UNIVERSITY OF MISKOLC
FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

ELABORATING AND OPTIMIZING METHODS TO INVESTIGATE HEAT TRANSFER PROBLEMS IN BUILDING COMPONENTS

PHD THESES

Prepared by

Ali Habeeb Askar
Mechanical Engineering Department (BSc),
Mechanical Engineering Department (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 Supervisor

Dr. Endre Kovács

Dr. Betti Bolló

Miskolc
2025

CONTENTS

SUPERVISOR'S RECOMMENDATIONS

Date

Supervisor

LIST OF SYMBOLS AND ABBREVIATIONS

NOMENCLATURE

SYMBOLS			GREEK LETTERS	
C	*Heat capacity* [J/K]		ρ	*Mass density* [kg/m^3]
c	*Specific heat* [J/(kgK)]		Δ	*Difference*
DF	*Dufort and Frankel*		α	*Thermal diffusivity* [m^2/s]
h	*Convection heat transfer coefficient [W/(m^{2}K)]*		Δt	*Time step size* [s]
K	*Convection coefficient* [1/s]		σ^*	*Realistic values of the non-black body* [W/(m^2. K^4)]
k	*Thermal conductivity* [W/(m. K)]		σ	*Coefficient of the radiation term* [s^{-1}K^{-3}]
L	*Thickness* [m]		ε	*Emissivity*
LH	*Leapfrog–Hopscotch*		δ	*Factor to indicate daily mean temperature*
PI	*pseudo-implicit (PI)*		M	*The matrix index*
Q	*Heat transfer rate [*W*]*		r	*Mesh ratio*
q^*	*heat generation* [W/m^2]		τ_i	*Characteristic time or time constant*
q	*Heat source rate* [K/s]		***Subscripts***	
R	*Thermal resistance* [K/W]		a	*Ambient air*
S	*Area*		c	*Convection*
t	*Time* [s]		i,o	*Inlet, outlet*
u	*Temperature [K]*		l, r	*Left side, Right side*
U	*Overall heat transfer coefficient* [W/(m^{2}K)]		*up, low*	*Upper, lower*
v	*The air velocity [m/s]*		*sun*	*solar radiation*

The Methods	
ode15s	*A first- to fifth-order (implicit) numerical differentiation formulas with variable-step and variable order (VSVO), developed for solving stiff problems*
ode23	*Second (third) order Runge–Kutta–Bogacki–Shampine method*
ode23s	*A second-order modified (implicit) Rosenbrock formula*
ode23t	*Applies implicit trapezoidal rule using a free interpolant*
ode23tb	*A combination of backward differentiation formula and trapezoidal rule*
ode45	*A fourth (fifth) order Runge–Kutta–Dormand–Prince solver*
ode113	*A first- to 13th-order VSVO Adams–Bashforth–Moulton solver*

1. INTRODUCTION

1.1. Energy of Buildings

The advent of smart technologies and intelligent buildings has ushered in a new era of energy optimization. With buildings accounting for 40% of primary energy consumption and contributing to 24% of greenhouse gas emissions, the potential for energy efficiency in this sector is immense. This thesis explores the intersection of smart home technologies, energy efficiency, and thermal comfort, with a particular focus on heat transfer in building envelope and design. I delve into the crucial role of predicting energy requirements during the design stage of a building before its construction. This process is instrumental in ensuring that buildings can effectively utilize energy, achieve optimal economic results, and provide ease of use. According to the International Energy Agency, the construction industry is responsible for more than a third of the worldwide final energy consumption and is responsible for more than 40% of the world's direct and indirect CO_2 emissions (IEA) [1]. The installation of solar photovoltaic (PV) systems in Hungary has skyrocketed in recent years. To achieve its ambitious goal of producing 90% clean energy by 2030, Hungary will need to significantly expand its use of low-carbon sources of generation, including an early phase-out of the use of coal in power generation by the year 2025 [2]. In 2020, fossil fuels accounted for 68% of Hungary's total energy supply (TES). Natural gas accounted for 33%, oil for 27%, and coal for the remaining 7%. The largest non-fossil energy source is nuclear (16 % of TES), followed by bioenergy and waste (10 %), electricity imports (4 %), and other renewables (2%), including hydro, wind, geothermal, and solar see Figure 1.1 [2].

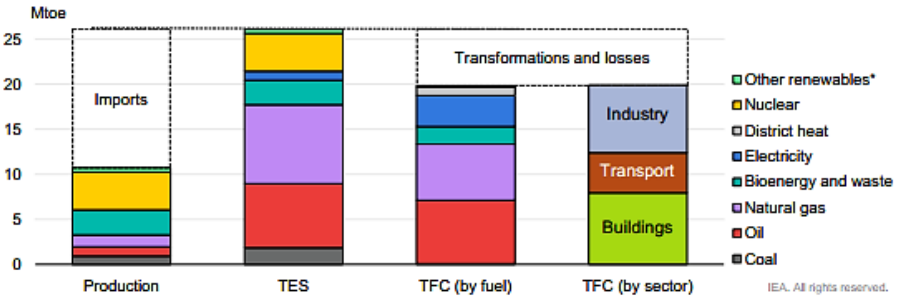


Figure 1.1. Hungary's energy supply demand and fuel sector, 2020 [2].

As it can be seen, the building sector consumes a large part of the total final consumption (TFC). Conduction heat transfer affects many issues related to buildings, such as exterior and interior wall conduction, conversion of heat gain/loss to cooling and heating load, heat loss or gain through roofs and ground heat loss from the slab-on-grade floor and basement walls. The rate of heat transfer through walls can be modified by changing the wall thickness or increasing the insulation layer thickness, and using different scenarios and types of smart materials [3]. Transient wall conduction heat transfer responds to weather conditions such as temperature variations, solar

radiation, air movement, etc. One of the most effective ways to increase the energy efficiency and reduce the energy consumption of a building is to increase the thermal insulation of the building envelope and decrease the heat loss through walls. Walls are responsible for about 35% of the heat loss in a house, and most of the heat enters and exits through them [4]. Heat is often lost by conduction or physical contact because the walls of a house are exposed to the colder air outside [5]. Thermal bridges are important to consider because they increase heat loss and lower the energy efficiency, durability, and air quality of buildings, which requires a combined thermal and structural design [6]. Thermal bridge models are especially relevant for architects, civil engineers, and insulation materials industry professionals. The study of thermal bridge effects aims to show the possibility of applying energy renovation techniques to existing buildings.

1.2. Building Envelope

The energy efficiency of buildings is a critical aspect of creating a more sustainable economy. Therefore, enhancing their energy efficiency is a key strategy. The construction sector, in particular, holds significant potential for reducing energy usage and greenhouse gas emissions relative to other industries. Heat transfer calculations are integral to multiple facets of building design and performance. These calculations are used for estimating energy loss or gain through the building envelope (heat conduction), analyzing the indoor environment, and addressing issues with specific materials or structural elements.

Conduction heat transfer impacts numerous building-related matters. These include exterior wall conduction, interior mass conduction, conversion of heat gain/loss to cooling and heating load, and ground heat loss from slab-on-grade floors and basement walls. The rate of heat transfer through walls can be altered by modifying the wall thickness or augmenting the insulation layer thickness.

Transient wall conduction heat transfer responds to weather conditions, including temperature fluctuations, solar radiation, and air movement. Walls constitute a significant portion of a building's envelope and are designed to deliver both thermal and acoustic comfort within the premises, while simultaneously maintaining the structure's aesthetic integrity. The wall's thermal resistance, or R-value, is a critical determinant of a building's energy consumption, particularly in skyscrapers where the wall-to-envelope area ratio is substantial. Current market evaluations of center-of-cavity R-values and clear wall R-values account for the impact of thermal insulation. Nevertheless, the effects of the framing factor and interface connections are frequently overlooked [7].

In scenarios where thermal insulation is present, walls are more susceptible to surface condensation if the ambient air's relative humidity exceeds 80%, assuming that the convective and radiative heat transfer coefficients of the external wall are minimal. This issue intensifies during the winter season and in colder climates with elevated humidity levels [8]. Such moisture accumulation on the exteriors of buildings can foster the growth of microbes, potentially diminishing the lifespan of the wall and causing other adverse conditions within the structure.

Traditionally, walls are categorized based on the construction materials used, such as wood, metal, or masonry. However, innovative wall designs have been developed to enhance the energy efficiency and comfort of buildings. Subsequent sections will elaborate on these advanced wall technologies. Thus, Passive solar walls are designed to collect, store, and distribute solar energy in the form of heat in the winter and reject solar heat in the summer. They are an integral part of passive solar design, a method that uses the sun's energy for the heating and cooling of living spaces. Lightweight concrete (LWC) walls, on the other hand, are constructed using concrete with lower densities than traditional concrete, resulting in a structure that is both strong and lighter, which can improve thermal insulation and ease of installation. Ventilated or double skin walls feature a cavity between two wall layers that allows for air circulation, enhancing thermal comfort and energy efficiency. Lastly, walls with latent heat storage incorporate materials that absorb and release heat, which helps in stabilizing indoor temperatures throughout the day, making them highly effective for energy conservation [9].

