UNIVERSITY OF MISKOLC
FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

DESIGNING AND IMPLEMENTING NUMERICAL METHODS TO SOLVE NONLINEAR HEAT TRANSFER PROBLEMS TO IMPROVE BUILDINGS' ENERGY EFFICIENCY

PHD THESES

Prepared by

Humam Kareem Jalghaf Al-Janabi
Mechanical Engineering (BSc),
Mechanical Engineering (MSc)

ISTVÁN SÁLYI DOCTORAL SCHOOL OF MECHANICAL ENGINEERING SCIENCES

TOPIC FIELD OF BASIC ENGINEERING SCIENCES
TOPIC GROUP OF TRANSPORT PROCESSES AND MACHINES

Head of Doctoral School

Dr. Gabriella Bognár
DSc, Full Professor

Head of Topic Group

Dr. László Baranyi
Full Professor

Scientific Supervisors

Dr. Endre Kovács
Dr. Betti Bolló

Miskolc

2025

CONTENTS

SUPERVISOR'S RECOMMENDATIONS

Date

Supervisor

LIST OF SYMBOLS AND ABBREVIATIONS

Quantity	Meaning	Unit	Quantity	Meaning
u	Temperature	Kelvin (K)	*anal*	Analytical
u_{cr}	Melting Temperature	Kelvin (K)	*Num*	Numerical
Q_{gen}	Heat Generation Rate	Watt (W)	*Max*	Maximum
$Q_{convection}$	Heat Convection Rate	Watt (W)	*abs*	Absolute
$Q_{radiation}$	Heat radiation Rate	Watt (W)	f	Shape Function
ΔE	Change in Energy of an Element	Joule (J)	η	Reduced Variable
k	Thermal Conductivity	W/(m·K)	S, L	Solid, Liquid State
h_c	Convection Coefficient	W/(m^2·K)	Li	Liquid Phase
σ^*	Radiation Constant	W/(m^2·K^4)	So	Solid Phase
SB	Stefan–Boltzmann Constant	W/(m^2·K^4)	La	Latent
α	Thermal Diffusivity	m^2/s	Se	Sensible
ρ	Density	kg/m^3	g	Gaussian Function
c	Specific Heat	J/(kg·K)	λ	The Root of the Transcendental Equation
C	Heat Capacity	J/K	*erf*	Error Function
R	Thermal Resistance	K/W	FEM	Finite Element Method
t	Time	s	CR	Convection, Radiation
$h=\Delta t$	Time step size		BC	Boundary Condition
S	Surface Area	m^2	M	The System Matrix
V	Volume	m^3	ε	Emissivity
φ	Azimuthal angle	Rad, deg		
θ	Polar angle	Rad, deg		
u_{face}	Boundary Temperature	K		
$L_{x,z}$	Length	m		
r	Radius	m		
z	Height	m		
σ	Standard Deviation Temperature	K		
EHC	Effective Heat Capacity	J/K		
SHC	Sensible Heat Capacity	J/K		
LHC	Latent Heat Capacity	J/K		
H	Latent Heat	J/kg		
Q	Heat Energy	kJ/ m^2		

1. INTRODUCTION

1.1. General

Energy efficiency in the building sector is crucial for addressing the challenges of climate change and fostering a sustainable economy. Buildings are significant energy consumers and have the potential to make substantial contributions to reducing energy consumption and global warming emissions. Enhancing the buildings' energy efficiency requires a multifaceted approach, with a particular emphasis on optimizing the thermal behavior of building components. The efficient management of heat transfer within buildings is fundamental to achieving energy sustainability and cost-effectiveness. Building envelopes, including walls, roofs, and floors, exhibit different thermal performances depending on their position within the building. To optimize energy efficiency, it is essential to accurately calculate heat transfer within these building components.

The heat transfer in building components can be calculated by using the heat transfer equation, which depends on various parameters, most importantly, material properties and boundary conditions. Utilizing materials with excellent thermal properties such as thermal conductivity, density, and specific heat capacity determines how effectively heat is transferred through a material can significantly enhance heat transfer performance, thereby improving the overall energy efficiency. Boundary conditions, determined by the internal and external environments of the building, play a crucial role in heat transfer calculations. These conditions, including temperature, humidity and airflow, serve as input parameters for accurately modeling the thermal behavior of building components.

To conduct a precise thermal analysis of building walls, accurate numerical methods are essential. Several studies in the literature have focused on heat transfer through walls to analyze the thermal behavior of a multilayer medium in a transient regime. These studies have developed mathematical models that calculate temperature and thermal contact resistance distributions. Some research proposes MATLAB-based numerical solution models for simulations, while others utilize computational fluid dynamics (CFD) methods.

1.2. Literature Review

1.2.1. Literature Review of Numerical Methods

The diffusion equation, incorporating a diffusion term, has been extensively studied, resulting in numerous analytical solutions [1-6]. However, these solutions generally assume constant parameters such as the diffusion coefficient or heat conductivity, which do not vary with space, time, or the dependent variable *u*. A notable exception is the work of Zoppou and Knight, who derived analytical solutions for the two- and three-dimensional advection-diffusion equation with specific forms of spatially variable coefficients [7]. Nevertheless, for general cases with space-dependent coefficients, numerical methods are essential. This is particularly true for systems where physical properties vary significantly even within proximity [8]. Such variations often result in eigenvalues spanning several orders of magnitude, leading to severely stiff problems.

When partial differential equations (PDEs) are spatially discretized, they are transformed into systems of ordinary differential equations (ODEs). Solving these systems numerically becomes particularly challenging when dealing with many variables, especially in three-dimensional spaces. Traditional explicit methods, such as the Runge-Kutta method, are conditionally stable, meaning they require very small time steps to maintain stability, which can be computationally expensive for large-scale problems. In contrast, implicit methods are typically unconditionally stable, allowing for larger time steps, but they require solving systems of algebraic equations at each time step. These algebraic systems can be computationally intensive, especially when the matrices involved are non-tridiagonal or have complex structures. To address these challenges, significant efforts have been directed toward developing advanced modifications to improve the efficiency of implicit methods [9]. Currently, implicit methods with these extensions are commonly used to solve stiff problems, such as those involving rapid temperature changes, high thermal diffusivity materials, or systems with multiple heat transfer mechanisms (e.g., conduction, convection, and radiation). These methods are particularly effective for problems where stability and accuracy are critical, such as heat transfer in multilayer walls, phase change materials (PCMs), and other complex building components.[10-12]. Despite these advancements, parallelizing implicit methods remains challenging, though some progress has been made [13,14]. The shift towards increased parallelism in high-performance computing [15,16], driven by the stagnation in CPU clock frequency improvements, further emphasizes this issue.

Given these challenges, a part of my work focuses on developing novel, explicit, easily parallelizable, and unconditionally stable methods. A key example is the original odd-even hopscotch (OEH) algorithm, introduced by Gordon [17] and later reformulated and analyzed by Gourlay [18-20] (see also [21]). This method has been modified to enhance its reliability and accuracy, typically by increasing its implicitness. This has led to a hierarchy of algorithms, from the fully explicit OEH to the alternating direction implicit (ADI) hopscotch, each offering greater accuracy at the cost of increased programming complexity and runtime [19]. Morris and Nicoll applied these methods to thermal print head calculations and found that, while the OEH method was faster than its more implicit versions for isotropic media, it produced inaccurate results for anisotropic cases, necessitating the use of the ADI hopscotch for meaningful solutions [22].

The OEH method has since been applied to various problems, including the incompressible Navier-Stokes equations [23], the Frank-Kamenetskii [24] and Gray-Scott reaction-diffusion equations [25], and even the nonlinear Dirac equation [26]. Goede and Boonkkamp implemented a vectorized OEH scheme for the two-dimensional Burgers' equations, significantly increasing speed and solver performance [27]. Recently, Maritim et al. developed hybrid algorithms incorporating the hopscotch, Crank-Nicolson, Du Fort-Frankel, and other schemes for the two-dimensional Burgers' equations, finding their implicit algorithms stable and accurate [28,29].

In a series of papers [30-32], new hopscotch combinations were developed using alternative formulas to the original explicit and implicit Euler schemes. Tests showed [30] that for stiff systems, the original OEH method could produce significant inaccuracies for large time steps, with relative errors reaching up to 10^4, which could be more problematic than instability if unnoticed by inexperienced users. Two of the three new combinations, however, demonstrated much better performance.

1.2.2. Literature Review of The Energy Efficiency in Buildings

The efficient management of heat transfer in buildings is paramount for achieving energy sustainability and cost-effectiveness in the built environment. The comfortable interior environment of the building is a crucial issue for most people living or working inside, and it largely depends on the wall structure. So, to understand how the wall structure is affected in the interior zone, I will focus on the thermal analysis of the wall structure by using very efficient algorithms. The integration of advanced materials and technologies into building envelopes has gotten significant attention from researchers and practitioners alike. Among these innovations, the combination of phase change materials (PCMs) and thermal insulation holds great promise for enhancing energy efficiency and occupant comfort. PCMs are known for their high heat capacity and outstanding energy storage potential, as well as low heat transfer coefficient. The integration of PCM within building envelopes offers the ability to store and release latent heat during phase transitions, thereby mitigating temperature fluctuations and reducing the reliance on mechanical heating and cooling systems. Concurrently, thermal insulation serves to minimize heat transfer, further enhancing the overall energy performance of the building. Historically, numerical simulations have played an important role in understanding the nature of heat transfer within building structures.

X. Geng et al. [33] explored the optimization of the location combination for thermal insulation material (TIM) and PCM in multi-layer walls during both continuous and intermittent air-conditioning operations. These walls typically incorporate TIM or PCM layers to enhance thermal performance. Four wall models were constructed for evaluation, considering temperature and heat flow on inner surfaces. Placing the PCM layer inside the wall proves better for outdoor thermal environments during continuous air-conditioning, while situating the TIM layer inside is preferable for higher energy-saving contributions during intermittent operation. Despite intermittent operation yielding energy savings of 46.69–64.73%, it raises the peak load on the urban electricity system compared to continuous operation. Notably, for multi-layer walls with the TIM layer inside, this negative effect is negligible in comparison to their superior energy-saving benefits.

Z. Liu et al. [34] showed that the PCM can enhance lightweight building walls' (LBW) thermal performance, but optimal parameters vary by wall orientation due to outdoor thermal variations. A study tested a small-scale LBW in different orientations and analyzed PCM's impact using a heat transfer model. The results suggest that east and south-facing walls benefit from PCM in the middle temperature range (20–30°C), while west and north-facing walls perform best with inner (18–28°C) and outer (24–34°C) PCM placement. East and west-facing walls see the most significant thermal improvement, reducing peak and average heat flux by 62.8–66.4% and 28.2–29.5%, respectively, and increasing delay time by 5–5.34 hours compared to reference walls.

E. Tunçbilek et al. [35] explored combining PCMs and conventional thermal insulation for enhanced energy savings in building walls. PCM on the interior side with layer thicknesses

$L_{PCM} \leq 16$ mm outperformed insulation saving up to 38.2% more energy than insulation with layer thickness L_{INS} =6 mm. A parameter ψ defining the ratio of L_{PCM} to L_{PCM} + L_{INS} was introduced. Combining PCM and insulation (a configuration labeled by C5 in their paper, ψ = 0.05) saved up to 7.3% more energy compared to insulation alone. Overall, combined designs with $0 < \psi \leq 0.6$ showed improved energy savings compared to insulation only, with latent heat activation being crucial for better thermal performance.

Y. Cascone et al. [36] conducted a study on optimizing PCMs in retrofitting office buildings for energy efficiency in Mediterranean climates, crucial for achieving EU's 2020 sustainability goals. PCMs, with careful consideration of properties, quantity, and placement, are recommended for effective and economically feasible use. The paper presents multi-objective optimization analyses for retrofitting with PCM-enhanced opaque building envelope components. Objectives included minimizing primary energy consumption, global costs, building energy needs for heating and cooling, and investment costs. The research variables encompassed PCM properties, window type, insulation materials, and wall configuration. Post-optimization analyses provided insights for designers, revealing that optimal PCM properties are notably influenced by the HVAC system's operation.

R. F. Jam et al. [37] conducted a study for optimization of the PCMs location and thickness in building walls with an energy-economic analysis. The research emphasizes the significance of thermal insulation for reducing energy consumption in buildings. CMS are investigated as a form of insulation in an educational building at Hakim Sabzevari University, Iran. Through numerical simulations, the study explores the effects of PCM integration during the hot months of the year. Optimal PCM placement within the wall and various thicknesses (2, 3, 4, and 5 cm) are analyzed. Results indicate heat exchange reductions of 9.8%, 13.4%, 17.5%, and 20.4%, respectively, for different PCM thicknesses. Additionally, a thermo-economic analysis calculates energy savings and payback periods. The study identifies a 3 cm PCM thickness as optimal, resulting in a 50-month payback period through Pareto solutions and the TOPSIS method.

M. J. Abden et al. [38] conducted research on the combined use of PCM and thermal insulation to improve energy efficiency of residential buildings, applying thermal insulation to external walls and ceilings in standard practice. The study evaluates the approach by combining expanded polystyrene with PCM gypsum board in a typical Australian standalone house. Numerical simulations are conducted considering the house's location in three distinct Australian cities—Darwin, Alice Springs, and Sydney—representing tropical savanna, hot semi-arid, and humid subtropical climates, respectively. Results indicate significant cost savings over a 10-year lifecycle: AU\$167.0, \$162.3, and \$39.7/m^2 in Darwin, Alice Springs, and Sydney, respectively. Additionally, energy ratings improve by 3.5, 3.8, and 4.3 stars in the three cities. Payback periods for the renovation vary from 2.2 to 7.5 years, contingent on climate conditions.

E. Iffa et al. [39] conducted thermal energy storage systems in buildings serve to store cooling/heating energy during non-peak load hours or when renewable energy sources are available, aiding in peak load shaving, reducing electric grid burdens, and enhancing occupant thermal comfort. While thermal lag in systems like thermally activated building systems often leads to passive energy release, integrating active insulation systems can enhance flexibility in charging and discharging energy. That study designed a wall system equipped with both active insulation and thermally activated storage systems to evaluate its performance in contributing to active cooling energy. The results showed that the thermal properties of the storage core material and the spacing of embedded pipes in both the storage and active insulation systems significantly influenced wall performance. During discharging, heat flux into the wall reached up to 81.92W/m^2, with the dynamic R-value of the active insulation system varying from less than 1ft2·◦F·h/BTU (0.18 m^2·K/W) to 98% of equally thick foam insulation's R-value.

P. Arumugam et al. [40] aimed to optimize PCM and insulation placement in building envelopes for improved thermal performance and reduced cooling load demand in Indian office buildings across different climates. Models integrated with PCM or insulation on outer walls showed more comfortable indoor temperatures than those on inner walls. The selection of PCM and insulation depended on location temperatures. The recommended techniques resulted in cooling load reductions of 64%, 61%, 57%, 63%, and 58/59% for Bangalore, Delhi, Jodhpur, Pune, and Guwahati, respectively, compared to basic buildings.

1.3. The Aim of The Dissertation

The dissertation aims to design and implement new, efficient explicit numerical methods to solve the linear and nonlinear heat equations, encompassing heat conduction, convection, radiation, and heat generation across Cartesian, cylindrical, and spherical coordinate systems. The work builds on the modification of well-known numerical methods, such as the Explicit Euler based FTCS (forward time central space), the Implicit Euler method, the Crank Nicolson method, the Rational Runge–Kutta method, the Dufort–Frankel (DF) method, the UPFD (Unconditionally Positive Finite Difference) method, Heun’s method, and the original hopscotch method, to enhance their efficiency and stability. Building on these modifications, improved numerical schemes were developed, including the constant neighbor method, the two and three-linear neighbor method, and the CpC method. The core novelty of this work lies in the invention of entirely new numerical methods, such as the Shifted-Hopscotch method, Leapfrog-Hopscotch method, Asymmetric-Hopscotch method, Reversed-Hopscotch method, and Pseudo-Implicit method, which represent significant advancements over existing explicit numerical schemes, offering superior stability, accuracy, and computational efficiency for solving complex heat transfer problems. These methods were implemented and tested using MATLAB 2020b, rigorously verified and validated against analytical solutions and experimental measurements, and applied to real-life heat transfer problems in various engineering applications. They serve as powerful tools for thermal analysis, enabling the calculation of temperature and heat energy distributions in complex geometries and systems. In this dissertation, the methods were applied to analyze heat transfer in different building walls and heated cylinders, ranging from simple geometries (low stiffness systems), such as insulated walls, to highly complex geometries (high stiffness systems), such as multilayer walls composed of different materials (e.g., insulators, phase change materials (PCMs), and base materials like brick or concrete). The goal was to control the amount of heat transfer between indoor and outdoor environments, contributing to the development of sustainable and energy-efficient buildings. By achieving these objectives, this research advances ongoing efforts to improve energy efficiency and sustainability in building design.

2. THE HEAT TRANSFER EQUATION

In the current study, I aim to calculate the heat transfer in different geometries. First, I derive the heat transfer equation (encompassing conduction, convection, and radiation) based on energy balance in Cartesian coordinates, and then in cylindrical and spherical coordinates.

2.1. The Heat Transfer Equation in Cartesian Coordinates

For the Cartesian coordinate, consider a small rectangular element $\Delta x, \Delta y, \Delta z$, as shown in Fig.2.1. The energy balance for the differential control volume during a small-time interval Δt can be expressed as [41] :

$$\begin{pmatrix}\text{The heat conduction rate}\\ \text{at x, y, and z}\end{pmatrix}-\begin{pmatrix}\text{The heat conduction rate at}\\ \text{x+}\Delta\text{x,y+}\Delta\text{y, and z+}\Delta\text{z}\end{pmatrix}+\begin{pmatrix}\text{The heat generation rate}\\ \text{inside the element}\end{pmatrix}+\begin{pmatrix}\text{The covection rate at}\\ \text{x,y,z}\end{pmatrix}+\begin{pmatrix}\text{The radiation rate at}\\ \text{x,y,z}\end{pmatrix}$$
$$=\begin{pmatrix}\text{The change of energy}\\ \text{content rate of the element}\end{pmatrix}$$

Or

$$Q_x + Q_y + Q_z - Q_{x+\Delta x} - Q_{y+\Delta y} - Q_{z+\Delta z} + Q_{gen} + Q_{Convection} + Q_{Radiation} = \frac{\Delta E_{element}}{\Delta t} \tag{2.1}$$

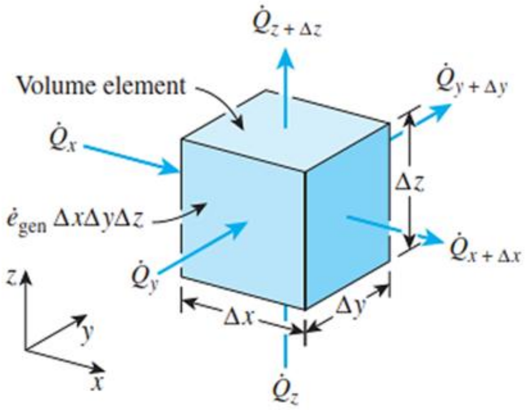

Figure 2.1. 3D rectangular element [41].

I use the following three well-known laws:

Fourier's law of heat conduction:

$$Q_x = -k.S_x.\frac{\Delta u}{\Delta x}, Q_y = -k.S_y.\frac{\Delta u}{\Delta y}, Q_z = -k.S_z.\frac{\Delta u}{\Delta z} \tag{2.2}$$

where $u = u(\vec{r},t)$ is the temperature, $k = k(\vec{r})$ is the thermal conductivity of material and the surface area of the element in *x,y* and *z* are $S_x = \Delta y.\Delta z$, $S_y = \Delta x.\Delta z$ and $S_z = \Delta x.\Delta y$ respectively .

Newton's law of heat convection:

$$Q_{convection} = h_c S \Delta u = h_c S(u_a - u), \tag{2.3}$$

where h_c represents convection heat transfer coefficient, S is the surface area, and the ambient temperature u_a does not depend directly on u, and the term hSu_a therefore it is included into the heat generation term.

The law of Stefan–Boltzmann for the incoming and outgoing radiation heat:

$$Q_{radiation} = \sigma^* S(u_i^4 - u^4) \quad , \tag{2.4}$$

where $\sigma^* = SB \cdot \varepsilon$, i.e., the Stefan-Boltzmann universal constant $SB = 5.67 \times 10^{-8}\,\mathrm{W/(m^2 \cdot K^4)}$ is multiplied by the proper constant of emissivity ε where the surface has a non-unity emissivity, u_i is the temperature of the incoming heat radiation, and u is the temperature of the outgoing radiation by the surface elements. Incoming heat radiation, such as direct sunlight, is incorporated into the heat source term q as the $\sigma^* S u_i^4$ term.

$$\Delta E_{element} = E_{t+\Delta t} - E_t = mc(u_{t+\Delta t} - u_t) = \rho c \Delta V (u_{t+\Delta t} - u_t)\,, \tag{2.5}$$

where $\rho = \rho(\vec{r}\,)$, $c = c(\vec{r}\,)$ and $\Delta V = \Delta x \Delta y \Delta z$ are the density, the specific heat and the elementary volume, respectively. Substituting Eqs (2.2-2.5). into Eq. (2.1), dividing by $\Delta x.\Delta y.\Delta z$ and limited as $\Delta x, \Delta y, \Delta z, \Delta t \to 0$ yields:

$$\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right) + \frac{Q_{gen}}{\Delta x.\Delta y.\Delta z} + \frac{h_c S(u_a - u)}{\Delta x.\Delta y.\Delta z} + \frac{\sigma^* S(u_i^4 - u^4)}{\Delta x.\Delta y.\Delta z} = \rho c \frac{\partial u}{\partial t} \tag{2.6}$$

Eq. (2.6) is divided by (ρc), and if k is constant, I obtain

$$\frac{k}{\rho c}\frac{\partial^2 u}{\partial x^2} + \frac{k}{\rho c}\frac{\partial^2 u}{\partial y^2} + \frac{k}{\rho c}\frac{\partial^2 u}{\partial z^2} + \frac{1}{\rho c}\frac{Q_{gen}}{\Delta x\,\Delta y\,\Delta z} + \frac{1}{\rho c}\frac{h_c S(u_a - u)}{\Delta x\,\Delta y\,\Delta z} + \frac{1}{\rho c}\frac{\sigma^* S(u_i^4 - u^4)}{\Delta x\,\Delta y\,\Delta z} = \frac{\partial u}{\partial t} \tag{2.7}$$

Where the property $\alpha = \dfrac{k}{\rho c}$ is the thermal diffusivity of the material. In this work do not consider volumetric heat generation, so the heat generation is represented by all incoming heat that crosses the surface element. I introduce q^*, which is the incoming heat by radiation and convection for a unit area. In all of our cases, the direction of the heat transfer is constrained to the horizontal direction, thus $S = \Delta y \Delta z$ and $Q_{gen} = \left(q^* + hu_a + \sigma^* u_i^4\right) S$. With these, I obtain

$$\alpha\frac{\partial^2 u}{\partial x^2} + \alpha\frac{\partial^2 u}{\partial y^2} + \alpha\frac{\partial^2 u}{\partial z^2} + \frac{1}{\rho c}\frac{q^* \Delta y \Delta z}{\Delta x\,\Delta y \Delta z} + \frac{1}{\rho c}\frac{h_c\,\Delta y \Delta z u_a}{\Delta x\,\Delta y \Delta z} + \frac{1}{\rho c}\frac{\sigma^* \Delta y \Delta z\, u_i^4}{\Delta x\,\Delta y \Delta z} - \frac{1}{\rho c}\frac{h_c \Delta y \Delta z\, u}{\Delta x\,\Delta y \Delta z} - \frac{1}{\rho c}\frac{\sigma^* \Delta y \Delta z\, u^4}{\Delta x\,\Delta y \Delta z} = \frac{\partial u}{\partial t} \tag{2.8}$$

After simplification, I have

$$\alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + \frac{q^*}{\rho c \Delta x} + \frac{h_c}{\rho c \Delta x}u_a + \frac{\sigma^*}{c\rho \Delta x}u_i^4 - \frac{h_c}{\rho c \Delta x}u - \frac{\sigma^*}{c\rho \Delta x}u^4 = \frac{\partial u}{\partial t} \tag{2.9}$$

The equation for the temperature, which includes the source of heat generation, conduction, convection, and radiation can be expressed as follows:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + q - Ku - \sigma u^4 \tag{2.10}$$

Where is the heat generation or heat source coming from the outside of the wall structure, $K = K(\vec{r}) = \frac{h_c}{c\rho\Delta x}$ is the heat transfer convection term, and $\sigma = \sigma(\vec{r}) = \frac{\sigma^*}{c\rho\Delta x}$ is the radiation heat transfer term, where K and σ are exist at the interface surfaces (interface between the solid and liquid or gas); otherwise, it is set as zero at the interior elements within the solid body. The terms q, Ku, and σu^4 in Eq. (2.10) are nonnegative and still in $[\mathrm{K/s}]$. If there is a multilayer wall, then the material properties depend on space, so an equation with a more general form can be used as follows:

$$\frac{\partial u}{\partial t} = \frac{1}{\rho c}\nabla(k\nabla u) + q - Ku - \sigma u^4 \tag{2.11}$$

2.2. The Spatial Discretization in Cartesian Coordinates

The standard central difference formula in two space dimensions is applied for the second-order derivative ($\nabla^2 u$). The space steps are Δx and Δz as shown in Figure 2.2. Now, for the nodes of a homogeneous material, one obtains

$$\frac{\partial^2}{\partial x^2}u(x_i) \approx \frac{\frac{u(x_{i+1}) - u(x_i)}{\Delta x} + \frac{u(x_{i\text{-}1}) - u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i\text{-}1} - 2u_i + u_{i+1}}{\Delta x^2} \tag{2.12}$$

$$\frac{\partial^2}{\partial z^2}u(z_i) \approx \frac{\frac{u(z_{i+Nx}) - u(z_i)}{\Delta z} + \frac{u(z_{i\text{-}Nx}) - u(z_i)}{\Delta z}}{\Delta z} = \frac{u_{i\text{-}Nx} - 2u_i + u_{i+Nx}}{\Delta z^2} \tag{2.13}$$

I obtain the spatially discretized form Eq. (2.11) in two dimensions:

$$\frac{du_i}{dt} = \alpha\frac{u_{i\text{-}1} - 2u_i + u_{i+1}}{\Delta x^2} + \alpha\frac{u_{i-Nx} - 2u_i + u_{i+Nx}}{\Delta z^2} + q - Ku_i - \sigma u_i^4 \tag{2.14}$$

Now, to be more realistic, let the k, c, and ρ quantities be functions of space. Then Eqs. (2.12) and (2.13) can be written using a two-dimensional, equidistant grid in the following form:

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \frac{1}{c(x_i)\rho(x_i)\Delta x}\left[k\left(x_i + \frac{\Delta x}{2}\right)\frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} + k\left(x_i - \frac{\Delta x}{2}\right)\frac{u(x_i - \Delta x) - u(x_i)}{\Delta x}\right] \tag{2.15}$$

$$\frac{\partial^2 u(z_i)}{\partial z^2} = \frac{1}{c(z_i)\rho(z_i)\Delta z}\left[k\left(z_i + \frac{\Delta z}{2}\right)\frac{u(z_i + \Delta z) - u(z_i)}{\Delta z} + k\left(z_i - \frac{\Delta z}{2}\right)\frac{u(z_i - \Delta z) - u(z_i)}{\Delta z}\right] \tag{2.16}$$

I now change from node to cell variables, which means that u_i, ρ_i, and c_i will be the temperature, density, and specific heat of cell i, respectively. Furthermore, since the material boundaries will always coincide with the cell borders, I write the average $\frac{k_i + k_{i+1}}{2}$ instead of $k\left(x_i + \frac{\Delta x}{2}\right)$. Now the discretized form of Equation (2.11) will take the form

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x}\left(\frac{k_i + k_{i+1}}{2}\frac{u_{i+1} - u_i}{\Delta x} + \frac{k_i + k_{i-1}}{2}\frac{u_{i\text{-}1} - u_i}{\Delta x}\right) + \frac{1}{c_i\rho_i\Delta z}\left(\frac{k_i + k_{i+N_x}}{2}\frac{u_{i+N_x} - u_i}{\Delta z} + \frac{k_i + k_{i-N_x}}{2}\frac{u_{i\text{-}N_x} - u_i}{\Delta z}\right)$$
$$+ q - Ku_i - \sigma u_i^4 \tag{2.17}$$

The distance between the centres of neighbouring cells is the same as the mesh spacing, and the interface area between cell *i* and its right neighbour is always *S*. Now I have

$$\frac{du_i}{dt}=\frac{1}{c_i\rho_i\Delta x\Delta y\Delta z}\begin{pmatrix}\Delta y\Delta z\dfrac{k_i+k_{i+1}}{2}\dfrac{u_{i+1}-u_i}{\Delta x}+\Delta y\Delta z\dfrac{k_i+k_{i-1}}{2}\dfrac{u_{i-1}-u_i}{\Delta x}\\ +\Delta x\Delta y\dfrac{k_i+k_{i+N_x}}{2}\dfrac{u_{i+N_x}-u_i}{\Delta z}+\Delta x\Delta y\dfrac{k_i+k_{i-N_x}}{2}\dfrac{u_{i-N_x}-u_i}{\Delta z}\end{pmatrix}+q-Ku_i-\sigma u_i{}^4 \tag{2.18}$$

The cell's heat capacity can be calculated as $C_i=c_i\rho_i V$. I calculate the thermal resistances in a horizontal and vertical direction between the neighbouring cells, as $R_{i,i+1}\approx\frac{\Delta x}{2k_i\Delta z}+\frac{\Delta x}{2k_{i+1}\Delta z}$, and $R_{i,i+N_x}\approx\frac{\Delta z}{2k_i\Delta x}+\frac{\Delta z}{2k_{i+N_x}\Delta x}$ respectively, where *i* and *j* represented the cells' index in the *x-axis* and *z-axis*. Semi-discretized form of Equation (2.14) can be expressed as below:

$$\frac{du_i}{dt}=\frac{u_{i-1}-u_i}{R_{i-1,i}C_i}+\frac{u_{i+1}-u_i}{R_{i+1,i}C_i}+\frac{u_{i-N_x}-u_i}{R_{i-N_x,i}C_i}+\frac{u_{i+N_x}-u_i}{R_{i+N_x,i}C_i}+q-Ku_i-\sigma u_i{}^4 \tag{2.19}$$

The time is discretized uniformly with a time-step size Δt and represents the temperature of cell *i* at the time $n\Delta t$, $n=0,1,...,T$. Now the formulae of the used methods are presented for the general discretization (2.19) only. For the simpler formula, I need to define the following quantities:

$$mr_i=\Delta t\sum_{j\neq i}\frac{1}{C_iR_{ij}}\ ,\quad A_i=\Delta t\sum_{j\neq i}\frac{u_j^n}{C_iR_{ij}}+\Delta t\cdot q_i\ \text{and}\ A_i^{\text{new}}=\Delta t\sum_{j\neq i}\frac{u_j^{\text{pred}}}{C_iR_{ij}}+\Delta t\cdot q_i$$

Where mr_i is the general mesh-ratio, while A_i shows the state and the effect of the neighbors of cell *i*.We prefer to use the ODE system for a general grid, which gives the derivative of the time of each temperature independently of any coordinate system

$$\frac{du_i}{dt}=\sum_{i\neq j}\frac{u_j-u_i}{R_{i,j}C_i}+q_i-Ku_i-\sigma u_i{}^4 \tag{2.20}$$

Which can be written in matrix form

$$\frac{d\vec{u}}{dt}=M\vec{u}+Q\,, \tag{2.21}$$

where $Q=\vec{q}_i-K\vec{u}_i-\sigma\vec{u}_i{}^4$, and the diagonal element of matrix *M* can be written as follows $m_{ii}=\sum_{j\in neighbour}\frac{-1}{R_{i,j}C_i}$. Off-diagonal elements $m_{ii}=1/R_{i,j}C_i$ of the global matrix *M* are nonzero only if the cells *i* and *j* are neighbours. From this point, all summations are performed over the neighbors of the current cell, denoted $j\in n(i)$. Unless specified, closed (zero Neumann) boundary conditions are applied, thermally insulating the domain's edges from conductive heat transfer. To help the reader imagine, we present the arrangement of the variables in Figure 2.2 for a 2D system. The framework is designed for generality, supporting both structured and unstructured meshes.

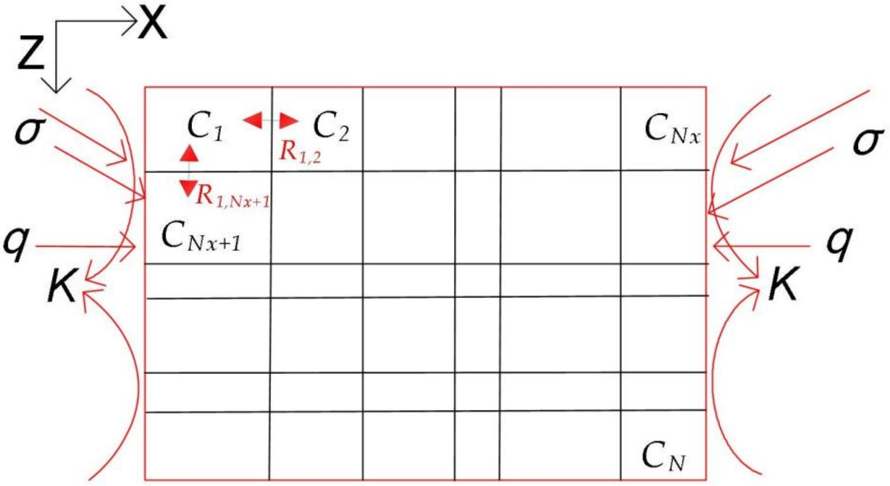

Figure 2.2. Arrangement of the generalized variables [Original].

For example this 4×4 system, the system of ODEs in matrix form can be written as

$$\frac{d}{dt}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\end{pmatrix}=\begin{pmatrix}\frac{-1}{C_1R_{12}}+\frac{-1}{C_1R_{13}} & \frac{1}{C_1R_{12}} & \frac{1}{C_1R_{13}} & 0\\ \frac{1}{C_2R_{21}} & \frac{-1}{C_2R_{21}}+\frac{-1}{C_2R_{23}}+\frac{-1}{C_2R_{24}} & \frac{1}{C_2R_{23}} & \frac{-1}{C_2R_{24}}\\ \frac{1}{C_3R_{31}} & \frac{1}{C_3R_{32}} & \frac{-1}{C_3R_{31}}+\frac{-1}{C_2R_{32}}+\frac{-1}{C_2R_{34}} & \frac{-1}{C_3R_{34}}\\ 0 & \frac{1}{C_4R_{42}} & \frac{1}{C_4R_{43}} & \frac{-1}{C_4R_{42}}+\frac{-1}{C_4R_{43}}\end{pmatrix}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\end{pmatrix}=\begin{pmatrix}Q_1\\Q_2\\Q_3\\Q_4\end{pmatrix}.$$

2.3. *The Heat Transfer Equation in Cylindrical and Spherical Coordinates*

In a similar way, the heat transfer equation in cylindrical coordinates can be obtained from an energy balance on a volume element in cylindrical coordinates, considering a small 3D cylindrical element as shown in Fig. 2.3. The energy balance in this element during a time interval can be expressed as [42]:

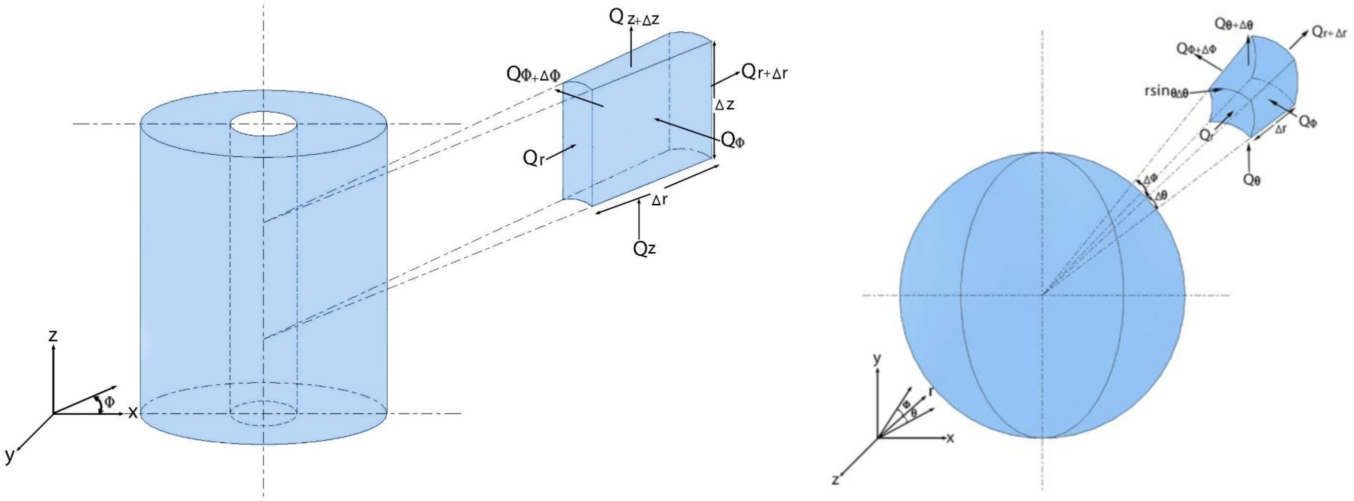

Figure 2.3. The cylindrical (left) and spherical (right) elements [42].

$$\begin{pmatrix}\text{The heat conduction rate}\\ \text{at } r\ ,\phi,\ z\end{pmatrix}-\begin{pmatrix}\text{The heat conduction rate at}\\ r+\Delta r\ ,\ \phi+\Delta\phi\ ,\ z+\Delta z\end{pmatrix}+\begin{pmatrix}\text{The heat generation rate inside}\\ \text{and on the surface of the element}\end{pmatrix}\pm\begin{pmatrix}\text{The convection rate}\\ \text{at the } r,\phi,z \text{ element}\end{pmatrix}$$
$$\pm\begin{pmatrix}\text{The radiation rate}\\ \text{at the } r,\phi,z \text{ element}\end{pmatrix}=\begin{pmatrix}\text{The change energy}\\ \text{content rate of the element}\end{pmatrix}$$

or briefly,

$$Q_r + Q_\phi + Q_z - Q_{r+\Delta r} - Q_{\phi+\Delta\phi} - Q_{z+\Delta z} + Q_{gen} \pm Q_{convection} \pm Q_{radiation} = \frac{\Delta E_{element}}{\Delta t} \tag{2.22}$$

To fill Eq. (2.22) with concrete formulas, the following three well-known laws are used.

Fourier's law of heat conduction:

$$Q_r = -kS\frac{\Delta u}{\Delta r},\ Q_\phi = -kS\frac{\Delta u}{\Delta \phi},\ Q_z = -kS\frac{\Delta u}{\Delta z} \tag{2.23}$$

The heat convection, radiation, and the change in energy of an element over a specific time interval are the same in Cartesian coordinates except the element volume being $\Delta V = \Delta\phi(r + {}^{\Delta r}\!/_{2})\Delta r \times \Delta z$. In the case of full cylindrical symmetry, it is better to choose a full ring-shaped element, which yields $\Delta V = 2\pi(r + {}^{\Delta r}\!/_{2})\Delta r \times \Delta z = \pi\left((r+\Delta r)^2 - r^2\right)\Delta z$.

From these equations, one can derive the heat-transport equation in a 3D cylindrical coordinate system, which can be written as:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(k\,r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial}{\partial\phi}\left(k\,r\frac{\partial u}{\partial\phi}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)+\frac{Q_{gen}}{\Delta V}-\frac{hSu}{\Delta V}-\frac{\sigma^* S u^4}{\Delta V}=\rho c\frac{\partial u}{\partial t} \tag{2.24}$$

In the case of spherical coordinates, a small 3D spherical element can be seen in Figure 2.2 The heat-transport equation for this case can be expressed as follows:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(k\,r^2\frac{\partial u}{\partial r}\right)+\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(k\,r\frac{\partial u}{\partial\phi}\right)+\frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(k\sin\theta\frac{\partial u}{\partial\theta}\right)+\frac{Q_{gen}}{\Delta V}-\frac{hSu}{\Delta V}-\frac{\sigma^* S u^4}{\Delta V}=\rho c\frac{\partial u}{\partial t} \tag{2.25}$$

2.4. The Spatial Discretization in Cylindrical and Spherical Coordinate

In the case of cylindrical geometry, I consider tube-shaped cells with height Δz and thickness Δr. For spheres, the cells have spherical-shell shapes with thickness Δr again. The temperature is considered at the middle of the cell layer, where the radial distance from the origin (the mean radius of the cells) is denoted by r_i, while the subsequent radius of the cell border is denoted by $r_i^* = r_i + \Delta r / 2$.

The cell's heat capacity in the cylindrical and in spherical cases is approximated as $C_i = c_i\rho_i\pi\left(r_{i+1}^{*2} - r_i^{*2}\right)\Delta z$ and $C_i = c_i\rho_i\,{}^{4}\!/_{3}\,\pi\left(r_{i+1}^{*3} - r_i^{*3}\right)$, respectively.

Let us denote the area of the cylindrical cell-surface perpendicular to *r* with S_r , which can be given as $S_r = 2\pi\ r\ \Delta z$. Now, for the thermal resistance in the *r*-direction, the approximate formula

$$R_{i,i+1} \approx \int_{r_i}^{r_{i+1}} \frac{dr}{k_{i,i+1}S_r} = \int_{r_i}^{r_{i+1}} \frac{dr}{k_{i,i+1}2\pi r\Delta z} = \frac{\ln r_{i+1} - \ln r_i}{2\pi k_{i,i+1}\Delta z} \tag{2.26}$$

is used. For the thermal resistance in the z-direction, the approximate formula $R_{i,i+N_x} \approx \frac{\left(z_{i+N_r} - z_i\right)}{k_i \pi (r_{i+1}^2 - r_i^2)}$ is used, where the cell $i + N_r$ is below the cell i.

