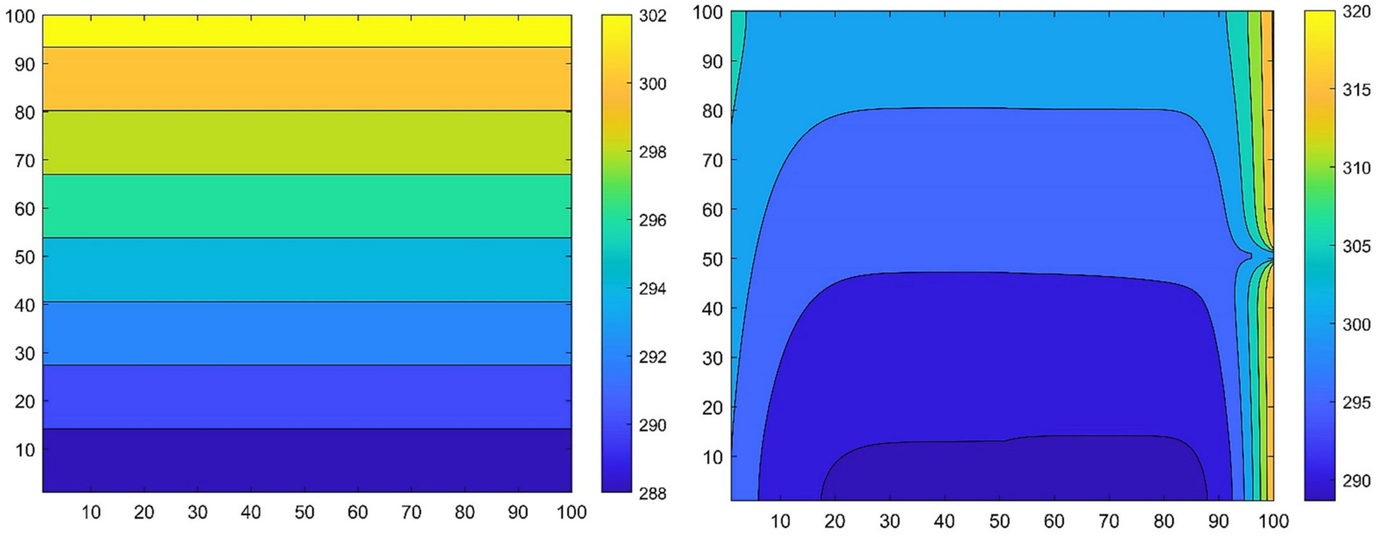

Figure 16. The temperature distribution contour for the equidistant mesh at initial (**left**) and final time (**right**) in the case of a multilayer cross-sectional area with thermal bridging.