Roofs play a pivotal role in the structural integrity of buildings, serving as a primary shield against solar radiation and environmental elements. They significantly affect the indoor comfort of occupants, with large structures like sports complexes and auditoriums being particularly vulnerable to heat gain or loss [10]. This progression underscores the importance of roof thermal performance in enhancing a building's overall energy efficiency.

In tropical climates, passive cooling strategies are essential for modifying roof architecture to improve comfort levels. Techniques range from compact cellular layouts that minimize solar exposure to domed and vaulted structures, along with ventilated roofs, both natural and mechanical, and even double-layered roofs. Innovations are also emerging, such as white-washed exteriors to decrease solar absorptivity, vegetative coverings for humidity control and shading, and the use of materials like concrete to reduce peak thermal loads. Roof shading, achieved through economical materials like terracotta tiles or inverted earthen pots, can lower indoor temperatures by up to 6°C [11]. Additionally, roof coatings with high solar reflectance and emissivity are selected to combat solar heat, avoiding those like aluminum-pigmented options due to their low infrared emittance. These coatings can cool a white concrete roof surface by 4°C on hot summer days and 2°C at night. The choice of roofing systems often depends on the local climate, with a diverse array of options available to meet specific environmental demands [11].

The roof is a pivotal element in a building's energy efficiency and overall energy consumption. The design of the roof, which can vary based on location and type such as plain, vault [12], or dome, significantly influences these factors [13]. By augmenting the thermal insulation of the roof, a critical component of the building envelope, I can markedly diminish heat loss [14]. This is attributed to the fact that a well-insulated roof serves as a barrier to heat flow, curtailing the amount of heat that escapes from the building during colder periods and inhibiting excessive heat from entering during warmer periods. As a result, this reduces the energy required for heating and cooling the building, thereby supporting the building's energy efficiency and curtailing its energy

consumption. Consequently, investment in superior roof insulation emerges as one of the most effective strategies for fostering energy-efficient buildings and advancing sustainability [15].

Fenestration, encompassing windows and doors, is crucial for thermal comfort and lighting within buildings, while also enhancing architectural aesthetics. Recent advancements in glazing technologies, such as solar control glasses and low-emissivity coatings, have significantly improved their performance. A study involving various glazing types across diverse climates revealed that a window's energy savings depend on factors like thermal conductivity (U-value), solar heat gain coefficient (SHGC), orientation, and building characteristics. A diverse array of technologies is available to cater to this need, including aerogel glazing, vacuum glazing, switchable reflective glazing, suspended particle devices (SPD) film, and holographic optical elements. Each of these solutions offers unique benefits and applications, reflecting the innovation and progress in the field of advanced glazing technologies [16].

1.3. The Governing Differential Equation of Heat Conduction, Convection and radiation

Within the Cartesian coordinate system, the differential control volume is delineated as a diminutive rectangular prism with dimensions δx, δy, and δz., as depicted in Figure 1.2 Assuming that the body's density is symbolized by ρ and its specific heat by *c*, the equations governing conduction, convection, and radiation are derived from the conservation of energy principle. Over a short interval Δ*t*, the energy balance for this elemental volume is mathematically expressed as follows:

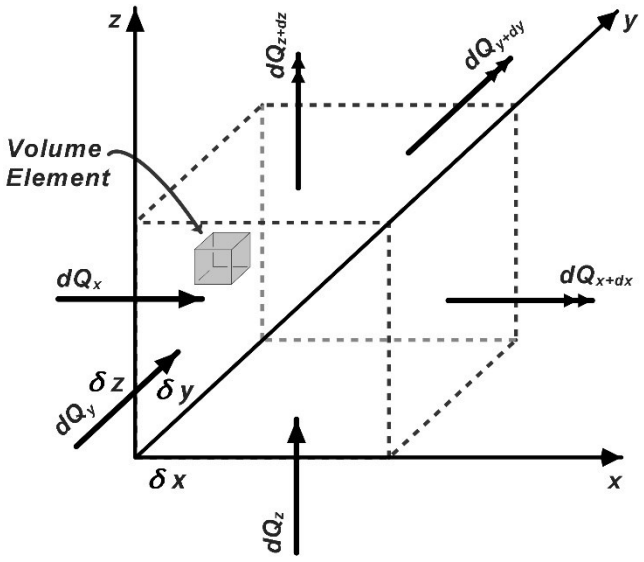


Figure 1.2. Which is used for deriving heat conduction in Cartesian coordinates, the volume is defined by δ*x*, δ*y*, and δ*z*.

$$\begin{pmatrix}\text{Rate of heat}\\ \text{transfer by}\\ \text{conduction}\\ \text{at } x, y, \text{ and } z\end{pmatrix} - \begin{pmatrix}\text{Rate of heat transfer}\\ \text{by conduction at}\\ x+\delta x, y+\delta y, \text{ and } z+\delta z\end{pmatrix} + \begin{pmatrix}\text{Rate of heat generation}\\ \text{on the element}\\ \text{surface and inside it}\end{pmatrix}$$

$$\pm\begin{pmatrix}\text{Rate of heat}\\ \text{convection}\end{pmatrix} \pm \begin{pmatrix}\text{Rate of heat}\\ \text{radiation}\end{pmatrix} = \begin{pmatrix}\text{Rate of change}\\ \text{in the element}\\ \text{energy content}\end{pmatrix}.$$

$$Q_x + Q_y + Q_z - Q_{x+\delta x} - Q_{y+\delta y} - Q_{z+\delta z} + Q_{gen} \pm Q_{convection} \pm Q_{radiation} = \frac{\Delta E_{element}}{\Delta t} \qquad (1.1)$$

The mass and volume of the differential control volume are determined, as follows:

$$\delta m = \rho \delta x \delta y \delta z, \quad \text{and} \quad \delta V = \delta x \delta y \delta z, \qquad (1.2)$$

where ρ is the mass density $[kg/m^3]$ of the control volume.

The Newton heat convection law:

$$Q_{convection} = hS\Delta u = hS(u_a - u)\ , \qquad (1.3)$$

where S is the surface area, h is the convection heat transfer coefficient, u_a is the ambient temperature and u the surface temperature of element.

Given that the ambient temperature, denoted as u_a, is not directly dependent on (u), the term hSu_a is incorporated into the heat generation expression. This aligns with the Stefan–Boltzmann law governing outgoing radiation heat:

$$Q_{radiation} = \sigma^* S u^4\ , \qquad (1.4)$$

where $\sigma^* = \varepsilon \cdot \sigma''$, i.e., considering that the surface is not a black body, the universal constant of Stefan-Boltzmann, denoted as $\sigma'' = 5.67 \times 10^{-8}\,\mathrm{W/(m^2K^4)}$, is scaled by the appropriate emissivity ε constant.

The rate of change of internal energy within the control volume can be expressed as $\rho c \delta x \delta y \delta z (u_{t+\Delta t} - u_t)$, where $E_{element}$ represents the energy content of the element. Mathematically, this can be written as:

$$\Delta E_{element} = E_{t+\Delta t} - E_t = mc(u_{t+\Delta t} - u_t) = \rho c \delta x \delta y \delta z (u_{t+\Delta t} - u_t)\,, \qquad (1.5)$$

substituting into Eq. (1.1), I get

$$Q_x + Q_y + Q_z - Q_{x+\delta x} - Q_{y+\delta y} - Q_{z+\delta z} + Q_{gen} - hSu - \sigma^* S u^4 = \rho c \delta x \delta y \delta z \frac{(u_{t+\Delta t} - u_t)}{\Delta t}\ ,$$

dividing by $\delta x \delta y \delta z$ gives

$$-\frac{1}{\delta y\delta z}\frac{Q_{x+\delta x}-Q_x}{\delta x}-\frac{1}{\delta x\delta z}\frac{Q_{y+\delta y}-Q_y}{\delta y}-\frac{1}{\delta x\delta y}\frac{Q_{z+\delta z}-Q_z}{\delta z}+\frac{Q_{gen}}{\delta x\delta y\delta z}$$
$$-\frac{hSu}{\delta x\delta y\delta z}-\frac{\sigma^* Su^4}{\delta x\delta y\delta z}=\rho c\frac{(u_{t+\Delta t}-u_t)}{\Delta t} \quad . \qquad (1.6)$$

Considering the heat transfer areas of the element for heat conduction in the *x*, *y*, and *z* directions are $S_x=\delta y\,\delta z$, $S_y=\delta x\,\delta z$, and $S_z=\delta x\,\delta y$ respectively, and taking the limit as $\delta x,\ \delta y,\ \delta z$ and $\Delta t\ \to 0$.