In the spherical case, S_r can be given as $S_r = 4\pi\, r^2$. Using this, the thermal resistance is calculated similarly as that in the cylindrical case, but now the integration yields $R_{i,i+1} \approx \frac{1}{4\pi k_{i,i+1}} \frac{r_{i+1} - r_i}{r_i\, r_{i+1}}$. From Equations (2.24) and (2.25) it is easy to obtain the ODE system

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} + \frac{Q_{gen}}{C_i} - \frac{hSu_i}{C_i} - \frac{\sigma^* S u_i^4}{C_i} \tag{2.27}$$

to determine the time evolution of the cell temperatures. Here, S is the area of the surface on which the convection and radiation occur, which will be the outer surface of the cylinder in Fig. 2.3. If one neglects the higher powers of Δr, one can easily derive that $C_i \,/\, S = c_i \rho_i \Delta r$ in both cases. Inserting these into (2.27), I can write Equation (2.27) in a simpler form:

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} + q_i - K u_i - \sigma u_i^4 \,, \tag{2.28}$$

which will be solved numerically.

3. SOLVE THE HEAT EQUATION NUMERICALLY

The numerical solution of the heat equation is frequently approached via discretization methods such as Finite Difference (FDM) [43-45] and Finite Element Methods (FEM) [44]. However, they can be extremely time-consuming since the examined system must be fully discretized both in space and time. Due to material inhomogeneities, which produce eigenvalue spectra spanning several orders of magnitude. This stiffness forces an exceedingly small CFL (Courant–Friedrichs–Lewy) limited time step for explicit schemes, rendering them inefficient or unstable. Consequently, implicit methods are traditionally employed for their unconditional stability, typically used for solving these kinds of equations, for example [47-53]. 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 powerful [54,55]. Thus, I think time is on the side of explicit methods because they can be much more straightforwardly parallelized. That is why I started to investigate explicit algorithms with improved stability properties. These explicit methods can also serve as a basis for implicit methods.

3.1. Existing Numerical Methods

Many explicit algorithms have been developed for heat conduction, convection, and radiation equations. While some methods adopt a purely explicit calculation strategy, others employ a mixed approach, integrating explicit and implicit calculations to balance computational efficiency and stability. Some of them are unconditionally stable for the linear heat conduction equation, and have special characteristics to deal with nontrivial cases. More details in the following:

3.1.1. The Explicit-Euler Method

The FTCS (Forward Time Central Space) scheme, the most common explicit method for the heat equation, uses the Explicit Euler method for time integration [44]. Now I adapt this to the heat transfer equation in the most standard way, thus the general formula is the following:

$$u_i^{n+1} = \left(1 - mr_i\right)u_i^n + A_i - \Delta t \cdot K_i \cdot u_i^n - \Delta t \cdot \sigma_i \cdot (u_i^n)^4 \quad . \tag{3.1}$$

In case the convection and radiation terms move to the denominator, then the Explicit Euler is called the Non-Standard Explicit Euler NS-ExpE.

3.1.2. The Crank-Nicolson Method

The Crank-Nicolson method [28] provides an alternative implicit scheme to provide accuracy. Difference approximations are developed at the midpoint of the time increment, and it is unconditionally stable, second-order accurate in both space and time, suitable for stiff systems. Thus the general formula is the following:

$$u_i^{\mathrm{n}+1} = \frac{\left(1 - {}^{mr_i}\!/_2\right) u_i^{\mathrm{n}} + A_i + 1 - \Delta t \cdot K_i \cdot u_i^n - \Delta t \cdot \sigma_i \cdot (u_i^n)^4}{1 + mr_i \left(1 - {}^{1}\!/_{2}\right)} \quad . \tag{3.2}$$

3.1.3. The Uniformly Positively Fractionalized Difference (UPFD) Method

The UPFD method is constructed by Chen-Charpentier and Kojouharov [56] for the linear advection - diffusion -reaction equation. Recently, I adapted it to the heat transfer equation as follows:

$$u_i^{n+1} = \frac{u_i^n + A_i}{1 + mr_i + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot (u_i^{\mathrm{n}})^3} \,. \tag{3.3}$$

3.1.4. The Dufort–Frankel Method

The Dufort–Frankel (DF) algorithm is a known but non-traditional explicit scheme [57] that is unconditionally stable for the linear heat equation. Now the formula for the case of Eq. (2.20) and (2.28) is as follows:

$$u_i^{\mathrm{n}+1} = \frac{\left(1 - mr_{ii}\right) u_i^{n-1} + 2A_i - 2 \cdot \Delta t \cdot K \cdot u_i^{\mathrm{n}} - 2 \cdot \Delta t \cdot \sigma \cdot (u_i^{\mathrm{n}})^4}{1 + mr_{ii}} \tag{3.4}$$

One can see that the formulas contain u_i^{n-1}, thus it is a two-step but one-stage method. As the method is not self-starting, an additional technique is required to initialize it by computing u_i^1. The UPFD formula (3.3). is employed for this initial calculation

In case of the convection and radiation terms move from the numerator to the denominator, then the DF is called Non-Standard Dufort–Frankel NS-DF.

3.1.5. The Rational Runge–Kutta Methods

From the family of the Rational Runge-Kutta methods, I chose a two-stage version [58] with the following definition. A full step is taken at the first stage, by the Explicit-Euler scheme, to obtain the predictor value. The increment for Eq. (2.20) is calculated as

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

and

$$g_i^1 = -mru_i^n + A_i - \Delta t \cdot K \cdot u_i^{\mathrm{n}} - \Delta t \cdot \sigma \cdot (u_i^{\mathrm{n}})^4 \,.$$

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

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

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

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

and $g_i^2 = -mr_i u_i^{\mathrm{pred}} + A_i^{\mathrm{new}} - \Delta t \cdot K \cdot u_i^{\mathrm{pred}} - \Delta t \cdot \sigma \cdot (u_i^{\mathrm{pred}})^4 \,.$

Now one needs to calculate the following scalar products

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

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

$$u_i^{n+1} = u_i^n + \frac{2p_1 g_i^1 - 2p_{12} g_i^1 + p_1 g_i^2}{4p_1 - 4p_{12} + p_2} \tag{3.5}$$

3.1.6. The Heun's Method

Heun's method, also referred to as the explicit trapezoidal rule, is a prevalent second-order Runge-Kutta (RK) algorithm for integrating ordinary differential equations ODEs and systems of ODEs [59], so it is straightforward to use it as a component of method of lines. It starts with a predictor step, which is an explicit Euler stage. In the cases of Eq. (2.20) and (2.28), it has the form:

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

A corrector step subsequently refines the solution by averaging the newly predicted and previous values of u :

$$u_i^{n+1} = u_i^n - mr_i \frac{u_i^n + u_i^{\text{pred}}}{2} + \frac{\Delta t}{2}\left(A_i + A_i^{\text{new}} - K \cdot \left(u_i^n + u_i^{\text{pred}}\right) - \sigma \cdot \left(u_i^n + u_i^{\text{pred}}\right)^4 \right) \tag{3.6}$$

3.1.7. The Original Odd-Even Hopscotch Method

To use an odd-even hopscotch method, a special, so-called bipartite spatial grid is necessary, where the cells are labelled 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 Fig. 3.1A. Originally, the standard Explicit Euler formula was applied in the first stage, and the Implicit-Euler formula was applied in the second stage [61]. The general formulas are the following:

$$\text{Explicit Euler:}\quad u_i^{n+1} = \left(1 - mr_i\right) u_i^n + A_i - \Delta t \cdot K_i \cdot u_i^{\text{n}} - \Delta t \cdot \sigma_i \cdot (u_i^{\text{n}})^4 \tag{3.7}$$

$$\text{Implicit Euler:}\quad u_i^{n+1} = \frac{u_i^n + A_i^{\text{new}}}{1 + mr_i + \Delta t \cdot K_i + \Delta t \cdot \sigma_i \cdot (u_i^{\text{n}})^3}\,, \tag{3.8}$$

This implicit formula becomes effectively explicit because the required u_j^{n+1} values are already available from Stage 1. This defines the Original Odd-Even Hopscotch (OOEH) method, which is unconditionally stable for the linear heat equation. However, for nonlinear cases with large r values, the update can yield negative temperatures. This instability arises when large negative values appear in the denominator term $(u_i^{\text{n}})^3$. To mitigate this, a simple stabilization is applied; negative values are prohibited using a conditional statement:

$$\text{if } u_i^{n+1} < 0 \text{ then } u_i^{n+1} = 0\,. \tag{3.9}$$

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

In the case of using the Non-Standard Explicit Euler, then the OOEH is called Non-Standard odd-even hopscotch NS-OEH.

3.1.8. *The Reversed Odd-Even Hopscotch Method*

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 (3.8), then the nonstandard explicit Euler formulas (3.7), with condition (3.9). 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 in the old time level, which is the same trick as the UPFD method uses.

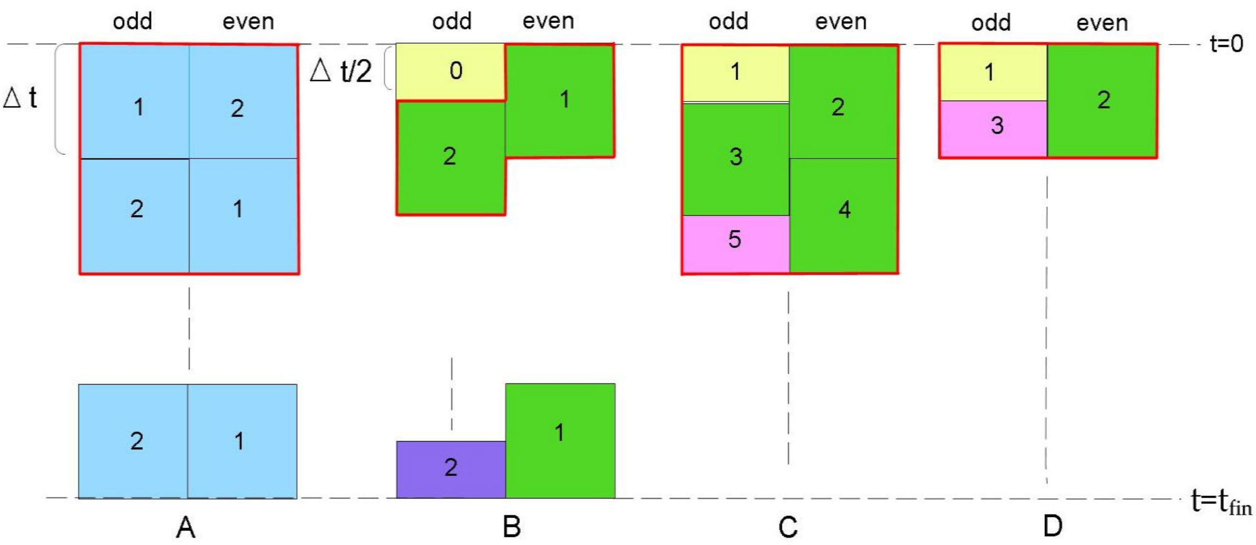

Figure 3.1. Space-time structure of (**A**) The original hopscotch and the reversed hopscotch methods. (**B**) The leapfrog-hopscotch method. (**C**)The shifted-hopscotch method. (**D**) The asymmetric hopscotch method [60].

3.2. *The Developed Numerical Methods*

In this section, I present some methods which were originally constructed for the heat conduction equation without my participation, but I took part to adapt them to the case with conduction and radiation.

3.2.1. *The Constant Neighbor Method*

The constant neighbor (CNe) method [28] for Equation (2.20) and (2.28) is:

$$u_i^{n+1} = u_i^n \cdot e^{-mr_i} + \frac{A_i}{mr_i}\left(1-e^{-mr_i}\right) + \Delta t \cdot q - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot (u_i^n)^4 . \tag{3.10}$$

3.2.2. *The Theta Method*

To proceed, the following general time discretization formula defines the theta method is used:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left[\theta\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + (1-\theta)\left(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}\right)\right] + q - K \cdot u_i^n - \sigma \cdot (u_i^n)^4 , \tag{3.11}$$

where $\theta \in [0,1]$. After rearrangement we have

$$u_i^{n+1} = u_i^n + mr\left[\theta\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + (1-\theta)\left(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}\right)\right] + \Delta t \cdot q - \Delta t \cdot K \cdot u_i^n - \Delta t \cdot \sigma \cdot (u_i^n)^4 , \tag{3.12}$$

For $\theta = 0, \frac{1}{2}, \text{and } 1$ yield the implicit Euler, Crank-Nicolson, and explicit Euler (FTCS) schemes, respectively [62]. If $\theta < 1$, the theta method is implicit. It can be modified to be explicit by taking the neighbors into account at the old-time level, where their values are already calculated. Thus, one can insert $u_{i\pm1}^{n}$ into the theta-scheme (3.12) instead of $u_{i\pm1}^{n+1}$ to obtain

$$u_i^{n+1} = u_i^n - 2mr\theta u_i^n - 2mr(1-\theta)u_i^{n+1} + mr\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t\cdot q - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4 \qquad (3.13)$$

With this modification, the final formula is completely explicit:

$$u_i^{\mathrm{n+1}} = \frac{(1-2mr\theta)u_i^n + mr\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right) + \Delta t\cdot q - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4}{1+2mr(1-\theta)} \qquad (3.14)$$

3.2.3. The Two-Stage and Three-Stage Linear-Neighbor Method

The subsequent algorithm is the two-stage linear-neighbor (LNe or LNe2) method [63]. This method uses the CNe scheme as a predictor to obtain new $u_{\mathrm{i}}^{\mathrm{pred}}$ values at the end of the time step.For the special case of an equidistant grid, these predicted values are used to calculate slopes:

$$s_i = \frac{mr}{\Delta t^2}\left(u_{i-1}^{\mathrm{pred}} + u_{i+1}^{\mathrm{pred}} - u_{i-1}^{\mathrm{n}} - u_{i+1}^{\mathrm{n}}\right)$$

and then the corrector values for the two-stage LNe method:

$$u_i^{n+1} = u_i^n e^{-2mr} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1-e^{-2mr}\right) + s_i\frac{\Delta t^2}{2mr}\left(1-\frac{1-e^{-2mr}}{2mr}\right) + \Delta t\cdot q - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4$$

For the general case, the corrector step is implemented as follows:

$$u_{\mathrm{i}}^{\mathrm{n+1}} = u_{\mathrm{i}}^{\mathrm{n}} e^{-mr_{\mathrm{i}}} + \left(A_{\mathrm{i}} - \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{mr_{\mathrm{i}}}\right)\frac{1-e^{-mr_{\mathrm{i}}}}{mr_{\mathrm{i}}} + \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{mr_{\mathrm{i}}} - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4 . \qquad (3.15)$$

The results from (3.13) can then be used to recalculate A_i^{new} again, Iterating the corrector step (3.15) with these updated slopes yields a new solution. This three-stage altogethe is designated the LNe3 method [63]. While remaining second-order accurate, the LNe3 algorithm provides improved accuracy over the two-stage LNe2 method.

3.2.4. The Two-Stage Constant-Neighbour

The two-stage constant neighbour *CpC* method [64] generally starts with a fractional time step of length $p\Delta t$, the constant-neighbour method with a full time step is briefly termed CpC. Here, I take $p = \frac{1}{2}$ because this version usually offers better accuracy than versions with other values of *p*. Consequently, in the first stage, I calculate new predictor values for the variables using the CNe formula, but with a $\Delta t_1 = \Delta t / 2$ time step:

$$u_i^{\mathrm{pred}} = u_i^n\cdot e^{-mr} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1-e^{-mr}\right) + \Delta t\cdot q - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4$$

And $u_{\mathrm{i}}^{\mathrm{pred}} = u_i^n e^{-mr_i/2} + \frac{A_i}{mr_i}\left(1-e^{-mr_i/2}\right) - \Delta t\cdot K\cdot u_i^{\mathrm{n}} - \Delta t\cdot\sigma\cdot(u_i^{\mathrm{n}})^4 .$ (3.16)

Then, in the second stage, I can use $A_i^{\text{new}} = \Delta t \sum_{j \neq i} \frac{u_j^{\text{pred}}}{C_i R_{ij}}$ with Δt_1 and take a full-time step size corrector step by applying the CNe formula again. The final values at the end of the time step are

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

$$u_i^{n+1} = u_i^n \cdot e^{-mr_{\text{i}}} + \frac{A_i^{\text{new}}}{mr_{\text{i}}}\left(1 - e^{-mr_{\text{i}}}\right) - \Delta t \cdot K \cdot u_i^{\text{n}} - \Delta t \cdot \sigma \cdot (u_i^{\text{n}})^4 \tag{3.17}$$

3.3. The Invented Numerical Methods

3.3.1. The Shifted-Hopscotch Method

The shifted-hopscotch (SH) method [65] is a new method I invented. It has a repeating block consisting of five stages: two half-time steps and three full-time steps, which altogether span two integer time steps for both the odd and even cells, as shown in Figure 3.1C. The first half-time step is taken for the odd cells using the general formula:

$$u_i^{n+\frac{1}{2}} = \frac{u_i^n + A_i + \Delta t \cdot q}{1 + 2mr + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^n\right)^3} \tag{3.18}$$

coloured by a yellow box with the number 1 in the figure. Full-time steps, taken strictly alternately, follow the formula:

$$u_i^{\mu+1} = \frac{\left(1 - mr_i / 2\right) u_i^{\mu} + A_i^{\mu+\frac{1}{2}} + \Delta t \cdot q}{1 + mr + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{\mu}\right)^3} \tag{3.19}$$

The upper index μ is n for the even nodes and n+1 for the odd nodes. These steps for the even, the odd, and the even cells are colorued by green boxes with the numbers 2, 3, and 4. The calculation is finally closed by a half-length time step (pink box with number 5 inside) for the odd cells with the formula:

$$u_i^{n+2} = \frac{\left(1 - mr_i\right) u_i^{n+1} + A_i^{n+\frac{3}{2}} + \Delta t \cdot q}{1 + \Delta t \cdot K + \Delta t \cdot \sigma \cdot \left(u_i^{n+1}\right)^3} \tag{3.20}$$

with condition (3.9) again.

3.3.2. The Leapfrog-Hopscotch Method

The architectural framework of the novel leapfrog-hopscotch (LH) algorithm [66], invented by our team with my participation, incorporates two half-time steps and several full-time steps, as one can see in Fig. 3.1B. Computation is initiated by applying the general formulae from (3.18) in the first stage (yellow box in the Figure). A strictly alternating sequence of full-time steps (green boxes) for even and odd nodes follows, governed by expressions (3.19) and subject to condition (3.9). A key principle of the method is that the most recent available data from neighboring points used (for example in $A_i^{\mu+\frac{1}{2}}$) must always be used to update the values of u, no matter the size of the time step being calculated. This alternation continues until the algorithm processes the last time step (purple box in Figure), which again uses (3.19) with a halved step size to ensure the odd nodes conclude at the same time as the even nodes.

3.3.3. The Asymmetric Hopscotch Method

The Asymmetric Hopscotch (ASH) Method is another novel algorithm, bearing similarity to the SH method but contains a reduced number of integer stages; it utilizes three stages instead of five, as shown in Fig. 3.1D. The computational process is initiated by a half-time step for the odd cell, governed by equation (3.18). This is followed by a full-time step applied to the even cell using formula (3.19) under condition (3.9). The calculation cycle is concluded by a final half-time step equation (3.20), with condition (3.9) again for the last odd cell.

3.3.4. The Pseudo-Implicit Method

I helped in the invention of the pseudo-implicit (PI) method, which is a new explicit method called Algorithm 5 in [67] with parameter $\lambda = 1$. For Eq. (2.20) and (2.28) the following two-stage algorithm is applied:

$$\textit{Stage 1}:\ u_i^{\text{pred}} = \frac{u_i^n + A_i/2}{1 + mr_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^{\text{n}})^3} \tag{3.21}$$

$$\textit{Stage 2}:\ u_i^{n+1} = \frac{(1 - mr_i)u_i^n + A_i^{\text{new}}}{1 + mr_i + \Delta t \cdot K + \Delta t \cdot \sigma \cdot (u_i^{\text{pred}})^2 \cdot u_i^{\text{n}}} \tag{3.22}$$

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

3.4. The Optimization of Shifted-Hopscotch Method Combinations

I constructed and tested innovative numerical algorithms to solve the transient diffusion equation (or heat conduction) equation [62], These methods represent a novel approach to addressing this class of problems

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \tag{3.23}$$

The new algorithms are fully explicit time-integrators obtained by a half-time step and applied different formulas in different stages. All of the algorithms consist of five stages, but they are one-step methods in the sense that when the new values of the unknown function *u* are calculated, only the most recently calculated *u* values are used, thus the methods can be implemented such that only one array of storage is required for the *u* variable, which means that the memory requirement is very low. I applied the conventional theta-method with 9 different values of θ and the non-conventional CNe method to construct 10^5 combinations in the case of small systems with random parameters, and examined the competitiveness of the best algorithms by testing them in the case of large systems against popular solvers.

The computational procedure is initiated by taking a half-length time step for the odd nodes (subset A) using the already calculated u_i^{n} values. This is followed by a sequence of operations: a full-step for the even nodes (subset B), a subsequent full-step for the odd cells, and another for the even nodes. A concluding half-interval step finalizes the value computation, as one can see in Figure 3.1.C. In each stage, I use the latest available *u* values of the neighbors, which means that the constructed methods are fully explicit and the previous values needn't to be stored at all. Thus, I have a structure consisting of 5 stages, which correspond to 5 partial time steps, that together span two complete time steps for all cells.

The application of the standard central difference formula (2.12) to Eq. (3.23) in one dimension produces a system of ordinary differential equations (ODEs) governing the nodes $i = 1,...,N-2$:

$$\frac{du_{\mathrm{i}}}{dt} = \alpha \frac{u_{\mathrm{i\text{-}1}} - 2u_{\mathrm{i}} + u_{\mathrm{i+1}}}{\Delta x^2}. \tag{3.24}$$

The form of this equation for the first and last node depends on the concrete boundary conditions which will be discussed later. I define a matrix *M* with the following elements:

$$m_{\mathrm{ii}} = -\frac{2\alpha}{\Delta x^2}\ \ (1 < i < N),\ \ m_{\mathrm{i,i+1}} = \frac{\alpha}{\Delta x^2}\ \ (1 \le i < N),\ \ \ m_{\mathrm{i,i-1}} = \frac{\alpha}{\Delta x^2}\ \ (1 < i \le N), \tag{3.25}$$

which is tridiagonal in the currently discussed 1D case. Now equation-system (3.24) can be expressed in a compact matrix form:

$$\frac{d\vec{u}}{dt} = M\vec{u} \tag{3.26}$$

I now introduce the following general time-discretization scheme

$$\frac{u_i^{\mathrm{n+1}} - u_i^{\mathrm{n}}}{\Delta t} = \frac{\alpha}{\Delta x^2}\left[\theta\left(u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) + (1-\theta)\left(u_{i-1}^{\mathrm{n+1}} - 2u_i^{\mathrm{n+1}} + u_{i+1}^{\mathrm{n+1}}\right)\right],$$

leads to the named theta-method:

$$u_i^{\mathrm{n+1}} = u_i^{\mathrm{n}} + mr\left[\theta\left(u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) + (1-\theta)\left(u_{i-1}^{\mathrm{n+1}} - 2u_i^{\mathrm{n+1}} + u_{i+1}^{\mathrm{n+1}}\right)\right], \tag{3.27}$$

where $r = \frac{\alpha\Delta t}{\Delta x^2} = -\frac{m_{ii}\Delta t}{2} > 0,\ 0 < i < N-1$ is the usual mesh ratio and $\theta \in [0,1]$. For $\theta = 0, \frac{1}{2}, \text{ and } 1$ one obtains the Implicit-Euler, the Crank-Nicolson and the Explicit-Euler (or, more concretely, the forward-time central-space, FTCS) schemes, respectively [62]. If $\theta > 0$, the theta-method is implicit. Now, in our shifted-hopscotch scheme, the neighbors are always taken into account at the same, latest time level, thus I insert $u_{i\pm1}^{\mathrm{m}}$ into (3.27) instead of $u_{i\pm1}^{n}$ and $u_{i\pm1}^{n+1}$, where $m = n, n+\frac{1}{2}, \text{ or } n+1$ at the first, middle, and last stages, respectively. Now, instead of (3.27), I can write

$$u_i^{\mathrm{n+1}} = u_i^{\mathrm{n}} - 2mr\theta u_i^{\mathrm{n}} - 2mr(1-\theta)u_i^{\mathrm{n+1}} + mr\left(u_{i-1}^{m} + u_{i+1}^{m}\right), \tag{3.28}$$

i.e. my final formula reads as follows:

$$u_i^{\mathrm{n+1}} = \frac{(1-2mr\theta)u_i^{\mathrm{n}} + mr\left(u_{i-1}^{m} + u_{i+1}^{m}\right)}{1 + 2mr(1-\theta)}. \tag{3.29}$$

In the case of $\theta = 0$, this formula gives back the UPFD method [37], [38] with *m*=*n*, which takes the form for a half and a full time step, respectively:

$$u_i^{\mathrm{n}+1} = \frac{u_i^{\mathrm{n}} + {}^{mr}\!/_{2}\left(u_{i-1}^m + u_{i+1}^m\right)}{1+mr}\ ,\ u_i^{\mathrm{n}+1} = \frac{u_i^{\mathrm{n}} + mr\left(u_{i-1}^m + u_{i+1}^m\right)}{1+2mr}\,. \qquad (3.30)$$

The other formula I use is the constant neighbor (CNe) method, which is introduced in section 3.2.1 and now briefly restated here. The starting point is Eq. (3.24), where an approximation is made: when the new value of a variable $u_{\mathrm{i}}^{\mathrm{n}+1}$ is calculated, I neglect the fact that the neighbors u_{i-1}^{n} and u_{i+1}^{n} are also changing during the time step. It means that the values of u_{j} $(\mathrm{j} \neq \mathrm{i})$ are considered as constants (that is why I call it constant-neighbor method). Taken into account the spatial discretization of heat equation in section 2.2 the general form of Equation (3.28) will be:

$$u_i^{\mathrm{n}+1} = u_i^{\mathrm{n}} - mr_i\theta u_i^{\mathrm{n}} - mr_i\left(1-\theta\right)u_i^{\mathrm{n}+1} + h\sum_{\mathrm{j}\neq\mathrm{i}}\frac{u_{\mathrm{j}}^m}{C_i R_{\mathrm{ij}}}\,;$$

thus, the generalized theta-method for integer time steps reads as follows:

$$u_i^{\mathrm{n}+1} = \frac{\left(1-mr_i\theta\right)u_i^{\mathrm{n}} + A_i}{1+mr_i\left(1-\theta\right)} \qquad (3.31)$$

Similarly, the generalized CNe formula is

$$u_i^{\mathrm{n}+1} = u_i^{\mathrm{n}}\cdot e^{-mr_{\mathrm{i}}} + \frac{A_i}{mr_{\mathrm{i}}}\left(1-e^{-mr_{\mathrm{i}}}\right) \qquad (3.32)$$

and of course, for halved time steps r_{i} and A_{i} must be divided by 2.

For the sake of brevity, I will use a compact notation of the individual combinations, where 5 data is given in a bracket, the numbers are the values of the parameter θ, while the letter ‘C’ is for the CNe constant neighbor method. For example (¼, ½, C, ½, ¾) means the following 5-stage algorithm, which will be selected from the top 5 algorithms in section 3.4, and named as A2.

Example 1. Algorithm A2 (¼, ½, C, ½, ¾), general from.

Stage 1. Take a half time step with the (3.31) formula with θ=¼ for odd cells:

$$u_i^{\mathrm{n}+1} = \frac{\left(1-{}^{mr_i}\!/_{8}\right)u_i^{\mathrm{n}} + A_{i,\mathrm{half}}}{1+{}^{mr_i}\!/_{2}\left(1-{}^{1}\!/_{4}\right)}\ ,\ A_{i,\mathrm{half}} = \frac{\Delta t}{2}\sum_{\mathrm{j}\neq\mathrm{i}}\frac{u_{\mathrm{j}}^m}{C_i R_{\mathrm{ij}}}\,.$$

Stage 2. Take a full-time step with the (3.31) formula with θ=½ for even cells:

$$u_i^{\mathrm{n}+1} = \frac{\left(1-{}^{mr_i}\!/_{2}\right)u_i^{\mathrm{n}} + A_i}{1+mr_i\left(1-{}^{1}\!/_{2}\right)}\ ,\ A_i = \Delta t\sum_{\mathrm{j}\neq\mathrm{i}}\frac{u_{\mathrm{j}}^m}{C_i R_{\mathrm{ij}}}\,.$$

Stage 3. Take a full-time step with the (3.32) formula for odd cells:

$$u_i^{\mathrm{n}+1} = u_i^{\mathrm{n}}\cdot e^{-mr_{\mathrm{i}}} + \frac{A_i}{mr_{\mathrm{i}}}\left(1-e^{-mr_{\mathrm{i}}}\right),\ A_i = \Delta t\sum_{\mathrm{j}\neq\mathrm{i}}\frac{u_{\mathrm{j}}^m}{C_i R_{\mathrm{ij}}}\,.$$

Stage 4. The same as Stage 2.

Stage 5. Take a half time step with the (3.31) formula with θ= ¾ for odd cells:

$$u_i^{n+1} = \frac{\left(1-{}^{3}\!/_{8}\, mr_i\right)u_i^{n} + A_{i,\text{half}}}{1+{}^{mr_i}\!/_{2}\left(1-{}^{3}\!/_{4}\right)}\,,\quad A_{i,\text{half}} = \frac{\Delta t}{2}\sum_{j\neq i}\frac{u_j^{m}}{C_i R_{ij}}\,.$$

All other combinations can be constructed in this manner straightforwardly.

3.4.1. General Definitions and Circumstances of The Examination

I examine 2-dimensional rectangular lattices with $N = N_x \times N_z$ cells similar to what can be seen in Figure 3.2. I solve Eq. (2.20) subjected to randomly generated initial conditions $u_i(0) = rand$, with *rand* being a MATLAB-generated, uniformly distributed (pseudo) random number in (0, 1) for each cell. Model parameters, including heat capacities and thermal resistances, were also randomized using a log-uniform distribution as follow:

$$C_i = 10^{(\alpha_C - \beta_C \times rand)},\ R_{x,i} = 10^{(\alpha_{Rx} - \beta_{Rx} \times rand)},\ R_{z,i} = 10^{(\alpha_{Rz} - \beta_{Rz} \times rand)}$$

where the coefficients $\alpha_C ,...,\beta_{Rz}$ in the exponents will be concretized later.

I use zero Neumann boundary conditions, i.e., the system is thermally isolated. This condition is implemented naturally within the framework of Eq. (2.19). Implementation simply requires the omission of any summation terms containing infinite resistivity in their denominator, a consequence of the isolated boundary. This implies that the system matrix *M* has one zero eigenvalue, belongs to the uniform distribution of temperatures, all other eigenvalues must be negative.

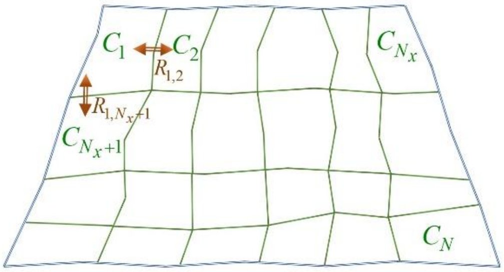

Figure 3.2. Arrangement of the generalized variables. The double-line red arrows symbolize conductive (heat) transport through the resistances R_{ij}. The blue line symbolizes thermal isolation at the boundaries of the system [65].

I calculate the numerical error by comparing our numerical solutions u_j^{num} with the reference solution u_j^{ref} at final time t_{fin}. In Subsection 3.4.5 the reference solution will be an analytical solution, otherwise it is a very accurate numerical solution which has been calculated by the ode15s built-in solver of MATLAB with very strict error tolerance. I use the following three types of (global) error. The first one is the maximum of the absolute differences:

$$\text{Error}(L_\infty) = \max_{0\le j\le N}\left|u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}})\right|. \tag{3.33}$$

The second one is the average absolute error:

$$\text{Error}(L_1) = \frac{1}{N}\sum_{0\le j\le N}\left|u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}})\right|. \tag{3.34}$$

The third one gives the error in terms of energy in case of the heat equation. It takes into account that an error of the solution in a cell with a large volume or heat capacity has more significance in practice than in a very small cell

$$\mathrm{Error}(Energy) = \frac{1}{N}\sum_{0\le \mathrm{j}\le N} C_{\mathrm{j}}\left|u_{\mathrm{j}}^{\mathrm{ref}}(t_{\mathrm{fin}}) - u_{\mathrm{j}}^{\mathrm{num}}(t_{\mathrm{fin}})\right| . \tag{3.35}$$

It is well known that the true solution always follows the maximum and minimum principles [62]. We say a method is positivity preserving if it never violates this principle, i.e., in our case no value of u is outside of the $[0,1]$ interval. I am interested in how these errors depend on the time step size in different concrete situations. As one can see in Figure 3.1C, there are 5-time steps (5 stages) altogether instead of 4 in the shifted hopscotch structure, so for the sake of honesty I must calculate the effective time step size as $\Delta t_{\mathrm{EFF}} = 4/5\,\Delta t$ and the errors will be plotted as a function of this quantity.

3.4.2. Preliminary Tests

I apply the following 9 different values for parameter theta: $\theta \in \{0, 1/5, 1/4, 1/3, 1/2, 2/3, 3/4, 4/5, 1\}$ in Eq. (3.31). It means that together with the CNe formula, I have 10 different formulas and I insert all of these into the shifted-hopscotch structure in all possible combinations. As there are 5 stages in the structure, I have 10^5=100000 different algorithm combinations. The code systematically constructs and tests all these combinations. After some tests, a few best combinations choose and continue the work only with them. For this an automatic assessment of the performance of the combinations is needed. The difficulty lies in the fact that methods which are very inaccurate or even unstable for large-time step sizes can be the most accurate for small time step sizes. Therefore, I choose two different final times $t_{\mathrm{fin}} = 0.1, 10$, the solution is first computed using a large time step (typically $t_{\mathrm{fin}}/4$). This calculation is then iteratively repeated for successively halved time step sizes until a minimal value is attained (typically approximately 2×10^{-6}). For the quantitative assessment of each previously defined error type, I introduce the so-called aggregated relative error (ARE), which can be calculated for the L_∞ error as follows:

$$\mathrm{ARE}(L_\infty) = \frac{1}{R}\sum_{i=1}^{R}\Big(\log\big(\mathrm{Error}(L_\infty)\big)_{\mathrm{OEH}} - \log\big(\mathrm{Error}(L_\infty)\big)_{\mathrm{shifted}}\Big), \tag{3.36}$$

which means that $\mathrm{ARE}(L_\infty)$ is the average of the difference between the error of the original OEH method and the actual shifted combination in terms of orders of magnitude. Then the code calculates the simple average of these errors:

$$\mathrm{ARE} = \frac{1}{3}\big(\mathrm{ARE}(L_\infty) + \mathrm{ARE}(L_1) + \mathrm{ARE}(Energy)\big), \tag{3.37}$$

and finally sorts the 100000 combinations in decresing order according to this quantity. In the obtained list usually positive ARE values have been assigned the first few thousands of combinations, the largest ones have been typically around 2, which means that some combinations are roughly two orders of magnitude more accurate than the original OEH method. I performed this procedure in case of 4 different small systems with $N_{\mathrm{x}} \times N_{\mathrm{z}} = 2\times 2, 2\times 6, 4\times 4, \text{and } 3\times 5$. The parameters α_C , β_C , α_{Rx} , β_{Rx} , α_{Rz} , β_{Rz} of the mesh-cell data distribution were selected to construct test problems exhibiting a range of stiffness ratios, defined as $\lambda_{\mathrm{MAX}}/\lambda_{\mathrm{MIN}}$. Here λ_{MIN}, and λ_{MAX} represent the smallest and largest non-zero absolute eigenvalues of the matrix *M*, respectively. The maximum possible time step size for the FTCS (Explicit Euler) scheme (from the point of view of stability) can be exactly calculated as $h_{\mathrm{MAX}}^{\mathrm{FTCS}}$, for example $\alpha_C = 1, 2, \text{or } 3$, $\beta_C = 2, 4, \text{or } 6$. I give the best 12 combinations in their short form:

$$\begin{gathered}(0, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, 1), \quad (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}), \quad (0, C, \tfrac{1}{2}, C, 1), \quad (0, C, C, C, 1), \\ (\tfrac{3}{4}, \tfrac{2}{3}, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}), \quad (\tfrac{1}{4}, \tfrac{1}{2}, C, \tfrac{1}{2}, \tfrac{3}{4}), \quad (\tfrac{1}{3}, \tfrac{2}{3}, C, \tfrac{1}{3}, \tfrac{2}{3}), \quad (C, \tfrac{1}{2}, C, \tfrac{1}{2}, C), \\ (\tfrac{1}{5}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{4}{5}), \quad (\tfrac{1}{4}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{3}{4}), \quad (\tfrac{1}{3}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{2}{3}), \quad (0, \tfrac{1}{2}, \tfrac{1}{2}, C, 1).\end{gathered} \tag{3.38}$$

Later preserves the positivity of the solution prove for formulas $\theta=1$ and CNe and therefore if only these two formulas are used in a combination, the whole algorithm will preserve positivity. Since this property is considered valuable [56], I repeated the above experiments for these $2^5=32$ combinations (instead of the 100000 above). I concluded that the (C, C, C, C, C) combination is the most accurate among these, therefore I further investigate 13 combinations altogether. I emphasize that these are the results of only preliminary (one might say tentative) tests, with the sole purpose of reducing the huge number of combinations into a manageable number, and I haven't stated anything exactly until this point.

3.4.3. Case study I and Comparison with Other Solvers

I test a mesh similar to Figure 3.2 with an isolated boundary. The mesh sizes were set to $N_x = 100$ and $N_z = 100$ (10,000 total cells) and a final time of $t_{fin} = 0.1$.

$$\alpha_C = 2,\ \beta_C = 4,\ \alpha_{Rx} = \alpha_{Rz} = 1,\ \beta_{Rx} = \beta_{Rz} = 2, \tag{3.39}$$

The exponents defined previously were assigned specific values, resulting in log-uniformly distributed capacities within the range of 0.01 to 100. The generated system was characterized by its stiffness ratio and $h_{\mathrm{MAX}}^{\mathrm{FTCS}}$ values, calculated as 3.1×10^{7} and 7.3×10^{-4}, respectively. A performance analysis was conducted, comparing the novel algorithms against established MATLAB solvers:

- ode15s: A variable-step, variable-order (VSVO) solver utilizing first- to fifth-order numerical differentiation formulas (NDFs), designed for stiff systems.
- ode23s: An implicit solver based on a modified second-order Rosenbrock formula.
- ode23t: An implementation of the implicit trapezoidal rule with a free interpolant.
- ode23tb: A solver combining the trapezoidal rule with backward differentiation formulas.
- ode45: An explicit Runge-Kutta solver based on the fourth/fifth-order Dormand-Prince method.
- ode23: An explicit second/third-order Runge-Kutta-Bogacki-Shampine method.
- ode113: A VSVO predictor-corrector solver implementing Adams-Bashforth-Moulton schemes of orders 1 to 13.

For each MATLAB solver, tolerance parameters were varied over several orders of magnitude, from a maximum of $\text{'AbsTol'} = \text{'RelTol'} \doteq \text{'Tol'} = 10^{3}$ to the minimum value $\text{'AbsTol'} = \text{'RelTol'} \doteq \text{'Tol'} = 10^{-5}$. The resulting L_∞ and energy errors were plotted against the effective time step size Δt_{EFF}. Based on this analysis, the top five combinations from the set defined in (3.38) were selected and after that:

S1 (C, C, C, C, C),

S2 (¼, ½, C, ½, ¾),

S3 (¼, ½, ½, ½, ¾),

S4 (0, ½, ½, ½, 1),

S5 (0, ½, ½, C, 1)

In Figures 3.3 and 3.4, I present the error and energy error functions only for these top five combinations, while Figure 3.5 plots the energy errors versus the total running times. Table 3.1 lists some results obtained by our numerical schemes and the "ode" routines of MATLAB. Notably, the results demonstrate that the best combination of the shifted-hopscotch method achieved a maximum error of 10−8, an energy error of 10−6, and a running time reaching 10−2, which is approximately four orders of magnitude better performance compared to the ordinary MATLAB routines.

