energies

MDPI

Article

Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges

Issa Omle [1,2,3], Ali Habeeb Askar [1,2,4], Endre Kovács [2,*] and Betti Bolló [1]

[1] Department of Fluid and Heat Engineering, University of Miskolc, 3515 Miskolc, Hungary; issa.j.omle@gmail.com (I.O.); 20156@uotechnology.edu.iq (A.H.A.)
[2] Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary
[3] Mechanical Power Engineering Department, Al-Baath University, Homs 77, Syria
[4] Mechanical Engineering Department, University of Technology—Iraq, Baghdad 10066, Iraq
[*] Correspondence: kendre01@gmail.com or endre.kovacs@uni-miskolc.hu

Abstract: Several previous experiments showed that the leapfrog–hopscotch and the adapted Dufort–Frankel methods are the most efficient among the explicit and stable numerical methods to solve heat transfer problems in building walls. In this paper, we extensively measure the running times of the most successful methods and compare them to the performance of other available solvers, for example, ANSYS transient thermal analysis and the built-in routines of MATLAB, where three different mesh resolutions are used. We show that the running time of our methods changes linearly with mesh size, unlike in the case of other methods. After that, we make a long-term simulation (one full winter month) of two-dimensional space systems to test the two best versions of the methods. The real-life engineering problem we solve is the examination of thermal bridges with different shapes in buildings to increase energy efficiency.

Keywords: heat conduction; thermal insulation; long-term simulation; types of thermal bridges; transient thermal Analysis; ANSYS

check for updates

Citation: 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. https://doi.org/10.3390/en16124604

Received: 14 April 2023
Revised: 3 June 2023
Accepted: 6 June 2023
Published: 8 June 2023

1. Introduction

Significant progress toward a sustainable economy may be made via improved building energy efficiency, where the buildings account for 40% of the primary energy use and 24% of the generation of greenhouse gases [1]. When compared to other industries, the construction industry has the ability to significantly reduce energy usage and, by extension, greenhouse gas emissions [2]. Heat transfer calculations are widely used in buildings, including determining how much energy is lost or gained through the building's envelope (heat conduction), conducting environmental analyses inside the building, and troubleshooting issues with specific materials or structural elements [3,4].

Exterior wall conduction, interior mass conduction, heat gain/loss conversion to cooling and heating load, and ground heat loss from the slab-on-grade floor and basement walls are all issues related to conduction heat transfer that arise in the context of buildings, so the rate at which heat moves through walls can be changed by changing the thickness of the walls or by making the layer of insulation thicker [5]. Transitory wall conduction heat transfer reacts to weather conditions such as temperature swings, sunshine, air movement, etc. Increasing the building envelope's thermal insulation and decreasing the heat loss via walls is one of the most efficient strategies to raise a structure's energy efficiency and decrease its energy consumption [6]. Walls account for around 35% of the heat loss in a house, with the vast majority of the heat entering and leaving them [7]. Since the walls of a house are in direct contact with the colder air outside, heat is often lost by conduction or physical contact.

EN ISO 13789 [8] is a commonly used standard for calculating transmission heat transfer through a building envelope; which in turn mentions two standards for thermal bridges; EN ISO 10211 [9] provides a structure for thorough calculations of thermal bridges in building construction and EN ISO 14683 [10] provides a simplified approach with default linear thermal transmittance values.

Because thermal bridges have a big effect on heat loss and reduce building energy efficiency, durability, and air quality, requiring integrated thermal and structural design [11], and the thermal bridge models that were looked into should be especially interesting to architects, civil engineers, and people who work in the insulation materials industry. The examination of the effects of thermal bridges is meant to illuminate the feasibility of implementing energy renovation techniques in existing structures [12]. Thermal bridge effects are analyzed using a stationary linear thermal transmittance method for a wide variety of foundation types and construction parameters. Validation tests on over a thousand real-world examples corroborated the validity of those correlations, allowing for a straightforward and practical analysis of thermal bridges in existing structures and providing practitioners with a resource that accounts for the vast majority of scenarios [13].

A thermal analysis may reveal the heat distribution patterns of a system or its individual parts. Most research into thermal quantities focuses on temperature distributions, thermal fluxes, and heat capacities. Since diverse heat transfer applications within engineering fields involve many thermal models, the analysis of transient heat transfer is an essential problem that is often solved using numerical rather than analytical methods. Analytic techniques provide precise results, but they can only be used for isotropic, homogeneous situations with straightforward geometries and boundary conditions [14].

Heat transfer across a layer is found to be proportional to the ΔT temperature difference across the layer and the heat transfer area, but inversely proportional to the layer thickness. It is called "transient conduction" when the mode of thermal energy transfer occurs throughout a time period in which temperatures vary at any location inside an item. Time-dependent temperature fields are referred to as "non-steady-state" conduction. Conduction of heat through the composite wall may be modeled if required. Assuming this is the case, we simply require boundary conditions at the outside wall surface, with the same conditions applied to the inside wall surface. Solving heat conduction equations across the wall layers yields their temperatures.

The heat conduction equation, often known as the "heat equation", is a partial differential equation (PDE) used to explain the conduction of heat through a solid. Innovative analytical approaches [15] exist for spatially homogeneous systems, and mathematicians create and evaluate most numerical techniques for homogeneous cases. However, some analytical solutions for steady-state and transient situations are also available for the one-dimensional inhomogeneous system, for example, multilayer systems [16]. Most of the time, these systems are used to determine how much heat is gained or lost through the outside envelope and how much heat is stored inside buildings [3]. Density, thermal conductivity, and specific heat are the key examples of material qualities that may change greatly in a building's system. Weather conditions also change over time. These imply that most heat conduction issues are multi-dimensional and transient [15]. As a result, we need to use numerical computer simulation.

Our research group works on transient heat transfer calculations using fundamental physical laws (ab initio approach). Therefore, it is expected that these results are much more accurate than those based on the usual (ISO) standards, which are steady-state calculations without solving the transient PDE and therefore cannot properly take into account, for example, the heat capacity of the envelope. Our long-term goal is to revolutionize these simulations (at this stage by the numerical methodology) in order to make transient simulations more available due to reduced computational cost and programming difficulty, as follows.

Heat transfer and other similar problems are routinely solved by lots of numerical methods. Most of them fall into one of two categories: explicit method or implicit method.

Compared to one another, each type has some key benefits and at least one serious drawback. The widely used explicit algorithms, such as the classical FTCS (forward time central space), execute time steps in a small amount of time and are quite straightforward to parallelize. The problem is that the so-called Courant-Friedrichs-Lewy (CFL) limit constrains their stability domain and causes them to become unstable if the time step size is above this limit [16]. Moreover, since an insulated wall consists of very different materials, the coefficients can change sharply from mesh point to mesh point; thus, this limit is rather tiny and the stiffness ratio is large.

Since the stability of the implicit algorithms is substantially greater, many scholars see them as superior and frequently employ them to solve these and other similar PDEs [17–19]. The drawback comes from the fact that in each time step, a system of algebraic equations must be solved, a procedure that is difficult to parallelize. In addition, the large amounts of the required RAM can also make calculations extremely slow. These problems are remarkable typically when the physical space is two or three-dimensional.

It has been shown that explicit approaches can be more effective even when tiny time step sizes are required [20]. Moreover, various ingenious combinations of explicit and implicit methods, such as semi-explicit or semi-implicit methods, have also been presented [21–24]. However, they do not really solve the problems and the drawbacks of the explicit and implicit approaches discussed above.

In light of the aforementioned data, it is reasonable to assume that explicit algorithms, particularly if they have improved stability qualities (see, for example, [25–32]), will have a growing comparative advantage over time. We started working on new explicit schemes a couple of years ago for determining heat conduction in any number of spatial dimensions. In our original articles [33–40], the novel algorithms were theoretically and experimentally investigated. They were proven to be stable for the linear diffusion- or diffusion-reaction equation, and their theoretical order of convergence is also stated. The algorithms were tested using analytical as well as numerical reference solutions. It was demonstrated that they deliver fairly accurate results at a remarkably greater speed than the widely used methods, for example, the built-in ODE solvers of MATLAB and that they can be used in a wide variety of situations involving random discontinuous parameters and initial conditions.

In our paper [38], we tested 12 explicit and stable numerical algorithms to simulate heat conduction in building's walls, with and without insulation, using equidistant and non-equidistant meshes. We obtained that the original odd–even hopscotch (OOEH), the leapfrog–hopscotch (LH), and the Dufort–Frankel (DF) are the most effective methods. Then, in [37] we adopted some of the above-mentioned 14 methods to include not only heat conduction but also convection and radiation. According to the results, the LH and the non-standard version of the OOEH are the most accurate if the system is not really stiff. However, the OOEH becomes less accurate when the stiffness is larger, which is the case if the mesh is non-equidistant and/or there are materials in homogeneity. In these cases, the LH takes the lead, but the DF, as well as the shifted-hopscotch (SH) and asymmetric hopscotch (ASH) methods, also perform well. Those methods which are unconditionally stable for the simple conduction case can be used without stability problems with fairly large time step sizes, so they outperform the conventional explicit time integrators. Since usually the LH was the best method, we devoted a whole paper [37] on the question of how to implement the convection and radiation term in an optimal way. However, in those papers [37,38], no running-time measurements were made, and the performance was evaluated only in terms of accuracy versus time step size.

In our current work, we continue the above-mentioned investigations. We use ANSYS' thermal analysis solutions that help engineers solve the most complex thermal challenges, and predict how their designs will perform with temperature changes. However, because simulation by this kind of software takes a long time and requires serious computer resources, we compare our methods with ANSYS to investigate runtime, stability, and other features [41]. Now the goal is to systematically evaluate how the performance of the various solvers (including MATLAB routines and ANSYS) depends on the mesh settings to see which one is optimal for certain accuracy requirements. Therefore, the paper is organized as follows: Section 2 presents the examined system with the equations, followed by the numerical methods and implementations of the convection and radiation terms, as well as the methods used for comparison purposes. Sections 3.1–3.3 display the preliminary conditions for the simulation of the wall: materials, mesh construction, initial, and boundary conditions.

In Section 3.4, we start by verifying our codes in a 2D system for three different cases and comparing the results against simple analytical solutions on a grid that is equidistant. Section 4 displays the results of the numerical tests performed to compare differences in errors and running times. Section 5 shows the second part of this study, where we chose our best methods to perform a long-term simulation of one month for a real wall in Miskolc city with and without insulation. Two kinds of the thermal bridges are included, and it is calculated how much energy is lost due to the thermal bridges. Section 6 finally concludes with a summary of our findings.

2. The Studied Cases

2.1. The Equation and Its Discretization

Using a linear parabolic PDE, we can characterize the phenomenon of the simplest Fourier-type heat conduction inside a homogeneous medium with a heat source as follows:

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

where $u = u\left(\vec{r}, t\right)$ is the temperature, q is the heat generation or heat source, and α is the thermal diffusivity, which is given as $\alpha = k/(c\rho)$, where $c = c\left(\vec{r}, t\right)$, $k = k\left(\vec{r}, t\right)$, $\rho = \rho\left(\vec{r}, t\right)$ are the specific heat, heat conductivity, and the density of the material, respectively.