The Fourier's law of heat conduction and the definition of the derivative:

$$\lim_{\delta x\to 0}\frac{1}{\delta y\delta z}\frac{Q_{x+\delta x}-Q_x}{\delta x}=\frac{1}{\delta y\delta z}\frac{\partial Q_x}{\partial x}=\frac{1}{\delta y\delta z}\frac{\partial}{\partial x}\left(-k\delta y\delta z\frac{\partial u}{\partial x}\right)=-\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)$$
$$\lim_{\delta y\to 0}\frac{1}{\delta x\delta z}\frac{Q_{y+\delta y}-Q_y}{\delta y}=\frac{1}{\delta x\delta z}\frac{\partial Q_y}{\partial y}=\frac{1}{\delta x\delta z}\frac{\partial}{\partial y}\left(-k\delta x\delta z\frac{\partial u}{\partial y}\right)=-\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)$$
$$\lim_{\delta z\to 0}\frac{1}{\delta x\delta y}\frac{Q_{z+\delta z}-Q_z}{\delta z}=\frac{1}{\delta x\delta y}\frac{\partial Q_z}{\partial z}=\frac{1}{\delta x\delta y}\frac{\partial}{\partial z}\left(-k\delta x\delta y\frac{\partial u}{\partial z}\right)=-\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)$$

Substituting into Eq. (1.6), the general heat equation in the rectangular coordinates

$$\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial u}{\partial z}\right)+\frac{Q_{gen}}{\delta x\delta y\delta z}-\frac{hSu}{\delta x\delta y\delta z}-\frac{\sigma^* Su^4}{\delta x\delta y\delta z}=\rho c\frac{\partial u}{\Delta t}. \qquad (1.7)$$

In the case of *k* is constant, Eq. (1.7) is divided by *k*, so it reduces to

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}+\frac{Q_{gen}}{k\delta x\delta y\delta z}-\frac{hSu}{k\delta x\delta y\delta z}-\frac{\sigma^* Su^4}{k\delta x\delta y\delta z}=\frac{1}{\alpha}\frac{\partial u}{\Delta t}\ , \qquad (1.8)$$

One can express the material's thermal diffusivity in terms of $\alpha=\frac{k}{\rho c}$, and q^* is the incoming radiative heat for a unit area. The conductive and radiative heat transfer directions will always be horizontal, thus $S=\delta y\delta z$ and $Q_{gen}=\left(q^*+hu_a\right)S$.

Based on these considerations, the heat conduction Eq. (1.9) can be extended to include heat convection, radiation, and source terms as Eq. (1.11).

$$\alpha\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\frac{q^*}{\rho c\delta x}+\frac{hu_a}{\rho c\delta x}-\frac{h}{\rho c\delta x}u-\frac{\sigma^*}{\rho c\delta x}u^4=\frac{\partial u}{\Delta t}\ . \qquad (1.9)$$

The temperature equation incorporates radiation, conduction, convection, and heat generation sources, utilizing vector notation [17], [18] to express the former equation:

$$\frac{\partial u}{\partial t}=\alpha\nabla^2 u+q-Ku-\sigma u^4\ , \qquad (1.10)$$

where ∇ is the differential vector operator, $\nabla = \vec{i}\left(\frac{\partial}{\partial x}\right) + \vec{j}\left(\frac{\partial}{\partial y}\right) + \vec{k}\left(\frac{\partial}{\partial z}\right)$, $q = \frac{q^*}{c\rho\delta x} + \frac{hu_a}{c\rho\delta x}$ refers to the outside heat source or heat generation, $K = K(\vec{r}) = \frac{h}{c\rho\delta x}$ is the term of convection heat transfer, and $\sigma = \sigma(\vec{r}) = \frac{\sigma^*}{c\rho\delta x}$ is the term of radiation heat transfer. The terms Ku, q and σu^4 in Eq. (1.10) are nonnegative and their unit is $[\mathrm{K/s}]$.

The heat equation states that heat transfer occurs when there is a temperature difference between two points, which leads to temperature equalization. If the material properties are non-homogeneous Eq. (1.10) can be expressed in a more general form:

$$\frac{\partial u}{\partial t} = \frac{1}{\rho c}\nabla(k\nabla u) + q - Ku - \sigma u^4 \ . \tag{1.11}$$

In various scientific disciplines, including biology, chemistry, and physics, mass movement can be modelled using generalized forms of the diffusion equation, such as the advection-diffusion-reaction equation. For instance, the atoms in carbon nanotubes, the charge carriers in semiconductors, and the proteins in developing embryos. Additionally, a system of equations or closely related equations has been employed to simulate the fluid flow through porous media, encompassing phenomena like ground water, moisture, and crude oil in subterranean reservoirs [19].

1.4. Numerical Methods for Solving Heat Conduction Equation

Analytical solutions, including recent advancements [20], are prevalent for spatially homogeneous systems. These systems often serve as the testing ground for the development and validation of numerical methods by mathematicians. Additionally, a selection of analytical solutions exists for one-dimensional multilayer problems, applicable to both steady-state and transient conditions. Such solutions are routinely employed in the computation of building envelopes heat gain and loss, as well as heat storage [21]. Nevertheless, most heat conduction issues within buildings are multi-dimensional and transient in nature. Furthermore, the properties of materials, such as density and heat conductivity, can exhibit significant variation within the system [22] (p. 15). Consequently, the use of numerical computer simulations is indispensable.

A thermal analysis may reveal the heat distribution patterns of a system or its individual parts. Most research into thermal quantities focuses on temperature distributions, thermal fluxes, and heat capacities. Since diverse heat transfer applications within engineering fields involve many thermal models, the analysis of transient heat transfer is an essential problem that is often solved using numerical rather than analytical methods. Analytic techniques provide precise results, but they can only be used for isotropic, homogeneous situations with straightforward geometries and boundary conditions [20]. In homogeneous space, the diffusion equation has analytical solutions,

old and new, as well, according to Barna et al. [23], but when the material properties are functions of space, numerical integration is necessary.

According to Newton's law of cooling, a term $K(u_a - u)$ can describe (free) convective heat transfer [24], where u_a is the ambient temperature (measured in Kelvin), which can be considered as independent from u. On the other hand, according to Stefan–Boltzmann 's law [25] the heat loss of a unit surface via electromagnetic radiation can be given by a term $-\sigma u^4$. The incoming radiation, the Ku_a heat gain due to the nonzero temperature of the ambient air, as well as other factors such as heat generated by electric currents, can be collected into a so-called heat source term denoted by q. Several numerical methods are usually classified as either explicit or implicit, but occasionally these two approaches are combined, which have been proposed to solve Eq. (1.7) and Eq. (1.9), belong to the wide group of finite difference schemes (FDM) [26], [27], [28]. Implicit methods have excellent stability properties; therefore, they are commonly used for these equations [29], [30]. The price for stability is that a system of algebraic equations must be solved at each time step. This can imply very slow calculations, particularly in cases when the number of space dimensions is more than one, thus one has very large-sized and non-tridiagonal matrices. In these cases, even the most trivial explicit (Euler) time integration can be considerably faster than the implicit one [29]. Moreover, explicit algorithms can be parallelized much more straightforwardly than in implicit methods. The main obstacle against the rise of the explicit algorithms is that they are typically only conditionally stable, i.e., the solutions can blow up if the time step size is below the so-called Courant–Friedrichs–Lewy (CFL) limit. For example, explicit Runge–Kutta methods can never be unconditionally stable [31] (p. 60). The coefficients c, k, or the diffusion constant can be highly non-uniform in space [32], e.g., when the physical properties have sharp discontinuities at the material boundaries. In these cases, the range of the eigenvalues of the system matrix has several orders of magnitude, the problem is very stiff, the CFL limit can be extremely small, thus the simulation can be unacceptably time consuming.

Analytical diffusion equation solutions are known and found in basic textbooks [33], [34]. These solutions are crucial to understanding diffusion. Second, these solutions test old and new numerical methods. I want to develop new algorithms for a modified version of PDE (1) with a reaction coefficient that depends on time and space. Semi-explicit or semi-implicit methods, which combine explicit and implicit approaches, can also work well. I must agree with scholars who create and study explicit and unconditionally stable algorithms. Chen-Charpentier and Kojouharov [35] proposed the UPFD (unconditionally positive finite difference) scheme for the linear diffusion–advection–reaction equation. Later Appadu [36] and Savovic et al. [37] study this algorithm. Pourghanbar et al. [38] solved a nonlinear PDE faster than implicit methods using the Alternating Direction Explicit (ADE) Saulyev Method. The ADE, ADI, and odd-even hopscotch methods were compared to solve diffusion-reaction equations by Al-Bayati et al. [39] They found that implicit methods are more accurate but slower than explicit methods. Our paper [40], compared 12 explicit and robust numerical methods for heat conduction in walls with and without insulation using equal and unequal meshes. I used 14 heat conduction, convection, and radiation

methods from [41] for heat conduction, convection, and radiation. I found that the best method was the leapfrog–hopscotch (LH). The LH is more accurate when the system is less stiff, but for high stiffness, it still behaves well. The Dufort–Frankel (DF), shifted-hopscotch (SH), and asymmetric hopscotch (ASH) methods also performed well. The unconditionally stable methods for the basic conduction problem could use very large time steps without stability issues, and they were better than usual explicit time integrators. I focused on the best way to include convection and radiation terms in our study, using the LH method as the best one [40] and using this method for long simulations of a building wall in summer and winter simulations.