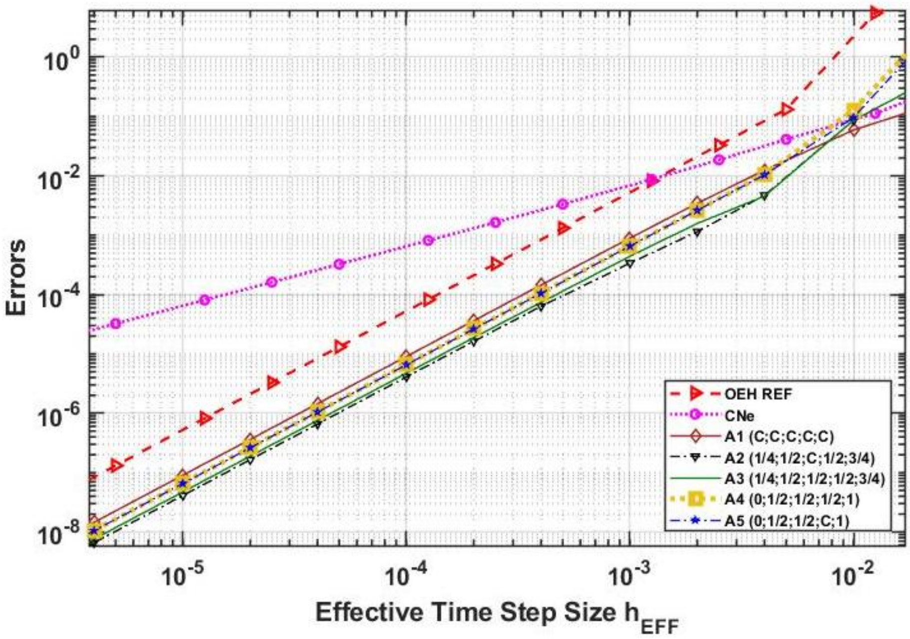

Figure 3.3. L_∞ errors as a function of the effective time step size for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), the original one stage CNe method, the new algorithms A1-A5 [65].

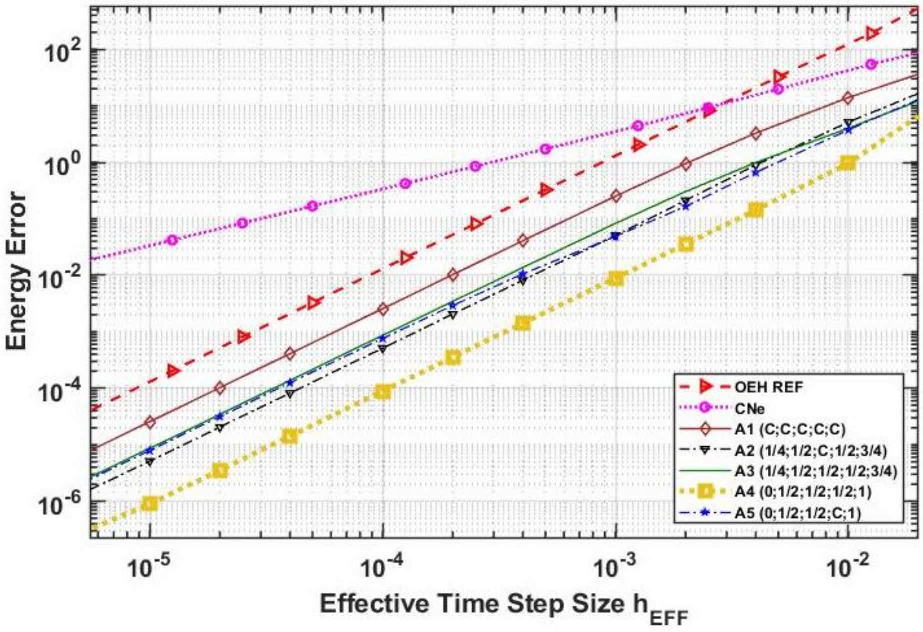

Figure 3.4. Energy errors as a function of the effective time step size for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), one stage CNe method, the new algorithms A1-A5 [65].

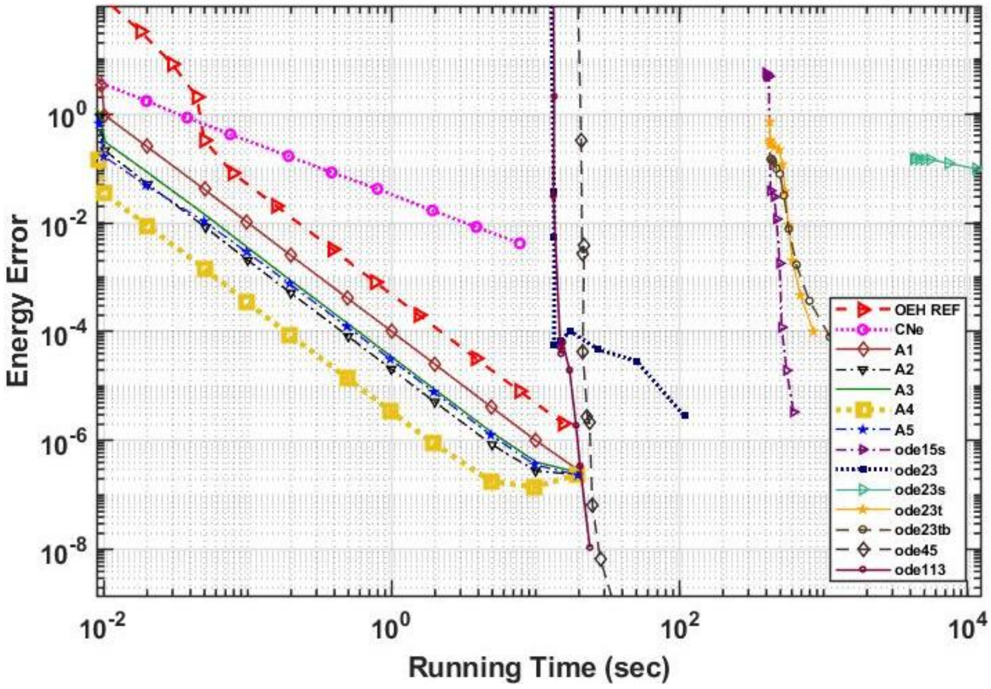

Figure 3.5. Energy errors as a function of the running time for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), one stage CNe method, the new algorithms A1-A5, and different MATLAB routines [65].

Table 3.1. Comparison of different shifted hopscotch algorithms and MATLAB routines for the moderately stiff system of ten thousand cells.

Numerical Method	**Running Time (sec)**	**Error**(L_∞)	**Error**(L_1)	**Energy Error**
ode15s, $\mathrm{Tol}=10^{3}$	3.97×10^{2}	1.3×10^{-2}	1.1×10^{-3}	5.62×10^{1}

ode23s, $\mathrm{Tol}=10^{3}$	4.346×10^{3}	4.2×10^{-4}	3.0×10^{-5}	1.5×10^{-1}
ode23t, $\mathrm{Tol}=10^{-8}$	8.49×10^{2}	2.9×10^{-7}	2.0×10^{-8}	1.0×10^{-4}
ode23tb, $\mathrm{Tol}=10^{2}$	4.28×10^{2}	4.1×10^{-4}	2.9×10^{-5}	1.4×10^{-4}
ode45, $\mathrm{Tol}=10^{-1}$	2.1×10^{1}	3.3×10^{-3}	6.5×10^{-5}	2.7×10^{-3}
ode23, $\mathrm{Tol}=10^{-6}$	2.7×10^{1}	3.7×10^{-7}	9.6×10^{-9}	4.8×10^{-5}
ode113, $\mathrm{Tol}=10^{-6}$	1.91×10^{1}	6.7×10^{-7}	4.2×10^{-10}	1.9×10^{-6}
A1, $\Delta t=1.25\times10^{-4}$	1.97×10^{-1}	9.06×10^{-6}	2.63×10^{-7}	2.56×10^{-3}
A2, $\Delta t=1.25\times10^{-3}$	2.02×10^{-2}	3.39×10^{-4}	6.93×10^{-6}	5.08×10^{-2}
A3, $\Delta t=2.5\times10^{-4}$	1.01×10^{-1}	1.88×10^{-5}	3.64×10^{-7}	3.44×10^{-3}
A4, $\Delta t=5\times10^{-4}$	5.03×10^{-2}	1.06×10^{-4}	1.07×10^{-6}	1.42×10^{-3}
A5, $\Delta t=2.5\times10^{-5}$	9.75×10^{-1}	2.62×10^{-7}	4.44×10^{-9}	3.15×10^{-5}

3.4.4. Case Study II and Comparison with Other Solvers

I tested our new algorithms and the conventional solvers for a harder problem as well. Consequently, new values were assigned to the α and β exponents, as defined by :

$$\alpha_C=3,\ \beta_\mathrm{C}=6,\ \alpha_{Rx}=3,\ \alpha_{Rz}=1,\ \beta_{Rx}=\beta_{Rz}=4\,. \tag{3.40}$$

This adjustment served to broaden the distribution of both the thermal capacitances and resistances, thereby introducing anisotropy into the system. On average, the resistances in the *x* direction became two orders of magnitude larger than those in the *z* direction. This modification resulted in a system characterized by a significantly higher stiffness ratio 2.5×10^{11}, while the maximum allowed time step size for the standard FTCS was $h_{MAX}^{EE}=1.6\times10^{-6}$. All remaining parameters and conditions were consistent with those detailed in Subsection 3.4.3. Figures 3.6 and 3.7 present the L_∞ and energy errors plotted against the total running time. The results indicate a performance improvement of approximately three to four orders of magnitude compared to conventional methods and the evaluated MATLAB routines.

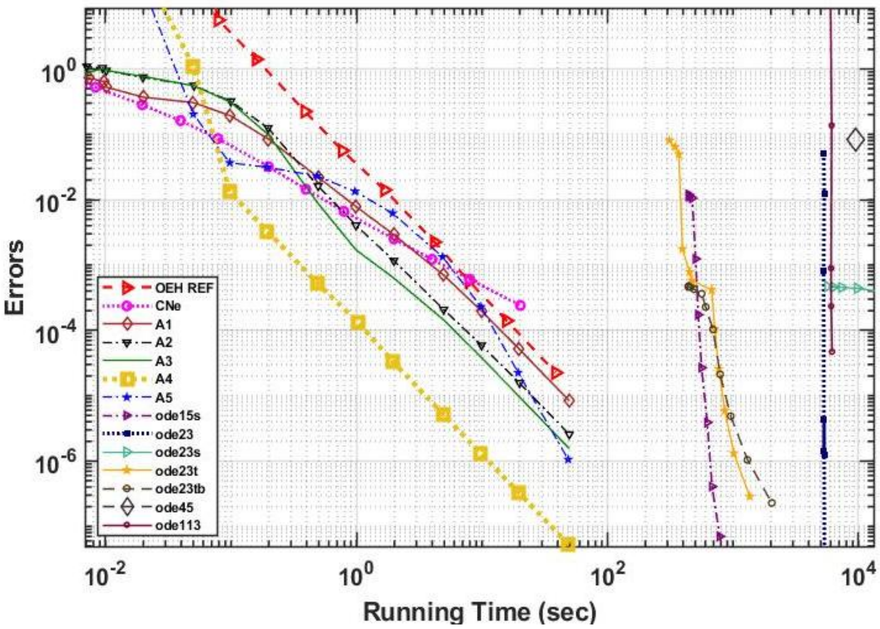

Figure 3.6. L_∞ errors as a function of the running time for the second (very stiff) system, for the original OEH method (OEH REF), the one stage CNe method, the new algorithms A1-A5, and different MATLAB routines [65].

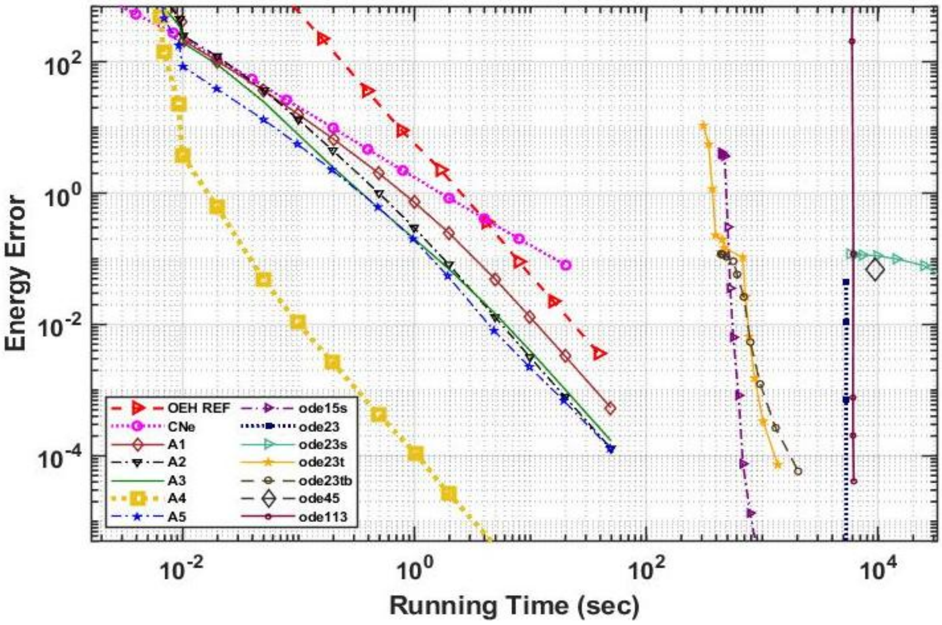

Figure 3.7. Energy errors as a function of the running time for the second system, in the case of the original OEH method (OEH REF), one-stage CNe method, the new algorithms A1-A5, and different MATLAB routines [65].

Table 3.2. Comparison of different shifted hopscotch algorithms and MATLAB routines for the very stiff system of ten thousand cells.

Numerical Method	**Running Time (sec)**	**Error(L_∞)**	**Error(L_1)**	**Energy Error**
ode15s, $\mathrm{Tol}=10^{3}$	6.8×10^{2}	4.1×10^{-7}	1.5×10^{-8}	7.5×10^{-5}
ode23s, $\mathrm{Tol}=10^{3}$	5.694×10^{3}	4.7×10^{-4}	2.4×10^{-4}	1.2×10^{-1}
ode23t, $\mathrm{Tol}=10^{3}$	3.1×10^{3}	8.1×10^{-2}	2.1×10^{-3}	1.06×10^{1}

ode23tb, $\text{Tol}=10^{3}$	2.037×10^{3}	2.3×10^{-7}	1.2×10^{-8}	5.8×10^{-5}
ode45, $\text{Tol}=10^{3}$	9.480×10^{3}	8.1×10^{-2}	1.5×10^{-5}	7.0×10^{-2}
ode23, $\text{Tol}=10^{3}$	5.317×10^{3}	1.2×10^{-6}	2.3×10^{-10}	1.1×10^{-6}
ode113, $\text{Tol}=10^{3}$	6.046×10^{3}	8.9×10^{-4}	1.7×10^{-7}	7.7×10^{-4}
A1, $\Delta t=1.25\times10^{-4}$	1.98×10^{-1}	8.46×10^{-2}	4.55×10^{-4}	6.72×10^{0}
A2, $\Delta t=5.0\times10^{-6}$	4.17×10^{0}	4.81×10^{-4}	3.69×10^{-6}	6.65×10^{-2}
A3, $\Delta t=2.5\times10^{-6}$	9.85×10^{0}	1.99×10^{-4}	7.65×10^{-7}	1.31×10^{-2}
A4, $\Delta t=1.25\times10^{-4}$	1.95×10^{-1}	3.28×10^{-3}	8.88×10^{-6}	2.68×10^{-3}
A5, $\Delta t=5\times10^{-7}$	4.95×10^{1}	1.55×10^{-6}	8.71×10^{-9}	1.69×10^{-4}

3.4.5. Verification by Comparison to Analytical Results

I consider very recent nontrivial analytical solutions of Eq. (3.23) found by Barna and Mátyás [4] by a similarity transformation technique. Both of them are given on the whole real number line for positive values of *t* as follows

$$u_1^{exact}(x,t)=\frac{x}{t^{3/2}}e^{-\frac{x^2}{4\alpha t}}, \tag{3.41}$$

and

$$u_2^{exact}=\frac{x}{t^{5/2}}\left(1-\frac{x^2}{6\alpha t}\right)e^{-\frac{x^2}{4\alpha t}}. \tag{3.42}$$

I reproduce these solutions only in finite space and time intervals $x\in[x_1,x_2]$ and $t\in[t_0,t_{\text{fin}}]$, where $x_1=-5, x_2=5, t_0=0.5, t_{\text{fin}}=1$. The space interval is discretized by creating nodes as follows: $x_j=x_1+j\Delta x,\ j=0,...,1000,\ \Delta x=0.01$. I prescribe the appropriate Dirichlet boundary conditions at the two ends of the interval:

$$u_1(x=x_b,t)=\frac{x_b}{t^{3/2}}e^{-\frac{x_b^2}{4\alpha t}}, \tag{3.43}$$

and

$$u_2(x=x_b,t)=\frac{x_b}{t^{5/2}}\left(1-\frac{x_b^2}{6\alpha t}\right)e^{-\frac{x_b^2}{4\alpha t}}, \tag{3.44}$$

where $x_b\in\{x_1,x_2\}$. I obtained that the new methods are convergent and the order of convergence is two. In Figure 3.8 the L_∞ errors as a function of the effective time step size h_{EFF} are presented for the case of the u_2 solution for the top 5 algorithms and a first-order "reference-curve" for the original CNe method. I note that very similar curves have been obtained for the u_1 solution, as well as for other space and time intervals.

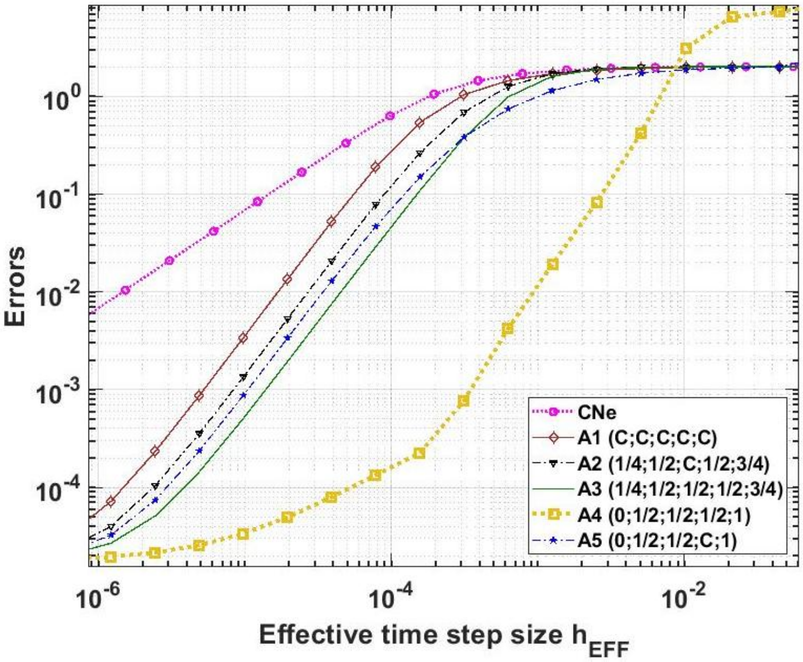

Figure 3.8. The L_∞ errors as a function of Δt_{EFF} for the u_2 solutions [65].

3.5. The Optimization of Leapfrog -Hopscotch Method Combinations

In a manner similar to the Shifted-Hopscotch method [65], the hopscotch spatial structure was integrated with leapfrog time integration [66]. This framework incorporated the theta method, evaluated at nine distinct θ values, and the recently developed CNe method, leading to the construction of 105 unique combinations. Through subsequent numerical experimentation, this large set decreased by eliminating underperforming variants; finally, only the top five algorithms of these remained. The evaluation was conducted on two-dimensional stiff systems comprising 10,000 cells with fully discontinuous random parameters and initial conditions; consequently, so the results are presented just for the best five algorithms.

The best algorithms were compared with other methods for a large, moderately stiff system with the same procedure of 3.4.3, and for a large, very stiff system with the same procedure of 3.4.4. for the same system size and final time. The following top 5 combinations are chosen based on the best performance of the maximum and energy error.

L1 (C, C, C, C, C),

L2 (0, ½, ½, ½, ½),

L3 (⅕, ½, ½, ½, ½),

L4 (¼, ½, C, ½, ½),

L5 (⅕, ½, C, ½, ½).

3.5.1.Verification by Comparison to Analytical Results Using a Non-Uniform Mesh

The nontrivial analytical solution [4] in section 3.4.5 of Eq. (3.23) is used here, given on the whole real number line for positive values of *t* as in Eq. (3.42), where the value $\alpha = 1$ is used, this solution was reproduced by prescribing the Dirichlet boundary conditions calculated using the analytical solution at the two ends of the interval. Now this kind of information is not used, but construct a large-scale non-equidistant spatial grid according to the following procedure. First the coordinates of the cell borders are define by the formula

$$x_j = x_{j-1} + \Delta x_{j-1}\,,\ x_0 = 0,\, \Delta x_0 = 0.01\,,\ \ \Delta x_j = \Delta x_0 \exp\left(\gamma\, j^4\right),\ j = 1,\dots,1000\,.$$

where $\gamma = 10^{-11}$. Thus I have a quite dense system of nodes close to the origin which becomes less and less dense as one is getting further from the origin, towards $+5922.3$, which is the right boundary of the mesh. Then the cell-centers are calculated straightforwardly:

$$X_j = X_{j-1} + \frac{\Delta x_j}{2}\,,\ \ X_0 = 0\,,\ j = 1,\dots,1000\ .$$

Now it is straightforward to reflect this structure to the origin to create the mirror image of the mesh at the negative side of the x-axis obtaining 2000 cells altogether. Now at the vicinity of the origin the diameter of a cells are 0.01, which are increasing as it is getting further from the origin, first very slowly, then more and more rapidly until it reaches $\Delta x_{\pm 1000} = 211.6$. The resistances and the cell capacities then can be calculated as:

$$C_i = \Delta x_i\ ,\quad i = 1,\dots,2000 \text{ and } R_i = X_{i+1} - X_i\ ,\quad i = 1,\dots,1999$$

zero Neumann boundary conditions are taken into account which is a good approximation because the values of the initial function are very close to zero far from the origin. The stiffness ratio is 5.7×10^{11} for this mesh, while $\Delta t_{\mathrm{MAX}}^{\mathrm{FTCS}} = 5\times 10^{-5}$. As in shifted hopscotch, the analytical solution is reproduce in finite time interval $t \in [t_0, t_{\mathrm{fin}}]$, where $t_0 = 0.5, t_{\mathrm{fin}} = 1$. In Figure 3.9 the L_∞ errors as a function of the time step size are presented for the case of the *u* solution for the top 5 leapfrog-hopscotch algorithms, a first-order “reference-curve” for the original CNe method and the Heun method. These results verify not only the second order convergence of the numerical methods, but the procedure of generalizing the calculations to non-uniform grids. One can also see that the L2 and L3 algorithms reach the minimum error (determined by the space discretization) for larger Δt than the CFL limit for the Heun method.

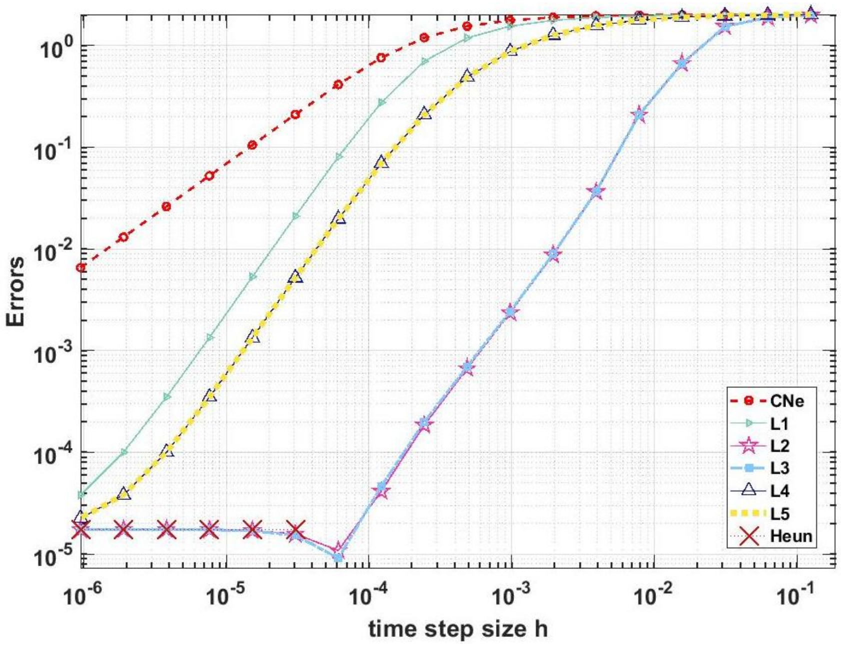

Figure 3.9. The L_∞ errors as a function of time step size for the space-dependent mesh to reproduce the exact solution given in (3.42) [66].

3.6. The Optimization of Pseudo-Implicit Method Combinations

By the iteration of the theta-formula and treating the neighbors explicitly, A novel two-stage explicit algorithm was developed for the solution of partial differential equations incorporating a diffusion term and two reaction terms [67]. One reaction term is linear, potentially modeling heat convection, while the second is proportional to the fourth power of the variable, representative of radiative effects. For the linear case, the method is analytically demonstrated to achieve second-order accuracy and unconditional stability. The diffusion-reaction equation (2.20) is going to be studied.

Algorithm 1, UPDF for the diffusion-convection-radiation equation

$$u_i^{n+1} = \frac{u_i^n + mr\left(u_{i-1}^n + u_{i+1}^n\right) + q_i\Delta t}{1 + 2mr + K_i\Delta t + \sigma\Delta t\left(u_i^n\right)^3}\,. \tag{3.45}$$

Similar to the original UPFD formula, this expression conserves the positivity property for arbitrary nonnegative values of r, q_i, K_i and σ, thus for the strongly nonlinear case as well. Its accuracy is not very good, thus a two-stage method proceeds to construct as well.

A combination of the UPFD idea and the θ-method is proposed for application to the diffusion term in the following manner :

$$u_i^{n+1} = u_i^n + mr\left[\theta\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + (1-\theta)\left(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}\right)\right], \tag{3.46}$$

Where $\theta \in [0,1]$. The FTCS scheme, equivalent to explicit Euler integration, is obtained for $\theta = 1$. The scheme with $\theta = 0, \frac{1}{2}$ corresponding to the Implicit-Euler and Crank-Nicolson methods, respectively [64]. Utilizing the previous trick and including the reaction and source terms leads to :

$$u_i^{n+1} = u_i^n + mr\left[-2\theta u_i^n - 2(1-\theta)u_i^{n+1} + u_{i-1}^n + u_{i+1}^n\right] - \Delta t K_i u_i^{n+1} + \Delta t q_i + \sigma u_i^{n+1}\left(u_i^n\right)^3\,. \tag{3.47}$$

The original UPFD formulation is recovers if $\theta = 0$. The key advantage is that this generalized formula can be easily rearranged to produce an explicit expression. For the 1D equidistant case, the new *u* value takes the form:

Algorithm 2, theta-generalization of Algorithm 2

$$u_i^{n+1} = \frac{(1-2mr\theta)u_i^n + mr\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t q_i}{1 + 2mr(1-\theta) + \Delta t K_i + \sigma\Delta t\left(u_i^n\right)^3}\,. \tag{3.48}$$

Since started from an implicit formula (3.46) formally but made it fully explicit, these methods started to be called *pseudo-implicit*. The main novelty of this study is that formula (3.48) is organized into a two-stage method as follows. The calculation starts with taking a fractional-sized time step using the already known u_i^n values, and then a full-time step is made.

Algorithm 3, 2-stage pseudo-implicit method for the diffusion-convection-radiation equation

Stage 1. Take a partial time step $\Delta t_1 = p\Delta t, \, p > 0$ using formula (3.48) with parameter θ_1:

$$u_i^{\text{pred}} = \frac{(1-2pmr\theta_1)u_i^n + pmr\left(u_{i-1}^n + u_{i+1}^n\right) + q_i\Delta t_1 - v_1 K_i \Delta t_1 u_i^n}{1 + 2pmr(1-\theta_1) + v_2 K_i \Delta t_1 + \sigma\Delta t_1 \left(u_i^n\right)^3} .$$

Stage 2. u_i^{pred} is redefine by calculating the linear combination with $0 < \lambda \le 1$:

$$u_i^{\text{pred}} = \lambda u_i^{\text{pred}} + (1-\lambda)u_i^n . \tag{3.49}$$

Take a full time step with the (3.48) formula with parameter θ_2:

$$u_i^{n+1} = \frac{(1-2mr\theta_2)u_i^n + mr\left(u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}\right) + q_i\Delta t - K_i\Delta t\left(w_1 u_i^n + w_2 u_i^{\text{pred}}\right)}{1 + 2mr(1-\theta_2) + (1-w_1-w_2)K_i\Delta t + \sigma\Delta t\left(u_i^{\text{pred}}\right)^2 u_i^n} , \tag{3.50}$$

Where v_1 , v_2 , w_1 , w_2 are real numbers that are considered as free parameters. The mathematically correct form of (3.49) would be $u_i^{\text{lin}} = \lambda u_i^{\text{pred}} + (1-\lambda)u_i^n$, however, it is directly transcribed into a form suitable for computer code to conserve memory. Furthermore, this handling of the nonlinear term yields a second-order method characterized by very favorable stability.

Algorithm 4: for the diffusion-convection-radiation equation

Stage 1. Take a partial time step $\Delta t_1 = \Delta t \big/ 2\lambda \, , \lambda > 0$:

$$u_i^{\text{pred}} = \frac{\left(1 + mr\left(1 - 1\big/\lambda\right)\right)u_i^n + mr\big/2\lambda\left(u_{i-1}^n + u_{i+1}^n\right) + q_i\Delta t_1}{1 + mr + K_i\Delta t_1 + \sigma\Delta t_1\left(u_i^n\right)^3} . \tag{3.51}$$

Stage 2. Calculate the linear combination $u_i^{\text{pred}} = \lambda u_i^{\text{pred}} + (1-\lambda)u_i^n$

Take a full-time step:

$$u_i^{n+1} = \frac{(1-mr)u_i^n + mr\left(u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}\right) + q_i\Delta t + K_i\Delta t\left(u_i^{\text{pred}} - u_i^n\right)}{1 + mr + K_i\Delta t + \sigma\Delta t\left(u_i^{\text{pred}}\right)^2 u_i^n} . \tag{3.52}$$

3.6.1. Analytical-Solution Based Verification

The following analytical solution of Eq. (2.20) is constructed for $\alpha = 1, \, K = 2$ and $q(x,t) = \sigma t^4 e^{4x-4t} + e^{x-t}$:

$$u^{\text{exact}}(x,t) = te^{x-t} . \tag{3.53}$$

Here this analytical solution numerically reproduces for $(t,x) \in [0.5, 1] \times [-1, 1]$ and $\sigma = 3$. The initial condition

$$u(x, t = 0.5) = 0.5e^{x-0.5} ,$$

and the Dirichlet boundary conditions at the ends of the interval

$$u(x = -1, \, t) = te^{-1-t} \, , \; \text{and} \;\; u(x = 1, \, t) = te^{1-t}$$

are obtained using the analytical solution. The numerical error is defined as the absolute difference, at the final time t_{fin} , between the numerical solution $u_{\text{j}}^{\text{num}}$ generated by the method under examination and the reference solution $u_{\text{j}}^{\text{ref}}$ (which is the analytical solution here). These individual nodes' or cells' errors are subsequently utilized to compute the maximum error according to Equation (3.33).

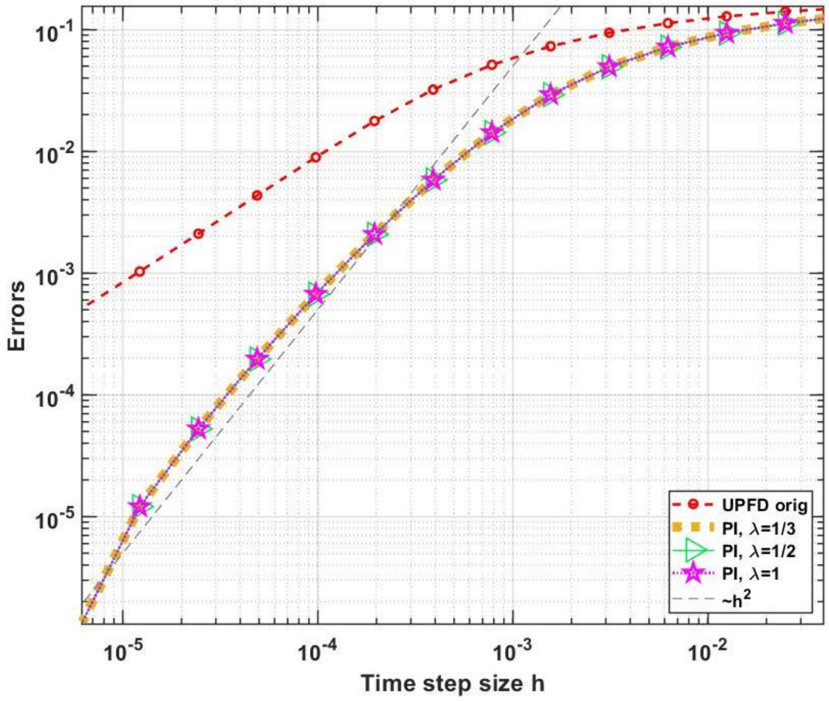

Figure 3.10. The L_∞ for the numerical solutions of the diffusion-convection-radiation equation in the case of Algorithm 1 and the new pseudo-implicit Algorithm 3 for three different values of the parameter λ [67].

4. USING EFFICIENT METHODS TO SOLVE REAL-LIFE HEAT TRANSFER PROBLEMS

The algorithms were tested in previous sections under general circumstances with discontinuous random parameters and initial conditions. I demonstrate that these methods can yield quite accurate results and are substantially faster than the professionally optimized MATLAB 'ode' routines. In this section, I perform a systematic examination of building walls by changing some system and mesh parameters. The aim is to assess how the performance of each method changes and to identify the most suitable algorithm under varying conditions.

4.1. *Calculate The Heat Conduction in an Insulated Wall*

Determining heat transfer through building elements is still a complex and critical challenge. To address this, in the present study we evaluates 13 numerical techniques (CN, UPFD, OOEH, ROEH, LNe2, LNe3, CpC, Heun, PI, DF, RRK, SH, and LH) for solving the heat conduction equation (2.20) in wall assemblies. Notably, eight of these methods are newly invented explicit algorithms with unconditional stability [68].

4.1.1. The Geometry and Mesh Generation:

Figure 4.1 illustrates a single-layer brick wall alongside two-layer configurations combining brick and glass wool insulation.

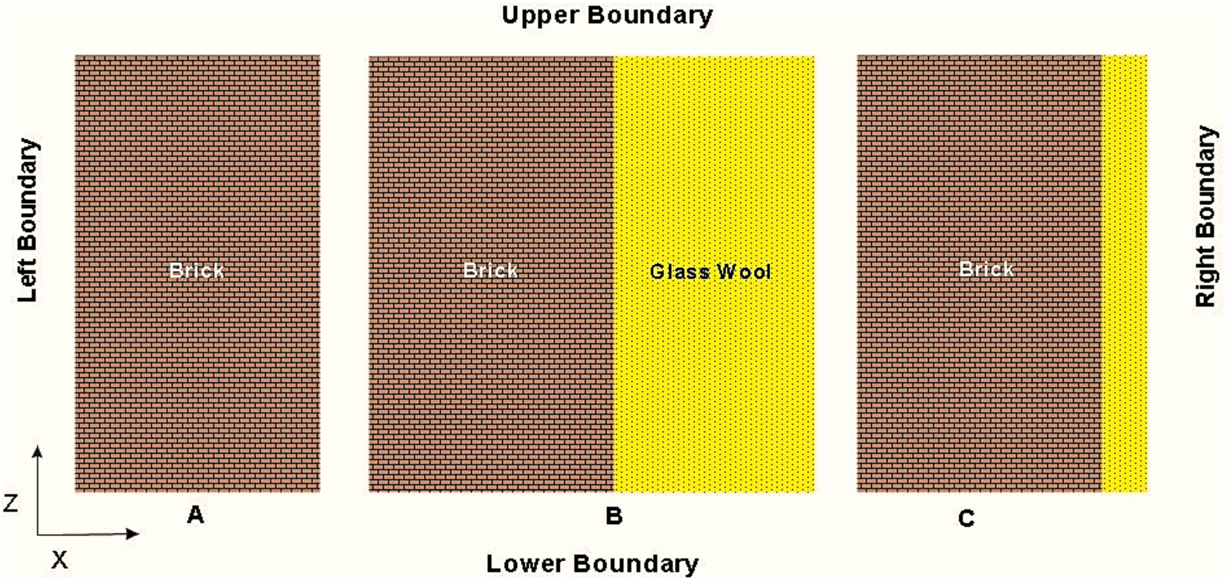

Figure 4.1. (**A**) single-layer (Brick) wall, (**B**) and (**C**) multilayer (Brick+Glass wool) wall [Original].

The study considers a piece of wall with volume (1 m × 1 m × 1 m). However, because all physical quantities remain constant in the *y*-direction (normal to the surfaces in Figures 4.1 and 4.2), this dimension can be disregarded. It means I deal only with a cross-section, which is a two-dimensional problem from the mathematical point of view and thus $\Delta y_i = 1$ can be used. So, several meshes of size 1m^2 are constructed, which means $(x,z)\in[0,1]\times[0,1]$. In the equidistant mesh, cells are square, while in non-equidistant meshes, they become rectangular. The heat capacity of the cells can be given as $C_i = c_i\rho_i\Delta x_i\Delta z_i$, while the thermal resistance in the *x*-direction has the approximate formula $Rx_i \approx \frac{\Delta x_i}{k_i Ax_i}$, where Ax_i is the surface element perpendicular to *x*. Since now it can be given as $Ax_i = \Delta y_i \Delta z_i = \Delta z_i$, the horizontal and vertical resistances can be given in case of a homogeneous material and uniform mesh as

$$Rx_i \approx \frac{\Delta x_i}{k_i \Delta z_i} \text{ and } Rz_i \approx \frac{\Delta z_i}{k_i \Delta x_i},$$

respectively. If the material properties or the sizes of the two neighboring cells are different, one can write

$$Rx_i \approx \frac{\Delta x_i}{2k_i \Delta z_i} + \frac{\Delta x_{i+1}}{2k_{i+1}\Delta z_{i+1}},$$

for the resistance between cells *i* and *i* + 1. If the cell *j* is below the cell *i*, I have

$$Rz_i \approx \frac{\Delta z_i}{2k_i \Delta x_i} + \frac{\Delta z_j}{2k_j \Delta x_j}$$

for the vertical resistance.

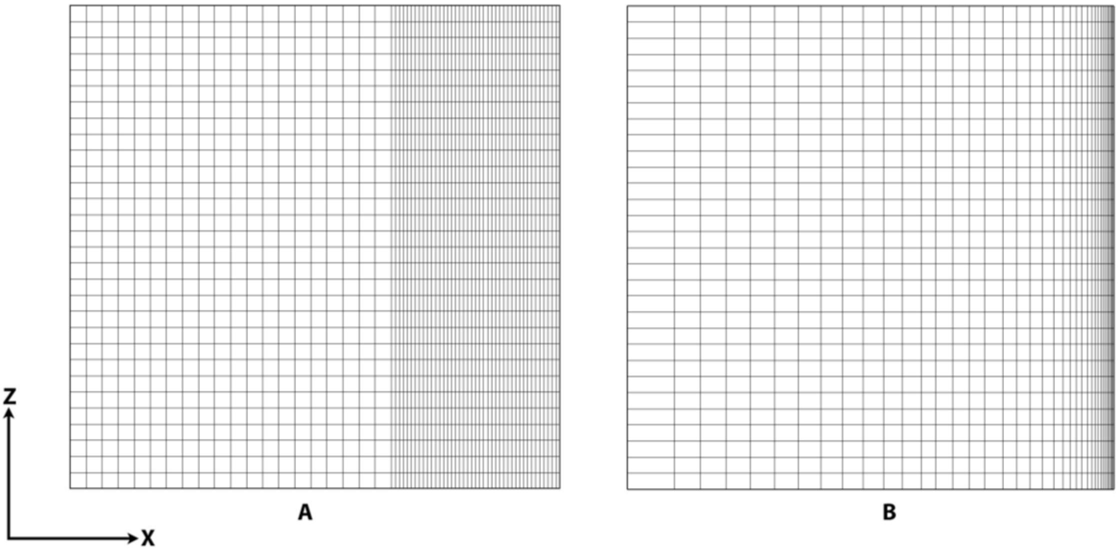

Figure 4.2. (**A**) Abrupt change in *x* direction, and (**B**) Gradual change in the *x* direction [Original].

Both equidistant and non-equidistant grids were employed to discretize spatial variables in single-layer and multilayer configurations. The axis *x* and *z* axes were divided into N_x = 100 and N_z = 100 respectively, except in Section 4.1.3.2, where $N_x = N_z = 80$. This resulted in a total cell count of $N = N_x N_z = 10000$ (with a modified $N = 6400$ in Section 4.1.3.2). It should be noted that the temperature in the middle of each cell was considered as the temperature of the cell.

For non-equidistant grids, cell sizes varied asymmetrically,

- Abrupt change: A coarse equidistant mesh $\Delta x = 0.0105$ covered the left 50% of the wall, while a fine equidistant mesh $\Delta x = 0.0097$ spanned the right 50%.
- Gradual change: Cell widths followed a geometric series, shrinking from $\Delta x_1 = 0.0234$ on the left to $\Delta x_{N_x} = 0.98^{99} \cdot \Delta x_1 = 0.00317$ on the right. For $\gamma \neq 1$ (common ratio)

and $n = N_x - 1$ (cell index) were selected to ensure smooth transitions. The series was defined by:

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

where $\gamma = 0.98$, and a = 0.0234.

In multilayer cases:

- Equidistant grids: Brick and insulator layers each occupied 50% of the volume (as in Figure 4.1B).
- Non-equidistant grids: Insulator thickness reduced when using abrupt/gradual x-direction changes (Figure 4.1C).

A uniform time step was applied throughout. The temperature in cell *i* at time $n\Delta t$ is denoted u_i^n.

4.1.2. The Materials and Boundary Conditions:

In the present work, real material properties are listed in Table 4.1.

This study employs real material properties (see Table 4.1) under distinct initial and boundary conditions for both single-layer and multilayer configurations:

Table 4.1. The properties of the used construction materials [68].