Newton's law of cooling suggests that a term $K(u_a - u)$ can describe convective heat transfer [42], where the ambient temperature u_a (measured in Kelvin) can be considered as independent from the unknown variable u. Consequently, it makes sense to include the Ku_a term in the expression q, which represents the heat source. Stefan-Boltzmann's law [43] says that a term $-\sigma u^4$ can be used to describe the surface's radiative heat loss, where now the proportionality constant σ is the product of the Stefan-Boltzmann constant and the surface area, which are all positive quantities. We can incorporate the incoming radiation, which may include direct sunshine, into the source term q similarly to the Ku_a term above. The following is an extension of the heat conduction Equation (1) to include the terms for convection, radiation, and the source of heat:

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

Note that all terms in Equation (2) are local, except the conduction term. In the case of Equation (1) in one space dimension, we apply to the $\alpha\, \nabla^2 u$ term the most common central difference Equation

$$\frac{\partial^2}{\partial x^2} u(x_i) \approx \frac{\frac{u(x_{i+1}) - u(x_i)}{\Delta x} + \frac{u(x_{i-1}) - u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \tag{3}$$

which is second order in Δx, where $i = 1, \ldots, N$ and N is the overall number of nodes. By doing this, we are able to derive the spatially discretized version of the heat transfer Equation (2) in one space dimension as follows:

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

Let us now present the discretization of the heat transfer equation assuming that the quantities describing the properties of materials, namely α, k, c, and ρ, are functions of the space, rather than a fixed value. Now in one space dimension, instead of the $\alpha \nabla^2 u$ term, we have to deal with the term

$$\frac{1}{c(x)\rho(x)} \frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}\right)$$

In this case, the heat conduction equation can be discretized as follows

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

Section 5 of the book [44] presents more details about this procedure for the case of underground reservoirs. The nodes are surrounded by cells, and $k_{i,i+1}$ is the heat conductivity between node i and its right neighbor. The discretized equation attains the following form

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x}\left(k_{i,i+1}\frac{u_{i+1} - u_i}{\Delta x} + k_{i-1,i}\frac{u_{i-1} - u_i}{\Delta x}\right) + q - Ku_i - \sigma {u_i}^4. \tag{5}$$

The dimensions of a cell, measured along its length and across its (typical) cross-section, are represented as Δx and S. Since the volume of the cell can be given as $V = S\,\Delta x$, the heat capacity is estimated as $C_i = c_i\rho_i V$. The thermal resistance between the two neighboring nodes can be determined as $R_{i,i+1} \approx \Delta x/(k_{i,i+1}S)$. Based on these new quantities, one can obtain the following expression for the time derivative of each cell variable:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i}C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i}C_i} + q - Ku_i - \sigma {u_i}^4 \tag{6}$$

It is not hard to generalize Equation (6) even more for the case of arbitrary number of neighbors to obtain the following spatially discretized version of Equation (2):

$$\frac{du_i}{dt} = \sum_{j\neq i}\frac{u_j - u_i}{R_{i,j}C_i} + q - Ku_i - \sigma {u_i}^4. \tag{7}$$

This ODE system is applicable to arbitrary (i.e., unstructured) grids with cells that may have different forms and characteristics. Naturally, uneven discretization might reduce spatial accuracy, but in this research, only cells with a rectangular form are employed.

2.2. The Leapfrog–Hopscotch Structure

It is necessary to discretize the space domain using a special, so-called bipartite mesh before applying the leapfrog–hopscotch or any other odd–even hopscotch algorithm.

Therefore, the mesh is split into two separate subgroups. The nodes or cells belonging to the first and second subgroups are designated as odd and even, respectively. The primary criterion is that, like on a checkerboard, all the near neighbors of the odd cells must be even and vice versa. We explain it using a 1D interval as an example $x \in [x_0, x_N]$, $L = x_N - x_0$ on which coordinates are used to create an equidistant grid $x_0, x_1, \ldots, x_N$ of nodes, so $x_j = x_{j-1} + \Delta x,\ j = 1,\ldots,N,\ \Delta x = L/N$. The time domain is $t \in \left[t^0, t^{\text{fin}}\right]$, and it is discretized as usual: $t^j = t^0 + jh\,,\ \ j = 1, \ldots,\ T\,,\ \ hT = t^{\text{fin}} - t^0$, where h is the time step size.

The leapfrog–hopscotch (LH) space-time structure [33] is presented in Figure 1. The calculation consists of two half and several full-time steps. The calculation starts by taking a half-sized time step for the odd nodes based on the initial values. This "zeroth" stage is symbolized by a blue half-circle in the figure. After that, full-sized time steps (red circles in the figure) are taken strictly alternately for the even and odd nodes up until the end. At the end, the last time step (purple half-circle) should be cut in half for odd nodes to reach the same final time point as the even nodes. It is essential that when a stage-calculation is executed for node i, the most recently obtained values of the neighbors $i - 1$ and $i + 1$ are used in the equations.

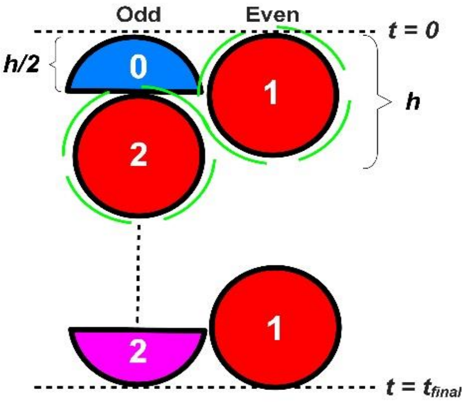

Figure 1. The leapfrog–hopscotch (LH) structure.

2.3. The Mathematical Equations We Use

In this paper, we give only the formulas applied for Equation (7). Note that the discretization and the formulas for a 1D equidistant mesh can be found in the given references.

It is well known that the theta-method for the ODE $y' = f(t, y)$ has the Equation

$$y^{n+1} = y^{n+1} + \Delta t\left[\theta f(t^n, y^n) + (1-\theta) f\left(t^{n+1}, y^{n+1}\right),\right] \tag{8}$$

where $\theta \in [0, 1]$. If we adapt it for Equation (7), we obtain

$$u_i^{n+1} = u_i^n + \theta\left[A_i^n - r_i u_i^n - \Delta t K u_i^n - \Delta t\sigma (u_i^n)^4\right] + (1-\theta)\left[A_i^{n+1} - r_i u_i^{n+1} - \Delta t K u_i^{n+1} - \Delta t\sigma \left(u_i^{n+1}\right)^4\right], \tag{9}$$

where

$$A_i^n = \Delta t \sum_{j \neq i} \frac{u_j^n}{R_{i,j} C_i} + \Delta t \cdot q_i, \ \text{ and } r_i = \Delta t \sum_{j \neq i} \frac{1}{R_{i,j} C_i}\ , \quad i = 1,\ \ldots, N,\ \ n = 0,\ \ldots, T. \tag{10}$$

The first quantity mediates the accumulated effect of the temperatures of the cell i's neighbors. The second quantity is the generalization of the often-used mesh ratio $r = \frac{\alpha \Delta t}{\Delta x^2}$, more specifically for Equation (4), the $r_i = 2r$ relation holds.

One should note that the sum in Equation (7) is separated into two terms: the first contains the actual cell variable, whereas the other, A_i^n, contains only its neighbors. The theta method given in Equation (9) is an implicit method if $\theta < 1$. It can be made explicit by the so-called pseudo-implicit trick: the neighbors in the second term at the r. h. s. of (9) must be taken at the n-th time level. Furthermore, 3 of the four powers of u_i^{n+1} in the nonlinear term can be replaced by u_i^n. With this, we obtain:

$$u_i^{n+1} = u_i^n + \theta\left[A_i^n - r_i u_i^n - \Delta t K u_i^n - \Delta t\sigma (u_i^n)^4\right] + (1-\theta)\left[A_i^n - r_i u_i^{n+1} - \Delta t K u_i^{n+1} - \Delta t\sigma u_i^{n+1} (u_i^n)^4\right], \tag{11}$$

which can be rearranged as

$$u_i^{n+1} = \frac{u_i^n + A_i^n - \theta\left[r_i u_i^n + \Delta t K u_i^n + \Delta t \sigma \left(u_i^n\right)^4\right]}{1 + (1-\theta)\left[r_i + \Delta t K + \Delta t \sigma \left(u_i^n\right)^3\right]}. \tag{12}$$

The value $\theta = 1$ yields the explicit Euler method, which has a low CFL limit. If $\theta = 0$ and q is non-negative, Equation (12) preserves the positivity of the temperature for arbitrary time step sizes. In this case, it can be considered as an adaptation of the so-called UPFD (unconditionally positive finite difference) scheme invented a decade ago [45]. Generally, smaller values of θ mean better stability, but it can imply worse accuracy as well. Now we fill the LH structure with this generalized theta-formula as follows.

In the pure conduction case (where K = 0 and $\sigma = 0$), the following theta values were obtained [39] during optimization: Stage 0: $\theta = 0$, all other stages: $\theta = 1/2$, which will be used everywhere in this work for the term $A_i^n - r_i u_i^n$. However, there is no reason to believe that for the convection and radiation terms, the optimal theta values are the same. In fact, in our last work [43], we examined the different treatments of these two terms, and according to our experience, only a few versions are competitive. For the convection term, $\theta = 1/2$ is always the best choice, since our analytical calculations showed that it preserves second order convergence and unconditional stability at the same time. We currently do not have analytical proofs in the presence of the nonlinear term, but we found that the three theta values are worth examining, namely $\theta = 0,\ 1/2,\ 1$. We exemplify these by presenting the "zeroth" stage equations as follows.

1. Pseudo-implicit treatment: $\theta = 0$ for the radiation term, which yields:

$$u_i^{1/2} = \frac{u_i^0 + A_i^0/2 - K\Delta t u_i^0/4}{1 + r_i + K\Delta t/4 + \sigma \Delta t \left(u_i^0\right)^3/2} \tag{13}$$

2. "Inside" treatment: $\theta = 1$ for the radiation term, which means that it is taken into account explicitly, which yields

$$u_i^{1/2} = \frac{u_i^0 + A_i^0/2 - K\Delta t u_i^0/4 - \sigma \Delta t \left(u_i^0\right)^4/2}{1 + r_i + K\Delta t/4} \tag{14}$$

3. Mixed treatment with an equal share of the previous two treatments, that is, $\theta = 1/2$ for the radiation term, which yields

$$u_i^{1/2} = \frac{u_i^0 + A_i^0/2 - K\Delta t u_i^0/4 - \sigma \Delta t \left(u_i^0\right)^4/4}{1 + r_i + K\Delta t/4 + \sigma \Delta t \left(u_i^0\right)^3/4} \tag{15}$$

2.4. Other Explicit and Stable Methods

One classic example of a scheme that meets both of the criteria of explicitness and unconditional stability is the one of Dufort and Frankel (DF) [46] (p. 313). As this algorithm does not start itself, u_i^1 has to be derived from u_i^0 using some other approach. Here, we use the UPFD calculation method [43] for this purpose:

$$u_i^1 = u_i^0 + A_i^0/1 + r_i + \Delta t K + \Delta t \sigma \left(u_i^0\right)^3 \tag{16}$$

After this first step, the procedure is described by a simple formula for heat conduction, but the terms of convection and radiation can be handled in several ways. We present here only the most promising treatments.

4. DF-D: the only place the Sigma and K terms appear is in the denominator:

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1 + r_i + 2\Delta t K + 2\Delta t \sigma \left(u_i^n\right)^3} \tag{17}$$

5. DF-M: the Sigma and K terms in a mixed way:

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t K u_i^n - \Delta t \sigma \left(u_i^n\right)^4}{1 + r_i + \Delta t K + \Delta t \sigma \left(u_i^n\right)^3} \tag{18}$$

6. DF-KD: the K term appears only in the denominator, and the Sigma term in a mixed way:

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t \sigma \left(u_i^n\right)^4}{1 + r_i + 2\Delta t K + \Delta t \sigma \left(u_i^n\right)^3} \tag{19}$$

7. DF-SD: the Sigma term appears only in the denominator, and the K term in a mixed way:

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t K u_i^n}{1 + r_i + \Delta t K + 2\Delta t \sigma \left(u_i^n\right)^3} \tag{20}$$

8. Over half a century has passed since the discovery of the original odd–even hopscotch (OOEH) algorithm [47]. Its temporal and spatial organization has been described in [45]. After the first step by the FTCS formula (which is based on explicit Euler time discretization, so $\theta = 1$) for the odd cells, the BTCS formula (which is based on implicit Euler time discretization) is used for the even cells. The labels odd and even are interchanged after each time step. We modify this method here to include the convection component, which is always considered at the new time level for enhanced stability. The radiation term is handled first explicitly and then implicitly [29]. These are the equations that are being used:

$$\text{First stage}: \qquad u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i - \Delta t \sigma \left(u_i^n\right)^4}{1 + \Delta t K} \tag{21}$$

$$\text{Second stage}: \qquad u_i^{n+1} = \frac{u_i^n + A_i^{\text{new}}}{1 + r_i + \Delta t K + \Delta t \sigma \left(+u_i^n\right)^3}. \tag{22}$$

9. NS-OOEH algorithm:

In addition, we try to alter the first (Explicit Euler) stage of the OOEH to make the terms for convection and radiation appear in the denominator to improve stability. The first stage, instead of (21), is as follows:

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

The second stage is the same as in Equation (22).