In my research, I collaborated with my supervisor to explore and enhance families of novel and traditional explicit methods for solving linear and nonlinear heat conduction equations. These methods were developed based on innovative approaches. Specifically, I adapted successful techniques, such as the pseudo-implicit and Leapfrog hopscotch methods, to scenarios involving heat transfer through convection and nonlinear radiation (like Stefan-Boltzmann-type radiation). These adaptations addressed real-world heat transfer challenges in building environments. Additionally, I conducted a comparative analysis of my findings with those obtained from neural networks

1.5. Outline of the Thesis

This thesis begins with an introduction outlining the background, motivation, objectives, and structure. Chapter 2 discusses the numerical discretization of the heat conduction equation, transforming PDEs to ODEs, and reviews explicit methods. Chapter 3 systematically tests nine numerical algorithms, analyzing their stability and accuracy. Chapter 4 develops the so called pseudo-implicit algorithm, highlighting its advantages. Chapter 5 explores free convection and radiation terms using the leapfrog-hopscotch method. Chapter 6 analyzes diffusion-reaction PDEs, comparing numerical algorithms. Chapter 7 applies neural networks to predict building energy, evaluating models, and simulating the thermal behavior of the walls of the building by using the recommended method, the leapfrog-hopscotch method. Chapter 8 optimizes roof inclination angles for energy efficiency in different climates. The thesis concludes with a summary of findings, contributions, and future work recommendations.

2. HEAT EQUATION DISCRETIZATION AND NUMERICAL METHODS

The focus of this chapter is to explain and illustrate the numerical discretization process that applies to the spatial variables present in the heat conduction equation. This process effectively transforms the equation from a partial differential equation (PDE) into a system of ordinary differential equations (ODEs). I will subsequently develop further procedural applications for the resulting system of ODEs. Furthermore, this chapter will include a brief review of a variety of explicit methods, both established and novel approaches. These explicit methods will serve as the foundational elements for all subsequent plans and applications delineated in the following sections. I will also review some of the software packages and methods built into them that I used in my work to simulate and validate methods in the end of this chapter.

2.1. The Equations and Its Discretization

Diffusion of particles and Fourier-type heat conduction are omnipresent mass or energy transport processes. In the simplest linear case, they are described by the following partial differential equation (PDE):

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} + q\,, \tag{2.1}$$

where $x,\, t \in \mathbb{R}$ are the independent variables, $u = u(x,t)$ is the unknown concentration of particles or the temperature in the case of heat transfer, and α is the coefficient of (thermal) diffusivity. The thermal diffusivity of a material can be given as $\alpha = k/(c\rho)$, where $c = c(\vec{r},t)$, $k = k(\vec{r},t)$, and $\rho = \rho(\vec{r},t)$ are the specific heat, the heat conductivity, and the density of the material, respectively. If these coefficients depend on space, one has to use the more general equation

$$\frac{\partial u}{\partial t} = \frac{1}{c\rho}\nabla\left(k\nabla u\right), \tag{2.2}$$

where it is assumed that the *c* and ρ functions are positive. This equation is now valid for more than one space dimension.

The heat conduction Eq. (2.1) can be extended to include heat convection, radiation, and source terms are added to the heat conduction Eq. (1.10). Note that all terms in Eq. (1.10) are local, except the conduction term. In the case of Eq. (2.1) in one space dimension, applying the most common central difference equation

$$\frac{\partial^2}{\partial x^2}u(x_i) \approx \frac{\dfrac{u(x_{i+1})-u(x_i)}{\Delta x} + \dfrac{u(x_{i-1})-u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i-1}-2u_i+u_{i+1}}{\Delta x^2}\,, \tag{2.3}$$

Which is second order in Δx, where $i = 1,\ldots,N$ and N is the overall number of nodes. By applying this, in one space dimension, I am able to derive the spatially discretized form of the heat transfer Eq. (1.10) in one space dimension as follows:

$$\frac{du_i}{dt} = \alpha \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + q - Ku_i - \sigma u_i{}^4 . \qquad (2.4)$$

Now, let us demonstrate the discretization of the heat transfer equation assuming that the variables α, k, c, and ρ, which describe the properties of materials, are functions of space rather than fixed values. In one space dimension, I now have to deal with the following instead of the term $\alpha \nabla^2 u$ for homogeneous materials:

$$\frac{1}{c(x)\rho(x)} \frac{\partial}{\partial x}\left(k(x) \frac{\partial u}{\partial x} \right) . \qquad (2.5)$$

I discretize the function k, and at the same time the space derivatives in Eq. (2.5) by the standard central difference formula to obtain:

$$c(x_i)\rho(x_i) \left.\frac{\partial u}{\partial t}\right|_{x_i} = \frac{1}{\Delta x}\left[k\left(x_i + \frac{\Delta x}{2} \right) \frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} + k\left(x_i - \frac{\Delta x}{2} \right) \frac{u(x_i - \Delta x) - u(x_i)}{\Delta x} \right]. \qquad (2.6)$$

Equations (2.5) and (2.6) are based on the node-picture, typically used by mathematicians. Instead of node-variables, let us introduce cell variables to arrive at a resistance-capacitance-type model of heat conduction. It means that u_{i}, c_{i}, and ρ_{i} are the approximation of the average temperature, specific heat, and density of cell i, by their value at the cell centre. Furthermore, $k_{\mathrm{i,i+1}}$ is the heat conductivity between cell i and its (right) neighbour, estimated by its value at the border of the cells. Now the previous formula will have the form:

$$\frac{du_i}{dt} = \frac{1}{c_i \rho_i \Delta x}\left(k_{i,i+1} \frac{u_{i+1} - u_i}{\Delta x} + k_{i-1,i} \frac{u_{i-1} - u_i}{\Delta x} \right) + q - Ku_i - \sigma u_i{}^4 . \qquad (2.7)$$

Let us now consider heat conduction of an element with cross-section S, divided into cells only along its length. The volume and the heat capacity of the cell in $[\mathrm{J/K}]$ can be given as $V = S\Delta x$, and $C_i = c_i \rho_i V$, respectively, the heat source term q,

$$q_{\mathrm{i}} = \frac{1}{V_{\mathrm{i}}} \int_{V_i} q dV \approx q \ \text{ in } \left[\frac{K}{s} \right] \text{units,}$$

On the other hand, the thermal resistance between two neighboring cells is estimated as $R_{i,i+1} \approx \Delta x / \left(S k_{i,i+1} \right)$ in (K/W) units. The distances between the cells centre in case of non-equidistant grid are $d_{\mathrm{i},i+1} = \left(\Delta x_{\mathrm{i}} + \Delta x_{i+1} \right)/2$ and the resistances can be determined by this approximation as $R_{i,i+1} \approx d_{\mathrm{ij}} / \left(k_{i,i+1} S_{\mathrm{ij}} \right)$

If the material properties or the sizes of the two neighboring cells are different, I can write for the resistance between cells i and i+1 that $Rx_i \approx \left[\Delta x_i / \left(k_i S_i \right) \right] + \left[\Delta x_{i+1} / \left(k_{i+1} S_{i+1} \right) \right]$, and if the cell j is below the cell i, I have $Rz_i \approx \left[\Delta x_i / \left(k_i S_i \right) \right] + \left[\Delta x_j / \left(k_j S_j \right) \right]$ for the vertical resistance.

Now the equation for the time derivative of the temperature of each cell in the element is as follows:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i}C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i}C_i} + q - Ku_i - \sigma u_i^4 \tag{2.8}$$

As a generalization of Eq. (2.8) one may construct the ODE system for the time derivative of the cell variables for a generic grid by using the above approximations as follows:

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j}C_i} + q - Ku_i - \sigma u_i^4 \tag{2.9}$$

The set of ordinary differential equations (ODEs) shown here can be used with a lot of different grids, even ones that are not structured and have cells that are different sizes, shapes, and properties. It is important to note that uneven discretization may potentially compromise spatial accuracy. However, for the purposes of this work, I have chosen to exclusively utilize cells of a rectangular configuration.

After spatial discretization as discussed above, Eq. (2.9) for $K = 0$ and $\sigma = 0$ can be written into a brief matrix-form:

$$\frac{d\vec{u}}{dt} = M\vec{u} + \vec{Q}, . \tag{2.10}$$

The system matrix *M* is tridiagonal (in the one-dimensional case), and it is the sum of two terms related to the diffusion and the linear reaction terms, respectively:

$$M = M^D + M^R \tag{2.11}$$

In the one-dimensional case of Eq. (2.4), the matrix *M* is tridiagonal with the following elements:

$$M_{ii}^D = -\frac{2\alpha}{\Delta x^2} \ , \ M_{i,i+1}^D = \frac{\alpha}{\Delta x^2} \ , \ M_{i,i-1}^D = \frac{\alpha}{\Delta x^2} , \ M_{ii}^R = -K_i \tag{2.12}$$

In the general case of Eq. (2.9), the nonzero elements of the matrix M^D introduced in Eq. (2.11) can be given as:

$$M_{ij}^D = \frac{1}{R_{ij}C_i} , \quad M_{ii}^D = -\sum_{j \neq i} M_{ij}^D \tag{2.13}$$

Figure 2.1 can help the reader to visualize these quantities the red double arrows are for conduction between cells with capacities C_i and C_j through the resistances R_{ij} . [42].