	$\rho\left(\mathrm{kg}\cdot\mathrm{m}^{-3}\right)$	$k\ \left(\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}\right)$	$c\ \left(\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}^{-1}\right)$
Brick	1600	0.73	800
Glass wool	200	0.03	800

Different initial and boundary conditions are applied for both the single-layer and the multilayer cases as follows [68]:

I. Sinusoidal initial condition with zero Dirichlet boundary conditions.

Initial condition: Defined by the product of sine functions:

$$u(x, z, t = 0) = \sin(\pi x)\sin(\pi z)\ . \qquad (4.1)$$

Boundary conditions: Zero Dirichlet (fixed temperature) on all edges:

$$u(x = 0, z, t) = u(x = 1, z, t) = u(x, z = 0, t) = u(x, z = 1, t) = 0\ . \qquad (4.2)$$

Analytical solution: Valid only for homogeneous (single-layer) walls:

$$u(x, z, t) = \sin(\pi x)\sin(\pi y)\,\mathrm{e}^{-2\pi^2 t}\ , \qquad (4.3)$$

II. Linearly changing initial condition with combined boundary conditions.

Initial condition: Linear variation along z :

$$u\left(x, z, t = 0\right) = 30 - 15z\ .$$

Boundary conditions:

- Top/bottom: Thermally insulated (Neumann condition):

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

- Left edge: Space-dependent temperature:

$$u\left(x = 0,\, z, t\right) = 30 - 15z$$

- Right edge: Time-dependent temperature:

$$.\ u\left(x = 1,\, z, t\right) = \mathrm{u}\left(x = 1,\, z = 0, t = 0\right) \cdot \mathrm{e}^{\lambda \mathrm{t}}$$

where λ = 0.00004. The final time was 10,000, ensuring the right boundary increased from $30\,°\mathrm{C}$ to $44.75\,°\mathrm{C}$.

In case II, complex boundaries were intentionally designed to rigorously test method performance under non-idealized scenarios.

4.1.3. The Simulation Results

Heun's method was employed as the reference solution for computing maximum and energy errors, utilizing an exceptionally small time step $\Delta t = 0.002$. This method was selected due to its extensive validation in existing literature compared to the other algorithms under investigation.

4.1.3.1. Analytical-Solution Based Verification

A single-layer brick wall configuration (Figure 4.1.A) was simulated under six distinct mesh conditions [68]:

(a) Uniform equidistant mesh
(b) Abrupt change in the *x*-direction with equidistant spacing in the *z*-direction
(c) Abrupt changes in both *x*- and *z*-directions
(d) Gradual change in the *x*-direction with equidistant spacing in the *z*-direction
(e) Gradual changes in both directions
(f) Abrupt change in *x*-direction with gradual change in *z*-direction

The simulations employed the sinusoidal initial condition Eq. (4.1) and zero Dirichlet boundary conditions Eq. (4.2), with validation against the analytical solution Eq. (4.3) with $t_{\mathrm{fin}} = 10000\left(\mathrm{s}\right)$. All cases demonstrated consistent results with spatial discretization errors below 10^{-4}, confirming the successful implementation of algorithms for both equidistant and non-equidistant mesh. Figure 4.3 presents time-step-dependent errors in log-log plots for case (f) as a sample. The results demonstrate that UPFD and CNe methods exhibit first-order accuracy in time step size, while all other methods achieve second-order convergence, as theoretically expected. Notably, the hopscotch algorithms (particularly the original OOEH) deliver superior accuracy compared to alternative approaches. Heun’s method performs accurately below the CFL limit but fails to produce valid results beyond this limit. Figure 4.4 illustrates the relationship between error and computational runtime for the same case as a sample, with averaged runtimes (over five runs) mitigating measurement fluctuations. As expected, runtime variations at fixed time steps primarily come from differences in method stages, e.g. the three-stage LNe3 method shows a slight rightward shift in its runtime curve relative to other methods.

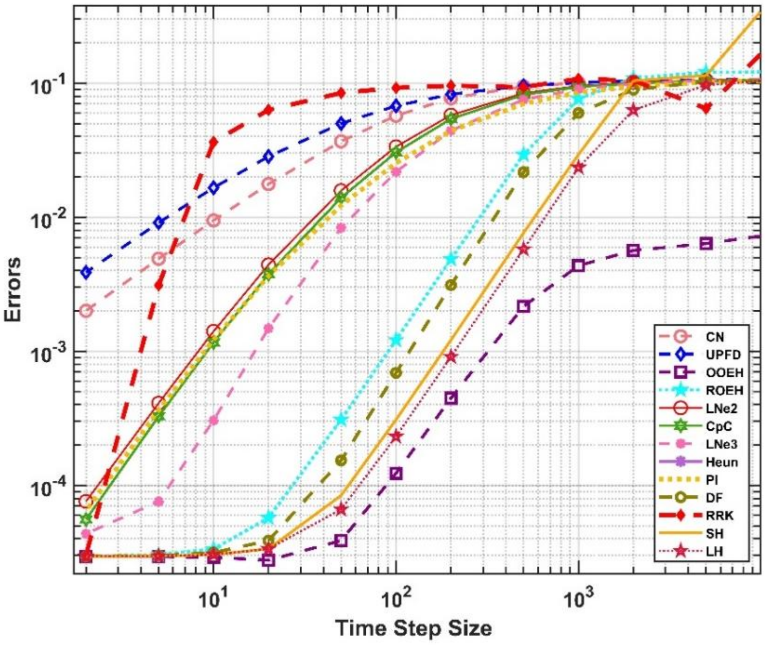

Figure 4.3. The maximum errors plotted with time step size in case of abrupt change in the *x*-direction and gradual change in the *z*-direction mesh [68].

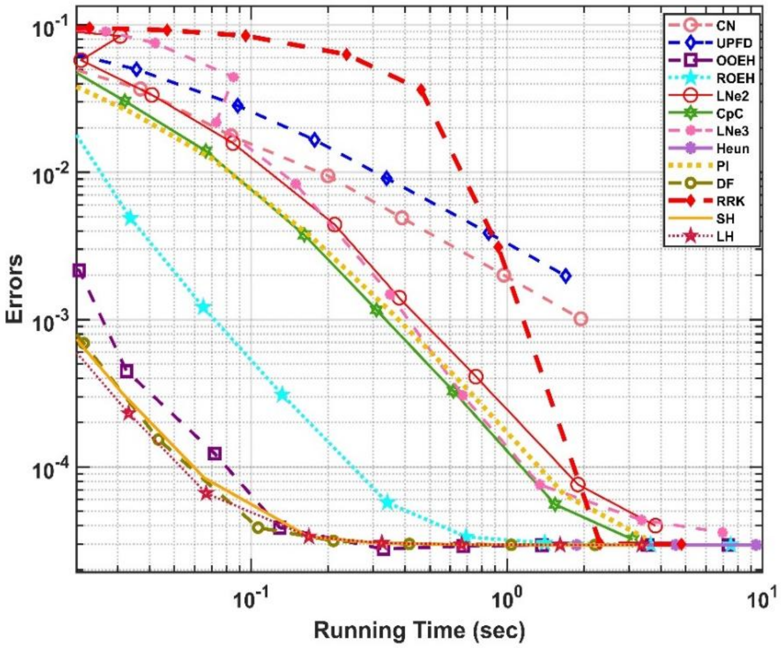

Figure 4.4. The maximum errors plotted with running time in case of abrupt change in *x*-direction and gradual change in the *z*-direction mesh [68].

4.1.3.2. Realistic Case with Nontrivial Boundary Conditions

In this subsection, the initial condition is a linear function of space, while the boundary conditions are complicated as it is written in point II. The Neumann boundary conditions for upper and lower boundaries are implemented by setting the appropriate resistances to infinity, implying that the matrix elements describing heat transfer through the boundary vanish. First, I perform the simulation for the one-layer wall for two different grids (equidistant and gradual change in both directions), and only then for the insulated wall.

In Figures 4.5 and 4.6, I present the maximum errors and energy error for a single-layer wall. The maximum and the energy error curves behave very similarly for both equidistance and non-equidistant mesh; the most significant change is that now the SH method performs better in terms of energy than the DF and the OOEH methods.

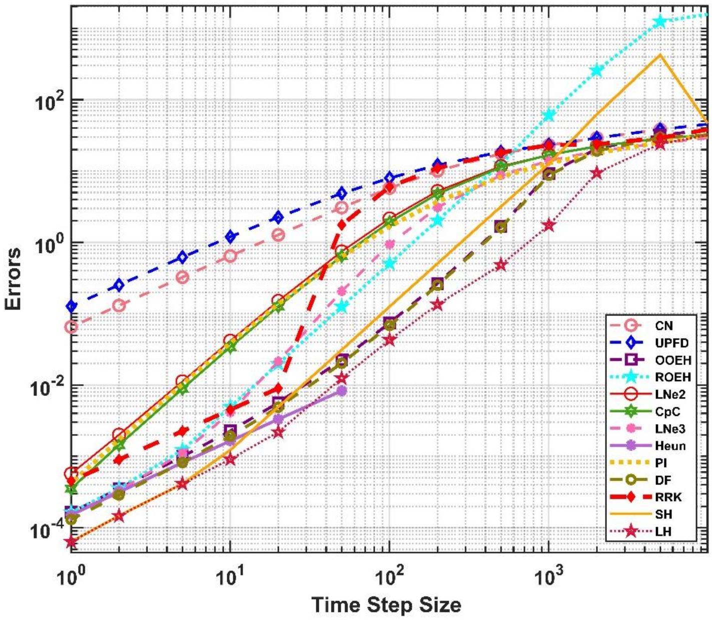

Figure 4.5. The maximum errors as a function of the time step size for a single-layer wall [68].

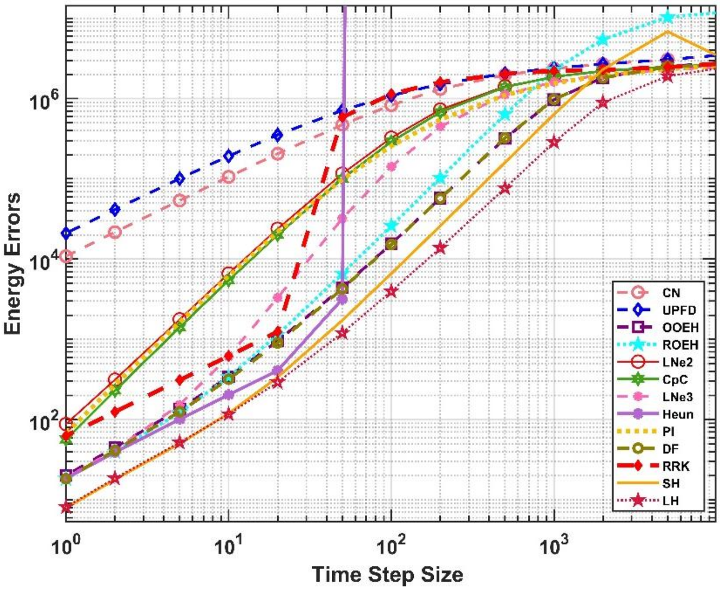

Figure 4.6. The energy errors as a function of the time step size for a single-layer wall [68].

In Figure 4.7, I present the maximum errors for a multi-layer wall with an equidistant mesh. For the non-equidistant mesh, the maximum errors and the energy errors are presented in Figures 4.8 and 4.9, respectively. From the figures, it is evident that the LH method can easily cope with this complicated heat-conduction problem as well.

Figure 4.10 presents the final temperature contours in the case of simple wall and insulated wall, while the right-side temperature profile at medium height can be seen in Figure 4.11. One can also observe that the heat from the outer side of the insulator penetrates more slowly into the wall in the case of the insulated wall.

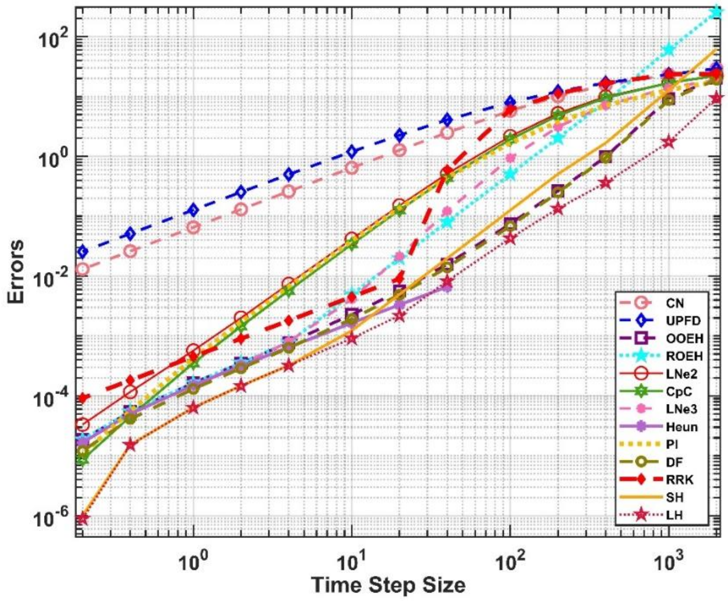

Figure 4.7. The maximum errors as a function of the time step size for the equidistant mesh for a wall with insulation [68].

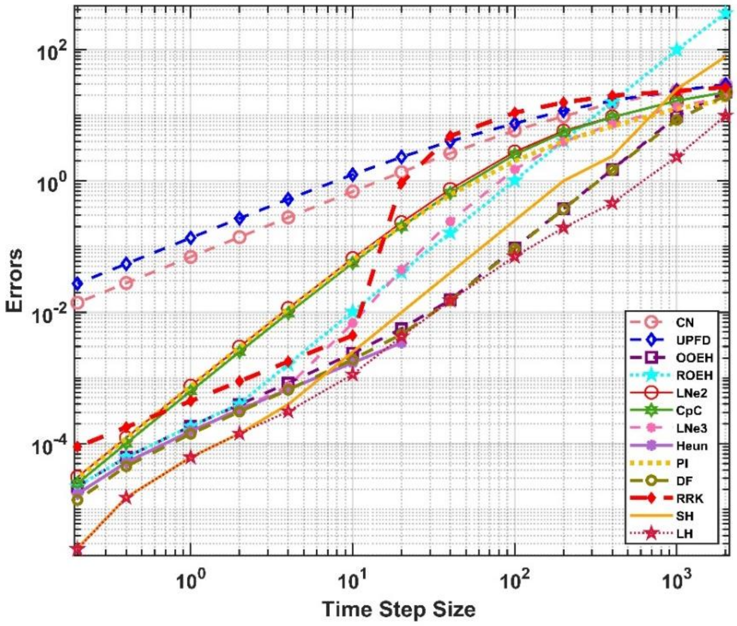

Figure 4.8. The maximum errors as a function of the time step size for the non-equidistant mesh for a wall with insulation [68].

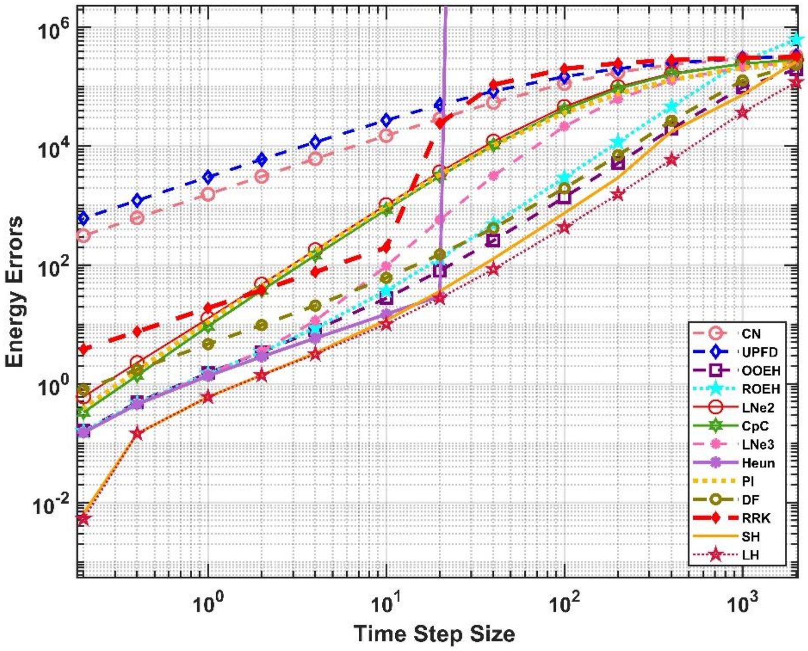

Figure 4.9. The energy errors as a function of the time step size for the non-equidistant mesh for a wall with insulation [68].

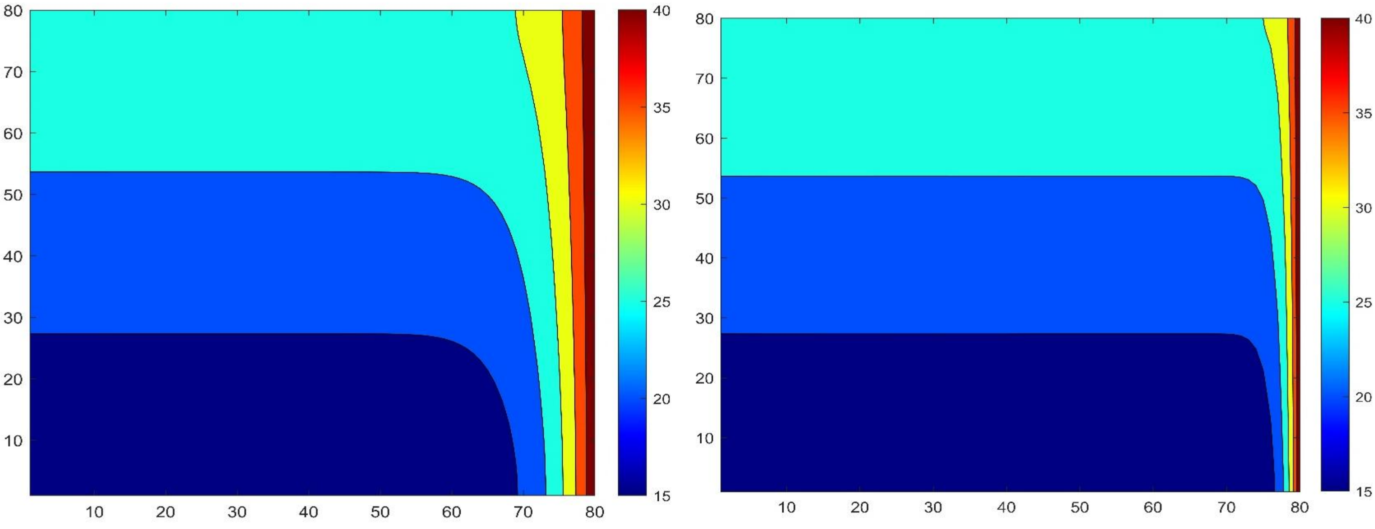

Figure 4.10. The contour of temperature distribution for the equidistant mesh at the final time in case of: a wall (left), and a wall with insulation (Right) [68].

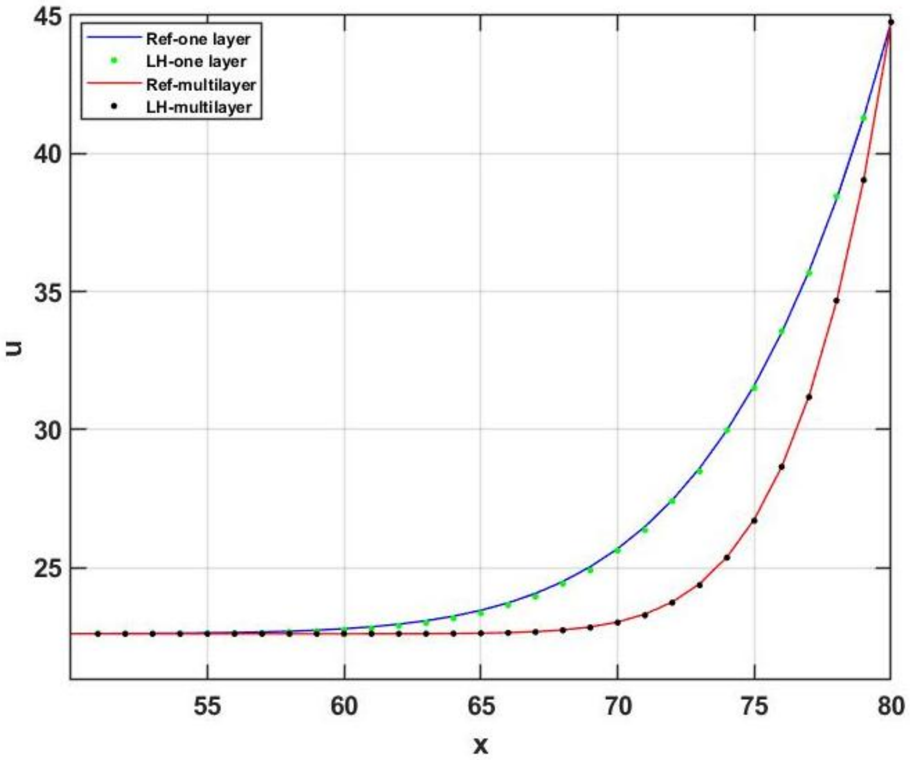

Figure 4.11. The temperature u °C as a function of the cell index in the x direction at the middle row $\left(z \approx 0.5\right)$ for $\Delta t = 400$, in the case of the brick and Brick+Insulator wall using an equidistant grid [68].

4.1.4. The Summary of The Present Section

The numerical investigation of transient heat conduction in 2D walls (both insulated and non-insulated configurations) employed eight novel and four traditional explicit stable algorithms, plus Heun's method for reference. Verification using analytical solutions across six grid types (one equidistant, five non-equidistant) confirmed all methods' convergence, though performance varied significantly:

1. CNe and UPFD showed first-order accuracy (less precise), while others achieved second-order (except RRK at medium/large time steps).
2. OOEH excelled in uniform cases but struggled with stiffness, whereas LH maintained high accuracy.
3. Heun's method proved conditionally stable (diverging beyond CFL limits, as expected), while others remained unconditionally stable.
4. CNe, UPFD, LNe2, LNe3, and CpC preserved positivity but have less accuracy at smaller time steps.
5. Hopscotch methods (OOEH, ROEH, SH, LH) required the partition of the mesh to two sub-meshes, but minimized memory by avoiding additional storage arrays.
6. Computational efficiency varied from one calculation per step (fastest: CNe, UPFD, OOEH, ROEH, DF, SH, LH) to three (slowest: LNe3).

For homogeneous materials with equidistant grids, OOEH or LH are recommended, but for general cases, LH,SH, and DF provide optimal accuracy with large time steps, while LNe3 remains essential for positivity-critical simulations.

4.2. *Calculate The Heat Transfer in an Insulated Wall with Thermal Bridging*

In the current work, I examined 14 numerical methods (ExpE, NS-ExpE, Heun, UPFD, DF, NS-DF, RRK, PI, OOEH, NS-OEH, ROEH, LH, SH, and ASH) to solve the heat equation (2.20) inside building walls. I considered heat conduction, convection, and radiation, in addition to heat generation. Five of the used methods are recently invented algorithms that are unconditionally stable for conducting problems [60].

4.2.1. The Geometry and Mesh Generation:

As one can see in Figure 4.12, I consider the following cases:

A) The surface of the wall is made of brick only.

B) Two-layer cross-section of a wall consisting of brick and insulator.

C) The same two-layer cross-section with a steel structure thermal bridge.

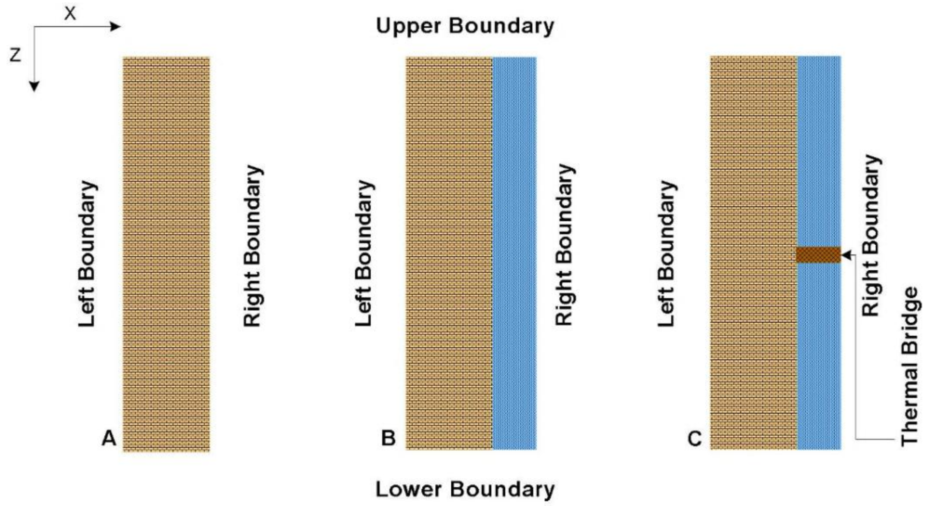

Figure 4.12. (**A**) One-layer wall, (**B**) wall with insulator, and (**C**) wall with insulator and thermal bridge [60].

In a similar way to generate the mesh in section 4.1.1, I generate the mesh of the current geometries. I apply an equidistant grid in the case of the surface of the wall, while 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 insulator for programming simplicity. It implies that the volume of the brick and the insulator is the same in the equidistant case. However, if I have a gradual change in the *x*-direction, the thickness of the insulator is smaller (0.269m). The thermal bridge has the same thickness as the insulator in the *x* direction, thus the horizontal position of the bridge is from *x*=0.5m to *x*=1m for equidistant and from *x*=0.731m to *x*=1m for the non-equidistant mesh. The height of the bridge is one cell (1cm) in the *z* direction, i.e., 0.01m, while it is positioned in row number 50 from *z*=0.49m to *z*=0.50m.

4.2.2. *The Materials and Boundary Conditions:*

In the present study, real material properties are taken into account. For the conduction term, they are listed in Table 4.2.

Table 4.2. The properties of the materials used [69].

	$\rho\left(\mathrm{kg\cdot m^{-3}}\right)$	$k\ \left(\mathrm{W\cdot m^{-1}\cdot K^{-1}}\right)$	$c\ \left(\mathrm{J\cdot kg^{-1}\cdot K^{-1}}\right)$
Brick	1600	0.73	800
Glass wool	200	0.03	800
Steel structure	7800	16.2	840

for all boundaries in all cases, I use zero Neumann boundary conditions, which forbid 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 Fig. 4.12.

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

$$u\left(x,z,t=0\right)=303-293z\ .$$

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

For the heat convection, I have used values from the literature [69] for the convection heat transfer coefficient *h_c*, as shown in Table 4.3. The universal Stefan-Boltzmann constant $5.67\cdot 10^{-8}\,\frac{\mathrm{W}}{\mathrm{m^2\cdot K^4}}$ is multiplied by the appropriate emissivity constant since the surface is not a black body. With this, I obtain realistic values for σ^*. The heat generation contains a fraction of the solar radiation, with which I obtain the value of q^* as shown below. The ambient temperature of the air is taken to be $30°\mathrm{C}\approx 303\mathrm{K}$.

Table 4.3. The heat source, convection, and radiation applied on the wall in case of surface area [69].

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

The term *q* contains also the convective heat gain due to the nonzero temperature u_a of the air (in Kelvin), with which I obtain the value of *q* as follows. The convective and radiative energy transfer is perpendicular to the surface, 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 equations (2.20) and (2.28) I obtain:

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

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

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

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

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, the heat can transfer by radiation and convection to the *x* direction with the values shown in Table 4.4.

Table 4.4. The heat source, convection, and radiation applied on both sides of wall elements in case of a cross-sectional area [60].

	$h_c\left(\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}}\right)$	$\sigma^*\left(\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}^4}\times10^{-8}\right)$	$q^*\ (\mathrm{W})$
Right Elements	2	5	500
Left Elements	4	4	500

I suppose that the right elements and left elements have the following heat source convection and radiation as follows [60]:

- For the left elements (interior side): $q=\frac{1}{c\rho}\times 500\frac{W}{m^2}+\frac{h_c}{c\rho\cdot\Delta x}\times 293\mathrm{K}$
- For the right elements (external side): $q=\frac{1}{c\rho}\times 500\frac{W}{m^2}+\frac{h_c}{c\rho\cdot\Delta x}\times 303\mathrm{K}$

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

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

4.2.3. The Simulation Results

4.2.3.1. In the Case of Surface Area of The Wall

I simulated a single-layer brick wall (see Figure 4.12.A). As I mentioned in point I. above, I applied linear initial and zero Neumann boundary conditions. I have performed the simulations with the equidistant mesh. In Fig. 4.13, 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 Fig. 4.14, I presented 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.

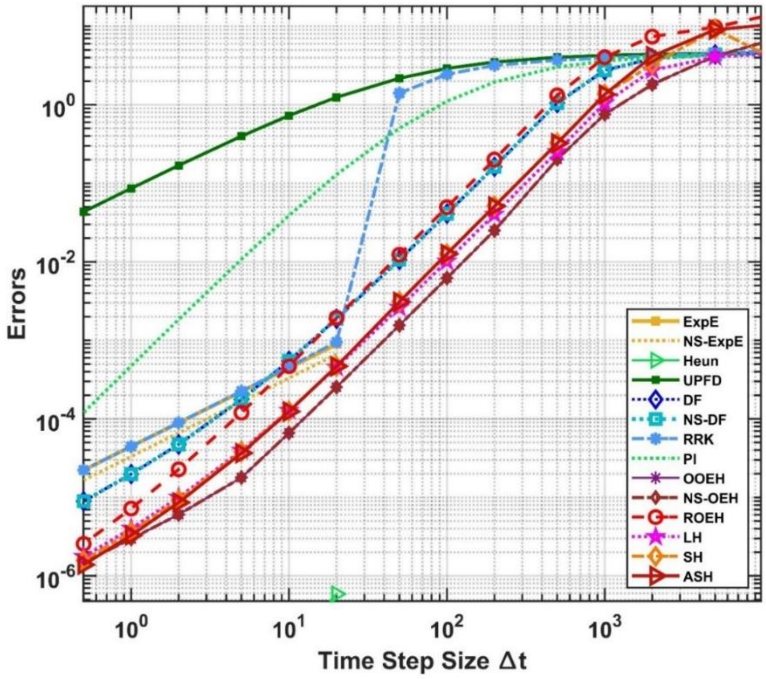

Figure 4.13. The maximum errors as a function of the time step size in the case of a surface area [60].

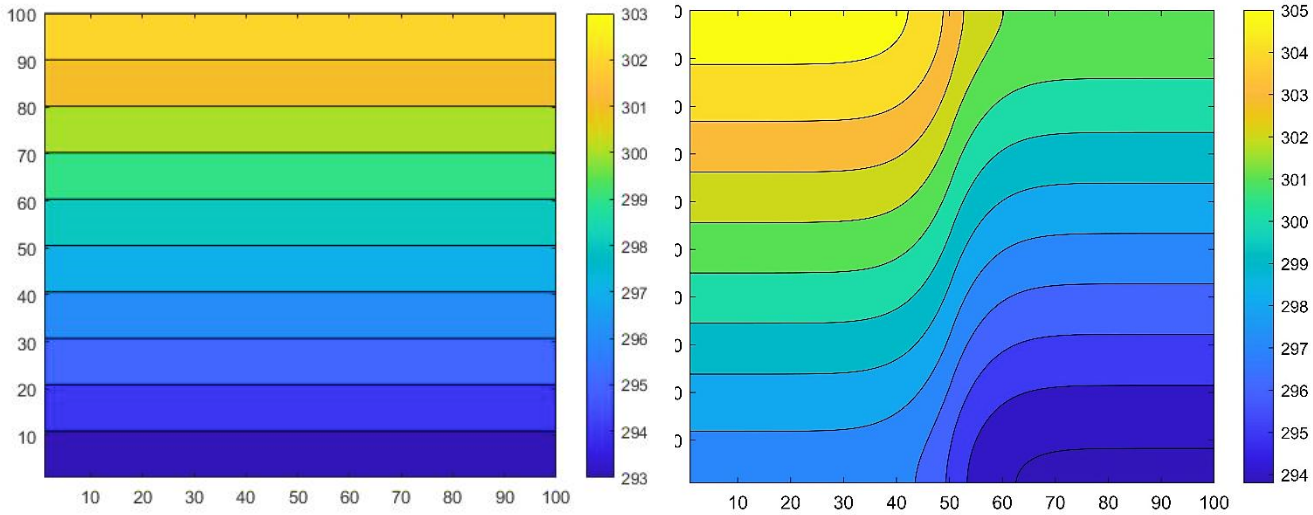

Figure 4.14. The contour of temperature distribution in Kelvin for the equidistant mesh at initial (left) and final time (right), in the case of a multilayer cross-sectional area. The numbers on the vertical and horizontal axes of the contours are the indices of the cells, which are the same as the coordinates in cm units [60].

4.2.3.2. The Results of the Cross-Section of a Brick Insulated Wall

I applied the linear initial and Neumann boundary condition of point II for the multilayer wall. The maximum errors are plotted for equidistant and non-equidistant meshes in Fig. 4.15 and 4.16, while the energy errors for the non-equidistant mesh can be seen in Fig. 4.17. The temperature distribution contours for the initial and final time moments are shown in Figure 4.18. 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.

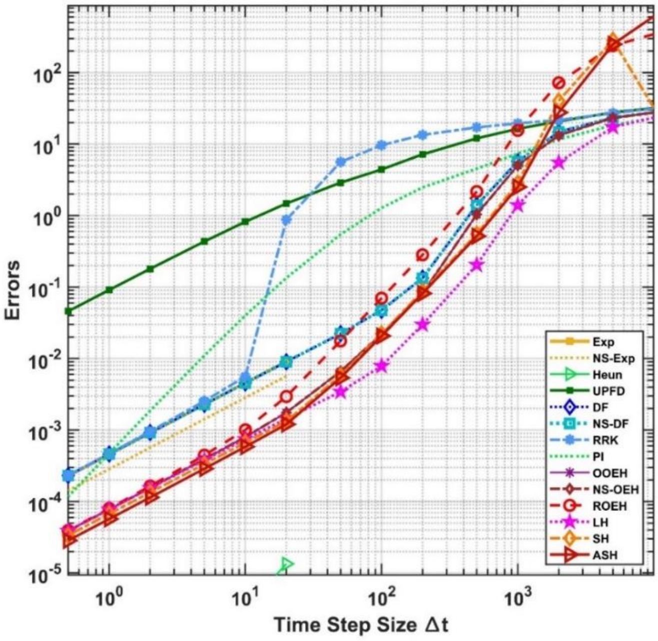

Figure 4.15. The maximum errors as a function of the time step size for the equidistant mesh [60].

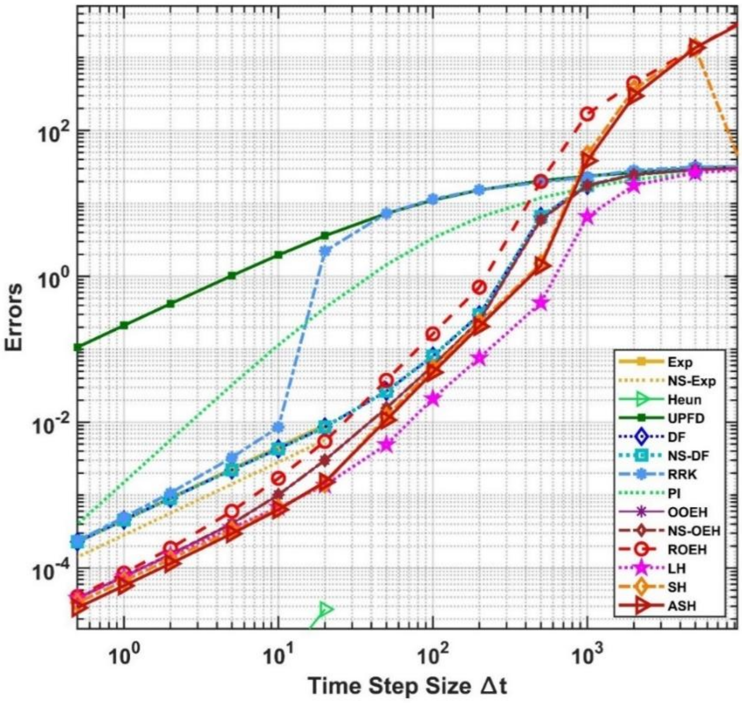

Figure 4.16. The maximum errors as a function of the time step size for the non-equidistant mesh [60].

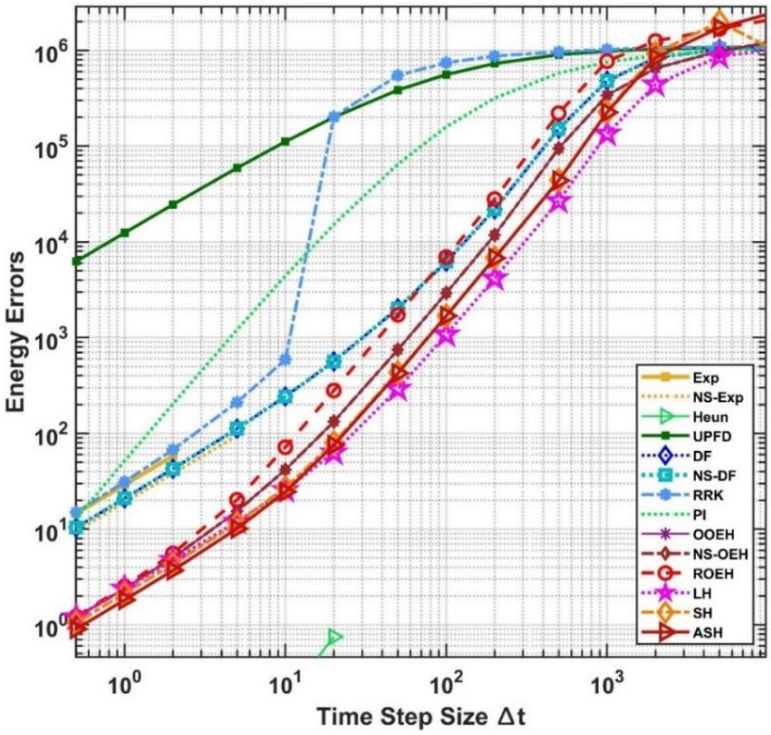

Figure 4.17. The energy errors as a function of the time step size for the non-equidistant mesh[60].

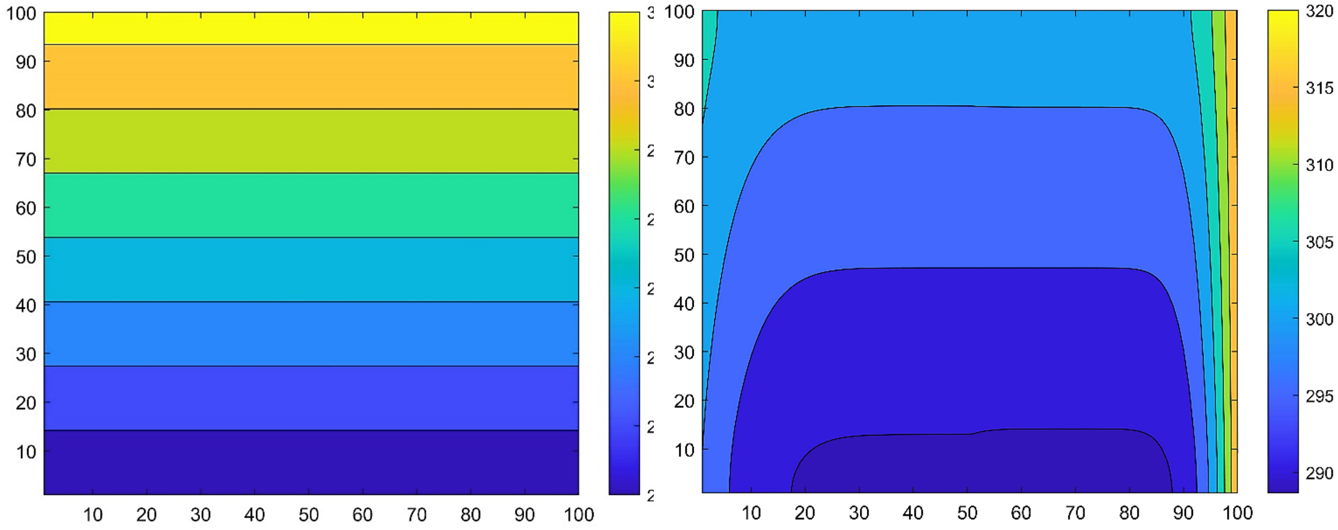

Figure 4.18. The contour of temperature distribution in Kelvin for the equidistant mesh at initial (left) and final time (right), in the case of the multilayer cross-sectional area. The numbers on the horizontal and vertical axes of the contours are the indices of the cells [60].

4.2.3.3. The Results of the Cross-Section of a Brick Insulated Wall and Thermal Bridging

I apply again the conditions listed in point II for the multilayer wall with thermal bridging. The maximum errors for equidistant and non-equidistant meshes are plotted in Fig. 4.19 and 4.20, respectively, while the energy errors for the non-equidistant mesh can be seen in Fig. 4.21. The maximum and the energy error curves are very similar; 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.

In Fig. 4.22, the temperature contour is presented for the initial and the final time moments, for the equidistant mesh. To highlight the thermal bridge's impact, I constructed Figure 4.23, which shows the final temperature at z=0.495 as a function of x, comparing results with and without the thermal bridge. This allows readers to clearly see its effect on temperature distribution.