2.5. Professional Solvers Used for Comparison Purposes

We compare our results with the direct, iterative, and programmable solvers available in student version of ANSYS 2023 R1 Academic software based on the finite element method (FEM).

(1) The direct solver gives the exact solution to the system of equations defining the FE model. For the system represented by the equation $[K] \cdot \vec{x} = \vec{b}$, the exact solution is

$\vec{x} = [K]^{-1} \cdot \vec{b}$, where $[K]^{-1}$ is the inverse of the matrix $[K]$. The high computational cost of finding the inverse of a matrix means that direct solvers do not usually calculate the inverse, but use LU decomposition to solve the equation;

(2) To provide an approximate solution within a certain convergence tolerance, iterative solvers assume an initial solution and iterate until they converge. Therefore, if the convergence tolerance is 0.01%, the solver will repeat until the difference between the current and past estimates of the solution is less than 0.01%.

The direct solutions tend to be more robust for complex systems and low-quality grids but require a relatively lot of memory. Iterative solvers are generally more efficient, because they use much less memory. Sometimes, however, they fail to converge even for a well-built model. Therefore, the requirements and capabilities should guide the choice of solver type. It is preferable to use an iterative solver to solve the model because that is ideal. The use of a direct solver is recommended if there is trouble with convergence. If the size of the model is quite large and the computer has limited RAM, it will need to use an iterative solution.

(3) The programmable solver is controlled by a program and employs a mixture of direct and iterative solvers.

MATLAB solvers have been used for comparison purposes, namely ode15s, ode23t, ode23tb, ode23, ode45, and ode113. While implicit solutions are used for the other odes, it is known that odes 45, 23, and 113 employ explicit methods. Since the time step sizes cannot be calculated explicitly for the MATLAB solvers, we instead define the tolerances, beginning with a large value, such as Tol $=10^{-1}$ until an extremely small minimum value, which is Tol $=10^{-12}$. We note that an additional solver, namely ode23s, is also available in MATLAB, but in our previous work [40,48], it is proved to be really slow for these problems; thus, we omitted it.

All the running times are measured on a desktop computer with an Intel Core i7-11700F (16 CPUs), and 64 GB RAM is used, whereas the program we used is MATLAB R2020b. The built-in tic-toc timer of that program is used to keep track of how long the algorithms have been running. We present a list of the names and a short description of the methods used in Table 1.

Table 1. Names and short descriptions of the methods used.

Abbreviation	Name/Description of the Method
DF-D	Dufort–Frankel, where the Sigma and K terms are only in the denominator
DF-M	Dufort–Frankel, where the Sigma and K terms are present in a mixed way
DF-KD	Dufort–Frankel, where the K term turns up only in the denominator, and the Sigma term is present in a mixed way
DF-SD	Dufort–Frankel, where the Sigma term turns up only in the denominator, and the K term is present in a mixed way
OOEH	Original odd–even hopscotch
NS-OEH	OOEH with the non-standard treatment of convection and radiation
LH Pseudo-Imp	Leapfrog–hopscotch scheme with pseudo-implicit treatment of the Sigma term
LH Inside	LH with inside treatment of the Sigma term (it appears in the numerator)
LH Mixed	LH with a combination of the pseudo-implicit and inside treatment
ANSYS-M	Mixed solver (controlled by ANSYS)
ANSYS-D	Direct solver
ANSYS-I	Iterative solver

Table 1. *Cont.*

Abbreviation	Name/Description of the Method
ode15s	A 1 to 5-order numerical differentiation formula with variable-step and variable order (VSVO), that was designed for stiff problems
ode23t	Applies the trapezoidal rule with a free interpolant
ode23tb	Uses trapezoidal rule in the first stage and a backward differentiation formula in the second one
ode23	Second (third) order Runge–Kutta–Bogacki–Shampine method
ode45	A fourth (fifth) order Runge–Kutta–Dormand–Prince solver
ode113	1-13 order VSVO Adams–Bashforth–Moulton solver

3. Numerical Simulation

3.1. Geometry and Material Properties

A piece of wall is considered with dimensions 1 m in the x and z directions and 0.02 m in the y direction, as can be seen in Figure 2. Two geometries are taken into consideration:

(a) One-layer of brick is examined only for verification case;

(b) Two-layers consisting of brick and rigid polyurethane foam insulation with a straight thermal bridge steel beam for running time measurements.

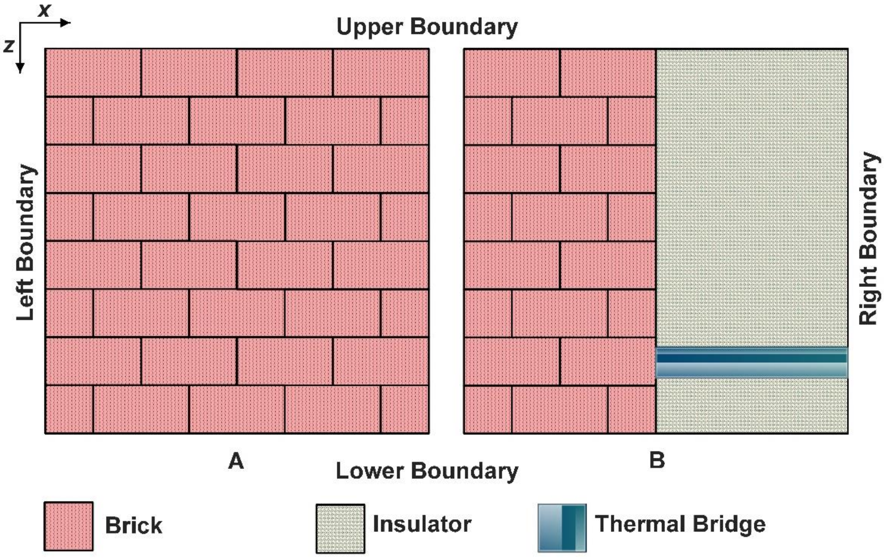

Figure 2. (**A**) One-layer wall. (**B**) Two-layer wall with brick-insulator and a thermal bridge.

In the current work, the real material properties listed in Table 2 are taken into account. Note that these coefficients are constants inside a material, that is, they do not change with time, space, or temperature, but they have a discontinuity at the border of the materials.

Table 2. The properties of the materials used.

	ρ $(\mathbf{kgm^{-3}})$	c $(\mathbf{Jkg^{-1}K^{-1}})$	k $(\mathbf{Wm^{-1}K^{-1}})$
Brick	1900	840	0.73
Rigid Polyurethane Foam	320	1400	0.023
Steel Beam	7800	840	16.2

3.2. Mesh Construction

We considered a dimension of wall 1 m × 1 m × 0.02 m in the calculations of Section 4. The y-direction is orthogonal to the surface of Figure 2A,B, and we use the approximation where no physical quantities change in that y-direction; as a result, this dimension from the mathematical point of view is irrelevant. In mathematical terms, this means that we only need to handle a two-dimensional problem (the cross-section) and can use $\Delta y_i = 0.02$ m. So, we have constructed three versions of meshes of size 1 m^2, which means $(x,z) \in [0,\ 1] \times [0,\ 1]$. The cells have a square or rectangular shape as shown in Figure 3.

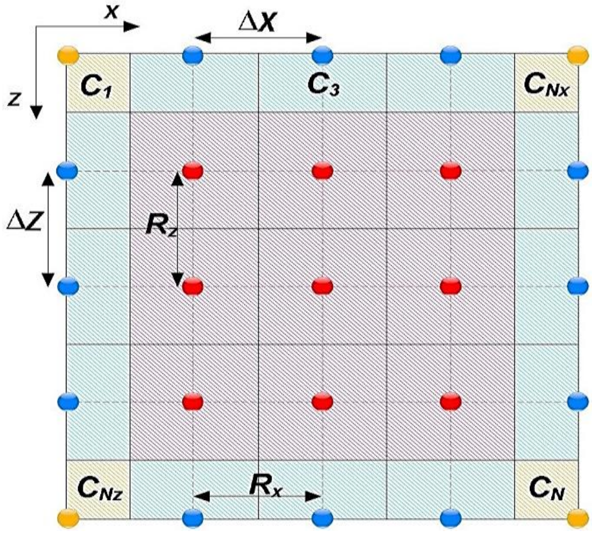

Figure 3. Visualization and Arrangement of the generalized variables.

We apply an equidistant grid to discretize the space variables in all cases. The computations were performed for three different uniform grids, namely, coarse = 40 × 40 = 1600, medium = 80 × 80 = 6400, and fine = 120 × 120 = 14,400. We have to note that the temperature is calculated at the nodes to be comparable to the ANSYS program, which is based on the finite element method, thus, we have a grid with these three grids.

The heat capacity of the cells around the nodes that are not at the boundary may be expressed in the case of a homogeneous material as $C_i = c_i\rho_i\Delta x_i\Delta z_i\Delta y_i$. If the material properties are different, we can write it as follows $C_i = c_i\rho_i\frac{\Delta x\Delta z\Delta y}{2} + c_{i+1}\rho_{i+1}\frac{\Delta x\Delta z\Delta y}{2}$. It is calculated at the upper, lower, left, and right borders by multiplying the heat capacity by a factor of $^1/_2$ and at the corners by multiplying it by a factor of 1/4 .

While the thermal resistance in the x-direction between two nodes has the approximate equation $Rx_i \approx \frac{\Delta x}{k_i Sx_i}$, where Sx_i is the surface element orthogonal to x, which can be given as $Sx = \Delta y\Delta z$ if the nodes are not at the boundary. The horizontal and vertical resistances can be calculated in the case of a homogeneous material as

$$Rx \approx \frac{\Delta x}{k\Delta z\Delta y} \text{ and } Rz \approx \frac{\Delta z}{k\Delta x\Delta y},$$

but it is calculated at the upper, lower, left, and right borders as

$$Rx \approx \frac{\Delta x}{k(\Delta z/2)\Delta y} \text{ and } Rz \approx \frac{\Delta z}{k(\Delta x/2)\Delta y},$$

respectively. If the material properties are different, we can write for the resistance between two nodes i and $i + 1$ that

$$Rx_{i,\,i+1} \approx \frac{\Delta x}{\left(\frac{k_i+k_{i+1}}{2}\right)\Delta z\Delta y},$$

and since the node $i + Nx$ is below the node i, we have

$$Rz_{i,\,i+Nx} \approx \frac{\Delta z}{\left(\frac{k_i+k_{i+Nx}}{2}\right)\Delta x\Delta y},$$

for the vertical resistance.

Also, it is calculated at the borders as

$$Rx_{i,i+1} \approx \frac{\Delta x}{\left(\frac{k_i+k_{i+1}}{2}\right)\frac{\Delta z}{2}\Delta y} \text{ and } Rz_{i,i+Nx} \approx \frac{\Delta z}{\left(\frac{k_i+k_{i+Nx}}{2}\right)\frac{\Delta x}{2}\Delta y}$$

3.3. The Initial and the Boundary Conditions

For running time measurements, zero Neumann boundary conditions have been applied to all borders, preventing the passage of conductive heat at the boundaries. To do this, we set zero for the matrix components describing heat conduction across the border by setting the relevant resistances to infinity.

On the left and right sides of the wall, convective and radiative heat transfer have been assumed. The interior components cannot lose or absorb heat via radiation or convection, and there is no heat source in those elements. Table 3 shows that elements on the left and right sides may transfer heat through radiation and convection in the x direction. The final time t_{fin} = 20,000 s indicates the end of the time interval that is examined; additionally, the time step size is specified in seconds.

Table 3. The convection, radiation, and heat source parameters on both sides of the wall components in the case of the two-layer wall.

	$h_c\ \left[\frac{\mathbf{W}}{\mathbf{m^2K}}\right]$	ε	$\sigma^*\ \left[\frac{\mathbf{W}}{\mathbf{m^2K^4}}\times\mathbf{10^{-8}}\right]$	$q^*\ \left[\mathbf{W/m^2}\right]$
Right Elements	22	0.8	4.5	435.39
Left Elements	9	0.9	5.1	360.95

As can be seen in Table 3, we have used numbers from publication [42] for the convection heat transfer coefficient h_c. The Stefan–Boltzmann constant is a universal constant for radiation: $5.67 \cdot 10^{-8} \frac{\text{W}}{\text{m}^2\text{K}^4}$. Since the surface is not a perfect black body, we have to multiply the Stefan-Boltzmann constant by an emissivity constant in order to get results that are more in line with realistic values for σ^*. The value of q^* is estimated for the definition of "heat source", which includes solar radiation in the table below. It is assumed that the ambient air temperature on the left side is $17\ ^\circ\text{C} \approx 290\ \text{K}$.