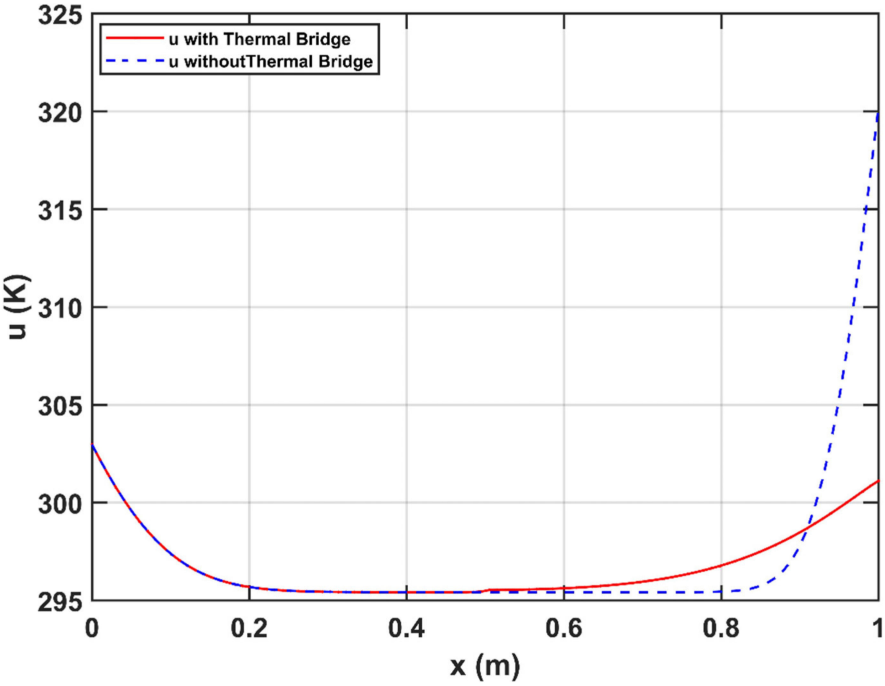

Figure 17. The temperature as a function of the space variable x at the middle row ($z \approx 0.5$) in the case of the multilayer insulated wall with and without thermal bridging using an equidistant grid.

6. Discussion and Summary

We adopted 14 fully explicit numerical algorithms to solve transient heat transfer problems, including heat conduction, convection, and radiation. First, the algorithms were verified using a simple analytical solution in one space dimension. Then we applied the algorithms to two-dimensional systems of a surface area and a cross-sectional area of a wall. This latter one consisted of a brick wall with a glass wool insulator layer, and it contained a thermal bridging steel structure. We used equidistant and non-equidistant grids for the cross-section area. Zero Neumann boundary conditions were applied, and the ode15 s MATLAB routine was used as a reference solution. We showed that all of the methods can be used for these simulations, but those which were proven to be unconditionally stable for the heat conduction equation have much better stability properties in this more general case as well. These methods can be used by quite large time step sizes without stability problems; thus, the traditional explicit time integrators are severely outperformed by them. For less stiff systems, the non-standard version of the odd-even hopscotch and the leapfrog-hopscotch methods are the most accurate. However, as stiffness increases due to material inhomogeneities or the non-equidistant grid, the odd–even hopscotch method becomes less accurate, and the leapfrog-hopscotch takes the lead, while the Dufort–Frankel scheme and the shifted- and asymmetric hopscotch methods also perform well. The UPFD method is the least accurate, but it has the advantage that it preserves the positivity of the temperatures for arbitrary time step size, even for the highly nonlinear case. We note that for very small time step sizes, Heun's method can be extremely accurate, but this level of accuracy is redundant in most fields of engineering, including building energetics.

In the near future, we are going to perform extensive tests of the most successful methods (especially the LH, the NS-DF, and the NS-OEH) by comparing their results and performance (including the running times) with other solvers, e.g., ANSYS. We also plan to validate the whole computation procedure via real physical experiments. After this, we will be ready to apply the methods to real-life engineering problems, the most important of those being smart wall envelopes and buildings containing phase change materials (PCM) to increase energy efficiency. We also started to work on the parallelization of the new algorithms on GPUs, which, due to the explicit nature of the algorithms, can be achieved without major difficulties.

Author Contributions: Supervision, E.K. and B.B.; conceptualization and methodology E.K. and H.K.J.; validation and software, E.K., investigation, visualization and writing—original draft preparation, H.K.J.; writing—review and editing, E.K. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the EU and the Hungarian State and co-financed by the ERDF in the framework of the GINOP-2.3.4-15-2016-00004 project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request.