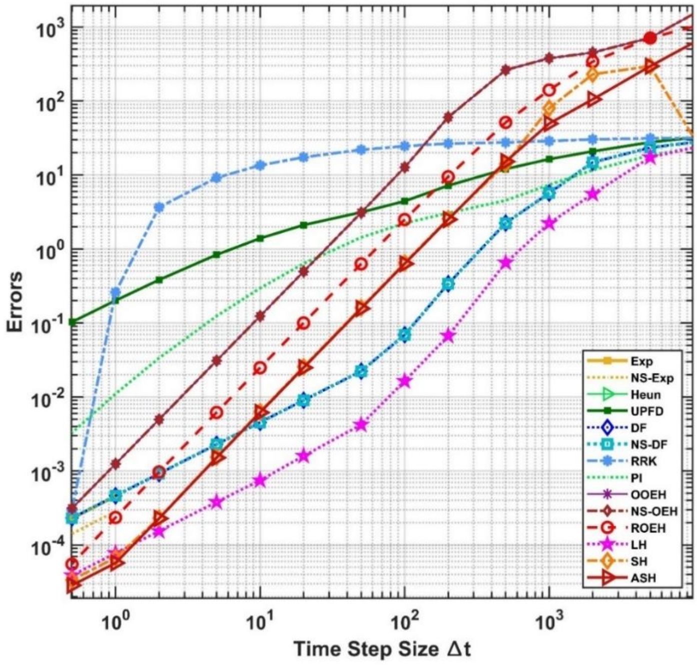

Figure 4.19. The maximum errors as a function of the time step size for the equidistant mesh and thermal bridging [60].

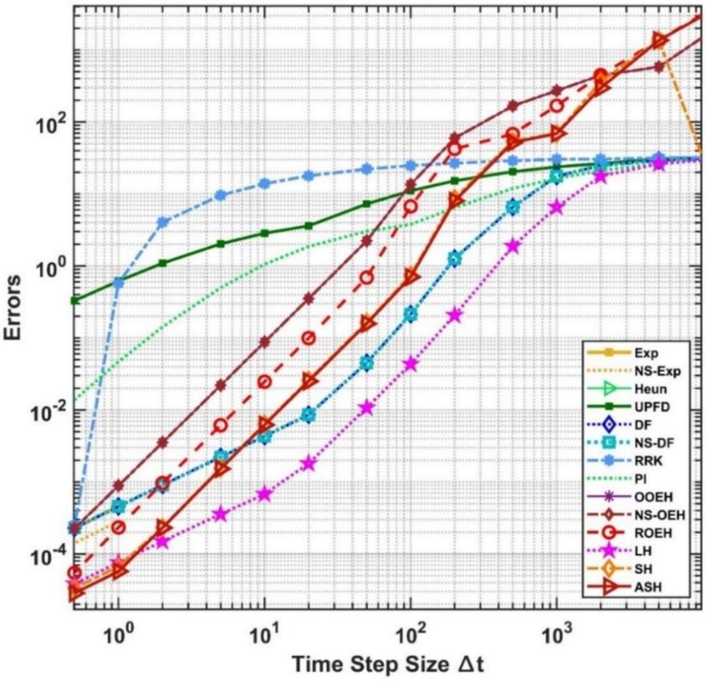

Figure 4.20. The maximum errors as a function of the time step size for the non-equidistant mesh and thermal bridging [60].

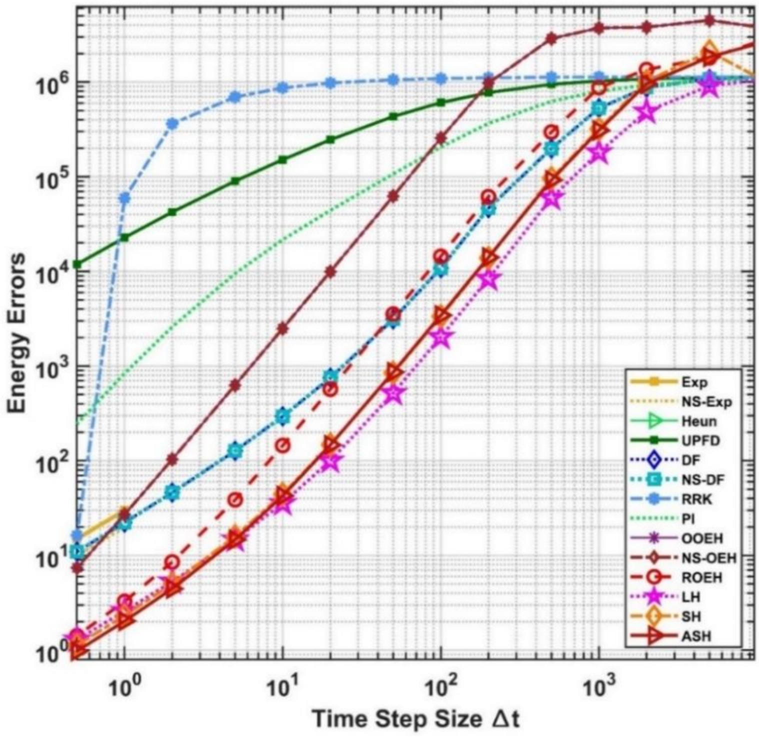

Figure 4.21. The The energy errors as a function of the time step size for the non-equidistant meshand thermal bridging [60].

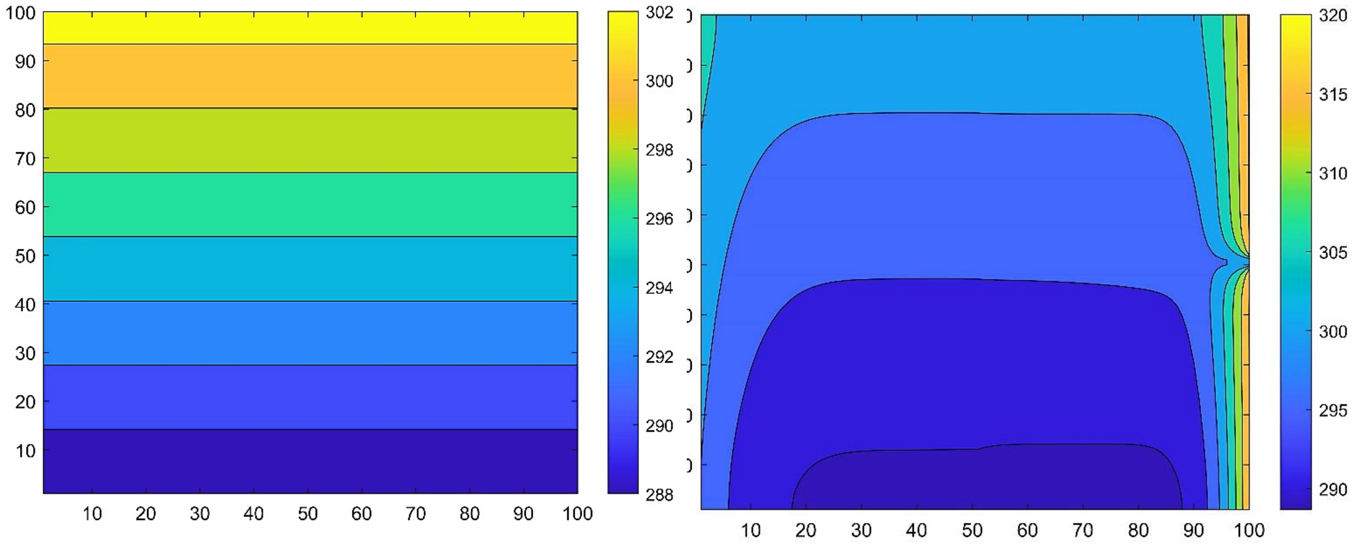

Figure 4.22. The contour of temperature distribution for the equidistant mesh at initial (left) and final time (right) in case of multilayer cross-sectional area with thermal bridging [60].

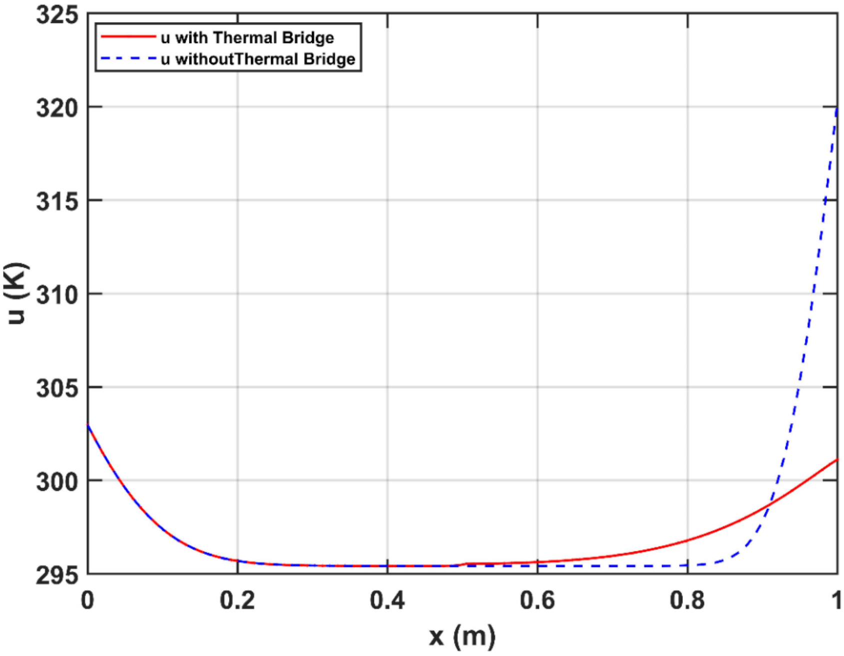

Figure 4.23. The temperature as a function of the space variable x at the middle row $\left(z \approx 0.5\right)$ in the case of the multilayer insulated wall with and without thermal bridging using an equidistant grid [60].

4.2.4. The Summary of The Present Section

I adopted 14 fully explicit numerical algorithms to solve transient heat transfer problems including heat conduction, convection, and radiation. I 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. I used equidistant and non-equidistant grids for the cross-section area. Zero Neumann boundary conditions were applied and the ode15s MATLAB routine was used as a reference solution. I showed that all of the methods can be used for these simulations, but those that 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 positivity of the temperatures for arbitrary time step size even for this highly nonlinear case. I 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.

4.3. *Calculate The Heat Transfer in Cylindrical and Spherical Shaped Bodies*

In this part, I reproduced new analytical solutions with high accuracy using recent explicit and unconditionally stable finite difference methods. After this, real experimental data from the literature regarding a heated cylinder are reproduced using the explicit numerical methods as well as using Finite Element Methods (FEM) ANSYS workbench. Convection and nonlinear radiation are also considered on the boundary of the cylinder [42].

The heat-transport equation in a 3D cylindrical coordinate system, which can be written as:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(k\,r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial}{\partial \phi}\left(k\,r\frac{\partial u}{\partial \phi}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)+\frac{Q_{gen}}{\Delta V}-\frac{hSu}{\Delta V}-\frac{\sigma^* S u^4}{\Delta V}=\rho c\frac{\partial u}{\partial t} \qquad (4.4)$$

In the case of spherical coordinates, a small 3D spherical element can be seen in Figure 2.3 The heat transport equation for this case can be expressed as follows:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(k\,r^2\frac{\partial u}{\partial r}\right)+\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial \phi}\left(k\,r\frac{\partial u}{\partial \phi}\right)+\frac{1}{r^2\sin\theta}\frac{\partial}{\partial \theta}\left(k\sin\theta\frac{\partial u}{\partial \theta}\right)+\frac{Q_{gen}}{\Delta V}-\frac{hSu}{\Delta V}-\frac{\sigma^* S u^4}{\Delta V}=\rho c\frac{\partial u}{\partial t}\,. \qquad (4.5)$$

If one does not consider the convection, radiation, and source terms in Equation (4.4) and assumes that the material properties are homogeneous, one obtains the form of the heat conduction equation in cylindrical and spherical coordinate systems. Symmetrical systems only are investigated, which means no relevant physical quantities depend on coordinate ϕ in the cylindrical and on coordinates ϕ and θ in the spherical case, which can be considered as a limitation of this study. If I temporarily also assume that nothing depends on the *z* coordinate in the cylindrical case, only the radius *r* remains as a spatial variable, which yields [42]:

$$\frac{\partial u}{\partial t}=\alpha\frac{1}{r^n}\frac{\partial}{\partial r}\left(r^n\frac{\partial u}{\partial r}\right), \qquad (4.6)$$

where n = 0, 1 and 2, which means Cartesian, cylindrical, and spherical coordinates, respectively, while $\alpha=\frac{k}{c\rho}$ is the (thermal) diffusivity. Equation (4.6) is also used for particle diffusion, where the diffusivity is usually denoted by *D*.

4.3.1. The Geometry, Materials, Mesh Generation, and Boundary Conditions

I am going to reproduce the experimental results of Cabezas et al. [70], where heat transfer was studied in a steel C45 cylinder of 168 mm total height with properties shown in Table 4.5 below.

Table 4.5. The properties of the steel used [70].

Material	$\rho\left(\mathrm{kg\cdot m^{-3}}\right)$	$k\ \left(\mathrm{W\cdot m^{-1}\cdot K^{-1}}\right)$	$c\ \left(\mathrm{J\cdot kg^{-1}\cdot K^{-1}}\right)$
Steel C45	7800	40	480

The bottom of the cylinder was heated for 30 s at the beginning of the experiment with *P* = 1500W power. However, in the original work [70], the position of the lowest thermocouple was 50mm higher than the heated surface. The top 118 mm and not the bottom 50 mm of the cylinder was examined either experimentally or numerically, and I followed this in my work. This means that the simulated volume of the cylinder segment is $V = 1.0087 \times 10^{-4}\ \mathrm{m}^3$, while $(r, z) \in [0, 0.0165\,\mathrm{m}] \times [0, 0.118\,\mathrm{m}]$. In my approximation, physical quantities did not change in the ϕ-direction, thus, that 3D was irrelevant and computationally, I dealt with a two-dimensional problem. The number of cells along the *r* axis and *z* axis were set to N_r = 15 and N_z = 100; thus, the total number of cells in the system was $N = N_r N_z = 1500$.

I used a constant initial condition in all cases.

$$u\left(r, z, t = 0\right) = 30.7\ ^\circ\mathrm{C}$$

I used different boundary conditions on different sides. On the left side, the center of the cylinder, I applied Neumann boundary conditions in all cases, which do not allow conductive heat transfer at the boundary

$$u_r(r = 0, z, t) = u_r(r = L_r, z, t) = u_z(r, z = L_z, t) = 0\,.$$

On the right (external) and upper boundaries, I used two types of boundary conditions. The first one was zero-Neumann, when there was no heat exchange with the environment. The second one, when there was a heat exchange with the environment via convection and radiation, considered the heat convection coefficient $h_c = 4.5\ \left(\mathrm{W} \cdot \mathrm{m}^{-2} \cdot \mathrm{K}^{-1}\right)$[38] and the emissivity constant as 0.85 to obtain realistic values for σ^*. The convective and radiative energy transfer was perpendicular to the surface. The interior elements cannot gain or lose heat by the heat source, heat convection, or radiation. On the lower boundary, I applied changing Dirichlet boundary conditions based on the temperature measurement results taken from a report I asked the authors of [70]. That report contained data from every two minutes, and I used linear interpolation between these data points in all cases to follow the experimental setup of the paper [70].

The heat generation contained incoming heat via convection and radiation, depending on the ambient temperature. Since the steel cylinder was placed in a closed box, the ambient temperature changed during the measurement. Instead of the ambient temperature functions, I used their averages taken from the report mentioned above. The ambient temperature of the air was taken as (30.7, 31.1, and 31.7 °C) in the cases of measurements at 20 min, 24 min, and 30 min duration, respectively.

4.3.2. Analytical-Solution Based Verification

In this section, I take the height of the cylinder as well as Δz unity. It means that, computationally, there is one space dimension only in both the cylindrical and the spherical cases. The solution parameters are:

$$N_r = 500, N_z = 1, N = N_r \times N_z = 500,\ r_0 = 0.0003,\ r_{max} = 0.999, \Delta r = 0.002, \alpha = 1,$$
$$a \in \left\{1, 1.2,\ 2\right\}, t^0 = 0.1,\ t^{\mathrm{fin}} = t^0 + 0.1.$$

Here, *N* represents the total number of cells, *a* self-similar exponent, while r_0 and r_{max} are the radial coordinates of the center of the first and last cells. The CFL limit (maximum allowed time-step size for the standard first-order forward Euler method) was around $2 \cdot 10^{-6}$ in all cases. The initial condition was obtained by substituting the initial *t* and boundary *r* values into the analytical solution, respectively. The Dirichlet boundary conditions on the right side (the circumference of the cylinder and sphere) were obtained simply by substituting the radius r_{max} into the analytical solution and calculating the function value at each time step. On the left side (the cylinder and sphere center, $r = r_0$), a zero-Neumann boundary was applied, since no heat can disappear from the center of the cylinder or the sphere. This boundary was applied only computationally and not physically. I remind the reader that the analytical solutions are constructed for Equation (4.6).

The obtained maximum errors are displayed as a function of the time-step size in Figures 4.24 and 4.25 for two values of parameter *a* in cylindrical coordinates, it is clear with *a=1,* the results are more accurate than *a=2*. Figure 4.26 presents the temperature value as a function of *r*. For the case of spherical coordinates, Figure 4.27 shows the maximum error as a function of the time step, and Figure 4.28 presents the temperature as a function of *r*. The fact that we obtained very small errors in all cases verifies not only the numerical algorithms, but also the equivalence of the two mathematical treatments of the physical problem.

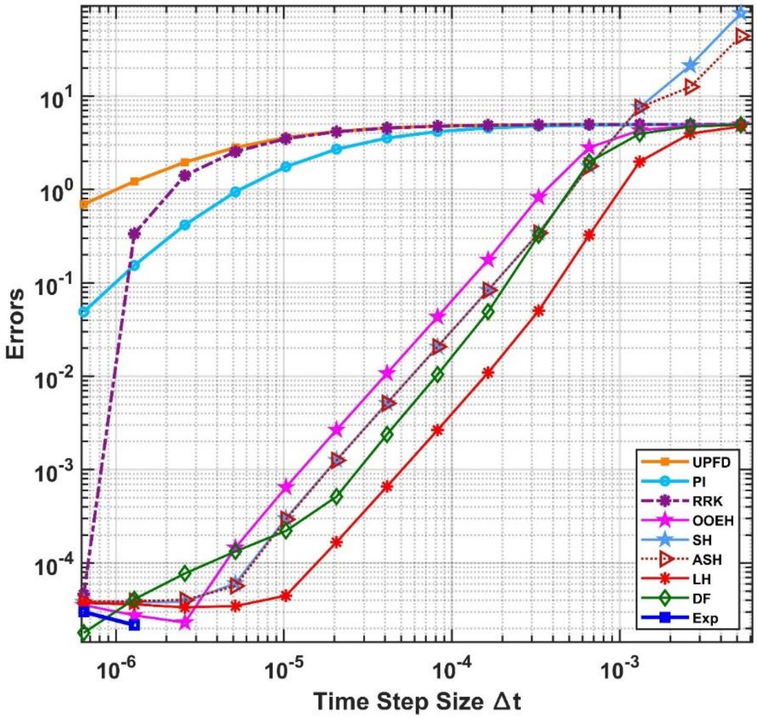

Figure 4.24. The maximum errors as a function of the time step size for the 9 numerical methods in case of cylindrical coordinates for *a* = 1 [42].

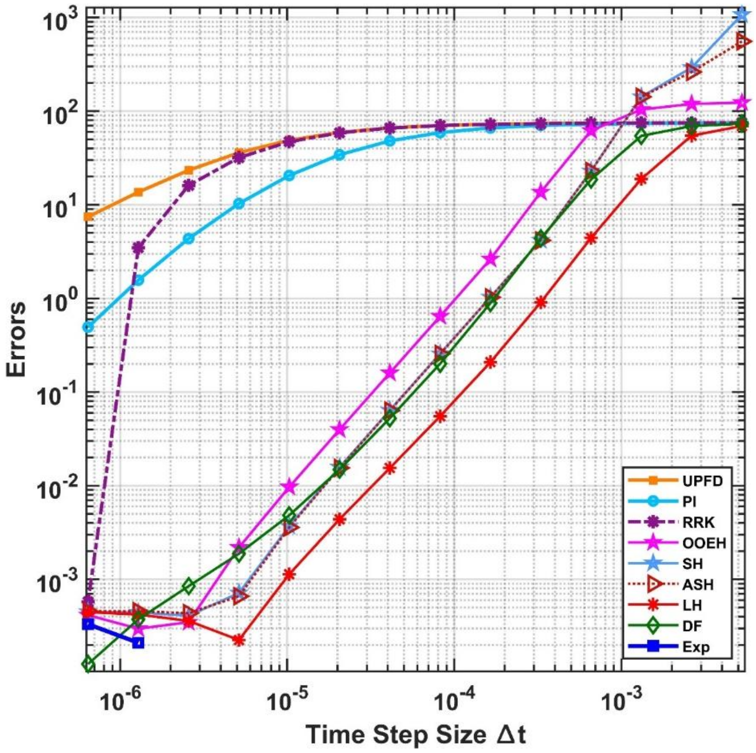

Figure 4.25. The maximum errors as a function of the time step size for the 9 numerical methods in case of cylindrical coordinates for *a* = 2 [42].

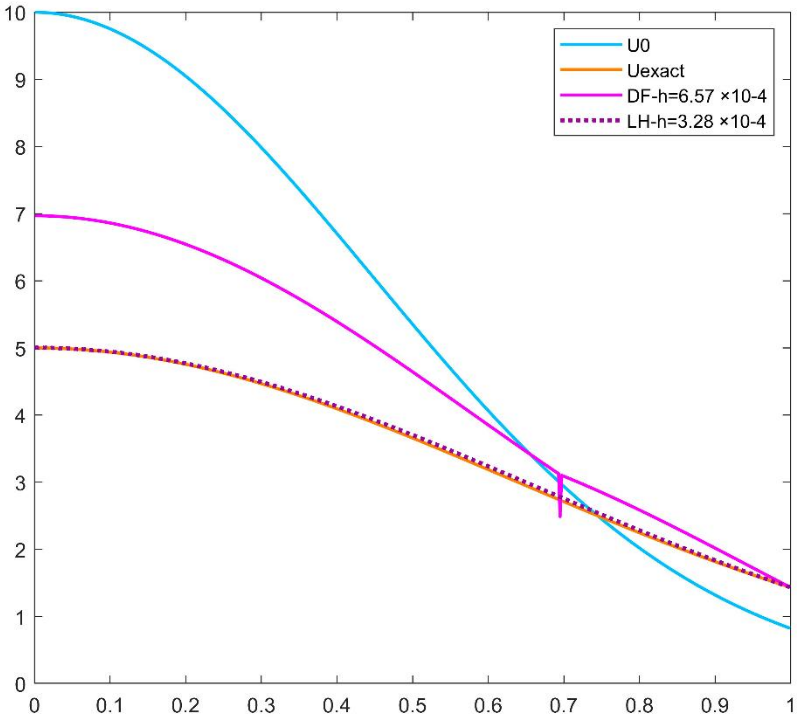

Figure 4.26. The values of temperature as a function of variable *r* in case of the initial function u^0, the analytical solution *U*exact, the DF method, and the LH method in case of cylindrical coordinates for *a* = 1 [42].

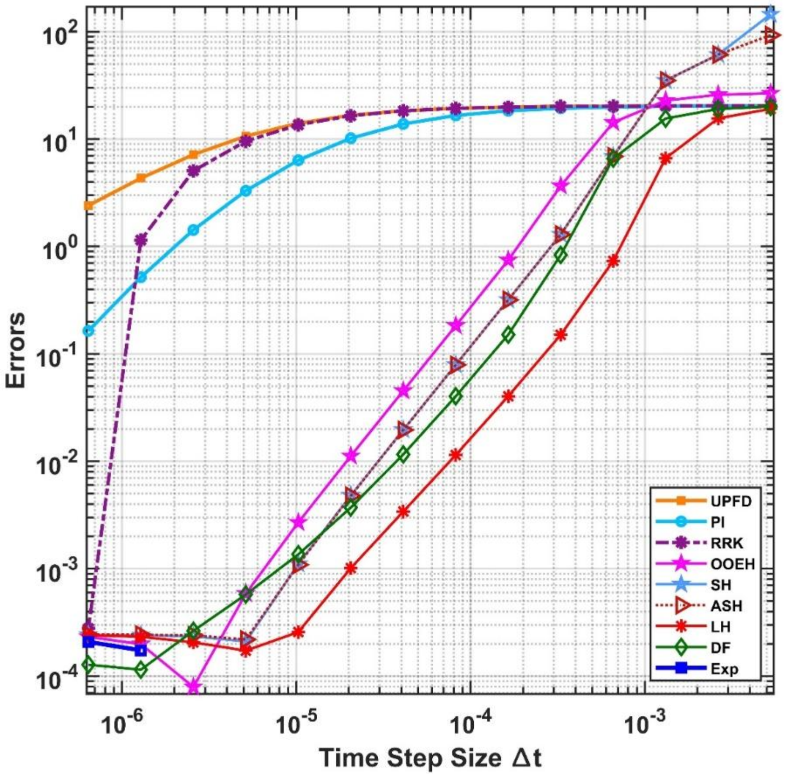

Figure 4.27. The maximum errors as a function of the time step size for the 9 numerical methods in the case of spherical coordinates for *a* = 1.2 [42].

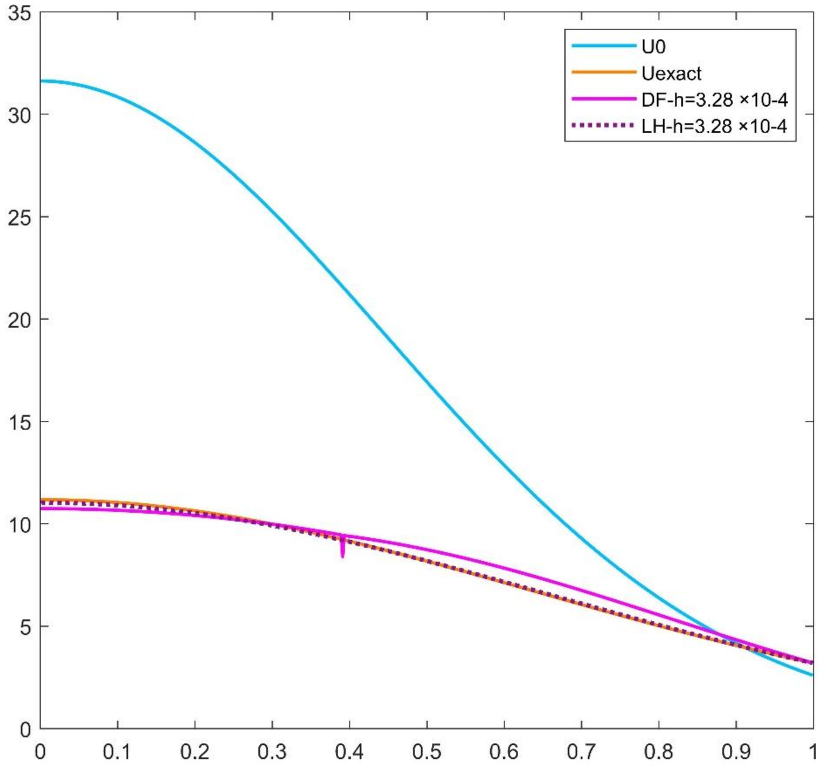

Figure 4.28. The values of temperature as a function of *r* variable in case of the initial function u^0, the analytical solution *U*exact, the DF method, and the LH method in case of spherical coordinates for *a* = 1.2 [42].

4.3.3. The Simulation Results

In this section, I present the results at the end of the examined time interval, which is defined as $t_{\text{fin}} = 1200, 1440 \text{ and } 1800\,\text{s}$ in both the numerical methods and Ansys simulation and then compare between them with the experimental results.

4.3.3.1. The Results of Numerical Methods

For the simulation, I chose the top five algorithms, namely DF, OOEH, LH, SH, and ASH. The simulation of a steel C45 cylinder was conducted using these selected algorithms considering different boundary conditions, as previously mentioned. Among these algorithms, the shifted-hopscotch method was chosen to visualize the temperature contour due to its high accuracy at small time-step size. Figures 4.29 and 4.30 display the final temperature distribution obtained from this method.

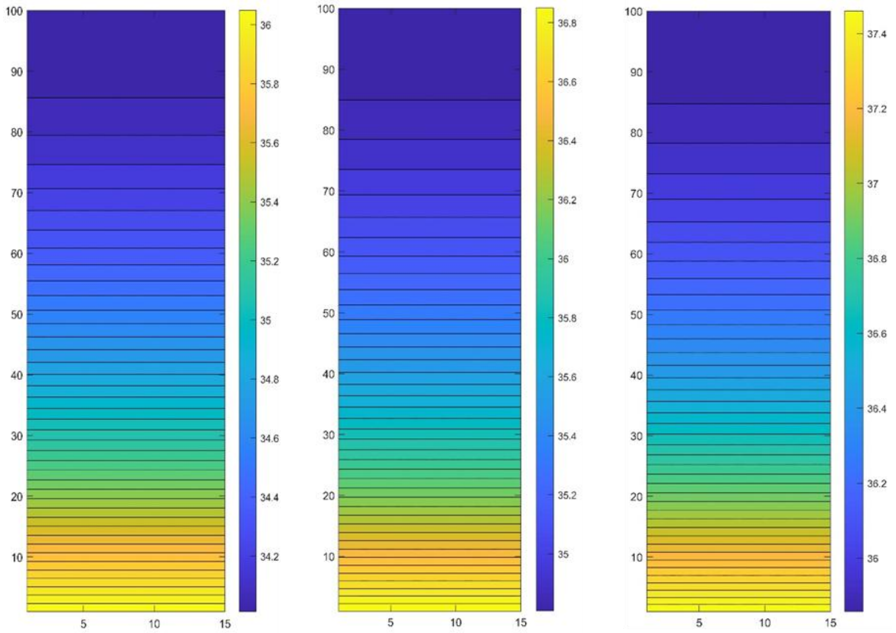

Figure 4.29. The final temperature distribution contour for different time values (t = 20, 24, and 30 min, respectively, from left to right) presented by the SH method when there is no heat exchange with the environment [42].

.

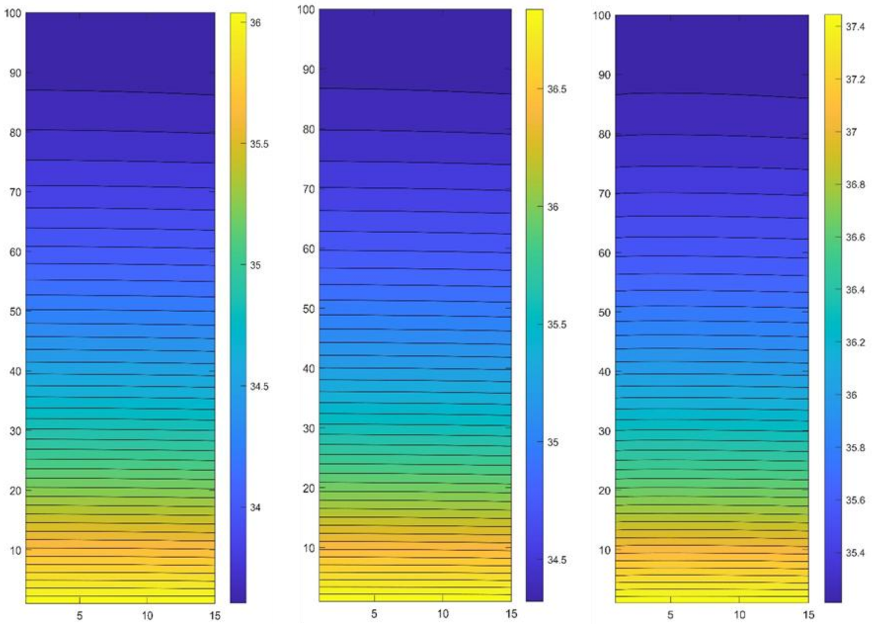

Figure 4.30. The contour of temperature distribution for different time values (*t* = 20, 24, and 30 min) presented by the SH method when there is heat exchange with the environment via convection and radiation [42].

4.3.3.2. The Results of Ansys Simulation

Ansys workbench 19.2 transient thermal analysis with Mechanical APDL solver was used to simulate the steel C45 cylinder. The mesh size was 1×10^{-3} , and the total number of elements was 197,183 since it was a computationally 3D problem. In Figures 4.31 I present the sample of temperature contour at the final time.

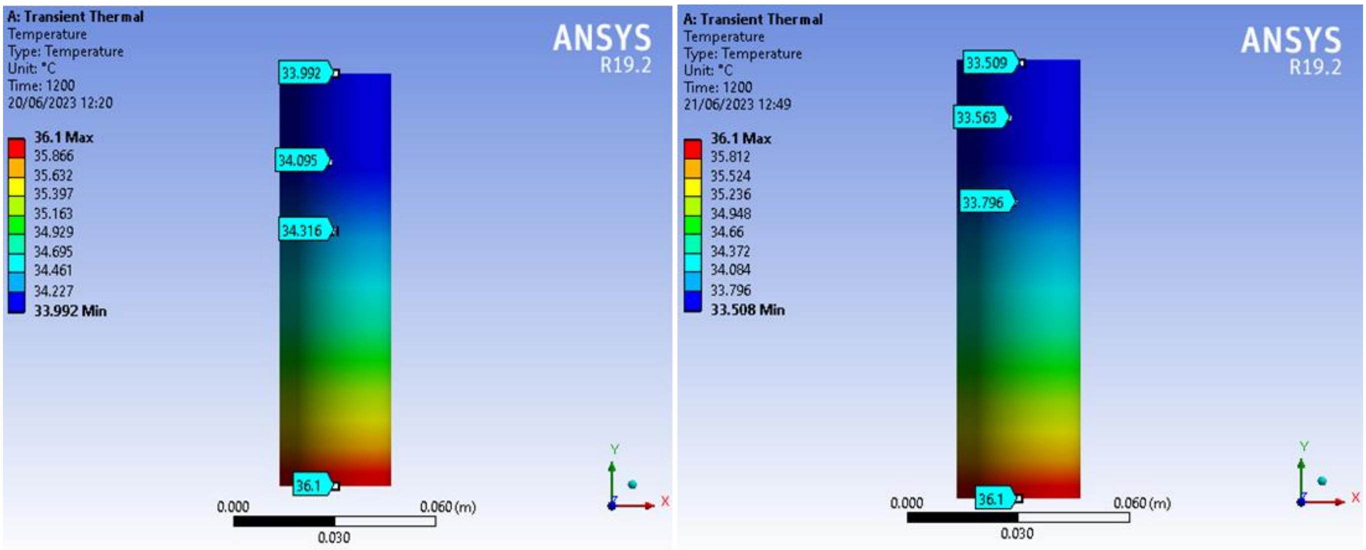

Figure 4.31. The temperature contour at time (t = 20 min) presented by Ansys when there is no heat exchange with the environment (left) and when there is a heat exchange (right) [42].

4.3.3.3. Comparison of The Results

The results of the experimental measurements, the finite element method (FEM) using Ansys Workbench, and the explicit numerical methods (exemplified by the shifted hopscotch method) were compared. Both FEM and SH were subjected to two types of tests, one considering convection and radiation effects, and the other excluding them. First, I employed steady-state thermal analysis using FEM Ansys Workbench to follow the original paper [70] to reach the same results. The maximum deviation was 0.07, which was a kind of verification for the setup. Then, I used transient thermal analysis to follow the real physical processes of the experiment. All results below are for this transient simulation. In Tables 4.6–4.7, the comparison was conducted at two specific spatial points (*z* = 75 and 95 mm, which are the distances from the bottom measurement point), and the results were measured at three different time moments. The temperatures are compared at two space points via plots in Figures 4.32–4.34.

Table 4.6. The temperature at *z* = 125 mm at three different time moments [42].

Time	**Temperature in °C, at *z* = 75 mm**				
	Experiment	**SH with CR**	**SH**	**FEM with CR**	**FEM**
20 min	33.9	33.941	34.298	33.796	34.316

24 min	34.6	34.668	35.087	34.534	35.128
30 min	35.7	35.514	36.07	35.283	36.036

Table 4.7. The temperature at *z* = 145 mm at three different time moments [42].

Time	Temperature in °C, at *z* = 95 mm				
	Experiment	**SH with CR**	**SH**	**FEM with CR**	**FEM**
20 min	33.7	33.71	34.099	33.563	34.095
24 min	34.5	34.427	34.88	34.285	34.88
30 min	35.5	35.30	35.92	35.093	35.856

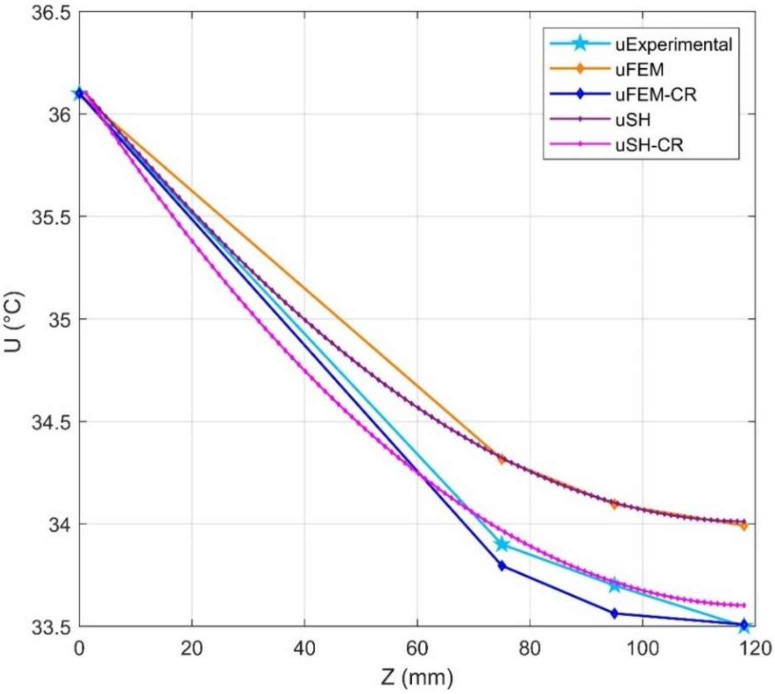

Figure 4.32. The temperature at the 4 selected measurement points in *z* at time t = 20 min [42].

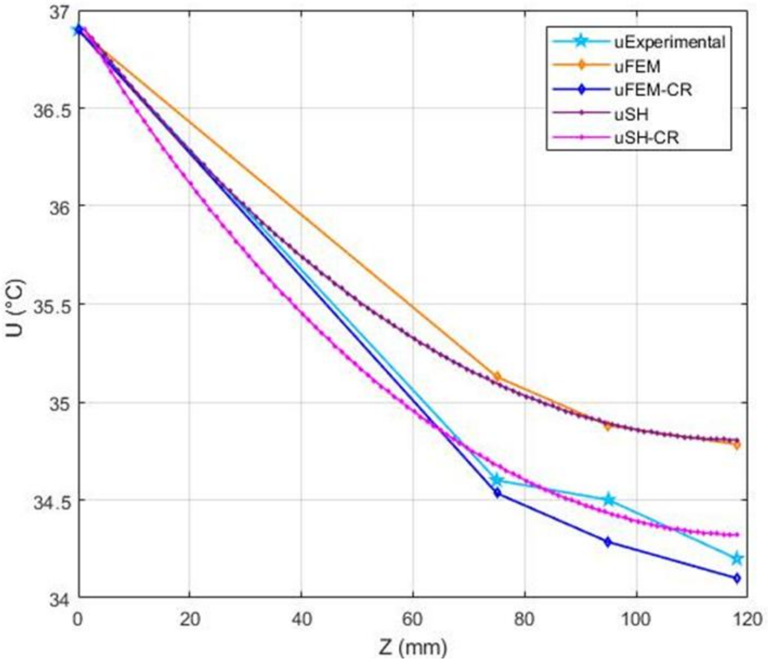

Figure 4.33. The temperature at the 4 selected measurement points in *z* at time t = 24 min [42].

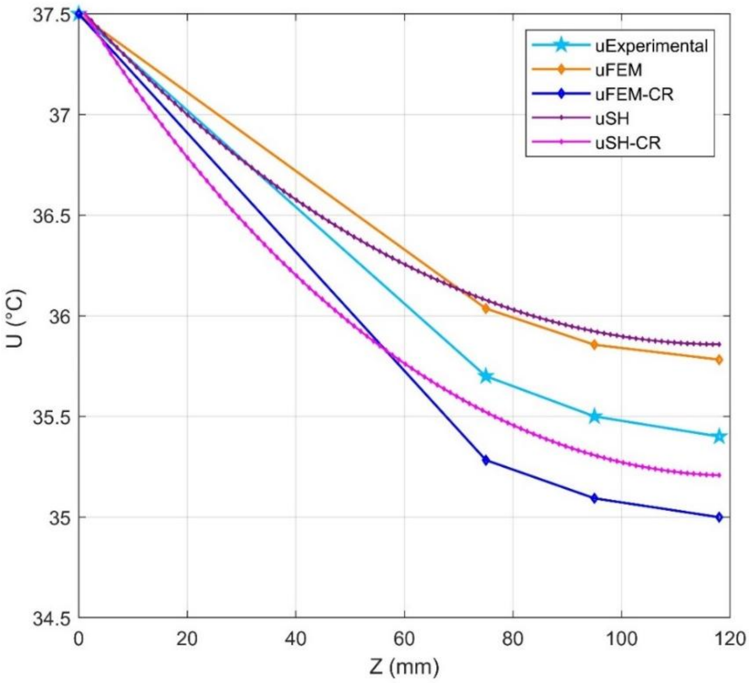

Figure 4.34. The temperature at the 4 selected measurement points in *z* at time t = 30 min [42].