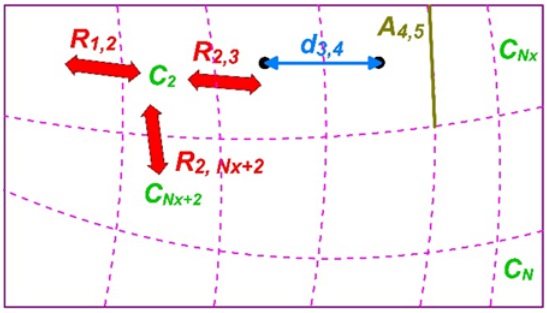


Figure 2.1 The Arrangement of the generalized variables for the case when the mesh is not regular.

I introduce the following notations:

$$\tau_{\mathrm{i}} = \frac{-1}{M_{ii}^{D}}, \quad r_i = \frac{\Delta t}{2\tau_i} \text{ and } \quad A_i = \Delta t \sum_{\mathrm{j}\neq i} M_{ij}^{D} u_{\mathrm{j}}^{n} = \Delta t \sum_{j\neq i} \frac{u_j^n}{C_i R_{ij}}. \tag{2.14}$$

where $\tau_{\mathrm{i}} \geq 0$ is the characteristic time or time constant of cell i, r_i is the generalization of $r = \frac{\alpha \Delta t}{\Delta x^2} = -\frac{m_{ii}\Delta t}{2}$ (the usual mesh ratio in the case of the diffusion equation), and A_i reflects the state of the neighbors of cell i.

The off-diagonal element, denoted as $m_{\mathrm{i,j}} = 1/\left(R_{\mathrm{i,j}} C_{\mathrm{i}}\right)$, of the M matrix may be nonzero exclusively when cells i and j are adjacent. Subsequent summations are performed over the neighbors of the actual cell, represented by $\mathrm{j} \in n(\mathrm{i})$. Unless specified otherwise, closed (zero Neumann) boundary conditions are considered, implying that the domain's boundary under investigation is thermally isolated concerning conductive heat transfer [42]. It is important to note that the cells' arrangement and shape do not necessarily have to be regular.

I define the eigenvalues of the system matrix, M, with the smallest and largest absolute values (excluding zero) as λ_{MIN} and λ_{MAX} respectively. The stiffness ratio of the system is then given by the ratio $\lambda_{\mathrm{MAX}} / \lambda_{\mathrm{MIN}}$. The Forward Time Central Space (FTCS), also known as the explicit Euler scheme, has a maximum possible time step size, denoted as $h_{\mathrm{MAX}}^{\mathrm{FTCS}} = |2 / \lambda_{\mathrm{MAX}}|$, beyond which the solutions are anticipated to diverge due to instability. This critical threshold is commonly referred to as the Courant–Friedrichs–Lewy (CFL) limit. It is noteworthy that this limit is also applicable to the second-order explicit Runge–Kutta (RK) method, as referenced in [43].

2.2. Some Explicit Methods

This part provides essential insights into several algorithms, encompassing both established methods and novel innovations. Initially, I introduce their equations within the context of a fundamental scenario a one-dimensional, equidistant mesh, Eq. (2.2), After that, I extend these equations to incorporate a comprehensive, arbitrary mesh. The primary rationale for presenting simplified numerical schemes in most textbooks is to facilitate comparative analysis. However,

this work primarily focuses on the more comprehensive forms, which exclusively feature in our study.

The conventional mesh ratio is $r_i = \frac{\alpha \Delta t}{\Delta x^2}$ for the 1D equidistant mesh, e.g. for Eq. (2.2). The r_i corresponds to an extension of *r*, while the other quantity considers not only the state of cell *i* but also that of its neighbouring cells. However, the subsequent notations are introduced for the general mesh:

$$r_i = \Delta t \sum_{j \neq i} \frac{1}{C_i R_{ij}} \text{ and } A_i = \Delta t \sum_{j \neq i} \frac{u_j^n}{C_i R_{ij}} \tag{2.15}$$

It is widely known that for the general first order ODE 1D, the theta-method has the formula of general time discretization

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

where $\theta \in [0,1]$. In the case of Eq. (2.2), it yields:

$$u_i^{\mathrm{n}+1} = u_i^{\mathrm{n}} + r\left[\theta\left(u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) + (1-\theta)\left(u_{i-1}^{\mathrm{n}+1} - 2u_i^{\mathrm{n}+1} + u_{i+1}^{\mathrm{n}+1}\right)\right], \tag{2.16}$$

where $u_{i\pm1}^n = u_{i+1}^n + u_{i-1}^n$. For $\theta = 0, \frac{1}{2},$ and 1, this is clearly an implicit Euler method, the Crank–Nicolson, and the explicit Euler (FTCS) schemes, respectively [44]. If $\theta < 1$, the theta method is implicit. By the so-called pseudo-implicit trick, we make it explicit: the neighbors $u_{i\pm1}^n$ in the second term at the right-hand side of Eq.(2.16) is taken into account at the old (*n*-th) time level. Using this, we obtain:

$$u_i^{n+1} = u_i^n - 2r\theta u_i^n - 2r(1-\theta)u_i^{n+1} + r\left(u_{i-1}^n + u_{i+1}^n\right), \tag{2.17}$$

which may easily be rearranged as:

$$u_i^{\mathrm{n}+1} = \frac{(1-2r\theta)u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right)}{1+2r(1-\theta)} \ . \tag{2.18}$$

2.2.1. The UPFD method

The UPFD method is the theta-method Eq. (2.2) for $\theta = 0$. In the case of Eq. (2.2), it reads as follows:

$$u_i^{\mathrm{n}+1} = \frac{u_i^{\mathrm{n}} + r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+2r}, \tag{2.19}$$

and the general form for Eq. (2.8) or (2.9) is:

$$u_i^{n+1} = \frac{u_i^n + A_i}{1+2r_i} \ . \tag{2.20}$$

2.2.2. *The Dufort–Frankel (DF) algorithm*

This method can be obtained from the so called leapfrog explicit scheme by a modification [45] (p. 313). It is a known explicit unconditionally stable scheme that has the formula in the special and general case:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+2r} \text{ and } u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1+r_i} \qquad (2.21)$$

As one can see, it is a one-stage but two-step method (the formula contains u_i^{n-1}), which is not a self-starter, so another method must be applied to start the method by the calculation u_i^1. For this purpose, we apply the UPFD formula twice (with halved time step size).

2.2.3. *The Leapfrog–Hopscotch method*

The leapfrog-hopscotch (LH) method [46]. We have a structure consisting of two half and several full time steps. The calculation starts again by taking a half-sized time step for the odd nodes using the initial values Stage 0 (not repeated, green box), which uses $\theta = 0$. Then, for the even and odd nodes, full-time steps are taken strictly alternately until the end of the last timestep, The intermediate stages as well as the last stage (light and dark orange boxes) use $\theta = 1/2$ (Figure 2.2). In this chapter, I used only the best already proven combination of formulas (L2 in [46]), which means that $\theta = 0$ and $\theta = 1/2$ are applied in formulas (LH) at the first and at all other time steps, respectively.

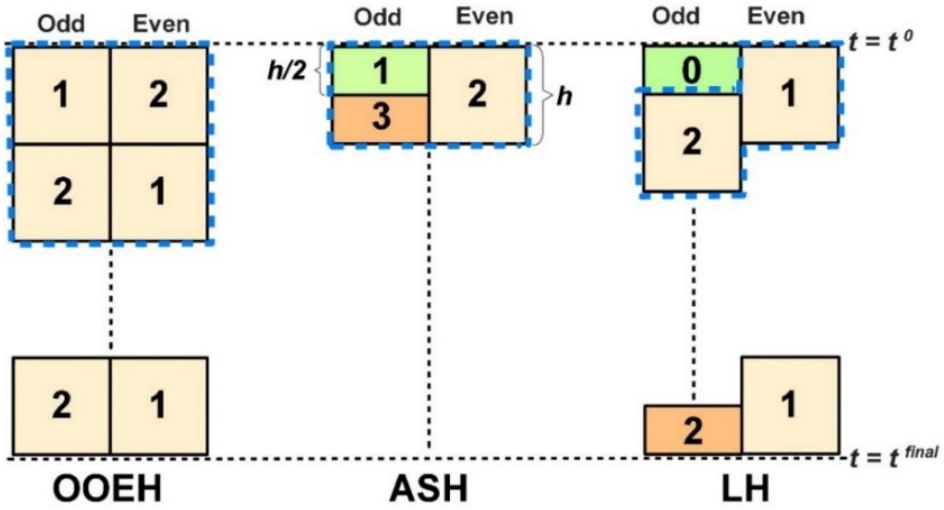


Figure 2.2. Hopscotch-type space-time structures. The time elapses from the top $\left(t = t^0\right)$ to the bottom $\left(t = t^{\text{fin}}\right)$.