Because of the nonzero temperature of the air u_a, the term q also includes heat gain from convection; this allows us to calculate the value of q as follows. The values of the coefficients in our Equations (2) and (3) are calculated using the values we obtain:

$$K = \frac{h_c}{c\rho\frac{\Delta x}{2}},\ \ \sigma = \frac{\sigma^*}{c\rho\frac{\Delta x}{2}},\ \ q = \frac{q^*}{c\rho\frac{\Delta x}{2}} + \frac{h_c}{c\rho\frac{\Delta x}{2}}u_{\text{a}}$$

We assumed that the left and right elements have the following heat sources as follows:

$$\text{In terms of the left} - \text{hand side}:\ q = \frac{1}{c\rho\frac{\Delta x}{2}} \times 360.95\frac{\text{W}}{\text{m}^2} + \frac{h_c}{c\rho\frac{\Delta x}{2}} \times 290\ \text{K}$$

$$\text{In terms of the left} - \text{hand side}:\ q = \frac{1}{c\rho\frac{\Delta x}{2}} \times 435.39\frac{\text{W}}{\text{m}^2} + \frac{h_c}{c\rho\frac{\Delta x}{2}} \times 313\ \text{K}\frac{\partial^2\Omega}{\partial u\partial v}.$$

The constant initial temperature is prescribed: $u(x,z,\ t=0) = 290\ K$.

3.4. Verification Using Analytical Solutions

For verification, we used three different cases. In the first one, we considered a one-layer wall made of homogeneous material (see Figure 2A) and applied a sinusoidal initial condition with a zero Dirichlet boundary condition to examine the conduction term only. Then in the second and third cases, we considered spatially homogeneous initial temperature to exclude conduction, which means we have only one temperature in the system, that is, the behavior can be described by an ODE. In the second case, we took into account only convection, which is perpendicular to the surface and happens in the y direction, and in the third case, we considered only radiation perpendicular to that surface. After that, we compared our results in MATLAB and ANSYS with the analytical solution.

- First verification (for conduction):

1. Two sine functions are multiplied to provide the initial condition:

$$u(x,z,t=0) = \sin(k_x\pi x)\sin(k_z\pi z)$$

2. Zero Dirichlet boundary conditions are used:

$$u(x=0,z,t) = u(x=1,z,t) = u(x,z=0,t) = u(x,z=1,t) = 0$$

3. One can quickly verify that the problem's analytical solution is

$$u(x,z,t) = \sin(k_x\pi x)\sin(k_z\pi y)\mathrm{e}^{-\alpha({k_x}^2+{k_z}^2)\pi^2 t} \tag{23}$$

where the wave numbers are fixed to $k_x = 1$, $k_z = 1$, and substituting the physical properties of the brick, we obtained α. We used the analytical solution Equation (23) at $t_{\text{fin}} = 2000(\text{s})$ as the reference solution.

- Second verification (for convection and heat generation)

Since the temperature is spatially homogeneous, we have the simple ODE, valid for each node:

$$\frac{du}{dt} = -K\cdot u + Q$$

Its analytical solution is:

$$u(x,z,t) = \left(\frac{Q}{K}\right) + \left[u_0 - \frac{Q}{K}\right]\cdot \mathrm{e}^{-Kt}$$

where the initial condition here is a constant temperature that equals to 290 K.

- Third verification (for radiation)

We have the ODE:

$$\frac{du}{dt} = -\sigma\cdot u^4$$

with the analytical solution:

$$u(x,z,t) = \left({u_0}^{-3} + 3\sigma T\right)^{-\frac{1}{3}}$$

The error is defined as the largest difference in absolute terms between the reference temperature $u_{\mathrm{i}}^{\mathrm{ref}}$, which is the analytical solution and the temperature $u_{\mathrm{i}}^{\mathrm{num}}$ obtained by the studied numerical method at $t_{\text{fin}} = 2000$ s:

$$MaxError = \max_{1\le i\le N}\left|u_i^{\mathrm{ref}}(t_{\text{fin}}) - u_i^{\mathrm{num}}(t_{\text{fin}})\right|. \tag{24}$$

For all systems, the obtained results in the case of MATLAB and ANSYS are very similar to the analytical solution, and the error is below 10^{-5}. This means that the MATLAB code using the ode15s solver, as well as the ANSYS solver, has been verified. The temperature contour is presented in the Supplementary Materials (see Figure S1) as a function of the x- and z-coordinates in the case of ANSYS and the analytical solution. The temperature as a function of the x-coordinate for z = 0.5 m was plotted in the case of ode15s, ANSYS, and the analytical solution for the coarse grid. These lines are so close to one another that they are indistinguishable by the naked eye, (see Figure S2 in the Supplementary Materials). For other system sizes, the same behaviour is experienced. In the case of the second and third verification, the maximum error is around 10^{-9} between ode15s and the analytical solution, but it is slightly larger than 10^{-5} between ANSYS and the analytical solution, as shown in Table 4. Since the MATLAB ode15s solver is proven to be more accurate in all cases, we choose this as a reference solution when we measure and compare the running times in the next section.

Table 4. The comparison between MATLAB and ANSYS at different mesh sizes and boundary conditions.

The Type of Boundary Condition	Grid Type	The Maximum Error between Analytical Solution	
		MATLAB	ANSYS
Zero Dirichlet	coarse	9.114×10^{-6}	3.577×10^{-5}
	medium	2.279×10^{-6}	2.097×10^{-5}
	fine	1.013×10^{-6}	1.863×10^{-5}
Only convection on the surface	medium	1.24×10^{-10}	8.49×10^{-5}
Only radiation on the surface	medium	4.04×10^{-9}	6.344×10^{-5}

4. Results: Comparison of Performances by Measuring the Running Times

4.1. Comparison with MATLAB Methods and ANSYS Solvers for the Coarse Mesh System

The two-layer wall (brick + insulation) with the straight thermal bridge (see Figure 2B) is simulated using the coarse mesh and the conditions in Section 3.3, and values from Table 2. The maximum errors are shown in Figure 4 as a function of the time step size. The comparison of the running times of the tested methods and three solvers of ANSYS are shown in Figure 5, and the temperature distributions are shown in Figure 6. We can see that for the coarse mesh, the LH pseudo-implicit treatment of radiation was the most accurate, but two of the DF versions and the three solvers of ANSYS are not really accurate. Regarding speed, we can say that the explicit and stable schemes are clearly faster than ANSYS, but they are more efficient than the MATLAB routines only if small or moderate accuracy is required.

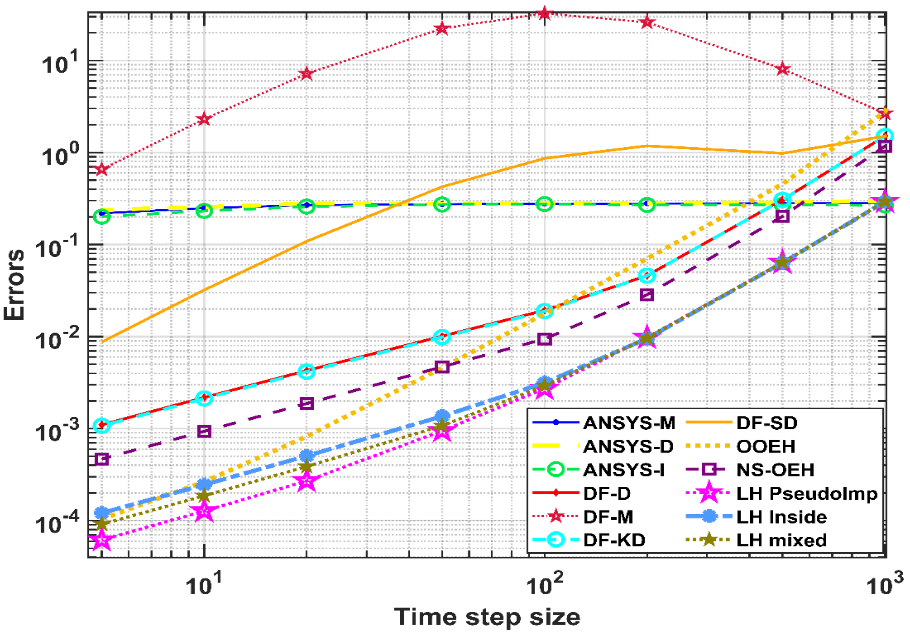

Figure 4. The maximum errors as a function of the time step size for the tested methods in the case of the coarse mesh.

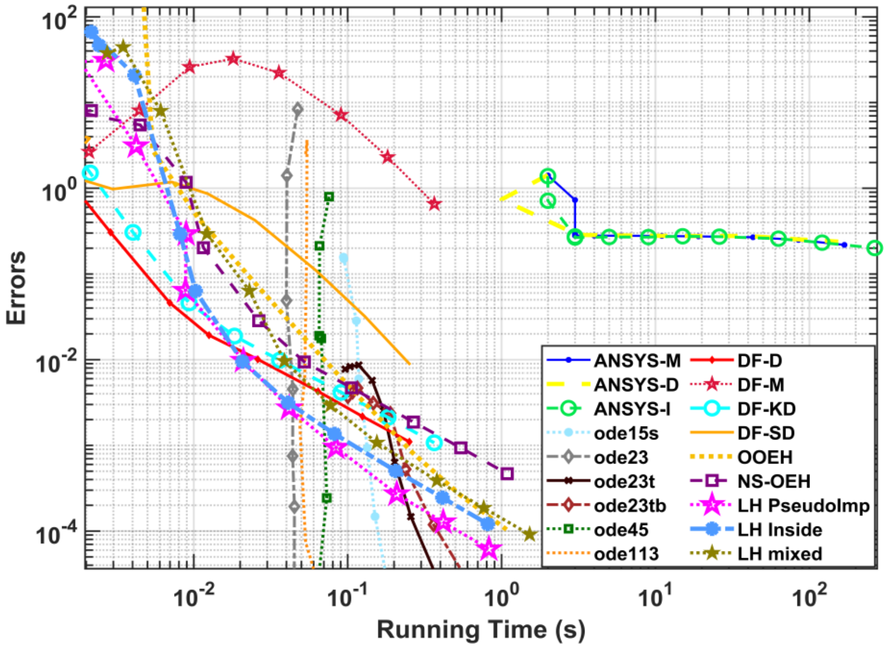

Figure 5. The maximum errors as a function of the running time for the tested methods in the case of the coarse mesh.

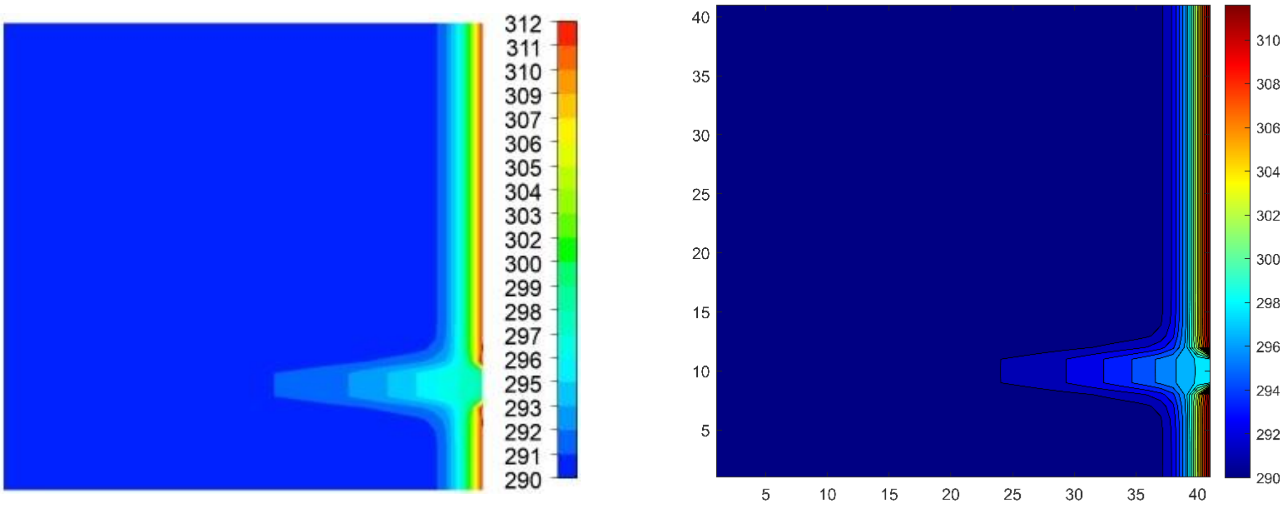

Figure 6. The temperature distribution contour in Kelvin units for the measurement of the running time in the case of the coarse mesh (**left**) for ANSYS and MATLAB on (**right**).