The figures and tables presented above illustrate a comparison of results obtained from the current numerical methods and the FEM ANSYS, utilizing experimental data from the literature study [37]. The findings indicate that the numerical methods employed in this study demonstrate superior accuracy compared with the FEM ANSYS used in both the current investigation and the same literature study [72] [37].

4.3.4. The Summary of The Present Section

This work was devoted to solving heat transfer problems in cylindrical and spherical geometries. Using the self-similar Ansatz, novel analytical solutions of the heat-conduction PDE were constructed, which contained the Kummer’s functions. Nine numerical algorithms were presented, most of which are recently introduced unconditionally stable explicit methods. To perform the verification, the novel analytical solutions of the heat-conduction PDE containing the Kummer’s functions were reproduced by these methods with high accuracy.

After these, experimental work was considered from the literature where a cylinder is heated from below, and the results were attempted to be reproduced using Ansys commercial software, but without considering convection and radiation on the surface of the cylinder. In contrast to that, I reproduced the experimental results by considering convection and radiation as well, not only using Ansys, but also the explicit methods. Since, in reality, convection and radiation are present, taking them into account makes the results closer to the experimental ones, especially for the first two measurement times. Moreover, the explicit and stable schemes were more accurate and effective than the finite element software in all cases. The LH algorithm was usually the most accurate among the studied methods. However, similarly to all hopscotch methods, it needs a special mesh, which can be hard or maybe impossible to implement for problems with irregular shapes. This limitation of these methods is probably more restrictive in complicated 3D problems.

4.4. *Calculate The Heat Transfer in Multilayer Walls with Photovoltaic Cell and Air.*

This section explores the simulation of the non-linear transient heat transfer equation (2.20) in multilayer walls subjected to various heat loads using efficient numerical algorithms (UPFD, NS-DF, PI, NS-OEH, LH, SH, and ASH). The study considers conduction, free and forced convection, and nonlinear radiation involving a two-phase material composed of solid (wall construction) and fluid (air). Different wall geometries and heat load scenarios are examined, encompassing both cooling and heating cases. The objective is to evaluate algorithm performance for outdoor surface convection and an air gap between insulation and PVC [73].

4.4.1. Geometry Model and Mesh Generation

Figure 4.35 helps to visualize the geometry and the environment for inside and outside of the wall section, with zooming on the selection cross-sectional area in the middle of the wall (the upper half is sunny and the lower half is in shadow) that will be simulated.

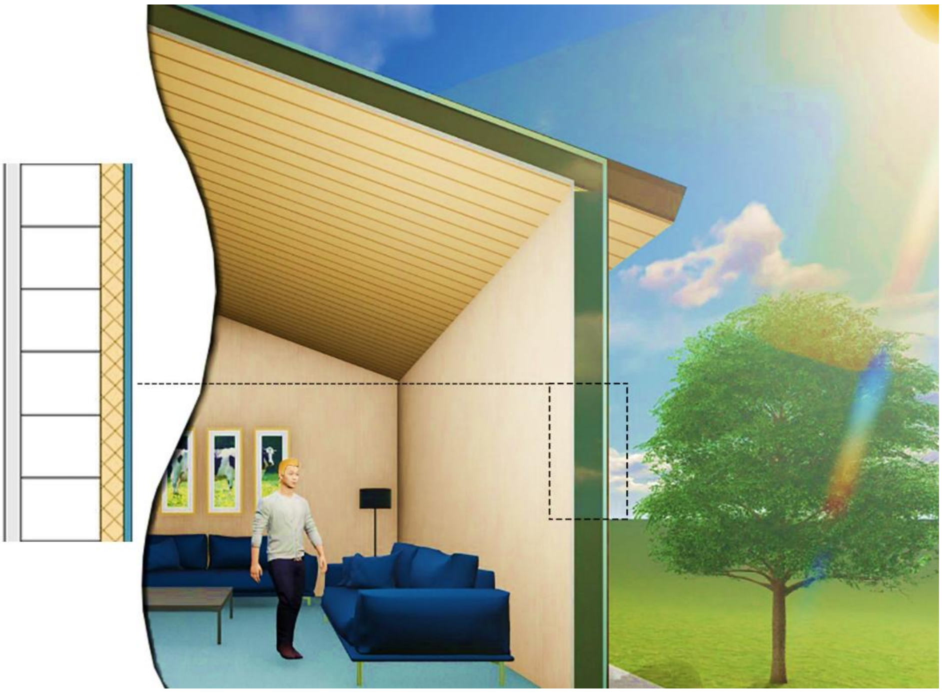

Figure 4.35. Visualization of the studied case, the selected wall cross-section [73].

The geometry is a multilayer wall with an air gap. The order of the media is the following: gypsum board, brick, glass wool, air gap, Photovoltaic Cell (PVC), and then air. I also investigated free and forced convection with cooling as well as heating processes. In this scenario, there are different kinds of convection depending on the air status on both sides of the PVC, when the air is moving (forced convection) and when the air is stationary (free convection). From this point of view, there are three subcases:

A. free-free convection means the air is stationary on both sides of the PVC,

B. free-forced convection: the air is stationary in the air gap zone, and the air is moving on the other side of the PVC.

C. forced-forced convection: the air is moving on both sides of the PVC, so in this case, there is forced convection on both sides of the PVC. This case is useful in cooling of photovoltaic cells when it warm up due to the hot weather, especially in summer or in a hot climate area, when the solar cell's temperature exceeds the optimum operation temperature.

I consider a wall segment with a cross-sectional area $S = L_X \times L_Z$ with value 0.5 m $\times$0.5 m.$(x,z) \in [0, 0.5] \times [0, 0.5]$, thus the mesh's total area is 0.25m^2. I have constructed an equidistant grid with square cells shaped for all cases. The number of cells along the *x* and *z* axes are set to N_x = 100 and N_z = 100; thus, I have a mesh with a total cell number $N = N_x N_z = 10{,}000$. The cells are indexed as a linear sequence, starting from the top left corner horizontally and ending at the bottom right corner. Due to this, the cell indexed by $i + N_x$ is just below the cell labelled by *i*, etc.

4.4.2. *Materials and Boundary Conditions*

In the current study, I use real material properties for wall construction, as shown in Table 4.8.

Table 4.8. The materials used properties [71]-[74] .

Material	$\rho\left(\mathrm{kg \cdot m^{-3}}\right)$	$k\ \left(\mathrm{W \cdot m^{-1} \cdot K^{-1}}\right)$	$c\ \left(\mathrm{J \cdot kg^{-1} \cdot K^{-1}}\right)$
Gypsum Board	805	0.292	977
Brick	1600	0.730	800
Glass wool	200	0.030	800
PVC, Silicon	2330	148	710.08
PVC, Glass	2500	1.7	780.33
Air, at 283K	1.2474	0.024840	1005.8
Air, at 288K	1.2257	0.025219	1005.9
Air, at 303K	1.1649	0.026341	1006.5

The initial conditions are constant for all cases as follows:

- Cooling case: solid temperature = 303 K, air gap temperature = 288K, air temperature= 283 K.
- Heating case: solid temperature = 283 K, air gap temperature = 288K, air temperature= 303 K.

I apply zero Neumann boundary conditions in all cases for the right, the top and the bottom boundary, which do not allow any heat transfer at those boundaries.

There are two types of incoming radiation: one of them is coming from outside of the studied system and it is independent of the temperatures in the system, thus I denote it by $q^*_{\text{from out}}$. The second type is coming from another part of the system and thus it is a temperature time-dependent variable, which can be denoted by $q^*_{\text{from in}}$. At the left-hand side of the system, the conduction is neglected, but the wall loses heat by radiation and convection to the interior of the building, and also gain the appropriate heat, which are included into the heat generation term. The intensity of the incoming radiation will be considered as a constant $q^*_{\text{from out}} = 400\text{W/m}^2$. The ambient temperature of the room is always $u_a = 293\text{K}$ which is considered as a comfortable temperature for a living space. The interior elements of the solid material cannot lose or gain heat by the heat convection, radiation, and heat source, only by conduction.

In case of free convection boundary the elements on the left and right sides in the interface between solid and fluid can transfer heat by convection and radiation with the values shown in Tables (4.9) [75]. I use realistic values for σ^* as it was explained above. The heat source generation contains a part of the solar radiation, with which I obtain the value of $q^*_{\text{from out}}$ as shown in table below. For the heat generation for the interface elements 1 and 2, I put (-) in the table because it receives $q^*_{\text{from in}}$ type radiation. The air ambient temperature (on the right side) is taken to be $30°\text{C} \approx 303\text{K}$ in case of heating and $10°\text{C} \approx 283\text{K}$ in case of cooling. Here interface 1 is the interface between the insulator and the air gap, and interface 2 is the interface between the PVC and airgap, while interface 3 is the interface between the PVC and the surrounding air.

Table 4.9. The heat convection, radiation, and source parameters on right and left sides of the wall elements [73].

	$h_c \left(\frac{\text{W}}{\text{m}^2 \cdot \text{K}}\right)$	$\sigma^* \left(\frac{\text{W}}{\text{m}^2 \cdot \text{K}^4} \times 10^{-8}\right)$	$q^*_{\text{from out}} \left(\text{W}/\text{m}^2\right)$
Left Elements	2	5	400
Interface 1 (insulator)	4	4	-
Air gap elements	4	0	0
Interface 2 (silicon)	4	4	-
Interface 3 (glass) sunny part	4	4	600
Interface 3 (glass) shaded part	4	4	100
Surrounding air elements	4	0	0

There is an air gap between the insulator and PVC, those two surfaces radiate each other with a $q^*_{\text{from in}}$ type radiation. The quantity of the radiative heat transfer changes with the temperature of each surface. In this case the heat generation (incoming heat) of the surface elements can be calculated as follows :

- For Interface elements 1: $q = \frac{\sigma^*}{c\rho\Delta x} \cdot u^4_{Silicon} + \frac{h_c}{c\rho\Delta x} \cdot u_{air\_gap}$.
- For air gap: $q = \frac{h_c}{c\rho\Delta x} \cdot u_{air\_gap}$.
- For Interface 2: $q = \frac{\sigma^*}{c\rho\Delta x} \cdot u^4_{Insutator} + \frac{h_c}{c\rho\Delta x} \cdot u_{air\_gap}$.

While in the case of forced convection, all the boundaries have the same expressions for heat transfer. However, the heat transfer coefficient in forced convection is not a constant but depends on air velocity, which I take in the *z* direction. The convection coefficient *h* for the air elements depends on the nondimensional parameters *Nu* and *Re*, which are derived based on the energy balance at the thermal boundary layer of air (for more details, see [76]). The procedure is as follows:

The heat transfer coefficient: $h_c = Nu\left(\frac{k}{L_z}\right)$, where L_z is the length of the surface in the *z* direction.

Nusselt number: $Nu = 0.664\, Re^{1/2}\, Pr^{1/3}$

Reynolds number: $Re = \frac{\rho v L_z}{\mu}$, where *v* is the air velocity, which is 0.5m/s, and μ is the dynamic viscosity.

Prandtl number: $Pr = \frac{v}{\alpha} = \frac{\mu c_p}{k}$, I can get it from an air properties table.

For the forced convection, the values of Reynolds Number Re are as follows:

- free-forced; the surrounding air moving at velocity v=0.5m/s, and Re= 17596.95 for cooling, Re= 15589.9 heating.
- forced-forced the air moving on both sides of the PVC at velocity v=0.5m/s, Re= 17056.69 for the air gap zone, Re= 17596.95 cooling, and Re= 15589.9 heating for the surrounding air.

4.4.3. The Simulation Results

I applied the initial condition and boundary conditions of section 4.4.2 with t_{fin} = 20,000s, the cases of study in both cooling and heating in free and forced convection. The maximum errors are plotted in Figures 4.36– 4.38, where it can be seen that the DF and the hopscotch methods lose their advantage with respect to the PI method if there is forced convection in the air gap. The main reason for this is that there is a rapid heat exchange between the air and the inner surface of the PVC, which consists of silicon that has a large heat conductivity, and this makes the required time step size smaller. Figures 4.39-4.42 show the contours of temperature distribution for the initial and final time step for both forced and free convection. It is shown clearly that the effect of the used insulator prevents the heat from penetrating inside, and the air gap is a key rule in circulating the air and enhancing the heat transfer performance. Figure 4.43 shows the effect of the air gap and that of forced convection in cooling down the PVC (silicon) layer.

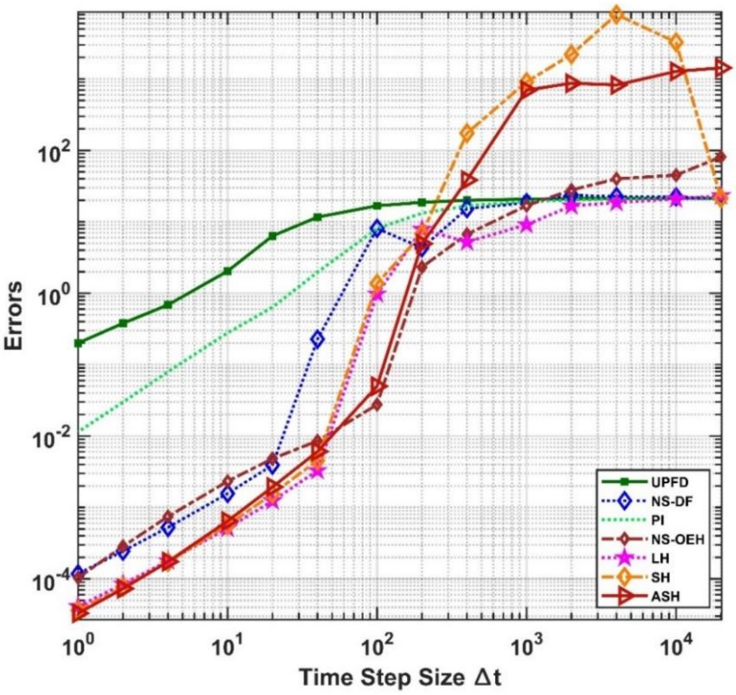

Figure 4.36. The maximum errors as a time step size function for the 7 tested methods in the case of free-free convection cooling [73].

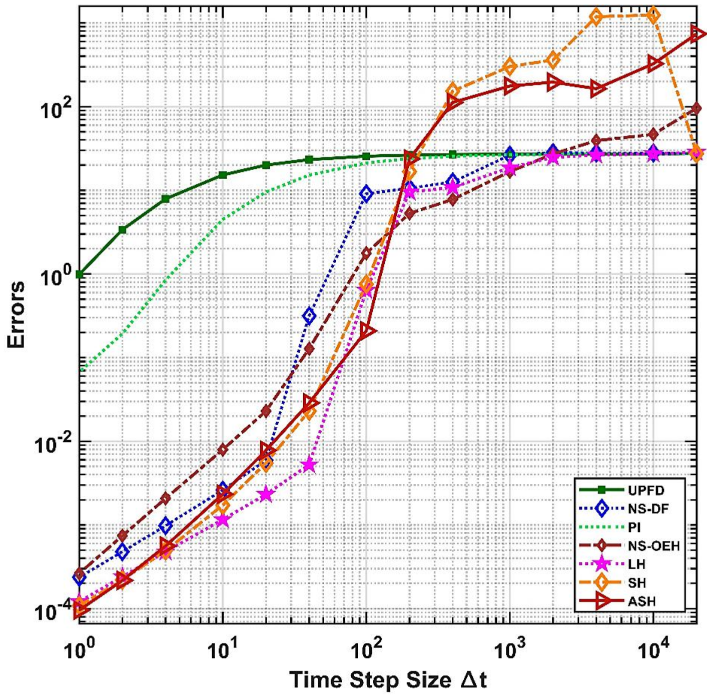

Figure 4.37. The maximum errors as a time step size function for the 7 tested methods in the case of free-forced convection cooling [73].

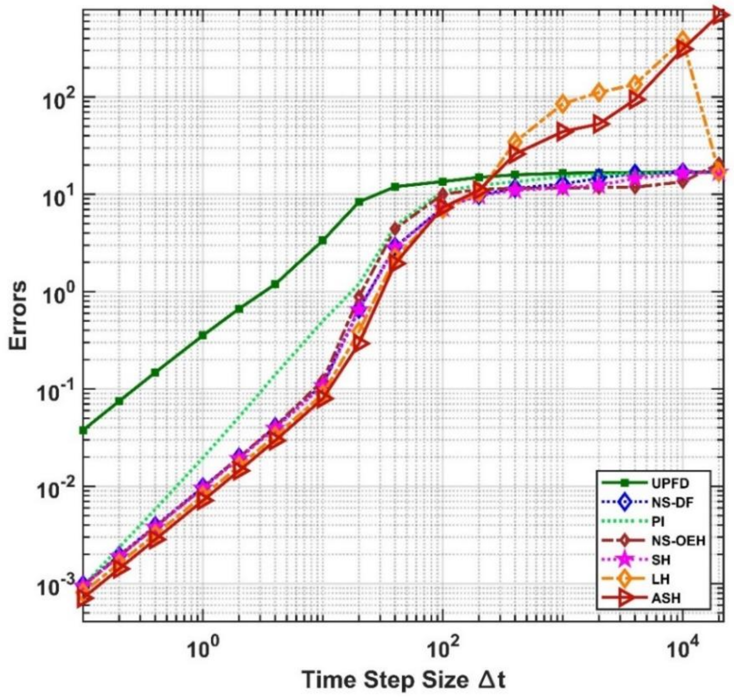

Figure 4.38. The maximum errors as a time step size function for the 7 tested methods in the case of forced-forced convection cooling [73].

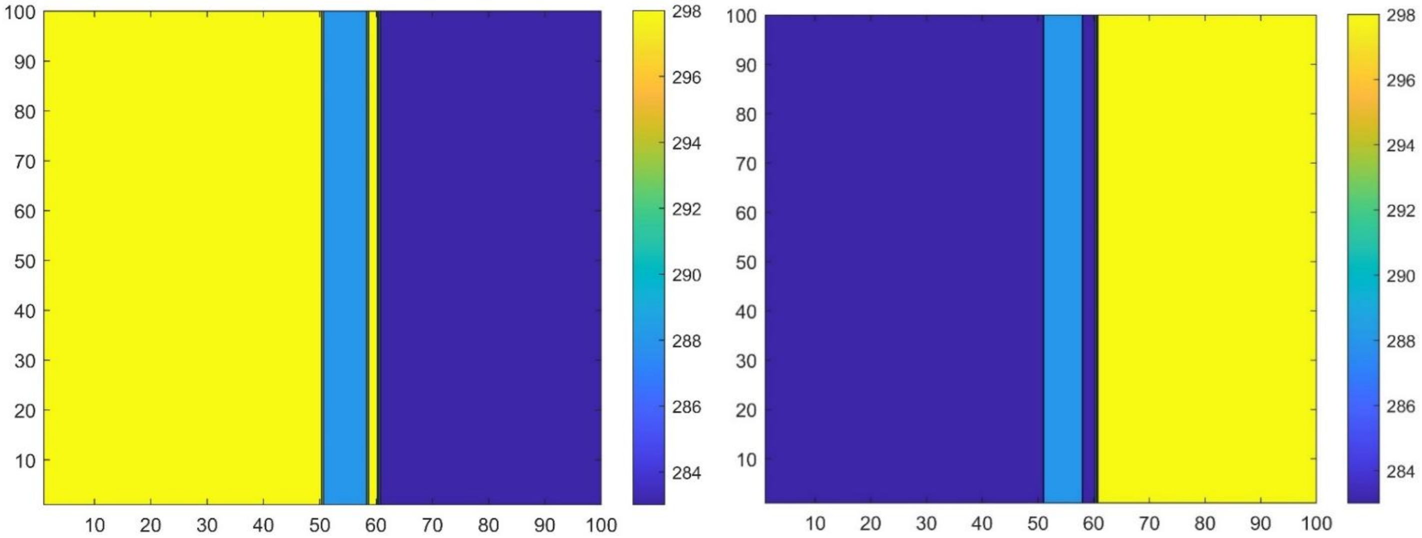

Figure 4.39. The contour of initial temperature distribution in Kelvin for free and forced convection in Case 3, in case of Cooling (left), and Heating (right) [73].

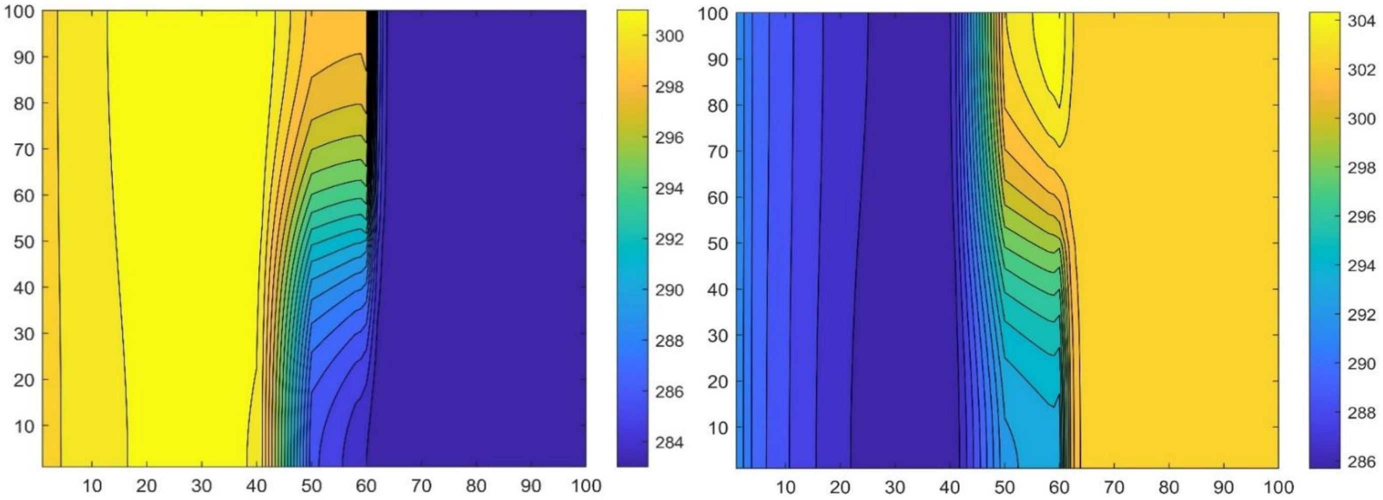

Figure 4.40. The contour of final temperature distribution in Kelvin for free-free convection in Case 3, in case of Cooling (left), and Heating (right) [73].

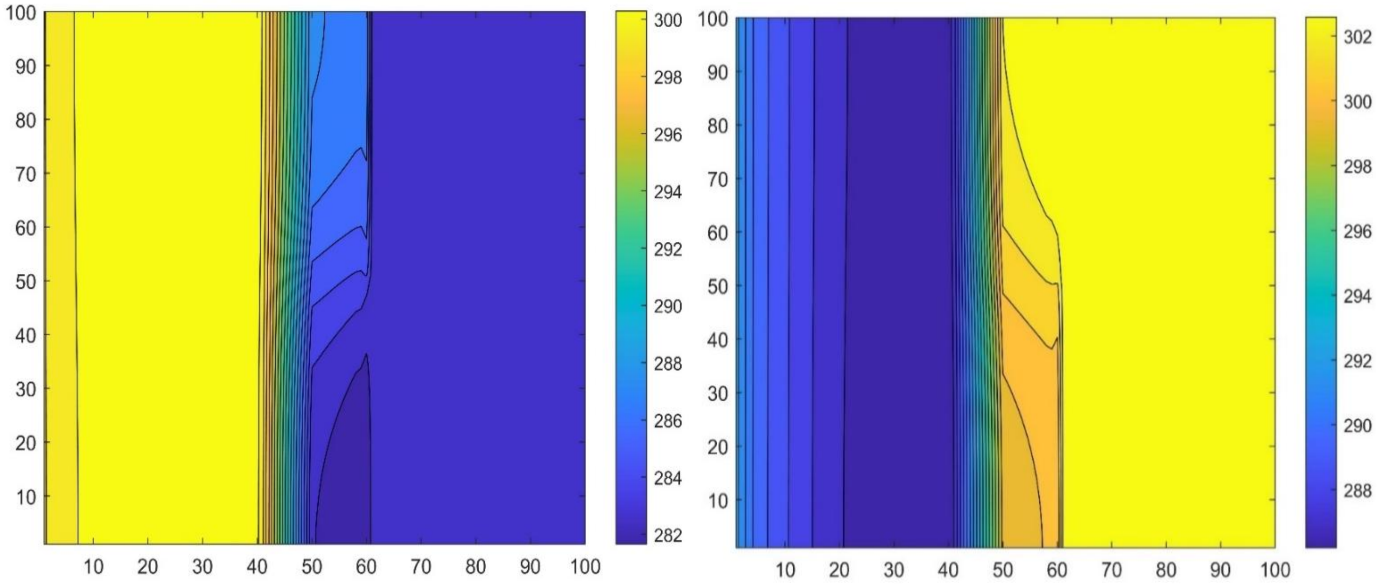

Figure 4.41. The contour of final temperature distribution in Kelvin for free-forced convection in Case 3, in case of Cooling (left), and Heating (right) [73].

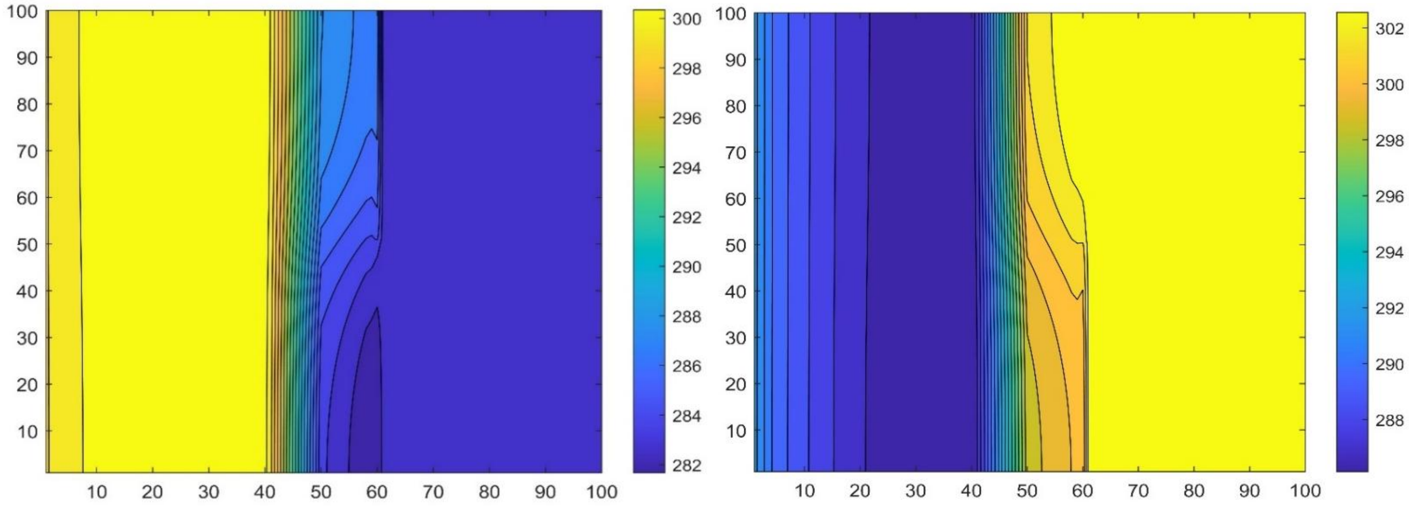

Figure 4.42. The contour of final temperature distribution in Kelvin for forced-forced convection in Case 3, in case of Cooling (left), and Heating (right) [73].

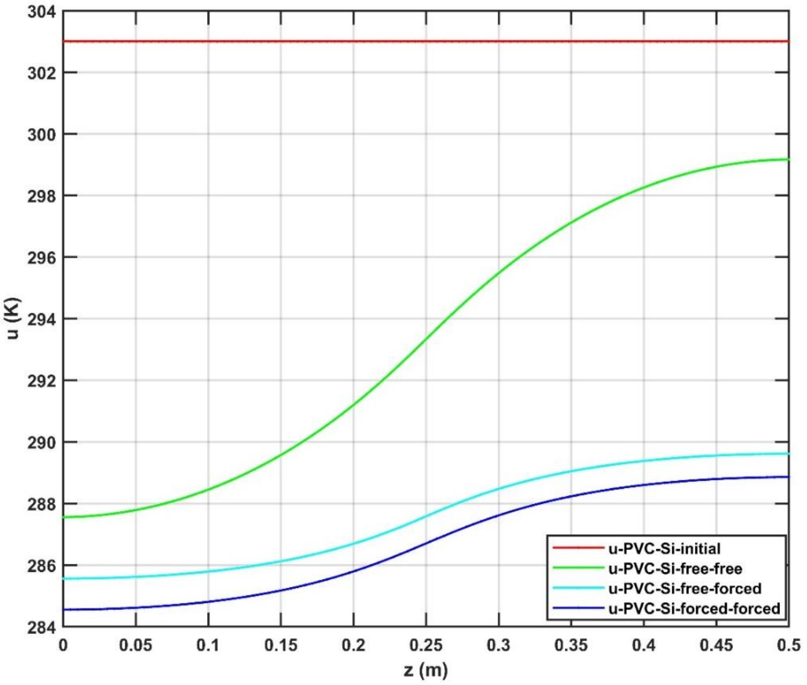

Figure 4.43. The temperature at the PVC (silicon) border in Kelvin for Case 3 in case of Cooling [73].

4.4.4. The Summary of The Present Section

I numerically studied transient heat transfer in the form of conduction, convection, and radiation in two-dimensional systems of gypsum board, brick, glass wool, air gap, PVC, and air. I used seven stable numerical algorithms for this purpose. The ode15s MATLAB routine served with the reference solution in all examined cases.

The tested methods’ advantages and disadvantages are listed in the following:

1. The UPFD is first order of magnitude, and generally not accurate enough, but it can treat convection and radiation terms very well. For an arbitrary time-step size, it is positivity preserving; all other methods are not. However, it is by far the least accurate for medium and small time step sizes.

2. The Hopscotch family (NS-OOEH, SH, LH, and ASH), the NS-DF, and the pseudo-implicit methods are second order, but the latter one is usually much less accurate due to the extra terms in its truncation error.

3. The LH is typically the most efficient algorithm to handle these kinds of problems. However, when there is forced convection in the air gap, the LH, as well as other accurate methods, lose most of their advantage, and the PI method can also be effectively used.

4. The current algorithms successfully deal with very stiff systems; thus, they are expected to be able to cope with any kind of materials or boundary conditions.

To conclude, the LH, ASH, and NS-DF algorithms can be proposed to solve these problems. All methods produce very accurate solutions and can utilize larger time steps. This capability results in superior computational speed over conventional explicit techniques that are constrained by stability limitations.

The conclusions from the engineering point of view are the following:

5. The used insulator on the outside of the brick prevents the heat from penetrating inside and, in this way, I keep the inside environment within a comfort limit.

6. The heat transfer in convection and radiation can be controlled at the boundary by applying forced convection.

7. The forced convection heat transfer has a significant effect on improving the heat transfer, especially in the case of cooling to cool down the PVC, which has a performance temperature limit to work in.

8. The temperature of the PVC exposed to sunshine is reduced significantly even by a light wind.

9. The air gap between the PVC and the insulator reduces the temperature at the insulator border and the PVC borders.

Related to the wall construction, I could recommend using both the insulator and the air gap to reduce the heat going inside the building due to hot weather and strong sunshine.

4.5. *Calculate The Heat Transfer in Building Walls with PCMs Using Effective Heat Capacity Model*

I employ efficient explicit numerical methods and validate my approach against established mathematical expressions and models in the literature. My research investigates various building wall geometries and boundary conditions, primarily focusing on employing the Effective Heat Capacity model to manage heat loads. The objective is to maintain interior temperatures within comfort zones. I compare two types of paraffin wax PCMs. The first one is characterized by a lower melting temperature and higher latent heat capacity; thus, it can efficiently store external heat when combined with brick or concrete [77].

4.5.1.Theory and Considerations of the Present Study

I perform the thermal analysis of PCM integrated with building components by using the effective heat capacity (EHC) model with two phases (solid: So and liquid: Li). It implies that the specific heat *c*, the heat conductivity *k*, and the density *ρ* depend not only on the space (due to material inhomogeneities) but on the temperature itself. In one dimension, the following PDE, the heat conduction equation, can be used to predict the behaviour of the temperature:

$$\frac{\partial u}{\partial t}=\frac{1}{\rho(x,u)c(x,u)}\nabla(k(x,u)\nabla u)+q\,. \tag{4.10}$$

To determine the heat capacity of the cell, I consider two types of heat capacity: sensible heat capacity (SHC) and latent heat capacity (LHC). Standard materials, such as brick and concrete, which cannot change their phase in normal conditions, have only sensible heat capacity. For PCMs, the EHC is computed as the sum of SHC and LHC at each phase, while taking phase transitions into account, as follows [77]:

$$C_i^{So}=c_i^{So}\rho_i^{So}V_i\text{ , and }\; C_i^{Li}=c_i^{Li}\rho_i^{Li}V_i\,. \tag{4.11}$$

Those represent the SHC for the liquid and solid states of the material.

For the EHC, I define *g* as the Gaussian function[78] centered at the melting temperature u_{cr} of the material with the standard deviation *σ* [77]:

$$g(i)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(u_i-u_{cr})^2}{2\,\sigma^2}\right). \tag{4.12}$$

At the phase transition region $u_{cr}-\sigma\le u\le u_{cr}+\sigma$, I used the following functions to represent the thermal properties [77]:

$$k_i=\frac{\left(\left(k_i^{Li}-k_i^{So}\right)(u_i-u_{cr})+\sigma\left(k_i^{Li}+k_i^{So}\right)\right)}{2\,\sigma}, \tag{4.13}$$

$$\rho_i=\frac{\left(\left(\rho_i^{Li}-\rho_i^{So}\right)(u_i-u_{cr})+\sigma\left(\rho_i^{Li}+\rho_i^{So}\right)\right)}{2\,\sigma}, \tag{4.14}$$

$$c_i=\frac{\left(\left(c_i^{Li}-c_i^{So}\right)(u_i-u_{cr})+\sigma\left(c_i^{Li}+c_i^{So}\right)\right)}{2\sigma}, \tag{4.15}$$

where *c* is the specific heat capacity of the material and it depends on the material’s state (solid, mixed solid and fluid, and liquid) and I used to calculate the sensible heat capacity as shown in the following equations. These functions are linear in the temperature variable and continuous in all of the parameters.

At the phase transition I used EHC as follow [77]:

$$EHC_i=SHC_i+LHC_i\;, \tag{4.16}$$

$$SHC_i=c_i\rho_iV_i\;, \tag{4.17}$$

$$LHC_i=H_ig_i\rho_iV_i\;. \tag{4.18}$$

All quantities calculated by equations (4.11-4.18) are updated at each time step and at each stage, contributing to the overall time consumption of the calculations. In Figure 4.44, there are some plotted examples of the Gaussian function at different σ values and u_{cr}=320 K.

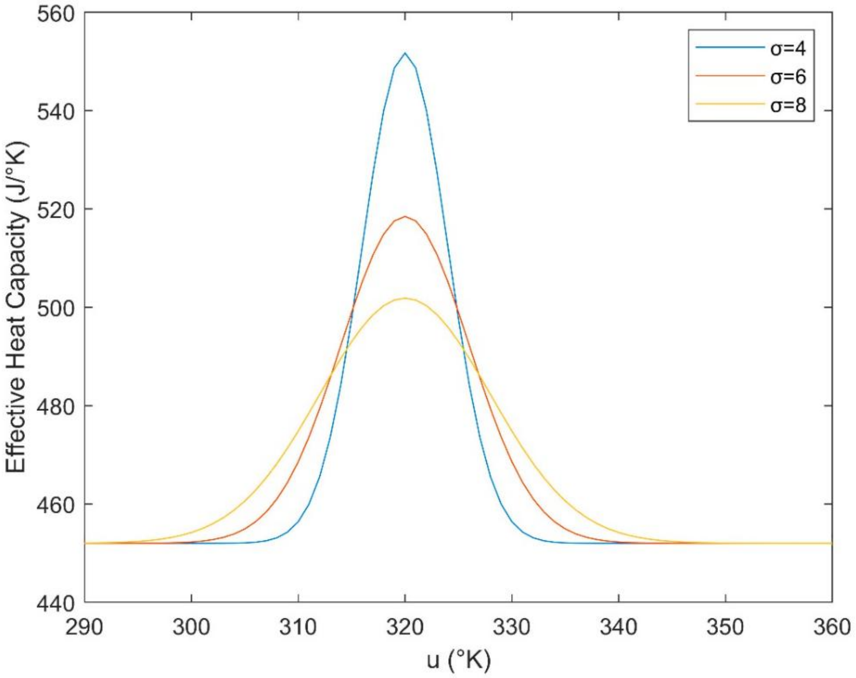

Figure 4.44. Gaussian function representative of the heat capacity with temperature [77].

Keeping in mind the above-mentioned considerations, the time-development of the temperatures can be calculated by solving the system of ordinary differential equations (ODEs):

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} + q_i \ , \tag{4.19}$$

which is the spatially discretized form of the nonlinear heat equation. After the temperatures are calculated, the total heat Q_T, sensible heat $Q_{Se,}$ and total latent heat Q_{La} of thermal systems can be given as follows [77]:

$$Q_T^t = Q_T^{t-1} + \sum_{i=1}^{N} EHC\left(u_i^t - u_i^{t-1}\right) \ , \tag{4.20}$$

$$Q_{Se}^t = Q_{Se}^{t-1} + \sum_{i=1}^{N} SHC\left(u_i^t - u_i^{t-1}\right) \ , \tag{4.21}$$

$$Q_{La}^t = Q_{La}^{t-1} + \sum_{i=1}^{N} LHC\left(u_i^t - u_i^{t-1}\right) \ . \tag{4.22}$$

Here, t is the index of the time level after the discretization of the time variable, which will be explained later. Those data measure the ability of PCM to store the energy during the phase transition, which has the advantage of reducing the energy consumption in the building and keeping the comfort indoor temperature. From this point of view, I also calculate the cooling load in terms of total heat transfer from outside to inside (x direction) as follows:

$$Q_{Cooling}^t = Q_{Cooling}^{t-1} + \frac{\Delta t\,(u_2 - u_1)}{R_1} \ , \tag{4.23}$$

where u_1 and u_2 are the first and second cells of internal surface layers respectively, thus they approximate well the heat transfer from the room to the wall and outside.

4.5.2. The 1D Analytical Solution

In the current section, I explain the heat transfer in a PCM by using Stefan-tape problems which have an explicit analytical solution [79]. The Paraffin wax PCM with properties shown in Table 4.10 is inside the container. Within this setup, a Paraffin wax slab, is presumed to possess a semi-infinite length along the *x*-axis. The boundary condition (BC) is zero-Neumann (insulated) on all boundaries except the left side (*x*=0), where it is Dirichlet BC with constant temperature (u-$_{face}$). This allows the heat to flow into or out of the system, which therefore undergoes the melting or solidification process from the left side toward the right, as shown in Figure 4.45. The thermal conductivity is the same for the solid and liquid states of the PCM. The interaction between the solid and fluid components depends upon the applied temperature and exposure duration. Consequently, temperature calculations are determined by the specific region within the body.

Table 4.10. The Paraffin wax PCM1 properties [79] .

Material	$\rho\left(\mathrm{kg\cdot m^{-3}}\right)$	$k\ \left(\mathrm{W\cdot m^{-1}\cdot K^{-1}}\right)$	$c\ \left(\mathrm{J\cdot kg^{-1}\cdot K^{-1}}\right)$	*Latent Heat* $\left(\mathrm{J\cdot kg^{-1}}\right)$
PCM$_{Solid}$	856	0.15	2210	247000
PCM$_{Liquid}$	778	0.15	2010	

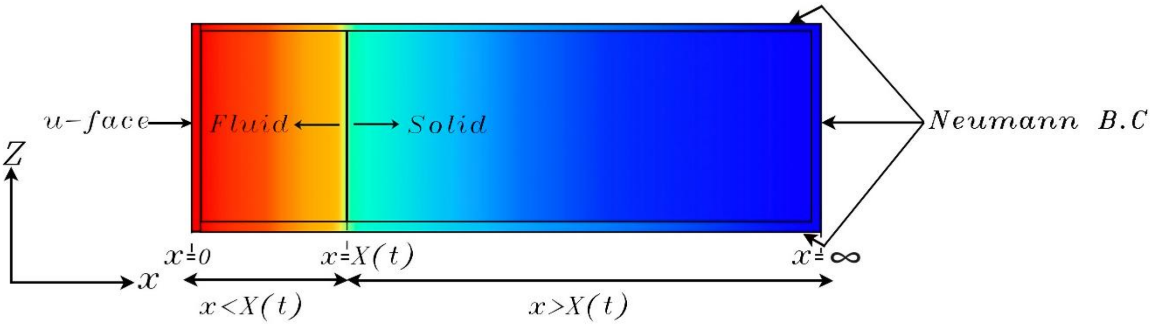

Figure 4.45. The paraffin wax inside the container [77].

The PCM initially has a temperature u_0, and has a melting or solidification temperature u_{cr}, where:

$$u_{face} > u_{cr} > u_0$$

in case of melting with initially solid PCM, and

$$u_{face} < u_{cr} < u_0$$

in case of solidification with initially liquid PCM.