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

References

1. IAEA. *Adapting the Energy Sector to Climate Change*; IAEA: Vienna, Austria, 2019; ISBN 978-92-0-100919.
2. Harish, V.S.K.V.; Kumar, A. A review on modeling and simulation of building energy systems. *Renew. Sustain. Energy Rev.* **2016**, *56*, 1272–1292. [CrossRef]
3. Li, T.; Xia, J.; Chin, C.S.; Song, P. Investigation of the Thermal Performance of Lightweight Assembled Exterior Wall Panel (LAEWP) with Stud Connections. *Buildings* **2022**, *12*, 473. [CrossRef]
4. Lienhard, J.H., IV; Lienhard, J.H., V. *A Heat Transfer Textbook*, 4th ed.; Phlogiston Press: Cambridge, MA, USA, 2017; ISBN 9780971383524.
5. Savović, S.M.; Djordjevich, A. Numerical solution of diffusion equation describing the flow of radon through concrete. *Appl. Radiat. Isot.* **2008**, *66*, 552–555. [CrossRef]
6. Suárez-Carreño, F.; Rosales-Romero, L. Convergency and stability of explicit and implicit schemes in the simulation of the heat equation. *Appl. Sci.* **2021**, *11*, 4468. [CrossRef]
7. Haq, S.; Ali, I. Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials. *Eng. Comput.* **2021**, *38*, 2059–2068. [CrossRef]
8. Lima, S.A.; Kamrujjaman, M.; Islam, M.S. Numerical solution of convection-diffusion-reaction equations by a finite element method with error correlation. *AIP Adv.* **2021**, *11*, 85225. [CrossRef]

9. Zhang, J.; Zhao, C. Sharp error estimate of BDF2 scheme with variable time steps for molecular beam expitaxial models without slop selection. *J. Math.* **2021**, *41*, 1–19.
10. Amoah-mensah, J.; Boateng, F.O.; Bonsu, K. Numerical solution to parabolic PDE using implicit finite difference approach. *Math. Theory Model.* **2016**, *6*, 74–84.
11. 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]
12. Aminikhah, H.; Alavi, J. An efficient B-spline difference method for solving system of nonlinear parabolic PDEs. *SeMA J.* **2018**, *75*, 335–348. [CrossRef]
13. Ali, I.; Haq, S.; Nisar, K.S.; Arifeen, S.U. Numerical study of 1D and 2D advection-diffusion-reaction equations using Lucas and Fibonacci polynomials. *Arab. J. Math.* **2021**, *10*, 513–526. [CrossRef]
14. Singh, M.K.; Rajput, S.; Singh, R.K. Study of 2D contaminant transport with depth varying input source in a groundwater reservoir. *Water Sci. Technol. Water Supply* **2021**, *21*, 1464–1480. [CrossRef]
15. Haq, S.; Hussaain, M.; Ghafoor, A. A computational study of variable coefficients fractional advection–diffusion–reaction equations via implicit meshless spectral algorithm. *Eng. Comput.* **2020**, *36*, 1243–1263. [CrossRef]
16. Reguly, I.Z.; Mudalige, G.R. Productivity, performance, and portability for computational fluid dynamics applications. *Comput. Fluids* **2020**, *199*, 104425. [CrossRef]
17. Gagliardi, F.; Moreto, M.; Olivieri, M.; Valero, M. The international race towards Exascale in Europe. *CCF Trans. High Perform. Comput.* **2019**, *1*, 3–13. [CrossRef]
18. Appadu, A.R. Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *Proc. Int. J. Numer. Methods Heat Fluid Flow* **2017**, *27*, 1412–1429. [CrossRef]
19. Karahan, H. Unconditional stable explicit finite difference technique for the advection-diffusion equation using spreadsheets. *Adv. Eng. Softw.* **2007**, *38*, 80–86. [CrossRef]
20. Sanjaya, F.; Mungkasi, S. A simple but accurate explicit finite difference method for the advection-diffusion equation. *J. Phys. Conf. Ser.* **2017**, *909*, 1–5. [CrossRef]
21. 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*, 1–12. [CrossRef]
22. Harley, C. Hopscotch method: The numerical solution of the Frank-Kamenetskii partial differential equation. *Appl. Math. Comput.* **2010**, *217*, 4065–4075. [CrossRef]
23. Al-Bayati, A.; Manaa, S.; Al-Rozbayani, A. Comparison of Finite Difference Solution Methods for Reaction Diffusion System in Two Dimensions. *AL-Rafidain J. Comput. Sci. Math.* **2011**, *8*, 21–36. [CrossRef]
24. Nwaigwe, C. An Unconditionally Stable Scheme for Two-Dimensional Convection-Diffusion-Reaction Equations. Ph.D. Thesis, University College of Swansea, Swansea, UK, 2022.
25. 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. Mat.* **2022**, *71*, 245–252. [CrossRef]
26. Berger, J.; Gasparin, S.; Mazuroski, W.; Mendes, N. An efficient two-dimensional heat transfer model for building envelopes. *Numer. Heat Transf. Part A Appl.* **2020**, *79*, 163–194. [CrossRef]
27. Ndou, N.; Dlamini, P.; Jacobs, B.A. Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations. *Mathematics* **2022**, *10*, 2639. [CrossRef]
28. Kovács, E.; Gilicz, A. New stable method to solve heat conduction problems in extremely large systems. *Des. Mach. Struct.* **2018**, *8*, 30–38.
29. Saleh, M.; Nagy, Á.; Kovács, E. Part 1: Construction and investigation of new numerical algorithms for the heat equation. *Multidiszcip. Tud.* **2020**, *10*, 323–338. [CrossRef]
30. Nagy, Á.; Saleh, M.; Omle, I.; Kareem, H.; Kovács, E. New stable, explicit, shifted-hopscotch algorithms for the heat equation. *Math. Comput. Appl.* **2021**, *26*, 61. [CrossRef]
31. Nagy, Á.; Omle, I.; Kareem, H.; Kovács, E.; Barna, I.F.; Bognar, G. Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation. *Computation* **2021**, *9*, 92. [CrossRef]
32. Jalghaf, H.K.; Kovács, E.; Majár, J.; Nagy, Á.; Askar, A.H. Explicit stable finite difference methods for diffusion-reaction type equations. *Mathematics* **2021**, *9*, 3308. [CrossRef]
33. Kareem Jalghaf, H.; 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]
34. Gasparin, S.; Berger, J.; Dutykh, D.; Mendes, N. Stable explicit schemes for simulation of nonlinear moisture transfer in porous materials. *J. Build. Perform. Simul.* **2018**, *11*, 129–144. [CrossRef]
35. Ochoa, G.V.; Sanchez, W.E.; Truyoll, S.D.L.H. Experimental and theoretical study on free and force convection heat transfer. *Contemp. Eng. Sci.* **2017**, *10*, 1143–1152. [CrossRef]
36. Holman, J.P. *Heat Transfer*; McGraw-Hill Science: New York, NY, USA, 2010; ISBN 978-0-07-352936-3.
37. Munka, M.; Pápay, J. *4D Numerical Modeling of Petroleum Reservoir Recovery*; Akadémiai Kiadó: Budapest, Hungary, 2001; ISBN 963-05-7843-3.