4.2. Comparison with MATLAB Methods and ANSYS Solvers for the Moderate Mesh System

Now a moderately fine resolution mesh is applied, and the same conditions and values are used as for the previous coarse grid. Most of the proposed methods, especially the LH-PI is again better than ANSYS solvers and MATLAB routines for both accuracy and speed. Figure 7 compares the running times of the three ANSYS solvers with the tested methods in MATLAB. (We note that the errors as a function of the time step size can be seen in Figure S3 in the Supplementary Materials).

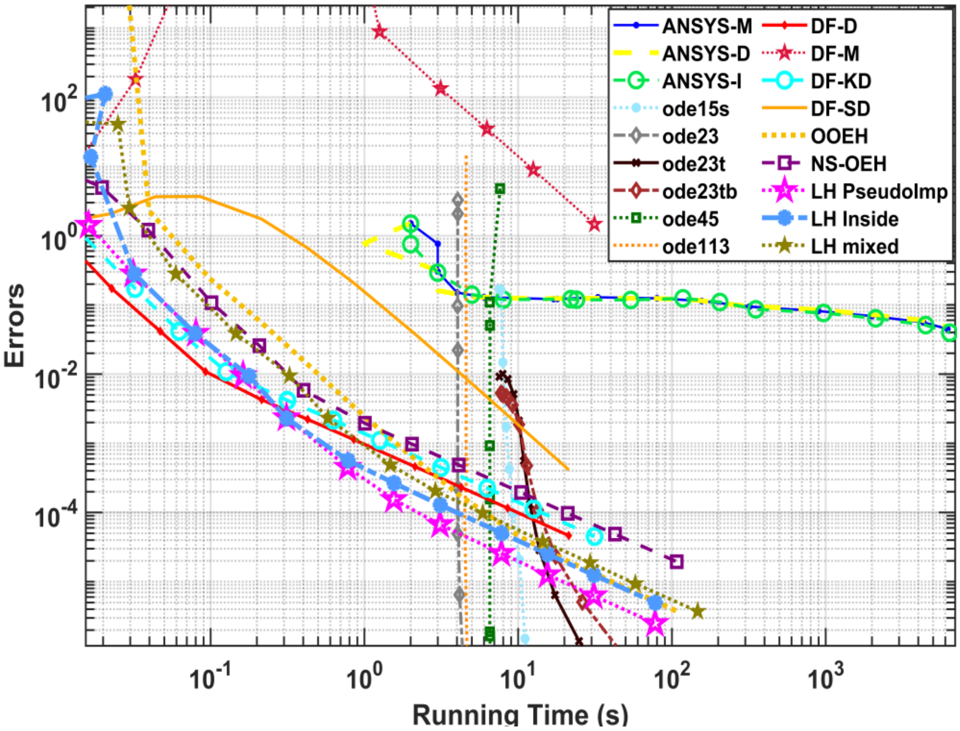

Figure 7. The maximum errors as a function of the running time for the tested methods in the case of the medium mesh.

4.3. Comparison with MATLAB Methods and ANSYS Solvers for the Fine Mesh System

For the fine mesh, the errors are plotted in Figure 8 as a function of the running time and in Figure S4 (Supplementary Materials) as a function of the time step size. We notice that the accuracy of the ANSYS solvers is positively affected by the smoothing of the mesh, the errors are decreasing with the time step size. In spite of this, some of the proposed explicit methods coded in MATLAB remain the best in both accuracy and speed. The solvers of ANSYS are very slow if we have a big system, but the built-in routines of MATLAB are getting slower at a larger rate with increasing system size, so we think that for even finer mesh, they would be the slowest. The leapfrog–hopscotch with the pseudo-implicit treatment of the radiation term (LH-PseudoImp) and the Dufort–Frankel schemes with the pseudo-implicit treatment of both the convection and the radiation term (DF-D) are the two best methods. Their advantages are in fact, increasing with increasing system size. That is why, in the next section, we choose these two methods to make a long-term simulation.

For mesh-independence, a horizontal line is considered in the middle of the thermal bridge (z = 0.225 m), and along this line, the temperature of the LH pseudo-Imp method is plotted for all three meshes. The result is presented in the Supplementary Materials (see Figure S5). As one can see, the values of the medium mesh are close to those of the fine mesh, where the maximum difference was quite small 0.097 in Kelvin units, so to reduce the computational cost, we can adapt the medium mesh in long-term calculations.

We also compared three of the solvers, the iterative ANSYS, the LH-PseudoImp, and ode15s using the medium mesh. The maximum difference between LH-PseudoImp and ode15s is equal to 7.3×10^{-7} in Kelvin units, but ANSYS has a little noticeable difference from them, which is equal to 0.2 K. The figure is presented in the Supplementary Materials (see Figure S6).

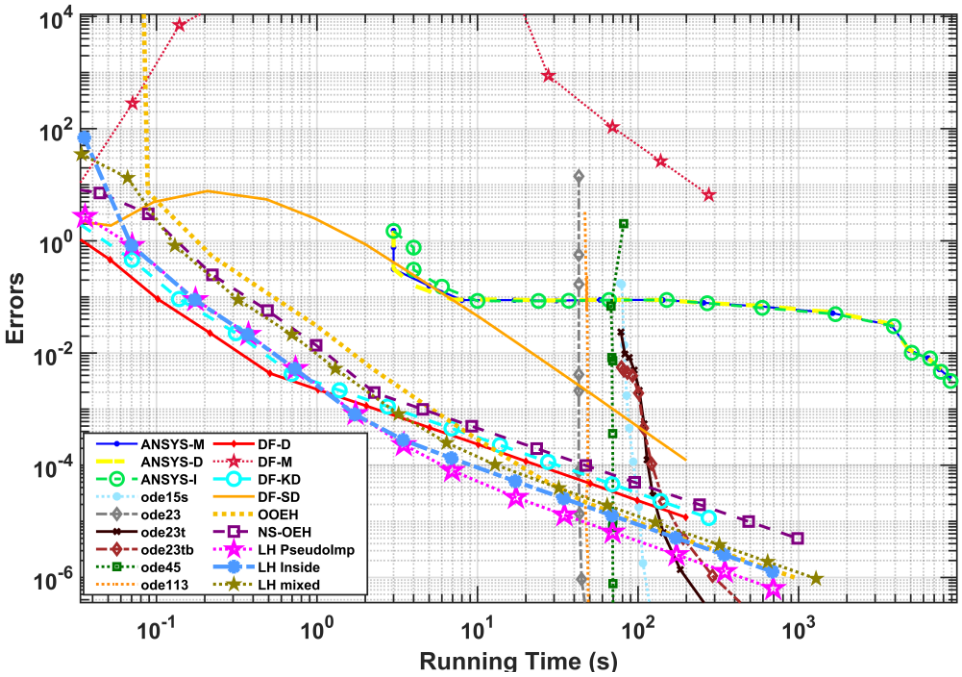

Figure 8. The maximum errors as a function of the running time for the tested methods in the case of the fine mesh.

Finally, we examined how the running times depend on the system size. We obtained that the running time of our methods increases linearly with the number of nodes, but it increases faster than linearly for other solvers. Figure S7 in the Supplementary Materials shows the concrete data for one of our best methods with one of the MATLAB routines ode23t, and ANSYS solvers.

Nevertheless, we absolutely do not intend to suggest that ANSYS and similar solvers are redundant since there are faster numerical algorithms than those they employ. This simulation software automatically generates the mesh and presents graphically the results, whereas if someone chooses to write their own code, it is tedious work, especially in the case of complicated geometries. Moreover, the explicit and stable methods have not been applied to cases involving the simulation of the motion of the air, for example, for the problem of air leakage in building components, whereas the simulation software routinely handles these problems. However, we think that in the already well-established cases, such as those types studied in this paper, the simulation software could incorporate the proposed algorithms as an option can be chosen by the users or by an artificial intelligence.

5. Long-Term Simulations

5.1. Geometry and Mesh

The simulations were performed for a standard residential wall in Miskolc, Hungary, and the outdoor temperature, convection coefficient, and solar radiation values for Miskolc were used in the simulations where the data was taken from the website every 3 h. Then we used linear interpolation to calculate the values for every 100 s.

The month of January is the coldest. The highest temperature measured in January 2022 was 11 °C, whereas the lowest was −6 °C. Depending on the topography, the predominant wind might blow in various different directions. The maximum measured wind speed was 10.8 m/s [49].

We applied the best methods, namely, the LH-PI and DF-D, to calculate temperatures and heat losses across the wall, with a fixed time step size of $\Delta t = 100$ s, where the total time was $T = 31 \cdot 24 \cdot 3600 = 2,678,400$ s. We chose this time step size because from Figure 4, it can be seen that the error of the best methods is around 0.01 °C, so sufficient accuracy is reached. We stress again that the CFL limit for explicit Runge-Kutta methods is orders of magnitude smaller than this time step size. The difference between the DF-D and LH-pseudoImp was continuously monitored during the simulation and found to be very small, so the presented results are for the LH-pseudoImp method. The running times for long-time simulation was 145.5 s for the case of two-layers with the straight thermal bridge case on the same computer which was used in the previous section.

Figure 9 illustrates the different models of the wall. First, one layer has been used, which is only brick; the dimension of the wall height, width, and thickness is (0.45, 1, 1 m) as shown in Figure 9A. In the second model, there are two layers: the brick wall has the same dimension as in the first model, and in addition to it, there is an insulation layer with a thickness of 0.15 m, as shown in Figure 9B. The third model is the same as the second one with a straight thermal bridge. The width of the thermal bridge is the same as that of the insulator, as shown in Figure 9C. The horizontal position of the thermal bridge is between $x = 0.45$ m and $x = 0.6$ m. The vertical position of the top of the thermal bridge is $z = 0.75$ m (25 cm from the bottom of the wall), and the thickness of the bridge is 5 cm in the z direction.

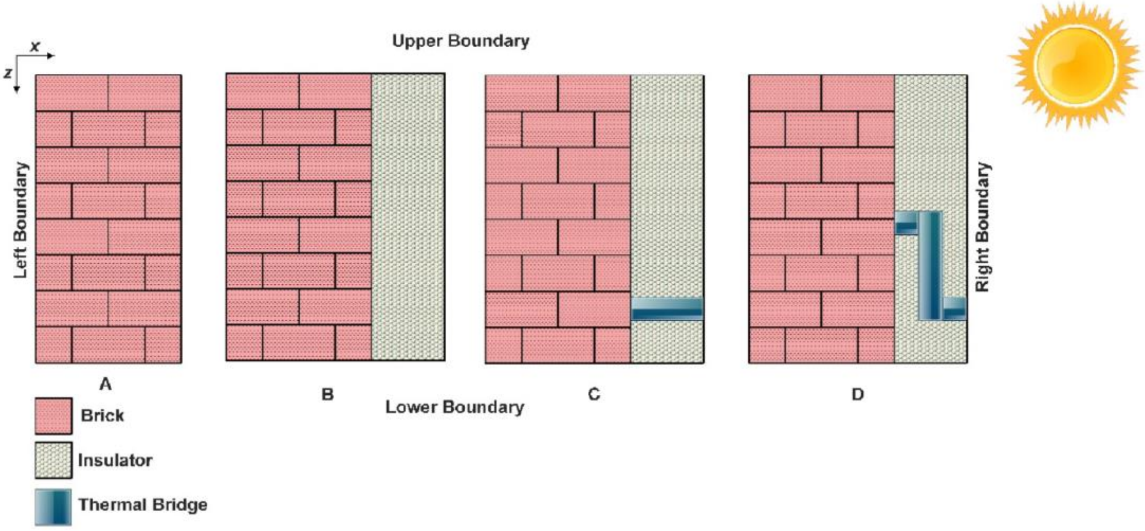

Figure 9. (**A**) One-layer wall, (**B**) Two-layers wall with brick-insulator, (**C**) Two-layers with straight thermal bridge, (**D**) Two-layers with bent thermal bridge.