2.2.4. *The original odd-even hopscotch (OOEH)*

The initial odd-even hopscotch (OOEH) technique was discovered nearly half a century ago [47]. Details regarding its geographical and temporal arrangement can be found in reference [48]. The primary objective behind designing this algorithm was to create a versatile solution that

minimizes input requirements from both humans and machines for any given task. The two-stage process, which remains fully transparent, has undergone continuous improvements and expansions to enhance precision, with a consistent trend toward greater implicitness. After the initial step, the FTCS formula, utilizing implicit Euler time discretization BTCS, is applied to even cells, while the FTCS formula, relying on explicit Euler time discretization, is used for odd cells. Following each time step, the designations 'odd' and 'even' are swapped. Figure 2.3 illustrates that the odd-even hopscotch method relies on a bipartite grid, where odd cells have even neighbors and vice versa. In this context, I have adapted the algorithm to incorporate convection, consistently considering it at the updated time level to improve stability. Before addressing implicit handling, we explicitly handle the radiation term [49]. The following equations are employed:

$$\text{First stage}: \quad u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i - \Delta t\sigma\left(u_i^n\right)^4}{1+\Delta tK} . \tag{2.22}$$

$$\text{Second stage}: \quad u_i^{n+1} = \frac{u_i^n + A_i^{\text{new}}}{1+r_i+\Delta tK+\Delta t\sigma\left(u_i^n\right)^3} , \tag{2.23}$$

where A_i^{new} is calculated in the same way as A_i in Eq. (2.15), but using the new values of the temperatures, which make the implicit formula effectively explicit.

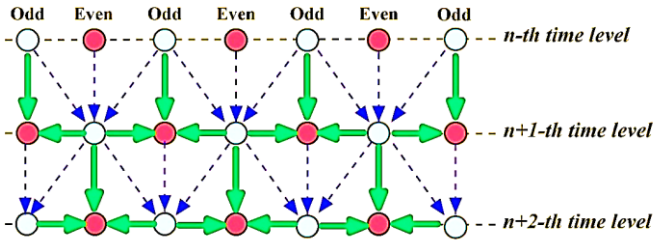


Figure 2.3. The stencil of the original odd-even hopscotch algorithm. Thin blue arrows and thick green arrows indicate operations at the first stage and second stage, respectively.

2.2.5. The asymmetric hopscotch method (ASH)

Asymmetric-hopscotch (ASH) scheme repeating unit consists of two half and one full-size stages. First a half-sized time step (light green rectangle with the number '1′) is taken for the odd nodes with $\theta = 0$, and then a full-length step for the even nodes (light orange rectangle) using $\theta = 1/2$, and finally a halved third stage (dark orange box) closes the calculation with $\theta = 1$.[50] .

2.2.6. The constant neighbor (CNe) method

The constant neighbor (CNe) method [51], [52] for Eq. (2.2) is:

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1-e^{-2r}\right), \tag{2.24}$$

While for general grids it is: $u_i^{n+1} = u_i^n \cdot e^{-r_i} + \frac{A_i}{r_i}\left(1-e^{-r_i}\right)$.

2.2.7. The Linear-Neighbour Method LNe

The two and three stage linear-neighbor (LNe and LNe3) methods, they are based on the CNe method, which is used as a predictor to calculate new $u_{\mathrm{i}}^{\mathrm{pred}}$ values valid at the end of the actual time step. [53]. Using them, we can calculate slopes

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

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

$$u_i^{n+1} = u_i^n e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-2r}\right) + s_i \frac{\Delta t^2}{2r}\left(1 - \frac{1 - e^{-2r}}{2r}\right). \tag{2.25}$$

$$\text{For the general case,} \quad A_i^{\mathrm{new}} = \Delta t \sum_{\mathrm{j}\neq\mathrm{i}} \frac{u_{\mathrm{j}}^{\mathrm{pred}}}{C_i R_{\mathrm{ij}}} \tag{2.26}$$

with which I can make the corrector step as follows:

$$u_{\mathrm{i}}^{\mathrm{n+1}} = u_{\mathrm{i}}^{\mathrm{n}} e^{-r_{\mathrm{i}}} + \left(A_{\mathrm{i}} - \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}\right)\frac{1 - e^{-r_{\mathrm{i}}}}{r_{\mathrm{i}}} + \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}. \tag{2.27}$$

The values given in Eq. (2.27) can be used to recalculate A_i^{new} again, which makes sense to repeat (2.27) to obtain new results. In this case, we have three stages altogether, thus the method is called the LNe3 method [53]. This algorithm is still second order, but more accurate than LNe2.

2.2.8. The CpC algorithm

The two-stage Constant-neighbor (briefly: CpC) algorithm [54] This generally starts with a fractional time step with length $p\Delta t$, but here we take $p = 1/2$, because it is the simplest and usually the most accurate choice. So, at the first stage, we can calculate new predictor values of the variables, but with a $\Delta t_1 = \Delta t / 2$ time step:

$$u_i^{\mathrm{pred}} = u_i^n \cdot e^{-r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-r}\right) \text{ and } u_{\mathrm{i}}^{\mathrm{pred}} = u_i^n e^{-r_i/2} + \frac{A_i}{r_i}\left(1 - e^{-r_i/2}\right).$$

In the second stage, we can use (2.26) with Δt_1 and take a full-time step size corrector step using the CNe formula again. Thus, the final values at the end of the time step are

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^{\mathrm{pred}} + u_{i+1}^{\mathrm{pred}}}{2}\left(1 - e^{-2r}\right) \text{ and } u_i^{n+1} = u_i^n \cdot e^{-r_{\mathrm{i}}} + \frac{A_i^{\mathrm{new}}}{r_{\mathrm{i}}}\left(1 - e^{-r_{\mathrm{i}}}\right).$$

2.2.9. Heun's method

This method also called explicit trapezoidal rule, may be the most common second-order RK scheme [55]. It starts with an explicit Euler stage as a predictor:

$$u_i^{\text{pred}} = (1-2r)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) \text{ and } u_i^{\text{pred}} = (1-r_i)u_i^n + A_i,$$

then using the average of the obtained and the old values a corrector-step follows:

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

$$\text{and } u_i^{n+1} = u_i^n - r_i\frac{u_i^n + u_i^{\text{pred}}}{2} + \frac{A_i + A_i^{\text{new}}}{2}.$$

2.3. MATLAB Built in Solvers

MATLAB solvers have been used for comparison purposes, namely ode15s, ode23t, ode23s, ode23tb, ode23, ode45, and ode113. While implicit solutions are used for the other odes, it is known that odes 45, 23, and 113 employ explicit methods. Also, we use preconditioned conjugate gradient (PCG) method [56] which has been implemented by the built-in routine of MATLAB called pcg. The conjugate gradient method is a non-classical iterative method which can be used for solving linear equation systems with symmetric, positive definite coefficient matrix. In general, the conjugate gradient method yields high accuracy numerical solutions in the so-called A-norm. However, the convergence rate strongly depends on the spectral features of the coefficient matrix, thus it can be very slow for stiff problems. Hence, one can apply preconditioning, i.e., transforming the linear equation system into another linear equation system which is equivalent in the sense that it has the same solution, but it has more favorable spectral features. As a consequence, one loses some accuracy but can reach more favorable convergence rates. Finally, the GMRES (generalized minimal residual) method [57], [58]. It is a non-classical iteration method for solving linear systems of equations which are not necessarily symmetric. The essence of the method is to find an approximate solution of linear equation system, which is the most accurate approximation in the Euclidean norm if we consider a Krylov sub-space with a given rank. The GMRES method has been implemented in the gmres built-in routine of MATLAB. Since the time step sizes cannot be calculated explicitly for the MATLAB solvers, we instead define the tolerances, beginning with a large value, such as $\text{Tol} = 10^2$ until an extremely small minimum value, which is $\text{Tol} = 10^{-6}$ [42], [59].

2.4. ANSYS Workbench

ANSYS software package is a powerful computational fluid dynamics (CFD) widely used for simulating fluid flow, heat transfer, and other related phenomena. Specifically, it excels in modeling heat transfer through various building components, such as walls and roofs. I use this package to compare and simulate my cases. ANSYS allows one to perform transient thermal analyses, which are essential for studying heat transfer over time. Transient simulations capture dynamic behavior, accounting for changes in temperature distribution as conditions develop. Setting up precise boundary conditions is crucial. The initial temperatures, heat sources (such as solar radiation), and convective heat transfer coefficients at surfaces should be defined. ANSYS

provides tools to specify these conditions accurately, ensuring realistic simulations. Heat transfer is governed by the energy equation. It accounts for conduction, convection, and radiation effects. To predict temperature distributions within building components, ANSYS solves this equation numerically [60].