The location of phase interaction between liquid and solid at time *t* is $x = X(t)$

$$X(t) = 2\lambda\sqrt{\alpha_L\, t}\ . \tag{4.24}$$

The temperature $u(x,t)$ of the liquid zone, where $0 < x < X(t)$, is

$$u(x,t) = u_{face} + \frac{u_{cr} - u_{face}}{erf(\lambda)} erf(x/_{2}\sqrt{\alpha_L t}) \ , \tag{4.25}$$

while the temperature of solid zone $x > X(t)$

$$u(x,t) = u_0 + \frac{(u_{cr} - u_0) - (1 - erf(x/_{2}\sqrt{\alpha_L t}))}{1 - erf(\lambda \frac{\alpha_L}{\alpha_s})} \ . \tag{4.26}$$

Here λ is the root of the transcendental equation

$$\frac{e^{-\lambda^2}}{erf(\lambda)} - \frac{k_S\sqrt{\alpha_L}\,(u_0 - u_{cr})}{k_L\sqrt{\alpha_S}\,(u_{cr} - u_{face})}\frac{e^{-\alpha_L\lambda^2/\alpha_s}}{\left(1 - erf(\lambda\sqrt{\alpha_L/\alpha_s})\right)} = \frac{\lambda H}{c_L}\frac{\sqrt{\pi}}{(u_{face} - u_{cr})} \ . \tag{4.27}$$

The total heat transfer into the system by the time t can be calculated by:

$$Q_T(t) = \int_0^t q(t)\,dt = \frac{2k_L\left(u_{face} - u_{cr}\right)\sqrt{t}}{\sqrt{(\alpha_L\pi)}\,erf\,\lambda} \ . \tag{4.28}$$

The total heat latent through the melting process:

$$Q_{La}(t) = \rho_{La} H X(t) \ . \tag{4.29}$$

Then the sensible heat is the difference between the total heat input and the latent heat

$$Q_{Se}(t) = Q_T(t) - Q_{La}(t) \ . \tag{4.30}$$

4.5.3. Geometry and Mesh Generation

In the current work, I have conducted multiple geometry studies focused on thermal analysis. The primary structural elements of the buildings under investigation predominantly consist of brick walls and concrete roofs, or both concrete walls and roofs in the case of precast construction. Additionally, I have integrated a PCM layer on the exterior surfaces of both the walls and roofs to enhance their thermal properties, as shown in Figure 4.46.

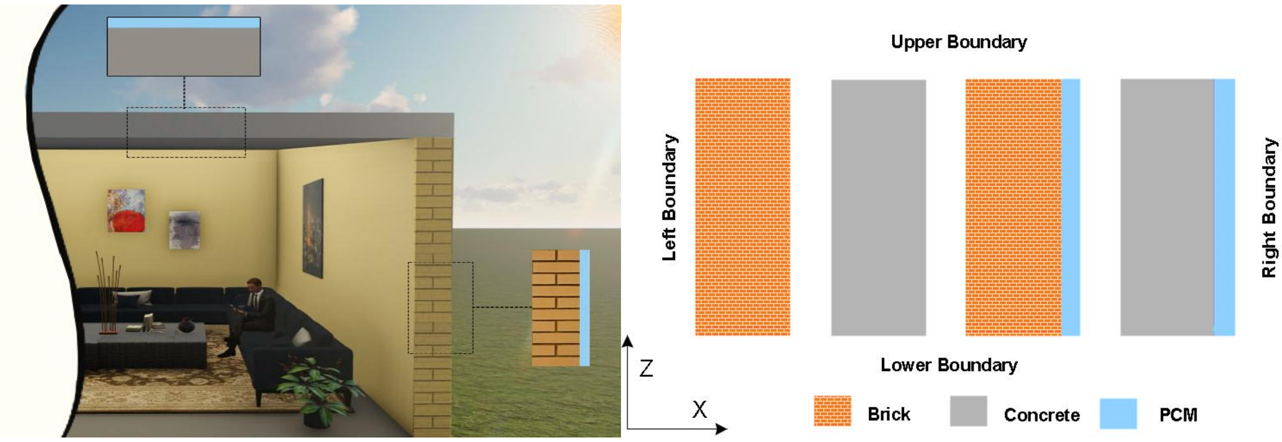

Figure 4.46. The selected section geometries of studies [77].

The value of mesh spacing will be $\Delta x = 3.334 \times 10^{-4}$ in all cases. It has been selected based on the mesh dependency study, whose details will be provided in Section 4.5.5.1.2. For brick and concrete, I consider a wall segment with a cross-sectional area $Area = L_X \times L_Z$ with value 0.2 m $\times$0.1m. I set $(x,z) \in [0, 0.2] \times [0, 0.1]$, thus the mesh's total area is 0.02m^2. The number of cells along the x and z axes are set to N_x = 600 and N_z = 1, thus, I have a mesh with a total cell number $N = N_x N_z = 600$. For the other cases, I consider a wall segment with a cross-sectional area $Area = L_X \times L_Z$ with value 0.25m $\times$0.1 m, where 0.2 is the wall thickness and 0.05 is the PCM layer thickness. $(x,z) \in [0, 0.25] \times [0, 0.1]$. Thus, the mesh's total area is 0.025m^2. I have constructed an equidistant grid for all cases. The number of cells along the x and z axes are set to N_x = 750 and N_z = 1, thus, I have a mesh with a total cell number $N = N_x N_z = 750$.

4.5.4. *Materials and Boundary Conditions*

Table 4.11 displays the material properties utilized in my current study, which primarily consist of structural materials like brick and concrete, with properties shown in Table 4.12. Meanwhile, I used two types of PCM, and both of them are a kind of Paraffin wax. The first one is PCM1 [79] with properties shown in Table 4.10 and a melting temperature of 309.7 K, and the second one is PCM2 [4] with properties shown in Table 4.12 and a melting temperature of 313 K, and with standard deviation σ=1 for both kinds of PCMs. I chose these two kinds of PCM due to the high environmental temperature outside, which requires the PCMs to have a high melting temperature (the time of temperature exposure range is high) and high latent heat properties.

Table 4.11. The Structural Materials Properties[77] .

Material	$\rho \left(\mathrm{kg \cdot m^{-3}}\right)$	$k \left(\mathrm{W \cdot m^{-1} \cdot K^{-1}}\right)$	$c \left(\mathrm{J \cdot kg^{-1} \cdot K^{-1}}\right)$
Brick	1600	0.73	800
Concrete	2300	1.70	840

Table 4.12 The Paraffin wax PCM2 with following Properties [77].

Material	$\rho \left(\mathrm{kg \cdot m^{-3}}\right)$	$k \left(\mathrm{W \cdot m^{-1} \cdot K^{-1}}\right)$	$c \left(\mathrm{J \cdot kg^{-1} \cdot K^{-1}}\right)$	*Latent Heat* $\left(\mathrm{J \cdot kg^{-1}}\right)$
PCM$_{Solid}$	830	0.48	2210	190000
PCM$_{Liquid}$	878	0.22	2300	

I used a linear relation to calculate the initial temperature by applying the recorded temperature at each one-hour taken by a weather-forecast website [80], for Basra-Iraq city on the 25 of August and on the 25 of September shown in Table 4.13 and Table 4.14 on different days (day1 is Case 1, and day2 is Case 2). I used linear relation of temperature changing with time to get the temperature distribution matrix:

$$u(x, z, t = 0) = Matrix\ values\ .$$

I applied different BCs on different sides. On the upper and lower sides, I applied Neumann boundary conditions in all cases:

$$u_z(x, z = 0, t) = u_z(x, z = L_z, t) = 0$$

On the left side I applied Dirichlet BC with constant temperature which represented the interior comfort temperature with value equal to 298 K:

$$u_x(x=0,z,t)=298$$

On the right side I applied Dirichlet BC, by applying the same recorded temperature that I used to calculate the initial temperature to get the right boundary temperature distribution array:

$$u_x(x=L_x,z,t)=Array\ values$$

Table 4.13 The Right Boundary Temperatures in Case1.

Time/h	1	2	3	4	5	6	7	8	9	10	11	12
$u_{Right/K}$	309	308	307	306	305	304	303	305	307	310	311	313
Time/h	13	14	15	16	17	18	19	20	12	22	23	24
$u_{Right/K}$	314	316	317	316	316	314	313	311	310	309	308	308

Table 4.14 The Right Boundary Temperatures in Case 2.

Time/h	1	2	3	4	5	6	7	8	9	10	11	12
u_{Right}/K	302	300	298	297	296	297	299	304	308	313	315	316
Time/h	13	14	15	16	17	18	19	20	12	22	23	24
u_{Right}/K	317	318	318	317	316	313	311	309	308	307	306	305

So, with these two cases and different scenario I simulated the wall section with 12 subcases, as shown in Table 4.15.

Table 4.15. Subcases for each Main Case (day)

Left Dirichlet Boundary
Brick
Concrete
Brick+PCM1
Concrete+PCM1
Brick+PCM2
Concrete+PCM2

4.5.5. The Results of the Current Study

4.5.5.1. The Numerical Methods Verification with Two Steps for PCM

4.5.5.1.1. First Step of Verification

In the initial verification step, I validated the numerical methods by employing the analytical solution given in Eqs. (4.24), (4.25), and (4.26). To use the analytical solution, one needs to solve the complicated transcendental equation (4.27). To enable myself to change the parameters, I first analytically reproduced that solution, but it could be done by some small error. This error, i.e. the difference between the literature and my analytical values, is much smaller than the difference between the exact analytical and approximate solution used in the literature [79], as one can see in Table 4.16.

The system considered was one-dimensional, with the height of the geometry as well as the corresponding space step set to unity. The key parameters for the solution were as follows:

$$L_x = 1,\ L_z = 1,\ N_x = 3000,\ N_z = 1,\ N = 3000,\ \Delta x = 0.00033,\ \Delta z = 1,\ u_0 = 294,\ u_{cr} = 309.7,\ u_{face} = 368$$

$$\alpha_L = 9.59\times10^{-8}, \alpha_s = 7.92\times10^{-8},\ t_0 = 0,\ t_{fin} = 3600, \Delta t = 0.036\,\mathrm{s}$$

The results I got by solving equations (4.24), (4.27), (4.28), (4.29), and (4.30) are shown in Table 4.16, while the results for the verification of the numerical methods are shown in Table 4.17.

Table 4.16. The results of verification .

Parameters	**Present Analytical values**	**Present Numerical values**	**Literature Analytical Values** [11]	**Literature Approximate Values [8]**
λ	0.4037	-	0.4033	0.42
$X(t)$ m	1.5×10^{-2}	1.5×10^{-2}	1.5×10^{-2}	1.56 ×10^{-2}
$Q_T(KJ/m^2)$	4426.6	4459.7	4446	4252
$Q_{La}(KJ/m^2)$	2882.6	2977	2883	2998
$Q_{Se}(KJ/m^2)$	1544	1482.6	1563	1254

Table 4.17. The results of the verification of the numerical methods.

Parameters	**Explicit**	**ASH**	**SH**	**LH**
$X(t)$ m	1.5×10^{-2}	1.5×10^{-2}	1.5 ×10^{-2}	1.5 ×10^{-2}
$Q_T(KJ/m^2)$	4459.701	4459.704	4459.706	4459.708
$Q_{La}(KJ/m^2)$	2977.010	2977.012	2977.013	2977.014
$Q_{Se}(KJ/m^2)$	1482.690	1482.692	1482.693	1482.694
MaxError	1.0580	1.0576	1.0579	1.0582

I calculated the maximum error (maximum absolute temperature differences along the *x*-axis between the analytical reference solution and the numerical solution) depending on the time step size. As shown in the table above, the current values are close enough to the literature values, which means I successfully verified the numerical methods based on the literature. The results of the four numerical algorithms are very close to one another, thus the deviation from the analytical values are mostly the consequence of the discretization and the EHC model.

The errors are presented in Figure 4.47. It is evident that all numerical methods exhibit an acceptable accuracy in handling PCM scenarios. This outcome instills confidence in my ability to address similar heat-related challenges in future endeavors [81].

In Figure 4.48, temperature values along the *x-axis* are depicted with a focused view on the transient phase zone for both the analytical result and the numerical methods (Explicit, ASH, SH, and LH). Remarkably, the values align closely, with differences seldom exceeding 1 degree.

Additionally, Figure 4.49 illustrates the EHC values in conjunction with temperature (on the left side) and along the *x-axis* (on the right side) for the numerical solution, where σ is set to 1 and the melting temperature is 309.7 K.

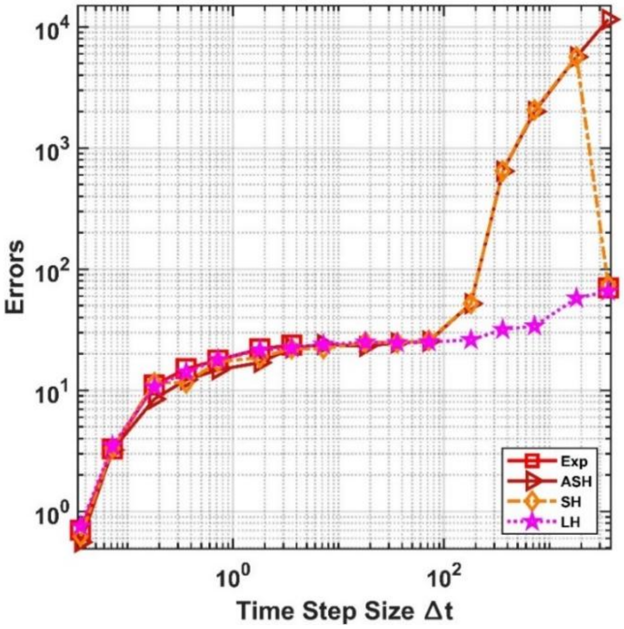

Figure 4.47. The maximum error as a time step function for numerical methods Explicit, ASH, SH and LH [77] .

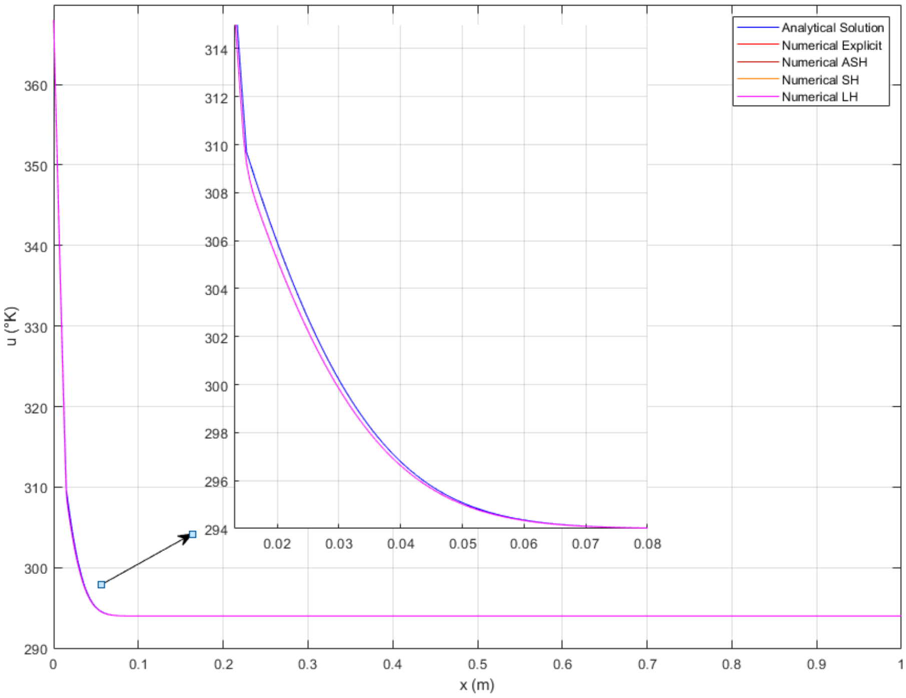

Figure 4.48. The values of temperature *u* along *x-axis* in case of the analytical solution and the numerical methods (Explicit, ASH, SH, and LH) [77].

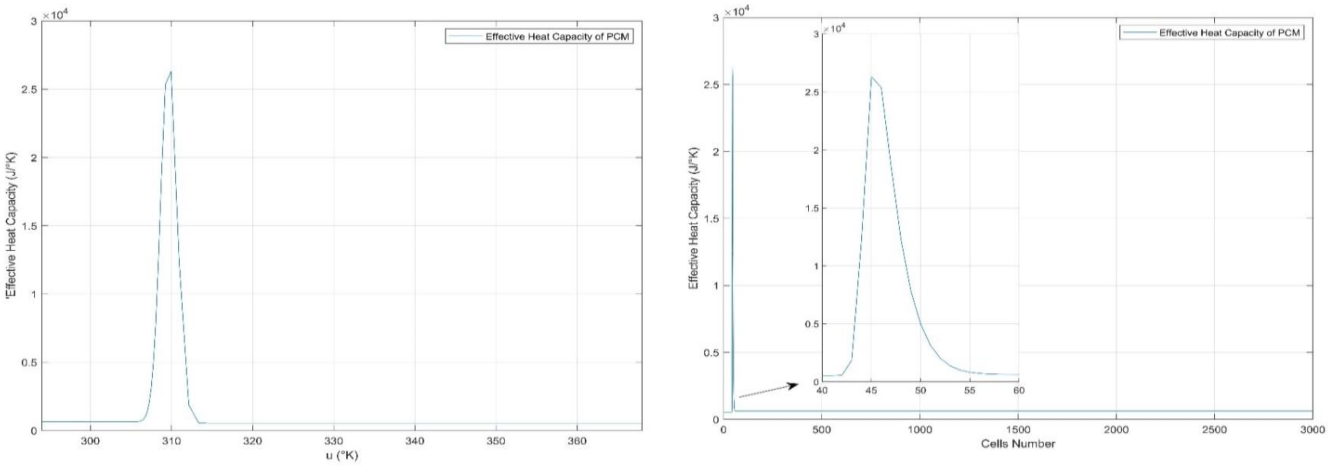

Figure 4.49. The *EHC* for PCM slab plotted with the final temperature *u* (left) and with *Cell Number* (right) [77].

4.5.5.1.2. Second Step of Verification

In the second phase of verification, the explicit method was employed as a reference solution, serving as a benchmark to evaluate other numerical methods. This comprehensive assessment aimed to measure mesh dependency, time dependency, and validate the applicability of a new PCM. Mesh dependency was scrutinized to understand the impact of mesh size on result accuracy. Following extensive analysis, an optimal mesh size of 3000 elements was identified, as demonstrated in Figure 4.50, and was subsequently applied across all study cases.

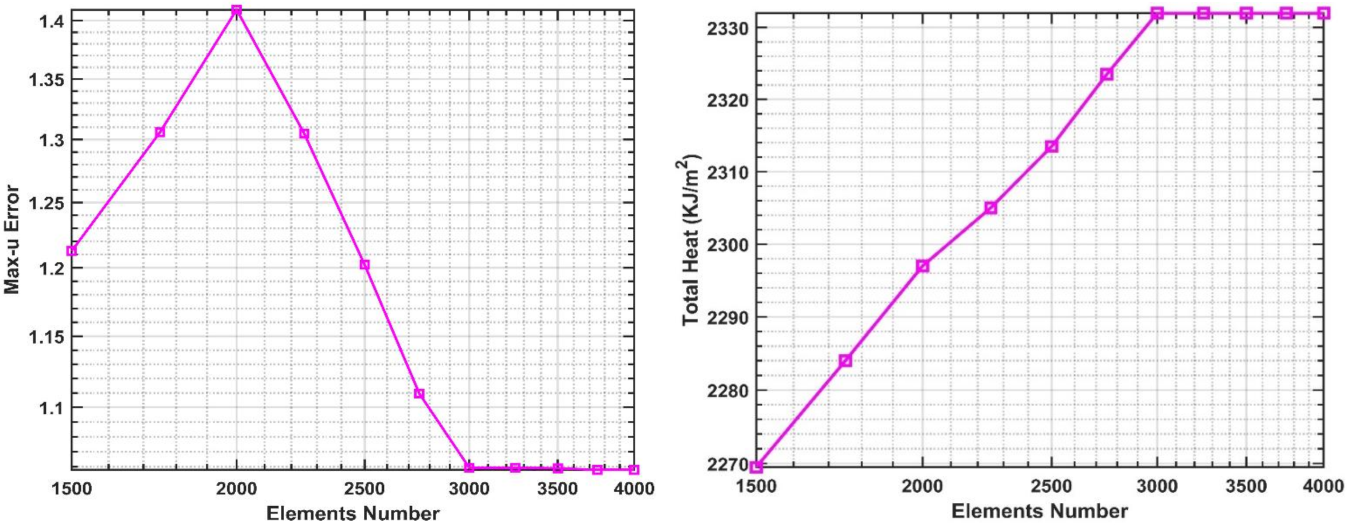

Figure 4.50. The mesh dependency examination, number of mesh elements with max error (Left), and with total penetrated heat (Right) [77].

Regarding time discretization dependency, meticulous analysis was conducted to select an appropriate time step size to meet stringent engineering precision standards, which corresponds to errors less than 10^{-2}. As illustrated in Figure 4.51 (Left), a time step size of Δt=0.86s is enough and it is implemented across all ongoing study cases.

In addition, a secondary verification step was undertaken for the new PCM material within the same computational framework, conditions and system scales. Remarkably, Figure 4.51 (Right) illustrates the maximum error over time step size, displaying striking similarities to curves associated with the other PCM material depicted in Figure 4.51 (Left). This observation underscores the versatility of the numerical methods within my updated framework, making them highly effective for various types of PCM materials, irrespective of their specific properties or the boundary conditions.

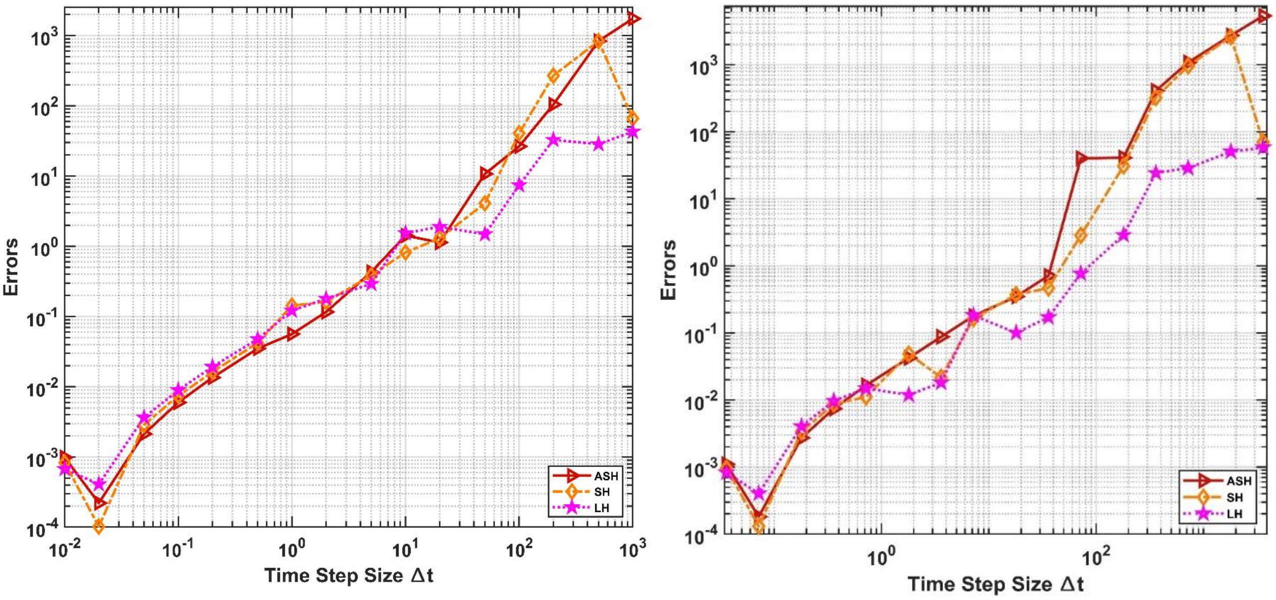

Figure 4.51. The maximum error as a function of time step size of three numerical methods for PCM1 (Left), and for PCM2 (Right) [77].

4.5.5.2. The Simulation Results

In this section, I present the results of my study in terms of total heat, heat storage, and heat transfer from the outside to the inside. These factors signify the cooling load or the amount of heat that needs to be removed using electric devices or other methods to maintain the interior environment at a comfortable zone temperature, set at 298 K. Figures 4.52-4.60 displays the results of Case 1 and Case 2 using two types of PCMs. In Figure 4.52 and 4.53 I provide samples of Effective Heat Capacity (EHC), Latent Heat Capacity (LHC), and Sensible Heat Capacity (SHC) plotted along the *x*-axis for the comparison of walls made of brick and PCMs. These graphs illustrate the behavior of melting and the storage heat hump. Similar results were obtained for concrete and PCMs. Due to the consistency in behavior across various cases, additional figures are unnecessary. Instead, I have compiled the data, including total, latent, and sensible heat values, in Tables 4.18-4.21 for reference and further analysis.

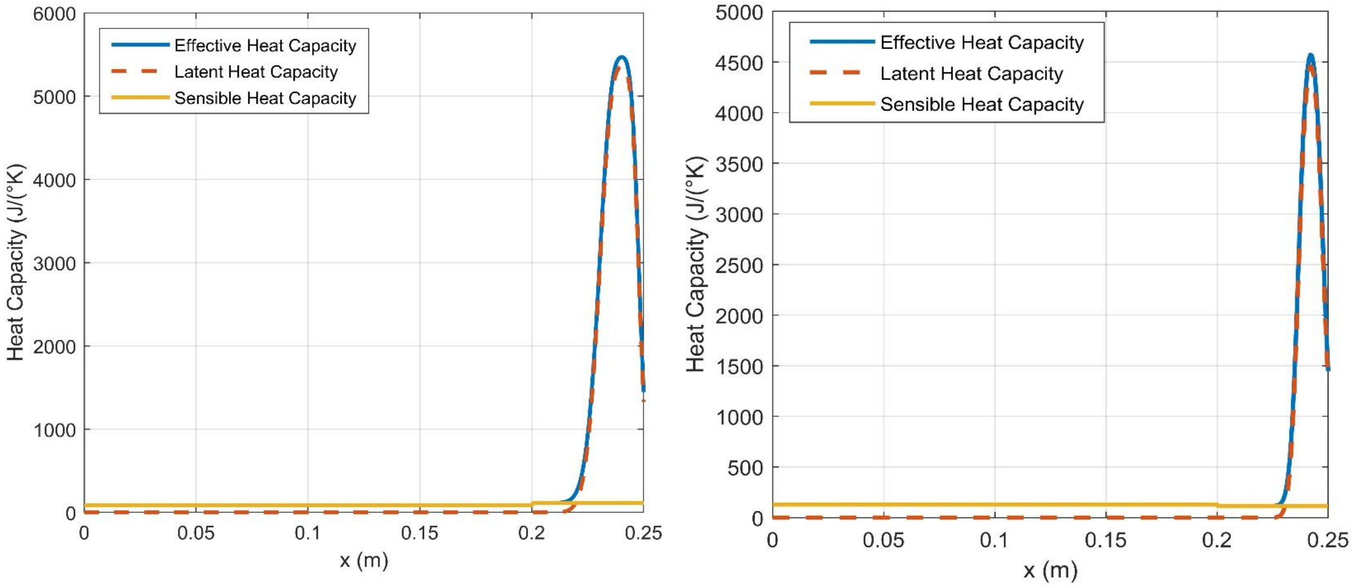

Figure 4.52. The heat capacity in Case1 for Brick+PCM1 (Left), Concrete +PCM1 (Right) [77].

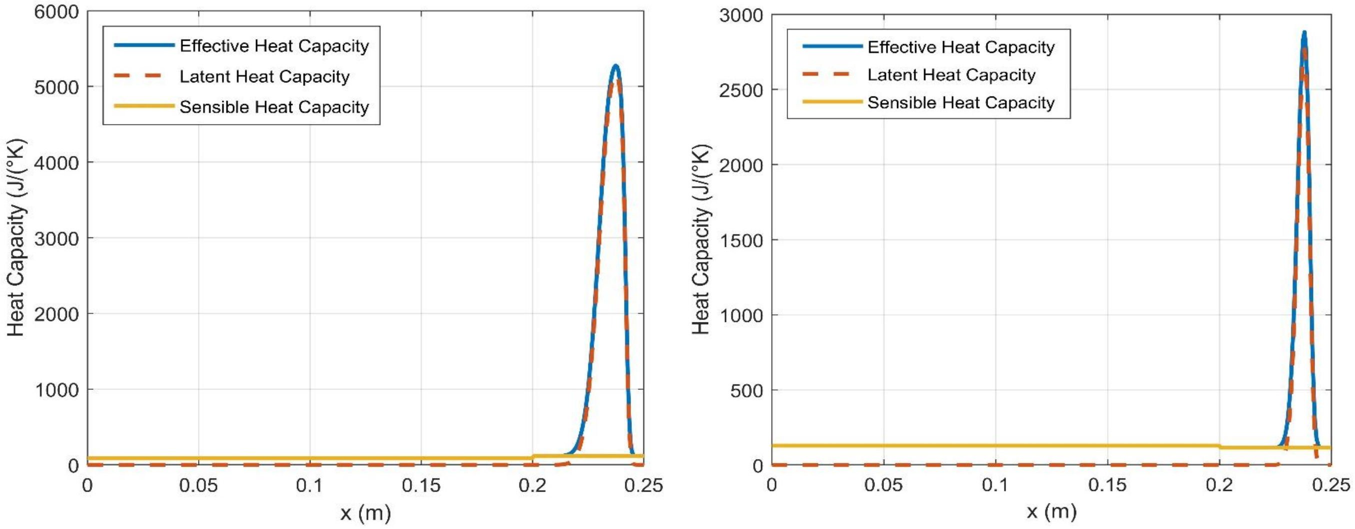

Figure 4.53. The heat capacity in Case of Brick+PCM2 (Left), Concrete +PCM2 (Right) [77].

Figure 4.54 illustrates samples of the effective heat capacity history plotted at selected points through the PCM (*x*=0.246, 0.233, and 0.2166 m) to allow the reader understanding the mechanism of heat saving during the phase transition which considering as a latent heat.

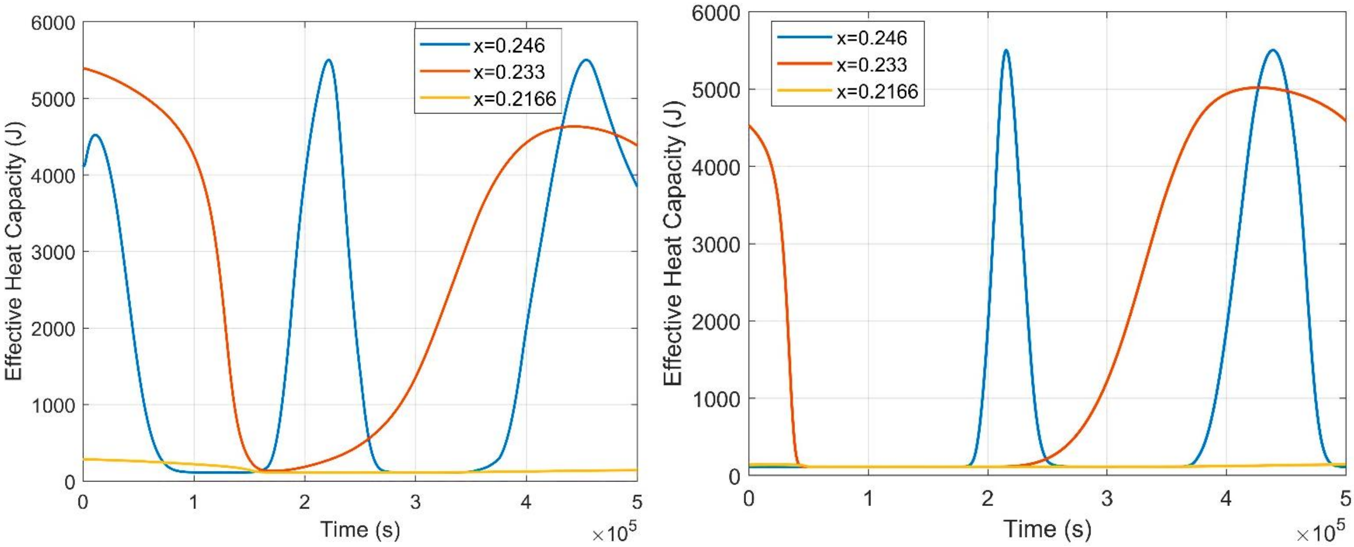

Figure 4.54. The Effective Heat Capacity history through PCM for Brick+ PCM1 in Case1 (Left), and Case2 (Right) [77].

Figures 4.55 and 4.56 illustrate the temperature history profiles at the middle of brick part of the wall. I can notice that the construction wall without PCM the temperature profile follows the outdoor temperature profile (applied boundary) with a small-time delay which represent the time of heat transfer against place. The utilization of PCM1 demonstrates a remarkable effect in maintaining the interior temperature close to comfort zone temperature and the initial values (308 K and 302 K) [82]. This indicates that a significant portion of the heat originating from the outside is efficiently stored inside PCM1 in the form of latent heat. Conversely, in the case of PCM2, the storage of heat is not as efficient due to its higher melting temperature (313 K) compared to the maximum applied temperature (317 K). Additionally, the latent heat capacity of PCM2 is considerably lower than that of PCM1. Consequently, a portion of the heat from the outside transfers indoors, leading to a noticeable impact on the indoor temperature.

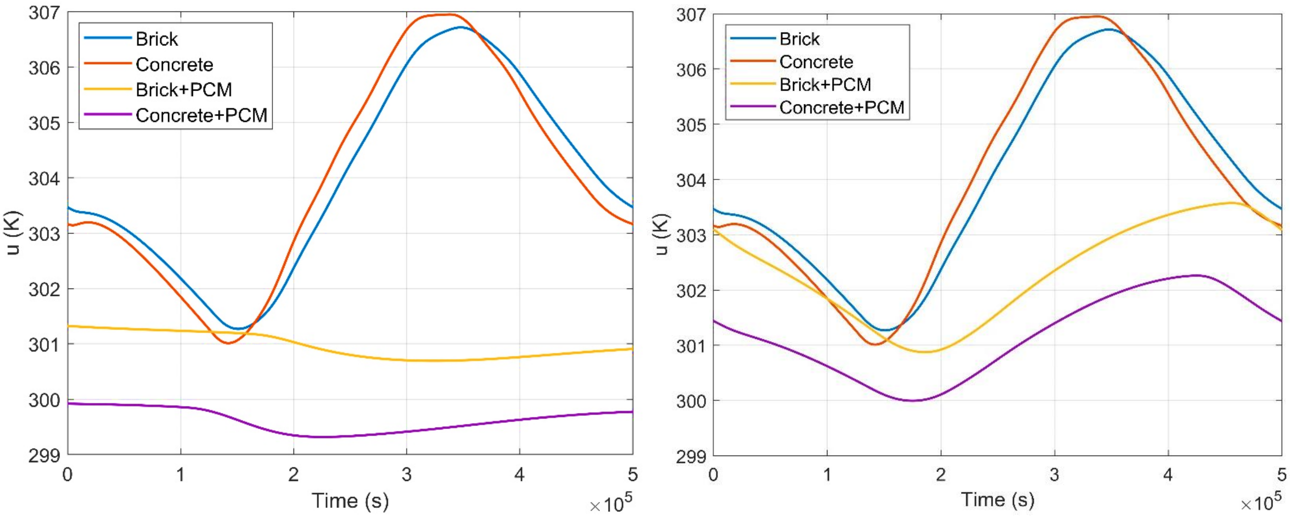

Figure 4.55. The temperature history in the middle of brick or concrete part in Case1 with PCM1 (Left), PCM2 (Right) [77].

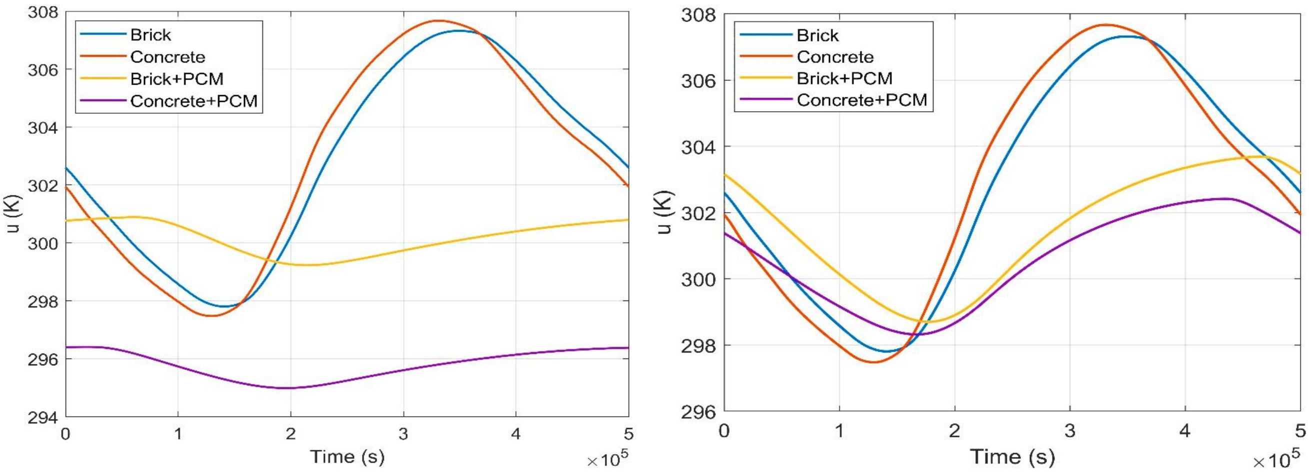

Figure 4.56. The temperature history in the middle of brick or concrete part in Case2 with PCM1 (Left), PCM2 (Right) [77].

Figures 4.57 and 4.58 depict the instantaneous total heat transfer observed throughout the duration of the study for both cases involving PCMs. These figures highlight a significant disparity in heat transfer between concrete and brick. Notably, the majority of this energy is directed inward, contributing to the interior environment. The influence of PCM usage on energy storage during the melting process is evident. However, it is essential to note that this increase in heat transfer does not imply that all of this energy directly infiltrates the interior space. A substantial portion of this heat is retained within the PCM during the transition phase, primarily in the form of latent heat.

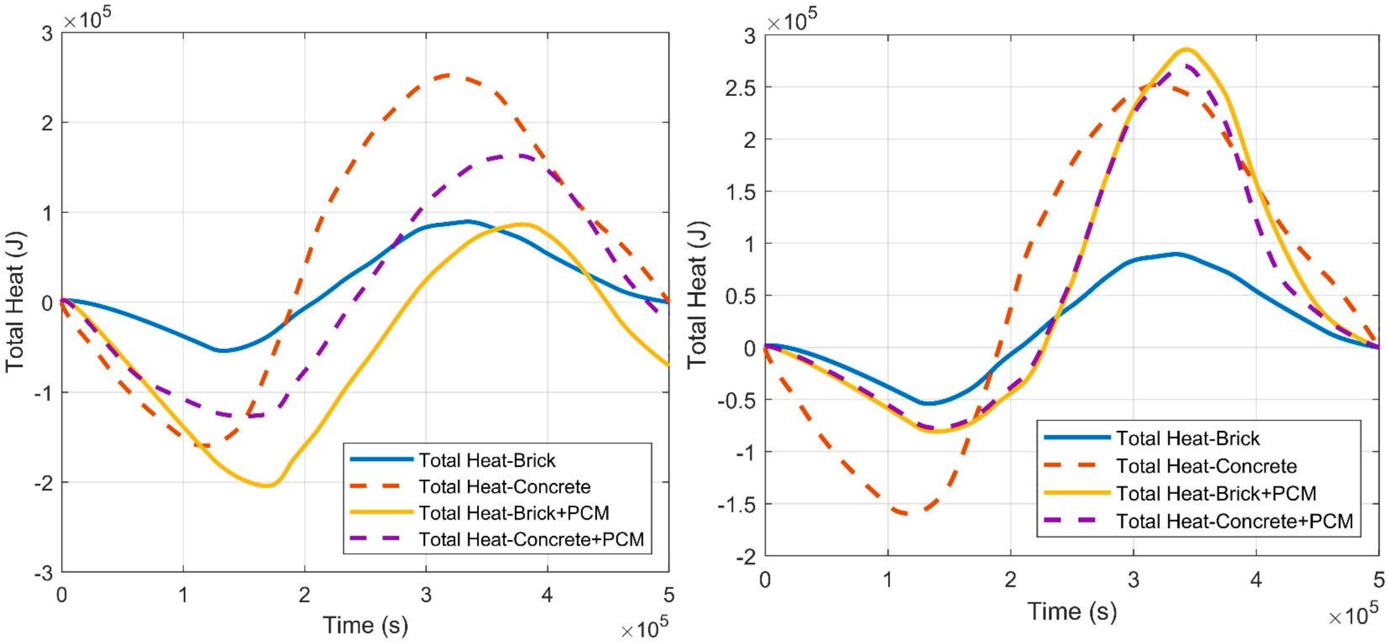

Figure 4.57. The total heat content in Case 1 with PCM1 (Left), PCM2 (Right) [77].