The fourth model contains the same two layers as the second one, but the bar thermal bridge has a curved shape. The thermal bridge is made up of three straight bars, two horizontal and one vertical. The first horizontal bar is connected from the outside at the same vertical position as the straight bridge above. It goes horizontally with a length 0.05 m. It is connected to the vertical bar, which has a length 0.3375 m in the z-direction. It then connects to the other horizontal bar, which has the same dimensions as the first horizontal bar, whereas the vertical position of the bottom of this piece is 0.5125 m as shown in Figure 9D. The medium mesh is applied for all cases, but in the first case, it means $\Delta x = 0.0057$, whereas in the other cases, the widths of the cells are larger, $\Delta x = 0.0076$, since the wall is thicker.

We are going to examine which thermal bridge yields a higher rate of heat transfer and more extra heating cost. It is a nontrivial question since, in the case of the straight bridge, the way of heat is shorter, but in the case of the bent bridge, the average conductivity of the wall is higher.

5.2. Initial and Boundary Conditions

Zero Neumann boundary conditions have been applied to all borders and wall cases, and convective as well as radiative heat transfer are assumed on the right and left sides. As in Section 4, the interior parts cannot lose or absorb heat via convection, radiation, and heat source. As it is indicated in Table 5, elements on the right and left sides can transfer heat through convection and radiation in the x direction. The left elements have constant values for the convection, radiation, and heat source parameters, whereas the right elements have changing values depending on the external conditions. In this part, the final time is t_{fin} = 2,678,400 s, which indicates the simulation of a full winter month.

Table 5. The convection, radiation, and heat source parameters are on both sides of the wall for all the previously mentioned wall types [42].

	$h_c \cdot \left[\frac{\mathbf{W}}{\mathbf{m^2K^4}}\right]$	ε	$\sigma^* \cdot \left[\frac{\mathbf{W}}{\mathbf{m^2K^4}} \times \mathbf{10^{-8}}\right]$
Right Elements	0.6–22.45	0.9	5.1
Left Elements	9	0.7	3.97

We obtain the values of the coefficients in our equations as follows: [42],

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

We also supposed that the left and right elements have the heat sources as follows:

$$\text{For the interior side elements}: q_l = \frac{1}{c\rho} \times q_l^* + \frac{h_{cl}}{c\rho \cdot \Delta x} \times 295\ \ \text{K}$$

$$\text{For the external side elements}: q_r(t) = \frac{1}{c\rho} \times q_r^*(t) + \frac{h_{cr}(t)}{c\rho \cdot \Delta x} \times u_r(t)$$

$$\text{And } K(t) = \frac{h_{cr}(t)}{c\rho\,\Delta x},\ \ \sigma = \frac{\sigma_r{}^*}{c\rho\,\Delta x},\ \ q(t) = \frac{q_r{}^*(t)}{c\rho\,\Delta x} + \frac{h_{cr}(t)}{c\rho \cdot \Delta x} \cdot u_r(t)$$

where $q_l^* = \varepsilon_l \sigma_l (295)^4$ and $q_r^*(t) = \alpha_{\text{sun}} G_{cr}(t) + \alpha_{\text{Low}} \varepsilon_r \sigma_r [u_r(t)]^4$ [50].

The convection heat transfer coefficient for outside elements as a function of air velocity is estimated as follows [51]:

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

$v(t)$: The air velocity is taken for each 100 s in January [m/s].
$u_r(t)$: The outside air temperature for each 100 s in January month [°C].
u_l: The inside air temperature [°C] on the left side.
$G_{cr}(t)$: The solar radiation was taken for each 100 s in January [W/m^2].
α_{sun}: The absorptivity of brick surface to solar radiation.
α_{Low}: The absorptivity of brick surface to low-temperature thermal radiation.

The environment air temperature is taken to be 22 °C ≈ 295 K inside, and changing depending on weather conditions outside.

We calculated the initial temperature inside the wall using the assumption that before the simulation time, a stationary heat flow with constant flux evolved between the given boundary values of the internal and external air temperatures. In the case of one-layer, it yields a linear function of the x variable for the initial condition:

$$u(x,z,\ t=0) = (u_{\text{r,initial}} - u_l)x/L_x + u_l$$

where $u_{\text{r,initial}} = 278\ \ K$.

For the remaining three cases, the assumption of stationary heat conduction with the initial values at the boundaries implies that we have to use two linear functions of the x variable for the initial condition:

For the brick part: $u(x,z,\ t=0) = (u_{\text{mid}} - u_l) \cdot x/L_b + u_l$

For the insulator part:

$$u(x,z,\ t=0)=((u_{\text{r,initial}}-u_{\text{mid}})x/L_{\text{ins}})-((u_{\text{r,initial}}-u_{\text{mid}})\cdot L_b/L_{ins})+u_{\text{mid}}$$
$$\text{where } u_{\text{mid}}=u_l-(q_{\text{flux}}L_b/k_b)$$
$$\text{and } q_{\text{flux}}=(u_l-u_{\text{r,initial}})/((L_b/k_b)+(L_{\text{ins}}/k_{\text{ins}}))$$

5.3. Result for the Simulation of the Wall

The results will be displayed for specific points depicted in Figure 10, and the subfigures contain the following information.

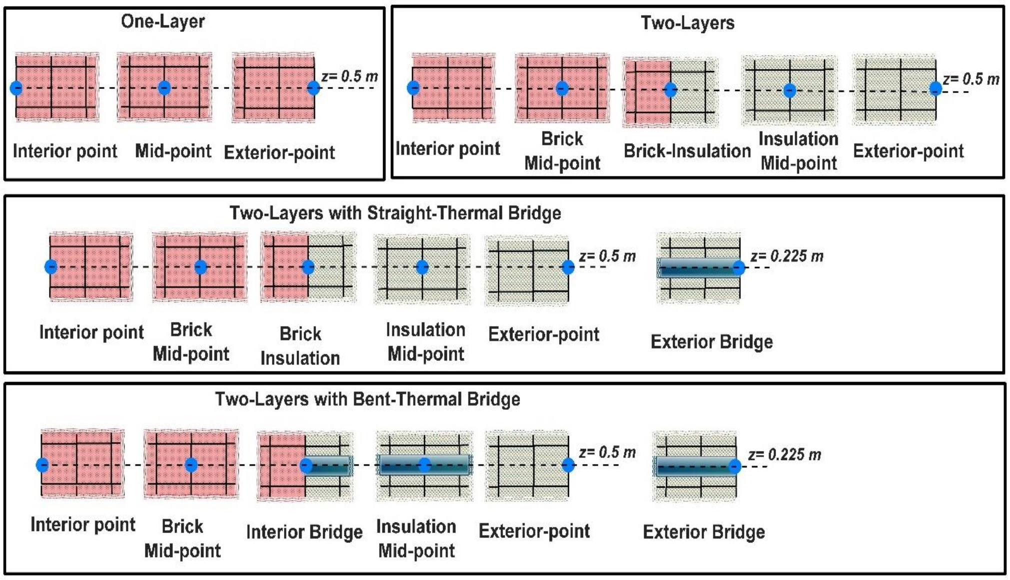

Figure 10. Illustration of the points and grid arrangements that present the results for the different cases.

(A) There are 3 points denoted for the one-layer wall: on the interior surface, at the middle of the wall, and on the exterior surface;

(B) Two-layer wall (brick + insulator) with five points on the interior surface, the middle of the brick, the border of the two layers, the middle of the insulation, and the exterior surface;

(C) Two-layers with a straight thermal bridge with 6 points: on the interior surface, the middle of the brick, the border of the two layers, the exterior surface of the straight thermal bridge, the middle of the insulation, and the exterior surface of the insulation;

(D) Two-layers with the bent thermal bridge with six points: on the interior surface, middle of the brick, the border of the two layers with bent thermal bridge, the exterior surface of the bent thermal bridge, the middle of the insulation, and the exterior surface of the insulation.

5.3.1. One Layer (Brick)

The temperatures as a function of the time are presented in Figure 11. Note that without the presence of the insulator, the temperature of the inner surface of the wall is initially equal to the temperature of the internal air (22 °C), then decreases because of the cooling effect of the wall and the outside cold weather. As for the temperature of the middle of the wall, it is low at first, and then it rises due to heat transfer by conduction from the inside. The external surface temperature is greater than the external air temperature due to the heat flowing to the surface through conduction.

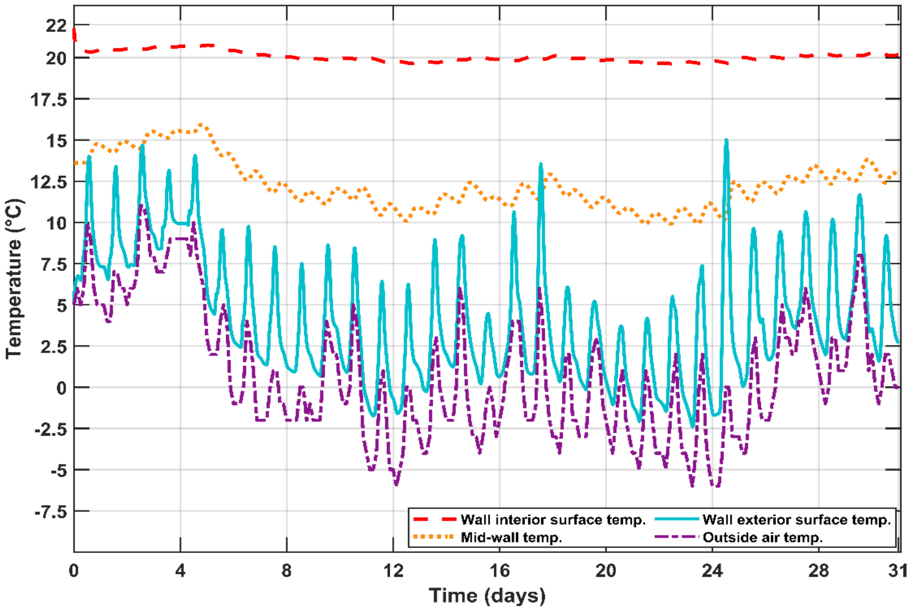

Figure 11. The temperature distribution in Celsius units as a function of time in days for the long-term simulation of the one-layer wall.

5.3.2. Two Layers (Brick and Insulation) without Thermal Bridge

The temperature-time functions are presented in Figure 12. The temperature in the brick part is changing very slowly, but in the insulator part it is changing very sharply. We note that in the presence of an insulator, the temperature of any point of the brick follows the temperature of external air only very slightly. As for the temperature of the middle of the insulator, it rises and falls due to the effects of external conditions, and the minimum values of the outer surface temperatures of the wall are slightly less than the inside air temperature because the insulator limits the flow of heat from inside to outside; for higher values, it is greater owing to the effects of solar radiation. The comparison between the one-layer and two-layer wall cases in terms of the final temperature is presented in the Supplementary Materials (see Figure S8).

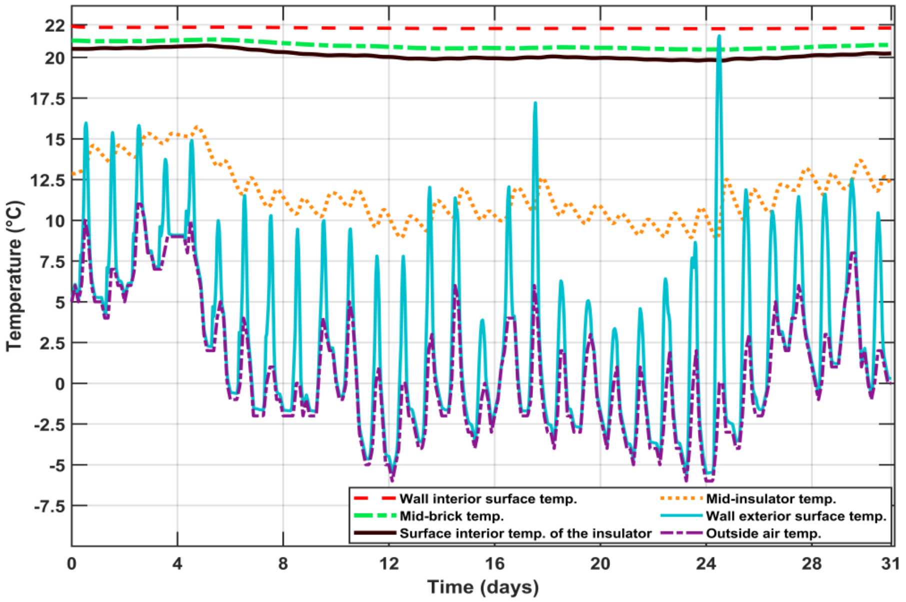

Figure 12. The temperature distribution in Celsius units as a function of time in days for the long-term simulation of the two-layer wall.