2.5. Hourly Analysis Program (HAP)

Carrier HAP is a versatile software package designed for evaluating the efficiency and cost-effectiveness of HVAC system designs in commercial buildings. By integrating energy analysis tools into a single, user-friendly bundle, significant time savings can be achieved. Notably, input data and results from system design simulations can often be seamlessly used for energy assessments without additional adjustments. With features such as weather data customization, space selection, building type specification, and component/material input, Carrier HAP streamlines the process. Users simply input relevant data to achieve desired design conditions and ensure occupant comfort [60].

3. SYSTEMATIC TESTING OF EXPLICIT POSITIVITY PRESERVING ALGORITHMS FOR THE HEAT-EQUATION

In this chapter, I performed systematic tests of recently invented stable and explicit algorithms which preserve the positivity of the solution for the linear heat equation. It is well known that the widely used explicit finite difference schemes are typically unstable if the time step size is below the so called CFL limit, and even if they are stable, they can produce negative temperatures. However, the numerical solutions should satisfy the same properties as the exact solution, such as positivity. Thus, I collected the available explicit positivity preserving methods, most of them created by us recently to examine their performance and relative competitiveness. I tested them in the case of several 2D systems to find how the errors depend on the stiffness ratio and the CFL limit of the system for each algorithm. Then I created an anisotropic but equidistant grid by shrinking the vertical dimension of the 2D system and investigated how this kind of anisotropy effects the errors [61].

3.1. General definitions and investigation circumstances

I consider the 2D system thermally isolated (zero Neumann boundary conditions), for matrix system see section 2.1.

In the first type of experiments, randomly generated cell capacities and thermal resistances following a log-uniform distribution

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

have been given to the cells. The parameters α_C, β_C, α_{Rx}, β_{Rx}, α_{Rz}, β_{Rz} of the distribution of the mesh-cells data have been chosen to construct test problems with various stiffness ratios. More concretely, I used the parameters shown in Table 3.1.

Table 3.1. The parameters used in algorithms.

Type	No.	C_{min}	C_{max}	R_{xmin}	R_{xmax}	R_{zmin}	R_{zmax}
Non-Stiff	1	0	0	0	0	0	0
	2	-1	1	0	0	0	0
Moderately Stiff	3	-1	1	-1	1	0	0
	4	-1	1	-1	1	-1	1
Medium Stiff	5	-2	2	-1	1	-1	1
	6	-2	2	-2	2	-1	1
	7	-2	2	-2	2	-2	2
Very Stiff	8	-3	3	-2	2	-2	2
	9	-3	3	-3	3	-2	2
	10	-3	3	-3	3	-3	3

The size of the grid is fixed to $N_x = 50$ and $N_z = 50$, thus the total cell number is 2500, while the final time is $t_{\text{fin}} = 0.2$ s.

The numerical error is calculated by comparing my numerical solutions $u_{\text{j}}^{\text{num}}$ produced by the examined method with the reference solution $u_{\text{j}}^{\text{ref}}$ at final time t_{fin} .The reference solution is given by the "ode15s" routine of MATLAB with large prescribed accuracy. I use the three types of (global) error [62] the L_∞ maximum error defined in Eq. (3.1), I also use the average error in Eq. (3.2)

$$\text{Error}(L_\infty) = \max_{0\le \text{j}\le N} \left| u_{\text{j}}^{\text{ref}}(t_{\text{fin}}) - u_{\text{j}}^{\text{num}}(t_{\text{fin}}) \right| \tag{3.1}$$

$$\text{Error}(L_1) = \frac{1}{N} \sum_{1\le j\le N} \left| u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}}) \right| \tag{3.2}$$

and the so-called energy error:

$$\text{Error}(Energy) = \sum_{1\le j\le N} C_{\text{j}} \left| u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}}) \right| \tag{3.3}$$

which, in case of heat transfer, gives the error in terms of energy.

For different algorithms these errors depend on the time step size (and the system parameters) in different ways. Thus, I first calculated the solution with a large time step size (which was $t_{\text{fin}}/4$), then repeated the calculation for subsequently halved time step sizes *S=15* times until *h* reached a small value. Now the aggregated relative error (ARE) quantities for each type of errors defined above are calculated as an average of these errors. For example, in the case of the L_∞ error, it has the following form:

$$\text{ARE}(L_\infty) = \frac{1}{S} \sum_{i=1}^{S} \log\left(\text{Error}(L_\infty)\right) \tag{3.4}$$

Finally, the simple average of the three kinds of errors also can be calculated:

$$\text{ARE} = \frac{1}{3}\left(\text{ARE}(L_\infty) + \text{ARE}(L_1) + \text{ARE}(Energy)\right) \tag{3.5}$$

3.2. Comparison between positivity preserving methods for a mildly stiff system

First, I examined the following concrete parameter-combination,

$$\alpha_c = -1, \beta_c = +1,\ \alpha_{Rx} = -1, \beta_{Rx} = +1,\ \alpha_{Rz} = 0, \beta_{Rz} = 0,$$

which configuration has number 3 in Table 3.1 and categorized as mildly stiff.

I have plotted the L_∞ errors as a function of the effective time step size Δt_{EFF} and as a function of the running times for all methods. In Figure 3.1 I present the L_∞ error as a function of the effective time step size Δt_{EFF} for the nine positivity preserving methods defined in Section 3, while in Figure 3.2 one can see the L_∞ errors vs. the total running times.

One can see the LNe3 scheme is the most accurate, but the accuracy of the LH-CNe as well as the SH-CNe and ASH-CNe methods [61] approach it. However, since LNe3 is a three-stage method, it is slightly slower for the same accuracy than the LH-CNe, SH-CNe and ASH-CNe.

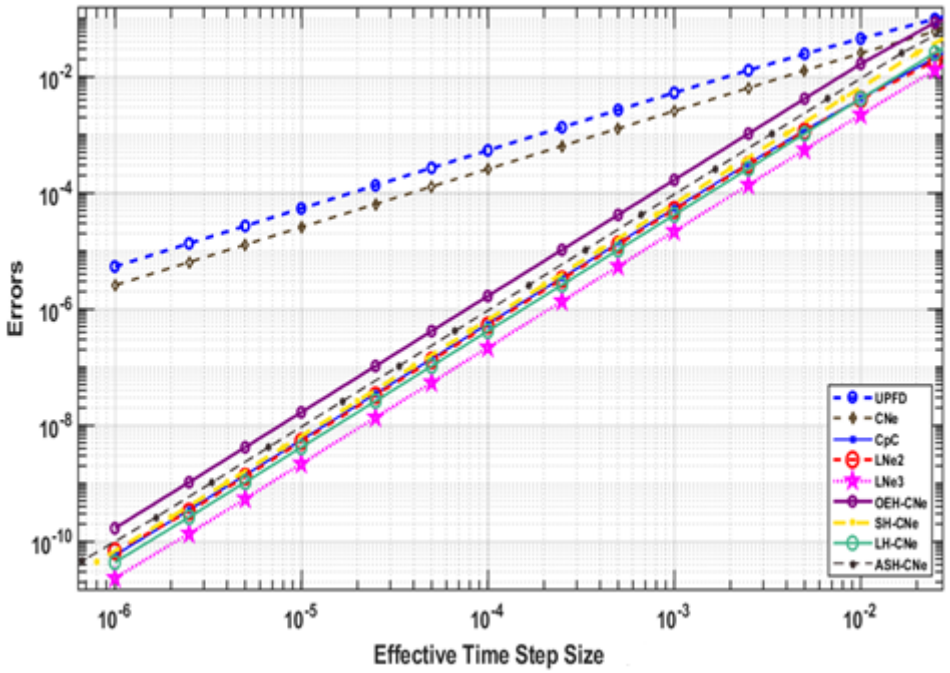


Figure 3.1. The L_∞ errors as a function of the effective time step size (Δt_{EFF}) for the (Mildly Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

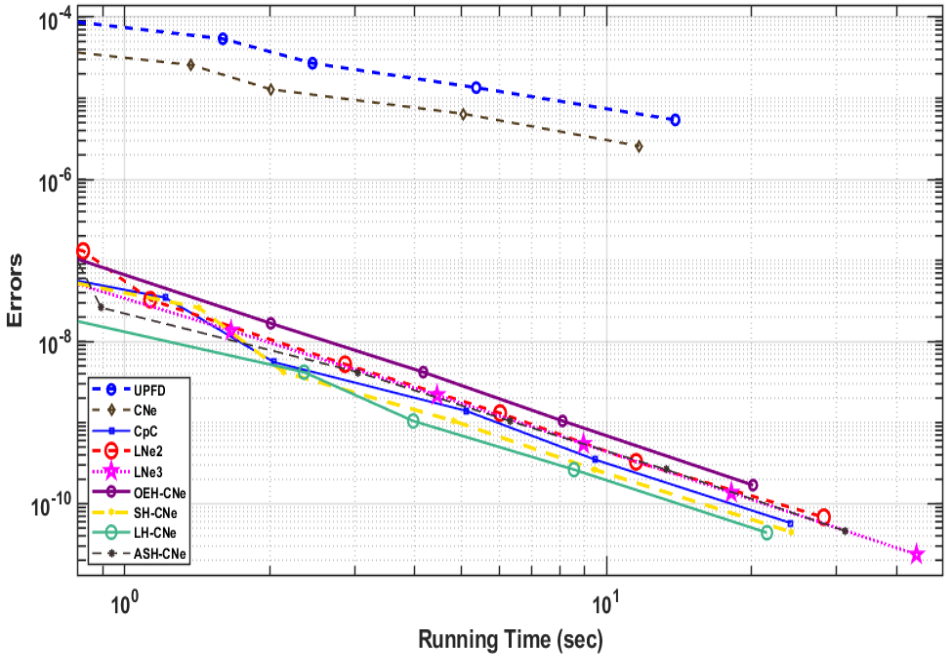


Figure 3.2. The L_∞ errors as a function of the running times for the (mildly stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