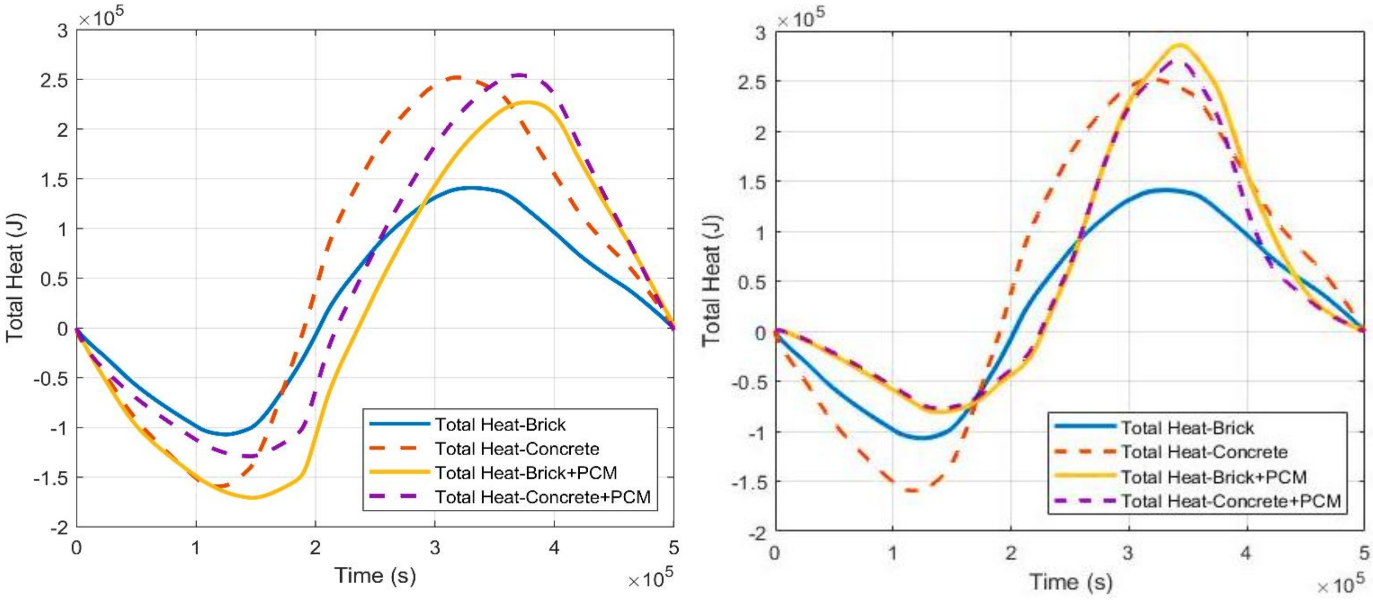

Figure 4.58. The total heat content in Case 2 with PCM1 (Left), PCM2 (Right) [77].

Figures 4.59 and 4.60 provide a comprehensive overview of the cooling load, representing the heat transfer from the outdoor environment to the indoor space across the wall structure. In Figure 4.59 (left), the significant impact of using PCM1 in conjunction with brick or concrete is evident. PCM1 efficiently stores the heat from the outside, preventing it from infiltrating the interior space. In contrast, Figure 4.59 (right) illustrates that PCM2 does not store as much heat due to its higher melting temperature, leading to a comparatively lower heat retention. Upon examining the values in the tables, it becomes apparent that the use of PCMs reduces the heat transfer to the interior, with the extent of reduction varying from total to partial. This reduction is contingent upon factors such as environmental temperature, melting temperature, and the latent heat properties of the PCM materials. For instance, Tables (4.18-4.21) list the concrete values of the results' parameters for all cases. I observed that integrating PCMs (PCMs) into construction walls significantly reduces the heat flow from outside to inside. PCM1, in particular, greatly decreases the interior heat flow due to its high latent heat capacity and appropriate melting temperature range, allowing it to melt and efficiently store energy. Additionally, it is noted that the cooling heat transfer values for both brick and concrete integrated with PCM are approximately halved. This demonstrates the effectiveness of PCM in storing most of the heat energy as latent heat, thereby minimizing heat transfer into the indoor environment.

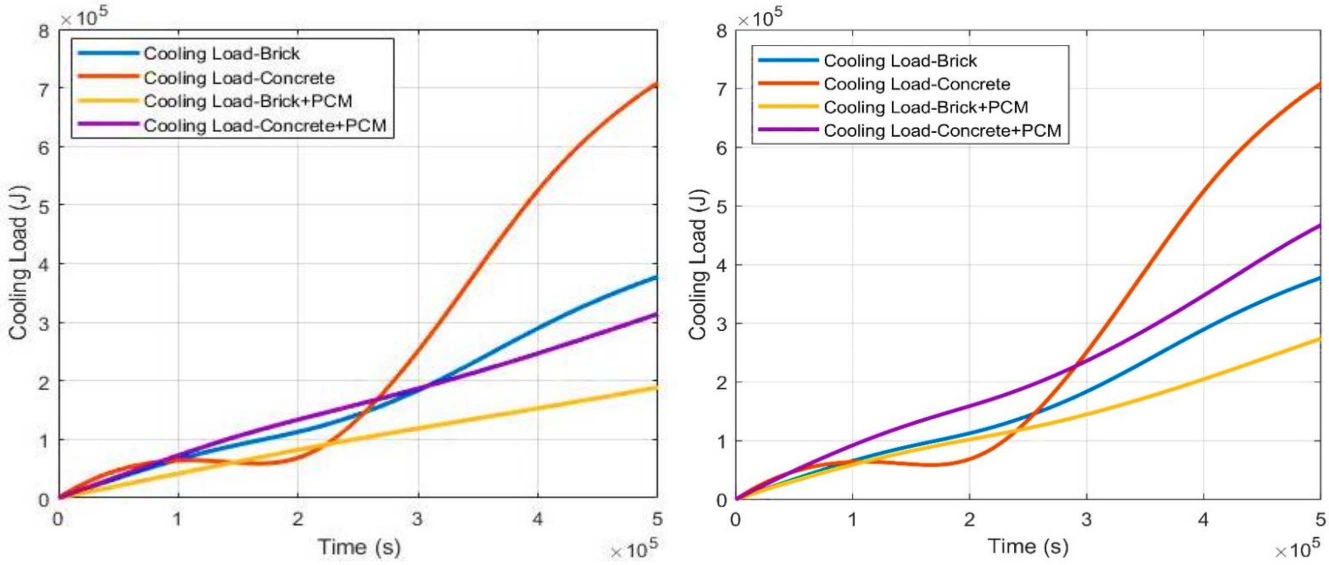

Figure 4.59. Cooling load in Case 1 with PCM1 (Left), PCM2 (Right) [77].

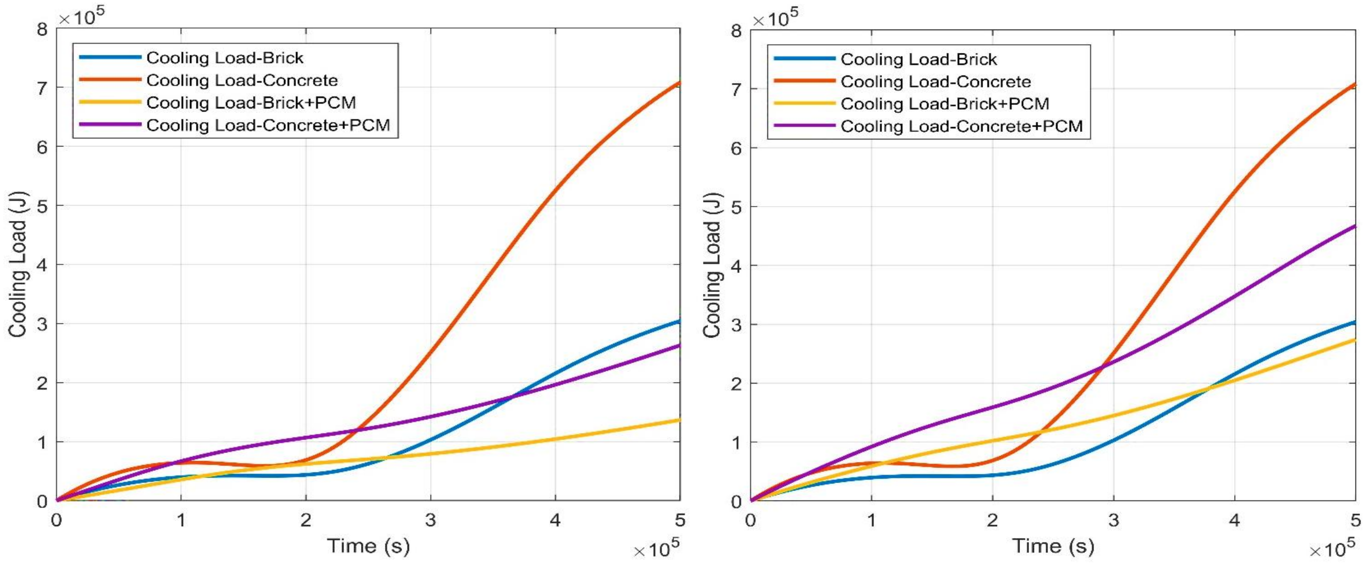

Figure 4.60. Cooling load in Case 2 with PCM1 (Left), PCM2 (Right) [77]

Table 4.18. The results in Case 1 with PCM1

Parameters	Brick	Concrete	Brick+PCM1	Concrete+PCM1
$\max(Q_{Total})\ kJ/m^2$	89.509×10^3	150.07×10^3	204.41×10^3	159.42×10^3
$mean(Q_{Total})\ kJ/m^2$	39.245×10^3	64.070×10^3	84.962×10^3	87.624×10^3
$\max(Q_{Sensible})\ kJ/m^2$	89.509×10^3	251.80×10^3	24.668×10^3	9.7919×10^3
$mean(Q_{Sensible})\ kJ/m^2$	39.245×10^3	131.98×10^3	14.752×10^3	6.8909×10^3
$\max(Q_{Latent})\ kJ/m^2$	0	0	182.59×10^3	149.63×10^3
$mean(Q_{Latent})\ kJ/m^2$	0	0	80.930×10^3	80.734×10^3
$\max(Q_{Cooling})\ J/m^2$	377.78×10^3	879.75×10^3	188.70×10^3	239.28×10^3
$mean(Q_{Cooling})\ J/m^2$	168.26×10^3	391.83×10^3	98.121×10^3	123.19×10^3

Table 4.19. The results in Case 1 with PCM2

Parameters	Brick	Concrete	Brick+PCM2	Concrete+PCM2
$\max(Q_{Total})\ kJ/m^2$	89.509×10^3	150.07×10^3	285.77×10^3	270.17×10^3
$mean(Q_{Total})\ kJ/m^2$	39.245×10^3	64.070×10^3	97.638×10^3	89.696×10^3
$\max(Q_{Sensible})\ kJ/m^2$	89.509×10^3	251.80×10^3	80.829×10^3	77.298×10^3
$mean(Q_{Sensible})\ kJ/m^2$	39.245×10^3	131.98×10^3	38.437×10^3	42.473×10^3
$\max(Q_{Latent})\ kJ/m^2$	0	0	239.81×10^3	204.90×10^3
$mean(Q_{Latent})\ kJ/m^2$	0	0	60.884×10^3	48.082×10^3
$\max(Q_{Cooling})\ J/m^2$	377.78×10^3	879.75×10^3	274.02×10^3	467.4×10^3
$mean(Q_{Cooling})\ J/m^2$	168.26×10^3	391.83×10^3	129.64×10^3	213.78×10^3

Table 4.20. The results in Case 2 with PCM1

Parameters	Brick	Concrete	Brick+PCM1	Concrete+PCM1
$\max(Q_{Total})\ kJ/m^2$	140.83×10^3	251.80×10^3	226.89×10^3	254.19×10^3
$mean(Q_{Total})\ kJ/m^2$	77.542×10^3	131.98×10^3	125.23×10^3	123.75×10^3
$\max(Q_{Sensible})\ kJ/m^2$	140.83×10^3	251.80×10^3	95.532×10^3	113.35×10^3
$mean(Q_{Sensible})\ kJ/m^2$	77.542×10^3	131.98×10^3	32.880×10^3	42.238×10^3
$\max(Q_{Latent})\ kJ/m^2$	0	0	229.94×10^3	255.16×10^3
$mean(Q_{Latent})\ kJ/m^2$	0	0	105.85×10^3	100.61×10^3
$\max(Q_{Cooling})\ J/m^2$	304.19×10^3	708.39×10^3	136.57×10^3	263.04×10^3
$mean(Q_{Cooling})\ J/m^2$	110.86×10^3	252.76×10^3	69.906×10^3	128.71×10^3

Table 4.21. The results in Case 2 with PCM2

Parameters	Brick	Concrete	Brick+PCM2	Concrete+PCM2
$\max(Q_{Total})\ kJ/m^2$	89.509×10^3	251.80×10^3	381.24×10^3	365.72×10^3
$mean(Q_{Total})\ kJ/m^2$	39.245×10^3	131.98×10^3	155.68×10^3	148.85×10^3
$\max(Q_{Sensible})\ kJ/m^2$	89.509×10^3	251.80×10^3	76.041×10^3	156.69×10^3
$mean(Q_{Sensible})\ kJ/m^2$	39.245×10^3	131.98×10^3	76.041×10^3	83.155×10^3
$\max(Q_{Latent})\ kJ/m^2$	0	0	301.16×10^3	258.61×10^3
$mean(Q_{Latent})\ kJ/m^2$	0	0	83.240×10^3	68.074×10^3
$\max(Q_{Cooling})\ J/m^2$	377.78×10^3	708.39×10^3	220.39×10^3	376.23×10^3

4.5.5.3. Computational Time of The Numerical Methods

Table 4.22 presents the computational time for recent numerical methods applied to two distinct geometries: brick and brick integrated with PCM. The inclusion of only these two types of geometries stems from the fact that computational times for brick and concrete are identical, a consistency maintained even when integrated with PCMs due to their equivalent system sizes. The tabulated values indicate that LH methods demonstrate quicker computational times. Nevertheless, it becomes evident that LH methods emerge as the optimal choice, striking a balance between speed and stability across all time step sizes.

Table 4.22. The computational time of numerical methods

Geometry	Computational time (s)		
	ASH	SH	LH
Brick or Concrete	29.1569	24.2810	19.7127
Brick or Concrete+PCM	549.5274	451.4683	373.3887

4.5.6. The Summary of The Present Section

The present work summarizes the following:

1. The novelty of recent numerical methods in effectively addressing the complexities associated with phase change,establishes the Effective Heat Capacity model, serving as a computational tool for simulating PCMs.
2. Emphasizing the substantial impact of PCMs on cooling loads and heat transfer dynamics between outdoor and indoor environments, particularly in diverse wall structures, the findings underscore the crucial role of PCMs in energy management.
3. Performance Disparities between PCM1 and PCM2: PCM1, distinguished by its lower melting temperature and higher latent heat, excels in proficiently storing external heat, thereby preventing its ingress into indoor spaces. In contrast, PCM2, characterized by higher melting temperature and lower latent heat, exhibits diminished efficiency in heat retention. PCM1, notably, achieves a significant reduction in heat transfer into interior spaces, approaching near-elimination due to its high latent heat and appropriate melting temperature range.
4. Impact on Interior Temperature Regulation: PCM1, especially when coupled with brick or concrete, sustains indoor temperatures near initial values, exemplifying its adeptness in efficient heat storage. Conversely, PCM2, while providing insulation, exerts a comparatively lesser influence on indoor thermal conditions.

In conclusion, this study accentuates the pivotal role of PCMs in mitigating cooling loads, preserving indoor temperatures, and impeding external heat intrusion. The judicious selection of PCMs, influenced by latent heat properties and melting temperature considerations, emerges as a critical factor in optimizing energy efficiency and elevating thermal comfort within architectural frameworks. For climates characterized by cold conditions, PCM2 is recommended, particularly when augmented by solar panels to harness and store daytime solar energy as latent heat for nocturnal cold periods. These discernments bear substantive implications for the formulation of energy-efficient structures, underscoring the strategic importance of PCM selection in building materials and construction methodologies.

5.THESIS POINTS – NEW SCIENTIFIC RESULTS

T1. I constructed and tested the Shifted-Hopscotch algorithms, which were fully explicit time-integrators obtained by applying half-time steps and full-time steps in the odd-even hopscotch structure. I applied the conventional theta method with 9 different values, and the non-conventional CNe method to construct 10^5 combinations and I chose the top five of them via numerical experiments. These experiments suggest that the proposed methods are, indeed, competitive, as they can give fairly accurate results orders of magnitude faster than the well-optimized MATLAB routines or the Crank–Nicolson method, and they are also significantly more accurate for stiff systems than the UPFD, the Dufort–Frankel, or the original odd-even hopscotch method. If high accuracy is required, the S4 (0, ½, ½, ½, 1) combination can be proposed; however, when preserving positivity is crucial, the S1 (C, C, C, C, C) algorithm should be used [65].

T2. To demonstrate the practical utility of these advanced numerical techniques, I investigated 13 algorithms to solve the problem of linear heat conduction in building walls. These included eight explicit, unconditionally stable algorithms invented by our research group, such as the Shifted-Hopscotch (SH) scheme. The validation process, where numerical results were compared against analytical solutions using both uniform and non-uniform spatial discretizations, was carried out as a teamwork. Then, I applied carefully designed nontrivial boundary conditions: spatially varying temperatures on the brick side and time-dependent temperatures on the outer surface of the insulation. I found that the classic Odd-Even Hopscotch (OEH) method delivers superior accuracy for homogeneous scenarios, while the Leapfrog-Hopscotch (LH) algorithm performs best in non-uniform configurations. Nevertheless, the Shifted-Hopscotch (SH) method also exhibited strong competitiveness across all test cases [68].

T3. I also examined 11 of the new methods to solve heat conduction, convection, radiation, and heat generation inside building walls' elements. These methods were tested on real-life applications involving surface area (one-layer brick) and cross-sectional area (two-layer brick and insulator) walls, with and without thermal bridging, to determine accuracy dependence on material properties, mesh type, and time step size. Neumann boundary conditions were applied to all boundaries, for surface area cases, the heat source, convection, and radiation inside all elements were considered, while for cross-sectional area cases only the right and left boundary elements containing heat source, convection, and radiation. The results indicate that the Original Odd-Even Hopscotch method is usually the best for uniform cases, while the Leapfrog-Hopscotch algorithm performs best for non-uniform cases [60].

T4. In addition to Cartesian coordinates, I developed 9 of the new methods to solve heat transfer problems in cylindrical and spherical geometries. I reproduced novel and nontrivial analytical solutions for the heat-conduction PDE with high accuracy. Furthermore, I verified the numerical methods in cylindrical and spherical coordinates, incorporating convection and radiation terms, by reproducing real experimental data of a heated cylinder and comparing it with Finite Element Methods (FEM) ANSYS workbench. Convection and nonlinear radiation were considered on the boundary of the cylinder. Verification results demonstrated the high accuracy of the numerical methods in dealing with cylindrical and spherical bodies. Additionally, temperature comparisons across all approaches revealed that explicit methods are more accurate than finite element software in all cases, with the Leapfrog-Hopscotch algorithm typically being the most accurate among the studied methods [42].

T5. I investigated the heat transfer through building walls, considering different wall geometries and heat load scenarios, encompassing both cooling and heating. My objective was to analyze how heat transfer depends on the wall materials and evaluate algorithm performance in cases involving heat transfer between solid surfaces and fluid (convection) on the outdoor surface, particularly across an air gap between the insulation and Photovoltaic Cells (PVC). The results of the study reveal that insulation prevents heat from entering the building, maintaining a comfortable indoor environment. Forced convection significantly enhances heat dissipation, especially during cooling operations to protect PVC with limited working temperature. Furthermore, the simulations highlight the air gap's efficiency in cooling PVC and reducing maximum temperatures on the insulation's outer surface, especially under forced convection conditions. The test results show that the Leapfrog Hopscotch algorithm offers the best solution for this highly stiff system, followed by the Asymmetric and Shifted-Hopscotch algorithms [73].

T6. I also simulated a multilayer wall integrated with PCMs using an effective heat capacity model and I employed the Leapfrog-Hopscotch methods for that. I validated my approach against established mathematical expressions and models in the literature, investigating various building wall geometries, two types of PCMs used in this investigation, and boundary conditions. The objective was to maintain interior temperatures within comfort zones. Regardless of the wall material, whether brick or concrete, my simulations consistently demonstrated the PCM's effectiveness in minimizing heat transfer into indoor environment [77].

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. Endre Kovács, whose support, guidance, and wisdom have been invaluable throughout my study. His knowledge has been instrumental in steering my research in the right direction. I am also sincerely thankful to my co-supervisor Dr. Betti Bolló for assessing my work and providing critical feedback that greatly enhanced the quality of my research.

I owe a profound debt of gratitude to my family, especially my father and mother, for their patience and understanding while I was far from them, even when they needed me most. My sisters deserve heartfelt thanks for their constant sacrifices, which allowed me to focus on my studies.

My deepest appreciation goes to my wife, who has endured the challenges of traveling just to be with me, providing unwavering support and companionship. The biggest gratitude goes to my little angel, my daughter Sidra, whose inspiration and daily energy gave me the strength to complete this journey.

Lastly, I extend my thanks to every person who has assisted me in my study. Your help and support have been crucial to my success.

Thank you all.

REFERENCES

[1] Mojtabi, A.; Deville, M.O. One-dimensional linear advection-diffusion equation: Analytical and finite element solutions. Comput. Fluids 2015, 107, 189–195, doi:10.1016/j.compfluid.2014.11.006.

[2] Barna, I.F.; Bognár, G.; Guedda, M.; Mátyás, L.; Hriczó, K. Analytic self-similar solutions of the Kardar-Parisi-Zhang interface growing equation with various noise terms. Math. Model. Anal. 2020, 25, 241–256, doi:10.3846/mma.2020.10459.

[3] Barna, I.F.; Kersner, R. Heat conduction: A telegraph-type model with self-similar behavior of solutions. J. Phys. A Math. Theor. 2010, 43, 375210, doi:10.1088/1751-8113/43/37/375210.

[4] Mátyás, L.; Barna, I.F. General self-similar solutions of diffusion equation and related constructions. arXiv 2021, arXiv:2104.09128.

[5] Bastani, M.; Salkuyeh, D.K. A highly accurate method to solve Fisher's equation. Pramana—J. Phys. 2012, 78, 335–346, doi:10.1007/s12043-011-0243-8.

[6] Agbavon, K.M.; Appadu, A.R.; Khumalo, M. On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term. Adv. Differ. Eq. 2019, 146, doi:10.1186/s13662-019-2080-x.

[7] Zoppou, C.; Knight, J.H. Analytical solution of a spatially variable coefficient advection-diffusion equation in up to three dimensions. Appl. Math. Model. 1999, 23, 667–685, doi:10.1016/S0307-904X(99)00005-0.

[8] Lienhard, J.H., IV; Lienhard, J.H., V A Heat Transfer Textbook, 4th ed.; Phlogiston Press: Cambridge, MA, USA, 2017; ISBN 9780971383524.

[9] Cusini, M. Dynamic Multilevel Methods for Simulation of Multiphase Flow in Heterogeneous Porous Media; Delft University of Technology: Delft, The Netherlands, 2019.

[10] Appau, P.O.; Dankwa, O.K.; Brantson, E.T. A comparative study between finite difference explicit and implicit method for predicting pressure distribution in a petroleum reservoir. Int. J. Eng. Sci. Technol. 2019, 11, 23–40, doi:10.4314/ijest.v11i4.3.

[11] Moncorgé, A.; Tchelepi, H.A.; Jenny, P. Modified sequential fully implicit scheme for compositional flow simulation. J. Comput. Phys. 2017, 337, 98–115, doi:10.1016/j.jcp.2017.02.032.

[12] Chou, C.S.; Zhang, Y.T.; Zhao, R.; Nie, Q. Numerical methods for stiff reaction-diffusion systems. Discret. Contin. Dyn. Syst.—Ser. B 2007, 7, 515–525, doi:10.3934/dcdsb.2007.7.515.

[13] Gumel, A.B.; Ang, W.T.; Twizell, E.H. Efficient parallel algorithm for the two-dimensional diffusion equation subject to specification of mass. Int. J. Comput. Math. 1997, 64, 153–163, doi:10.1080/00207169708804580.

[14] Xue, G.; Feng, H. A new parallel algorithm for solving parabolic equations. Adv. Differ. Eq. 2018, 2018, 1–6, doi:10.1186/s13662-018-1617-8.

[15] Gagliardi, F.; Moreto, M.; Olivieri, M.; Valero, M. The international race towards Exascale in Europe. CCF Trans. High Perform. Comput. 2019, 3–13, doi:10.1007/s42514-019-00002-y.

[16] Reguly, I.Z.; Mudalige, G.R. Productivity, performance, and portability for computational fluid dynamics applications. Comput. Fluids 2020, 199, 104425, doi:10.1016/j.compfluid.2020.104425.

[17] Gordon, P. Nonsymmetric Difference Equations. J. Soc. Ind. Appl. Math. 1965, 13, 667–673, doi:10.1137/0113044.

[18] Gourlay, A.R. Hopscotch: A Fast Second-order Partial Differential Equation Solver. IMA J. Appl. Math. 1970, 6, 375–390.

[19] 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.

[20] Gourlay, A.R. Some recent methods for the numerical solution of time-dependent partial differential equations. Proc. R. Soc. London. A Math. Phys. Sci. 1971, 323, 219–235, doi:10.1098/rspa.1971.0099.

[21] Hundsdorfer, W.H.; Verwer, J.G. Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations; Springer: Berlin, Germany, 2003;.

[22] Morris, J.L.; Nicoll, I.F. Hopscotch methods for an anisotropic thermal print head problem. J. Comput. Phys. 1973, 13, 316–337, doi:10.1016/0021-9991(73)90039-9.

[23] ten Thije Boonkkamp, J.H.M. The Odd-Even Hopscotch Pressure Correction Scheme for the Incompressible Navier–Stokes Equations. SIAM J. Sci. Stat. Comput. 1988, 9, 252–270, doi:10.1137/0909016.

[24] Harley, C. Hopscotch method: The numerical solution of the Frank-Kamenetskii partial differential equation. Appl. Math. Comput. 2010, 217, 4065–4075, doi:10.1016/j.amc.2010.10.020.

[25] 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, doi:10.33899/csmj.2011.163605.

[26] Xu, J.; Shao, S.; Tang, H. Numerical methods for nonlinear Dirac equation. J. Comput. Phys. 2013, 245, 131–149, doi:10.1016/j.jcp.2013.03.031.

[27] de Goede, E.D.; ten Thije Boonkkamp, J.H.M. Vectorization of the Odd–Even Hopscotch Scheme and the Alternating Direction Implicit Scheme for the Two-Dimensional Burgers Equations. SIAM J. Sci. Stat. Comput. 1990, 11, 354–367, doi:10.1137/0911021.

[28] Maritim, S.; Rotich, J.K.; Bitok, J.K. Hybrid hopscotch Crank-Nicholson-Du Fort and Frankel (HP-CN-DF) method for solving two dimensional system of Burgers' equation. Appl. Math. Sci. 2018, 12, 935–949, doi:10.12988/ams.2018.8798.

[29] Maritim, S.; Rotich, J.K. Hybrid Hopscotch Method for Solving Two Dimensional System of Burgers' Equation. Int. J. Sci. Res. 2018, 8, 492–497.

[30] Saleh, M.; Nagy, Á.; Kovács, E. Construction and investigation of new numerical algorithms for the heat equation: Part 1. Multidiszcip. Tudományok 2020, 10, 323–338, doi:10.35925/j.multi.2020.4.36.

[31] Saleh, M.; Nagy, Á.; Kovács, E. Construction and investigation of new numerical algorithms for the heat equation: Part 2. Multidiszcip. Tudományok 2020, 10, 339–348, doi:10.35925/j.multi.2020.4.37.

[32] Saleh, M.; Nagy, Á.; Kovács, E. Construction and investigation of new numerical algorithms for the heat equation: Part 3. Multidiszcip. Tudományok 2020, 10, 349–360, doi:10.35925/j.multi.2020.4.38.

[33] X. Geng, J. Wang, Y. Gao, and X. Meng, “Location combination optimization of thermal insulation material and phase-change material in multi-layer walls under air-conditioning continuous and intermittent operation,” *J. Energy Storage*, vol. 44, p. 103449, Dec. 2021, doi: 10.1016/J.EST.2021.103449.

[34] Z. Liu, J. Hou, D. Wei, X. Meng, and B. J. Dewancker, “Thermal performance analysis of lightweight building walls in different directions integrated with phase change materials (PCM),” *Case Stud. Therm. Eng.*, vol. 40, p. 102536, Dec. 2022, doi: 10.1016/J.CSITE.2022.102536.

[35] E. Tunçbilek, M. Arıcı, M. Krajčík, D. Li, S. Nižetić, and A. M. Papadopoulos, “Enhancing building wall thermal performance with phase change material and insulation: A comparative and synergistic assessment,” *Renew. Energy*, vol. 218, p. 119270, Dec. 2023, doi: 10.1016/J.RENENE.2023.119270.

[36] Y. Cascone, A. Capozzoli, and M. Perino, “Optimisation analysis of PCM-enhanced opaque building envelope components for the energy retrofitting of office buildings in Mediterranean climates,” *Appl. Energy*, vol. 211, pp. 929–953, Feb. 2018, doi: 10.1016/J.APENERGY.2017.11.081.

[37] R. F. Jam, M. Gholizadeh, M. Deymi-Dashtebayaz, and E. Tayyeban, “Determining the optimal location and thickness of phase change materials in the building walls: an energy-economic analysis,” *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 45, no. 10, p. 554, 2023, doi: 10.1007/s40430-023-04472-8.

[38] M. J. Abden, Z. Tao, M. A. Alim, Z. Pan, L. George, and R. Wuhrer, “Combined use of phase change material and thermal insulation to improve energy efficiency of residential buildings,” *J. Energy Storage*, vol. 56, p. 105880, Dec. 2022, doi: 10.1016/J.EST.2022.105880.

[39] E. Iffa, D. Hun, M. Salonvaara, S. Shrestha, and M. Lapsa, “Performance evaluation of a dynamic wall integrated with active insulation and thermal energy storage systems,” *J. Energy Storage*, vol. 46, p. 103815, Feb. 2022, doi: 10.1016/J.EST.2021.103815.

[40] P. Arumugam, V. Ramalingam, and P. Vellaichamy, “Optimal positioning of phase change material and insulation through numerical investigations to reduce cooling loads in office buildings,” *J. Energy Storage*, vol. 52, p. 104946, Aug. 2022, doi: 10.1016/J.EST.2022.104946.

[41] Yunus A. Cengel and Afshin J. Ghajar, *Heat and Mass Transfer: Fundamentals & Applications*, Fifth Edition. United States of America: McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121.

[42] H. K. Jalghaf, E. Kovács, I. F. Barna, and L. Mátyás, “Analytical Solution and Numerical Simulation of Heat Transfer in Cylindrical- and Spherical-Shaped Bodies,” *Computation*, vol. 11, no. 7, Jul. 2023, doi: 10.3390/computation11070131.

[43] Savović, S.; Djordjevich, A. Numerical solution of diffusion equation describing the flow of radon through concrete. Appl. Radiat. Isot. 2008, 66, 552–555.

[44] 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. https://doi.org/10.3390/app11104468.

[45] Haq, S.; Ali, I. Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials. Eng. Comput. 2021. Available online: https://link.springer.com/article/10.1007/s00366-021-01327-5 (accessed on 13 June 2022.).

[46] Lima, S.A.; Kamrujjaman; Islam, S. Numerical solution of convection-diffusion-reaction equations by a finite element method with error correlation. AIP Adv. 2021, 11, 085225. https://doi.org/10.1063/5.0050792.

[47] Ivanovic, M.; Svicevic, M.; Savovic, S. Numerical solution of Stefan problem with variable space grid method based on mixed finite element/ finite difference approach. Int. J. Numer. Methods Heat Fluid Flow 2017, 27, 2682–2695. https://doi.org/10.1108/hff-02-2014-0034.

[48] 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.

[49] 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. https://doi.org/10.1080/00207160.2021.1954621.

[50] Aminikhah, H.; Alavi, J. An efficient B-spline difference method for solving system of nonlinear parabolic PDEs. SeMA J. 2018, 75, 335–348; ISBN4032401701398.

[51] 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. https://doi.org/10.1007/s40065-021-00330-4.

[52] 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. https://doi.org/10.2166/ws.2021.010.

[53] Haq, S.; Hussain, 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. Available online: https://link.springer.com/article/10.1007/s00366-019-00760-x (accessed on 13 June 2022.).

[54] Reguly, I.Z.; Mudalige, G.R. Productivity, performance, and portability for computational fluid dynamics applications. Comput. Fluids 2020, 199, 104425. https://doi.org/10.1016/j.compfluid.2020.104425.

[55] Gagliardi, F.; Moreto, M.; Olivieri, M.; Valero, M. The international race towards Exascale in Europe. CCF Trans. High Perform. Comput. 2019, 1, 3–13. https://doi.org/10.1007/s42514-019-00002-y.

[56] Chen-Charpentier, B.M.; and Kojouharov, H. V. "An unconditionally positivity preserving scheme for advection-diffusion reaction equations," Math. Comput. Model., vol. 57, pp. 2177–2185, 2013, doi: 10.1016/j.mcm.2011.05.005.

[57] C. Hirsch, Numerical computation of internal and external flows, volume 1: Fundamentals of numerical discretization. Wiley, 1988.

[58] G. Sottas, "Rational Runge-Kutta methods are not suitable for stiff systems of ODEs," J. Comput. Appl. Math., vol. 10, pp. 169–174, 1984.

[59] A. H. Workie, "New Modification on Heun's Method Based on Contraharmonic Mean for Solving Initial Value Problems with High Efficiency," J. Math., vol. 2020, 2020, doi: 10.1155/2020/6650855.

[60] H. K. Jalghaf, E. Kovács, and B. Bolló, "Comparison of Old and New Stable Explicit Methods for Heat Conduction, Convection, and Radiation in an Insulated Wall with Thermal Bridging," *Buildings*, vol. 12, no. 9, Sep. 2022, doi: 10.3390/buildings12091365.

[61] A. R. Gourlay and G. R. McGuire, "General Hopscotch Algorithm for the Numerical Solution of Partial Differential Equations," IMA J. Appl. Math., vol. 7, no. 2, pp. 216–227, 1971.

[62] Kovács, E. A class of new stable, explicit methods to solve the non-stationary heat equation. Numer. Methods Partial Differ. Equ. 2020, 37, 2469–2489. https://doi.org/10.1002/num.22730.

[63] Holmes, M.H. Introduction to Numerical Methods in Differential Equations; Springer: New York, NY, USA, 2007; ISBN978-0387-30891-3.

[64] 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. Available online: https://www.mdpi.com/2227-7390/9/18/2284 (accessed on 3 November 2021).

[65] Á. Nagy, M. Saleh, I. Omle, H. Kareem, and E. Kovács, "New stable, explicit, shifted-hopscotch algorithms for the heat equation," Math. Comput. Appl., vol. 26, no. 61, 2021, [Online]. Available: https://www.mdpi.com/2297-8747/26/3/61/htm

[66] Á. Nagy, I. Omle, H. Kareem, E. Kovács, I. F. Barna, and G. Bognar, "Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation," Computation, vol. 9, no. 8, p. 92, 2021.

[67] H. K. Jalghaf, E. Kovács, J. Majár, Á. Nagy, and A. H. Askar, "Explicit stable finite difference methods for diffusion-reaction type equations," Mathematics, vol. 9, no. 24, 2021, doi: 10.3390/math9243308.

[68] H. K. Jalghaf, I. Omle, and E. Kovács, "A Comparative Study of Explicit and Stable Time Integration Schemes for Heat Conduction in an Insulated Wall," Buildings, vol. 12, no. 6, Jun. 2022, doi: 10.3390/buildings12060824.

[69] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*. Cambridge: Cambridge Univ. Press, 2009. ISBN9788490225370.

[70] Cabezas, S.; Hegedűs, G.; Bencs, P. Thermal experimental and numerical heat transfer analysis of a solid cylinder in longitudinal direction. Analecta Tech. Szeged. 2023, 17, 16–27. https://doi.org/10.14232/analecta.2023.1.16-27.

[71] A. Tutueva, T. Karimov, and D. Butusov, "Semi-implicit and semi-explicit Adams-Bashforth-Moulton methods," Mathematics, vol. 8, no. 5, 2020, doi: 10.3390/MATH8050780.

[72] J. Kettle et al., "Review of technology specific degradation in crystalline silicon, cadmium telluride, copper indium gallium selenide, dye sensitised, organic and perovskite solar cells in photovoltaic modules: Understanding how reliability improvements in mature technolog," Prog. Photovoltaics Res. Appl., vol. 30, no. 12, pp. 1365–1392, Dec. 2022, doi: 10.1002/pip.3577.

[73] H. K. Jalghaf, E. Kovács, and B. Bolló, "Simulation of transient heat transfer in multilayer walls with photovoltaic cell and air by using efficient numerical methods," *Results in Engineering*, vol. 21, Mar. 2024, doi: 10.1016/j.rineng.2023.101715.

[74] A. Uddin, M. B. Upama, H. Yi, and L. Duan, "Encapsulation of organic and perovskite solar cells: A review," Coatings, vol. 9, no. 2, p. 65, Jan. 2019, doi: 10.3390/coatings9020065.

[75] M. Samuel L., P. Suel-Hyun, M. Tensei, and D. P.Bentz, "Measurement of Thermal Properties of Gypsum Board At Elevated Temperatures," Proc. Fifth Int. Conf. Struct. Fire, pp. 656–665, 2008.

[76] Jack P. Holman, "Heat Transfer, Tenth Edition (McGraw-Hill Series in Mechanical Engineering)," McGraw-Hill, Inc, New York, vol. 9 MB, p. 758, 2010, [Online]. Available: www.mhhe.com

[77] H. K. Jalghaf and E. Kovács, "Simulation of phase change materials in building walls using effective heat capacity model by recent numerical methods," *Journal Energy Storage, Elsevier Ltd.*, vol. 83, Apr. 2024, doi: 10.1016/j.est.2024.110669.

[78] L. S. Athanasiou, D. I. Fotiadis, and L. K. Michalis, "8 - Propagation of Segmentation and Imaging System Errors," in Atherosclerotic Plaque Characterization Methods Based on

Coronary Imaging, L. S. Athanasiou, D. I. Fotiadis, and L. K. Michalis, Eds., Oxford: Academic Press, 2017, pp. 151–166. doi: https://doi.org/10.1016/B978-0-12-804734-7.00008-7.

[79] A. D. Solomon, “An easily computable solution to a two-phase Stefan problem,” Sol. Energy, vol. 23, no. 6, pp. 525–528, Jan. 1979, doi: 10.1016/0038-092X(79)90077-X.

[80] https://www.accuweather.com/en/iq/basra/206820/weather-forecast/206820.

[81] M. Saleh, E. Kovács, I. F. Barna, and L. Mátyás, “New Analytical Results and Comparison of 14 Numerical Schemes for the Diffusion Equation with Space-Dependent Diffusion Coefficient,” Mathematics, vol. 10, no. 15, p. 2813, Aug. 2022, doi: 10.3390/math10152813.

[82] K. Kant, A. Shukla, A. Sharma, Numerical simulation of building wall incorporating phase change material for cooling load reduction, Energy Clim. Chang. 1 (2020) 100008, https://doi.org/10.1016/j.egycc.2020.100008.

LIST OF PUBLICATIONS RELATED TO THE TOPIC OF THE RESEARCH FIELD

(1) Á. Nagy, M. Saleh, I. Omle, H. Kareem, and E. Kovács, “New Stable, Explicit, Shifted-Hopscotch Algorithms for the Heat Equation,” *Mathematical and Computational Applications, MDPI,* vol. 26, no. 3, p. 61, Aug. 2021, doi: 10.3390/mca26030061. **[65]**

(2) Á. Nagy, I. Omle, H. Kareem, E. Kovács, I. Ferenc Barna, and G. Bognar, “computation Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation,” *Computation, MDPI,* 2021, doi: 10.3390/computation.

(3) H. K. Jalghaf, E. Kovács, J. Majár, Á. Nagy, and A. H. Askar, “Explicit stable finite difference methods for diffusion-reaction type equations,” *Mathematics, MDPI,* vol. 9, no. 24, Dec. 2021, doi: 10.3390/math9243308.

(4) H. K. Jalghaf, I. Omle, and E. Kovács, “A Comparative Study of Explicit and Stable Time Integration Schemes for Heat Conduction in an Insulated Wall,” *Buildings, MDPI,* vol. 12, no. 6, Jun. 2022, doi: 10.3390/buildings12060824.

(5) H. K. Jalghaf, E. Kovács, and B. Bolló, “Comparison of Old and New Stable Explicit Methods for Heat Conduction, Convection, and Radiation in an Insulated Wall with Thermal Bridging,” *Buildings, MDPI,* vol. 12, no. 9, Sep. 2022, doi: 10.3390/buildings12091365.

(6) H. K. Jalghaf, E. Kovács, I. F. Barna, and L. Mátyás, “Analytical Solution and Numerical Simulation of Heat Transfer in Cylindrical- and Spherical-Shaped Bodies,” *Computation, MDPI,* vol. 11, no. 7, Jul. 2023, doi: 10.3390/computation11070131.

(7) H. K. Jalghaf, E. Kovács, and B. Bolló, “Simulation of transient heat transfer in multilayer walls with photovoltaic cell and air by using efficient numerical methods,” *Results in Engineering, Elsevier Ltd.,* vol. 21, Mar. 2024, doi: 10.1016/j.rineng.2023.101715.

(8) H. K. Jalghaf and E. Kovács, “Simulation of phase change materials in building walls using effective heat capacity model by recent numerical methods,” *Journal Energy Storage, Elsevier Ltd.,* vol. 83, Apr. 2024, doi: 10.1016/j.est.2024.110669.

(9) M.S. Alwan, H. K. Jalghaf, and E. Kovács, “ Numerical Study of Integrating the Phase Change Material with Building Envelop for Improved Indoor Thermal Comfort ”, Al-Iraqia Journal for Scientific Engineering Research, http://doi.org/10.58564/IJSER.3.3.2024.235.

(10) M.S. Alwan, and H. K. Jalghaf, “ Long Term Heat Transfer Simulation Through multilayer building wells with Phase Change Materials”, Al-Iraqia Journal for Scientific Engineering Research, http://doi.org/10.58564/IJSER.3.3.2024.226.