5.3.3. Two Layers with Straight Thermal Bridging

Figure 13 clearly shows that the temperature of the middle of the wall and the temperature of the interior surface of the insulator decreased slightly compared to the previous case due to the heat loss caused by the straight thermal bridge. Moreover, the external temperature of the thermal bridge is higher than the temperature of the exterior surface of the insulator due to the flow of heat from inside to outside as well as the physical properties of the material.

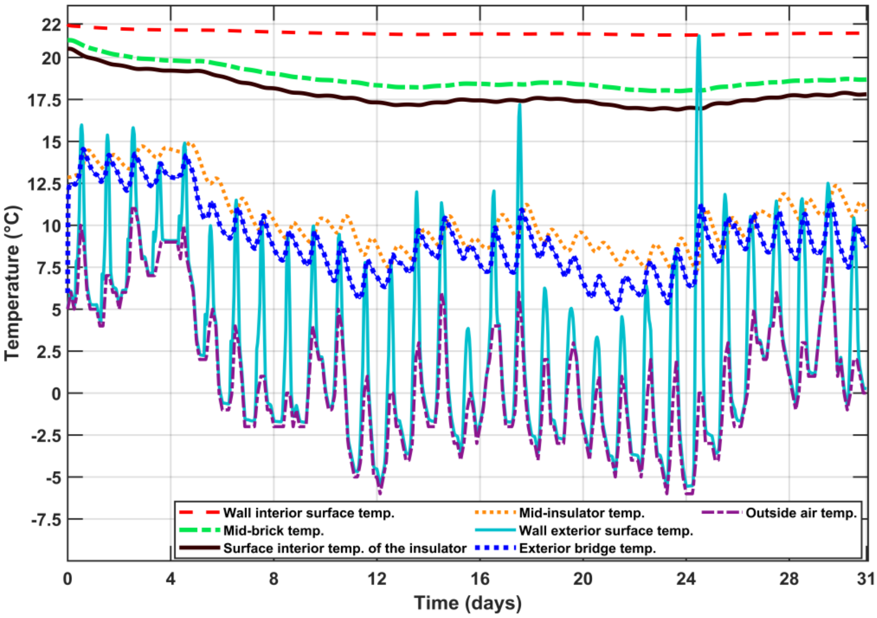

Figure 13. The temperature distribution in Celsius as a function of time in days for the long-term simulation of the wall with straight thermal bridging.

5.3.4. Two Layers with Bent Thermal Bridging

In this case, which includes the thermal bridge created by the bent thermal bridge, the mid-insulator temperature rises (see Figure S9 in the Supplementary Materials). Figure 14 shows the contours of both types of thermal bridge cases, where we note the way of the heat in the bent bridge is longer than in the straight bridge.

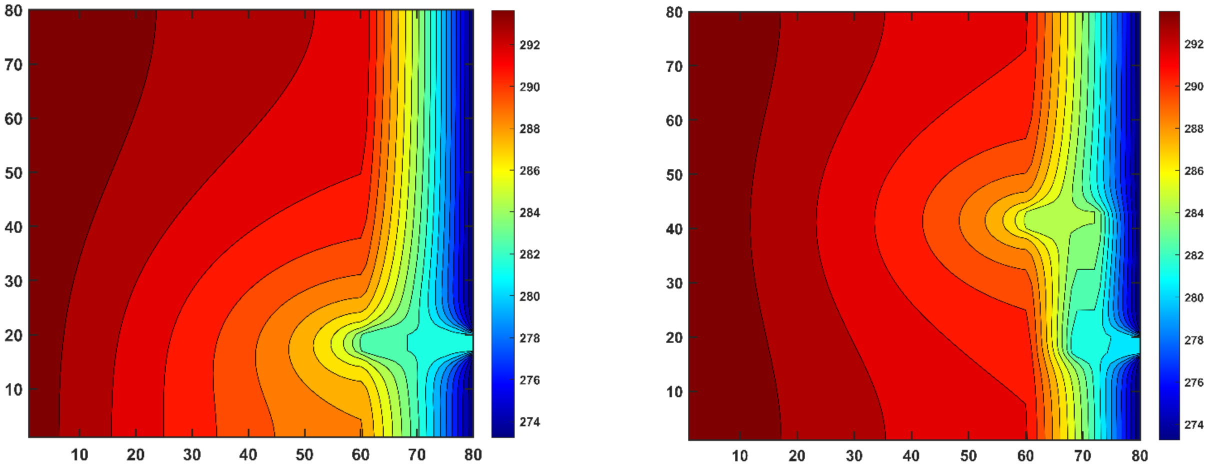

Figure 14. The temperature distribution contour in Kelvin units for the long-term simulation in the case of the medium mesh case (**left**) for straight thermal bridge and bent thermal bridge (**right**) at the end of the last day.

In Figure 15, we compared the temperatures of the external points of the two thermal bridges with the external points of the one- and two-layer cases. We concluded that the straight thermal bridge allows the heat to flow faster than the bent thermal bridge because the passage of heat in the bent bridge is longer than in the straight one.

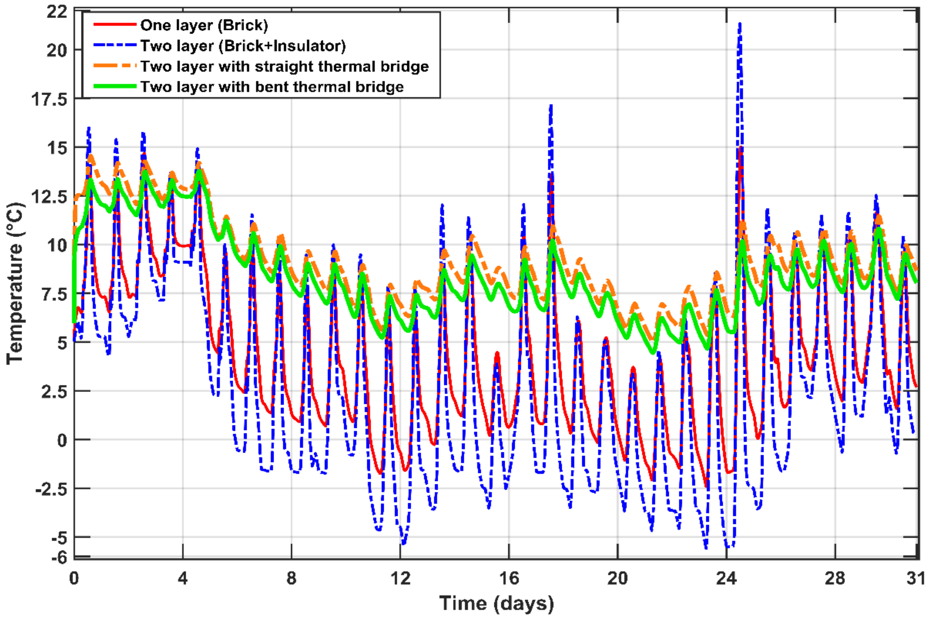

Figure 15. The temperature distribution in Celsius units with days for the long-term wall simulation in case of exterior points of thermal bridging (straight and bent) compared with one-layer and two-layer cases.

Figure 16 shows a comparison between the losses in energy. One can see the largest thermal loss in the case of a one-layer wall, and these losses are less with the presence of the insulator. When a thermal bridge is present, these losses are larger than without a thermal bridge, and the losses are slightly larger in the case of a straight thermal bridge than for the bent bar.

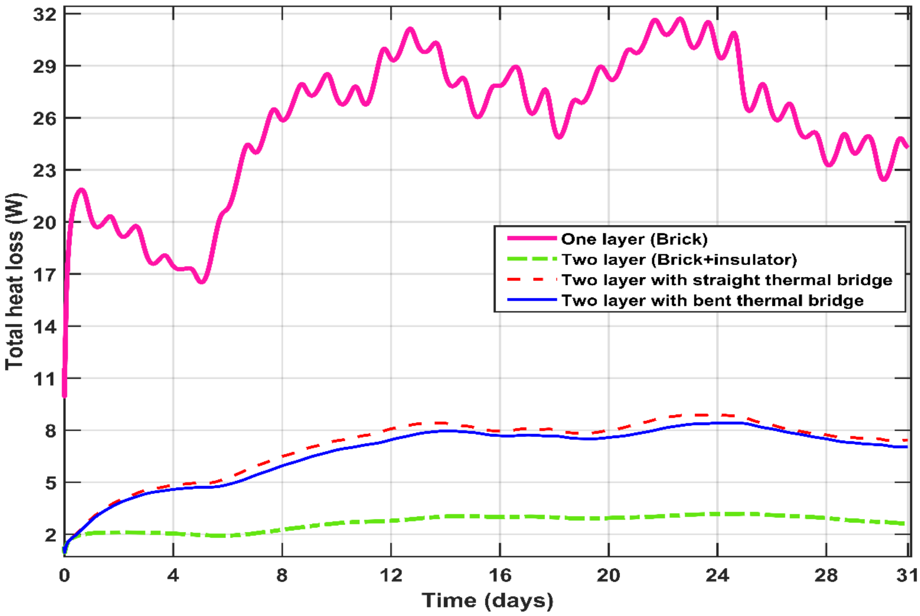

Figure 16. Total heat loss distribution in Watt units as a function of time in days for all cases of the long-term wall simulation for all cases.

5.3.5. Calculation of Heat Loss

The total energy requirement Q in kWh units was calculated from Equation (25)

$$Q = \sum_{t=1}^{Time} \left(\Delta t \sum_{i=1}^{N} Q_{\text{cell},i}(t) \right), \tag{25}$$

where $Q_{\text{cell}} = q^*{}_{\text{cell}} \cdot S$ and q^* is the current density of the heat loss, which can be calculated as $q^*{}_{\text{cell}} = \frac{((T_{i+1} - T_i)K_{\text{cond}})}{\Delta x}$.

Δt is the time step size, and S is the area of the wall in the direction of heat loss.

Now we can calculate the cost of the energy used to make up the energy loss by the multiplication of the current price of electricity (per kWh) used by the amount of heat losses. Table 6 below shows the heat loss and costs of energy consumption.

Table 6. Heat loss through the 1 m^2 piece of wall, and the energy cost in HUF and USD.

	One Layer	**Two Layers**	**Two Layers with a Straight Bridge**	**Two Layers with Bent Bridge**
Heat loss (full month, kWh)	19.14	1.99	5.29	5.01
The cost in HUF	717.19	74.63	198.24	187.8
The cost in USD	1.9	0.2	0.53	0.5

5.4. Simulation of the Coldest Day of the Month

For the design of the heating load, we adopt the hardest weather conditions, and the design is based on them. Therefore, we will review the coldest day of the month, which is 25 January, where the lowest recorded temperature is −6 °C. We also applied the previous best methods to calculate temperatures and heat losses across the wall, but with a new time step size of $\Delta t = 60$ s in this simulation, to track and record the temperature change every minute because the total simulation time here is short, only T = 24 × 3600 = 86,400 s. The temperature as a function of the hours of the day for the exterior points in all cases of the wall, and a comparison between the losses in energy on the coldest day of the month are presented in the Supplementary Materials (see Figures S10 and S11). One can see the largest losses in the case of one layer, and Table 7 contains the cost of energy consumption on this day.

Table 7. Heat loss through the wall and energy cost in HUF and USD for the coldest day.