38. Agbavon, K.M.; Appadu, A.R. Construction and analysis of some nonstandard finite difference methods for the FitzHugh–Nagumo equation. *Numer. Methods Partial. Differ. Equ.* **2020**, *36*, 1145–1169. [CrossRef]
39. Workie, A.H. New Modification on Heun's Method Based on Contraharmonic Mean for Solving Initial Value Problems with High Efficiency. *J. Math.* **2020**, *2020*, 6650855. [CrossRef]
40. 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]
41. Hirsch, C. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*; Wiley: Hoboken, NJ, USA, 1988.
42. Sottas, G. Rational Runge-Kutta methods are not suitable for stiff systems of ODEs. *J. Comput. Appl. Math.* **1984**, *10*, 169–174. [CrossRef]
43. 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]
44. Saleh, M.; Nagy, Á.; Kovács, E. Part 3: Construction and investigation of new numerical algorithms for the heat equation. *Multidiszcip. Tud.* **2020**, *10*, 349–360. [CrossRef]
45. Kovács, E.; Nagy, Á.; Saleh, M. A New Stable, Explicit, Third-Order Method for Diffusion-Type Problems. *Adv. Theory Simul.* **2022**, *5*, 2100600. [CrossRef]
46. Iserles, A. *A First Course in the Numerical Analysis of Differential Equations*; Cambridge University Press: Cambridge, UK, 2009; ISBN 9788490225370.