	One Layer	**Two Layers**	**Two Layers with a Straight Bridge**	**Two Layers with Bent Bridge**
Heat loss (one day, kWh)	0.713	0.076	0.213	0.2
The cost in HUF	26.7	2.85	7.96	7.56
The cost in USD	0.071	0.0076	0.022	0.02

6. Conclusions

The purpose of this article was to develop and test a simulation methodology based on fundamental physical principles and laws to study transient heat transfer in the wall. First, we numerically investigated transient heat transfer in a two-dimensional wall without an insulator, with an insulator, and two types of thermal bridges. For our purposes, we applied and tested nine explicit algorithms coded by us in MATLAB and three solvers included in the commercial software called ANSYS. Then the running time was measured for our methods and compared with the ANSYS solvers and six built-in MATLAB routines. We used three mesh sizes: 40 × 40, 80 × 80, and 120 × 120, which we applied to walls with 1600, 6400, and 14,400 cells, respectively. Three simple analytical solutions of the heat equation were used with an equidistant mesh for verification in the case of homogeneous material properties (one brick layer). All the methods used and the ANSYS solvers are confirmed to be convergent. These experiments suggest that our methods are better than all ANSYS solvers and MATLAB routines, whereas ANSYS was less accurate and slower, and it was observed that the best performance was achieved by the leapfrog–hopscotch and the Dufort–Frankel algorithms with the pseudo-implicit treatment of the nonlinear radiation term. Therefore, these two methods were applied to real problems, and a long-term simulation of four cases was performed. The temperature distribution and total heat losses of all cases were calculated. We found the straight thermal bridge to be energetically worse than others, and the total heat loss during the month (one-layer, two-layer, two-layer with a straight thermal bridge, and two-layer with a bent thermal bridge) was, respectively, 19.14, 1.99, 5.29, and 5.01 kWh for a 1 m^2 wall surface.

We can conclude that the numerical simulation methodology is established in this paper. In our next works, we will use these solvers and approaches to study real problems like real thermal bridges in blocks of flats, and we will use physical experiments as well as standards such as ISO 10211:2017 and ISO 14683:2017 to validate them.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/en16124604/s1, Figure S1: The temperature distribution contour for the sinusoidal initial temperature (left) for ANSYS and analytical solution on (right) in Kelvin units; Figure S2: The temperature comparison for the sinusoidal initial temperature in the case of the coarse mesh; Figure S3: The maximum errors as a function of the time step size for the tested methods in the case of the medium mesh; Figure S4: The maximum errors as a function of the time step size for the tested methods in the case of the fine mesh; Figure S5: Comparison between three types of mesh for the LH-PseudoImp method temperature; Figure S6: Comparison between three solvers for the medium mesh; Figure S7: Running time as a function of the total number of cells for the examined method and other solvers, where the left axis refers to LH-PseudoImp, and the right one refers to ode23t solver of MATLAB and ANSYS; Figure S8: The temperature in Celsius units for the long-term simulation, for one layer and two layers at the end of the last day; Figure S9: The temperature in Celsius units as a function of time in days for the long-term simulation of the wall with the bent thermal bridging; Figure S10: The temperature in Celsius units as a function of the hours of the day for the one day simulation of the wall on the exterior point of the thermal bridge (straight and bent) compared with the one-layer and two-layer cases; Figure S11: The rate of the total heat loss in W units with hours of the day for all cases of the one-day wall simulation.

Author Contributions: Conceptualization and resources: E.K.; methodology and supervision: E.K. and B.B.; software and visualization: I.O. and A.H.A.; investigation: I.O.; writing—original draft preparation, I.O.; writing—review and editing: I.O., E.K. and B.B. 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 on request from the first author.

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

Nomenclature

Symbols

c	*Specific heat* [J/(kgK)]
Δt	*Time step size* [s]
h_c	*Heat transfer coefficient* [W/(m^2K)]
K	*Convection coefficient* [1/s]
k	*Thermal conductivity* [W/(m·K)]
Q	*Heat transfer rate* [W]
q^*	*heat generation* [W/m^2]
q	*Heat source rate* [K/s]
t	*time* [s]
u	*Temperature* [K]
L	*Thickness* [m]

Greek Symbols

α	*Thermal diffusivity* [m^2/s]
Δ	*Difference*
P	*Mass density* [kg/m^3]
Σ	*Coefficient of the radiation term* [$s^{-1}K^{-3}$]
σ^*	*realistic values of the non-black body* [W/(m^2·K^4)]

Subscripts

A	*Ambient air*
L	*Left side*
R	*Right side*
b, ins	*Brick and Insulation*
c	*convection*

References

1. Berggren, B.; Wall, M. State of Knowledge of Thermal Bridges—A Follow up in Sweden and a Review of Recent Research. *Buildings* **2018**, *8*, 154. [CrossRef]
2. Liu, C.; Sharples, S. A Review of Building Energy Retrofit Measures, Passive Design Strategies and Building Regulation for the Low Carbon Development of Existing Dwellings in the Hot Summer—Cold Winter Region of China. *Energies* **2023**, *16*, 4115. [CrossRef]
3. Kusuda, T. Fundamentals of Building Heat Transfer. *J. Res. Natl. Bur. Stand.* **1977**, *82*, 97. [CrossRef] [PubMed]
4. Dai, B.; Qi, H.; Liu, S.; Zhong, Z.; Li, H.; Song, M.; Ma, M.; Sun, Z. Environmental and Economical Analyses of Transcritical CO_2 Heat Pump Combined with Direct Dedicated Mechanical Subcooling (DMS) for Space Heating in China. *Energy Convers. Manag.* **2019**, *198*, 111317. [CrossRef]
5. Cao, X.; Dai, X.; Liu, J. Building Energy-Consumption Status Worldwide and the State-of-the-Art Technologies for Zero-Energy Buildings during the Past Decade. *Energy Build.* **2016**, *128*, 198–213. [CrossRef]
6. 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]
7. 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*, 085225. [CrossRef]
8. *EN ISO 13789:2017*; Swedish Standards Institute Thermal Performance of Buildings—Transmission and Ventilation Heat Transfer Coefficients—Calculation Method. International Organization for Standardization: Geneva, Switzerland, 2017. Available online: https://www.iso.org/standard/65713.html (accessed on 20 April 2023).
9. *EN ISO 10211:2017*; Swedish Standards Institute Thermal Bridges in Building Construction—Heat Flows and Surface Temperatures—Detailed Calculations. International Organization for Standardization: Geneva, Switzerland, 2017. Available online: https://www.iso.org/%0Astandard/65710.html (accessed on 20 April 2023).
10. *EN ISO 14683:2017*; Swedish Standards Institute Thermal Bridges in Building Construction—Linear Thermal Transmittance—Simplified Methods and Default Values. International Organization for Standardization: Geneva, Switzerland, 2017. Available online: https://www.iso.org/standard/65706.html (accessed on 20 April 2023).
11. Alhawari, A.; Mukhopadhyaya, P. Thermal Bridges in Building Envelopes—An Overview of Impacts and Solutions. *Int. Rev. Appl. Sci. Eng.* **2018**, *9*, 31–40. [CrossRef]
12. Borelli, D.; Cavalletti, P.; Marchitto, A.; Schenone, C. A Comprehensive Study Devoted to Determine Linear Thermal Bridges Transmittance in Existing Buildings. *Energy Build.* **2020**, *224*, 110136. [CrossRef]
13. Karabulut, K.; Buyruk, E.; Fertelli, A. Numerical Investigation of the Effect of Insulation on Heat Transfer of Thermal Bridges with Different Types. *Therm. Sci.* **2016**, *20*, 185–195. [CrossRef]
14. Mátyás, L.; Barna, I.F. General Self-Similar Solutions of Diffusion Equation and Related Constructions. *arXiv* **2021**, arXiv:2104.09128v1.
15. Amiri, S.; Mazaheri, M.; Bavandpouri Gilan, N. Introducing a New Method for Calculating the Spatial and Temporal Distribution of Pollutants in Rivers. *Int. J. Environ. Sci. Technol.* **2021**, *18*, 3777–3794. [CrossRef]
16. Jejeniwa, O.A.; Gidey, H.H.; Appadu, A.R. Numerical Modeling of Pollutant Transport: Results and Optimal Parameters. *Symmetry* **2022**, *14*, 2616. [CrossRef]
17. 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]
18. Kumar, V.; Chandan, K.; Nagaraja, K.V.; Reddy, M. V Heat Conduction with Krylov Subspace Method Using FEniCSx. *Energies* **2022**, *15*, 8077. [CrossRef]
19. 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.* **2022**, *99*, 1139–1158. [CrossRef]
20. 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 Elem. Anal. Des.* **2022**, *211*, 103825. [CrossRef]
21. 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]
22. Fedoseev, P.; Pesterev, D.; Karimov, A.; Butusov, D. New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers. *Algorithms* **2022**, *15*, 275. [CrossRef]
23. Ndou, N.; Dlamini, P.; Jacobs, B.A. Enhanced Unconditionally Positive Finite Difference Method for Advection--Diffusion--Reaction Equations. *Mathematics* **2022**, *10*, 2639. [CrossRef]
24. Ji, Y.; Xing, Y. Highly Accurate and Efficient Time Integration Methods with Unconditional Stability and Flexible Numerical Dissipation. *Mathematics* **2023**, *11*, 593. [CrossRef]
25. 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]
26. Karahan, H. Unconditional Stable Explicit Finite Difference Technique for the Advection—Diffusion Equation Using Spreadsheets. *Adv. Eng. Softw.* **2007**, *38*, 80–86. [CrossRef]
27. Sanjaya, F.; Mungkasi, S. A Simple but Accurate Explicit Finite Difference Method for the Advection-Diffusion Equation. *Proc. J. Phys. Conf. Ser.* **2017**, *909*, 12038. [CrossRef]

28. 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]
29. Harley, C. Hopscotch Method: The Numerical Solution of the Frank-Kamenetskii Partial Differential Equation. *Appl. Math. Comput.* **2010**, *217*, 4065–4075. [CrossRef]
30. 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]
31. Nwaigwe, C.; An Unconditionally Stable Scheme for Two-Dimensional Convection-Diffusion-Reaction Equations. No. January 2022. Available online: https://www.researchgate.net/publication/357606287_An_Unconditionally_Stable_Scheme_for_Two-Dimensional_Convection-Diffusion-Reaction_Equations (accessed on 5 June 2023).
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. Mat.* **2021**, *71*, 245–252. [CrossRef]
33. 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]
34. Kovács, E.; Nagy, Á.; Saleh, M. A Set of New Stable, Explicit, Second Order Schemes for the Non-Stationary Heat Conduction Equation. *Mathematics* **2021**, *9*, 2284. [CrossRef]
35. 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]
36. 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]
37. Pirasaci, T. Investigation of Phase State and Heat Storage Form of the Phase Change Material (PCM) Layer Integrated into the Exterior Walls of the Residential-Apartment during Heating Season. *Energy* **2020**, *207*, 118176. [CrossRef]
38. 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]
39. 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]
40. Mahmoud, S.; Ádám, N.; Endre, K. Construction and Investigation of New Numerical Algorithms for the Heat Equation: Part III. *Multidiszcip. Tud.* **2020**, *10*, 349–360.
41. Alawadhi, E.M. *Finite Element Simulations Using ANSYS*; CRC Press: Boca Raton, FL, USA, 2009.
42. Holman, J. *Heat Transfer*, 10th ed.; McGraw-Hill Education: New York, NY, USA, 2010.
43. 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]
44. Munka, M.; Pápay, J. *4D Numerical Modeling of Petroleum Reservoir Recovery*; Akadémiai Kiadó: Budapest, Hungary, 2001; ISBN 9630578433.
45. 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]
46. Amando, R.S. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*; Hirsch, C., Ed.; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 1989.
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. Kovács, E. New Stable, Explicit, First Order Method to Solve the Heat Conduction Equation. *arXiv* **2019**, arXiv:1908.09500. [CrossRef]
49. Weather-Online Website. Available online: https://www.worldweatheronline.com/miskolc-weatherhistory/miskolc/hu.aspx (accessed on 20 November 2022).
50. Askar, A.H.; Kovács, E.; Bolló, B. ANN modeling for thermal load estimation in a cabin vehicle. In *Vehicle and Automotive Engineering 4: Select Proceedings of the 4th VAE2022, Miskolc, Hungary*; Springer: Cham, Switzerland, 2022; pp. 357–373.
51. 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.