3.3. *Comparison between positivity preserving methods for a large, very stiff system*

I put new values for the α and β parameters for the very stiff system in the second case study:

$$\alpha_c = -3, \beta_c = +3,\ \alpha_{Rx} = -2, \beta_{Rx} = +2,\ \alpha_{Rz} = -2, \beta_{Rz} = +2,$$

In Figure 3.3 and Figure 3.4, the L_∞ error is presented as a function of the effective time step size Δt_{EFF} and the total running time, respectively.

I can see that in this case, the LH-CNe method outperforms all other examined positivity preserving methods provided that not only the accuracy, but also the speed, is taken into account. In Table 3.2, the data related to this numerical experiment is reported.

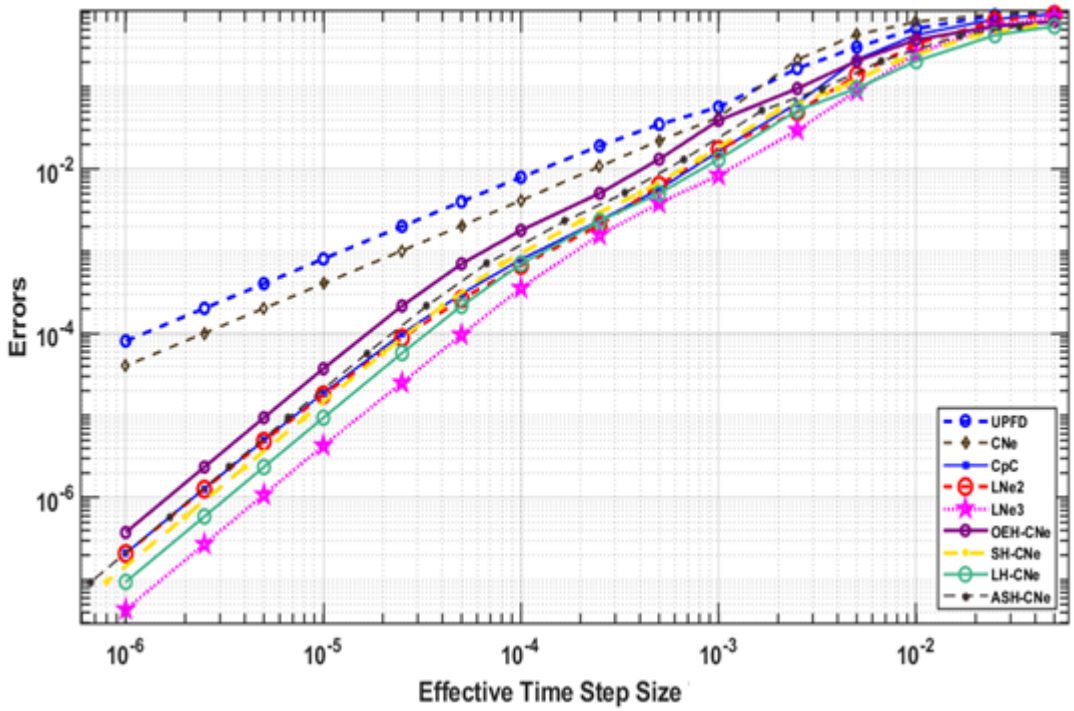


Figure 3.3. The L_∞ errors as a function of the effective time step size (Δt_{EFF}) for the (very Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

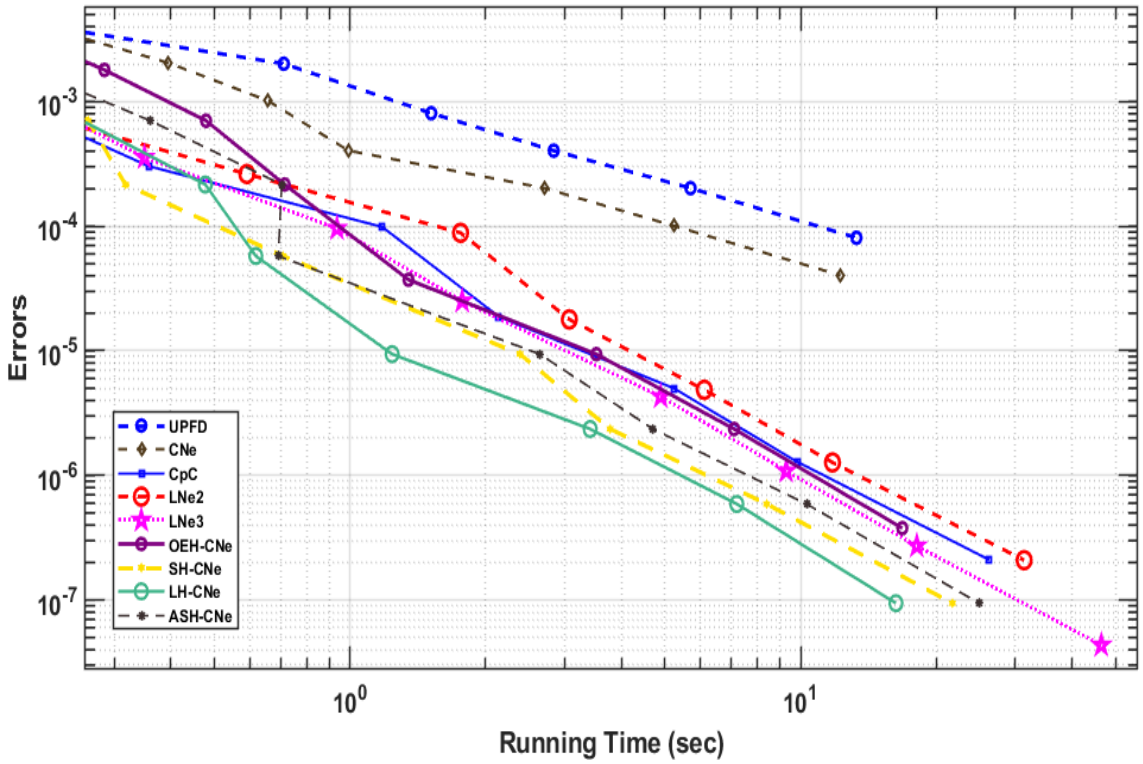


Figure 3.4. The L_∞ errors as a function of the running times for the (very Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

Table 3.2. Comparison the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods for the very stiff system of 2500 cells.

Numerical Method	Error(L_∞)	Error(L_1)	Energy Error	Running Time (sec)
UPFD, $\Delta t = 5\times10^{-5}$	4.012×10^{-3}	1.44×10^{-4}	0.753	0.2128
CNe, $\Delta t = 5\times10^{-5}$	2.046×10^{-3}	6.46×10^{-5}	0.4042	0.3947
CpC, $\Delta t = 5\times10^{-5}$	3.034×10^{-4}	7.093×10^{-6}	0.0333	0.357
LNe2, $\Delta t = 5\times10^{-5}$	2.622×10^{-4}	7.8522×10^{-6}	0.0448	0.5898
LNe3, $\Delta t = 5\times10^{-5}$	9.6×10^{-5}	3.22×10^{-6}	0.02698	0.935
OEH-CNe, $\Delta t = 5\times10^{-5}$	7×10^{-4}	1.845×10^{-5}	0.995×10^{-1}	0.4788
SH-CNe, $\Delta t = 5\times10^{-5}$	2.16×10^{-4}	5.49×10^{-6}	0.315×10^{-1}	0.317
LH-CNe, $\Delta t = 5\times10^{-5}$	2.16×10^{-4}	5.49×10^{-6}	0.315×10^{-1}	0.477
ASH-CNe, $\Delta t = 5\times10^{-5}$	9.4×10^{-6}	2.44×10^{-7}	1.5×10^{-3}	0.7048

3.4. Comparison the ARE errors between positivity preserving methods as a function of Δt_{MAX} and stiffness ratio

Figure 3.5 and Figure 3.6 show *ARE* errors as a function of Δt_{MAX} and stiffness ratio, respectively. I note the stiffness ratio affected the accuracy of methods when they increased, so the accuracy becomes worse compared to the cases of small stiff ratios.

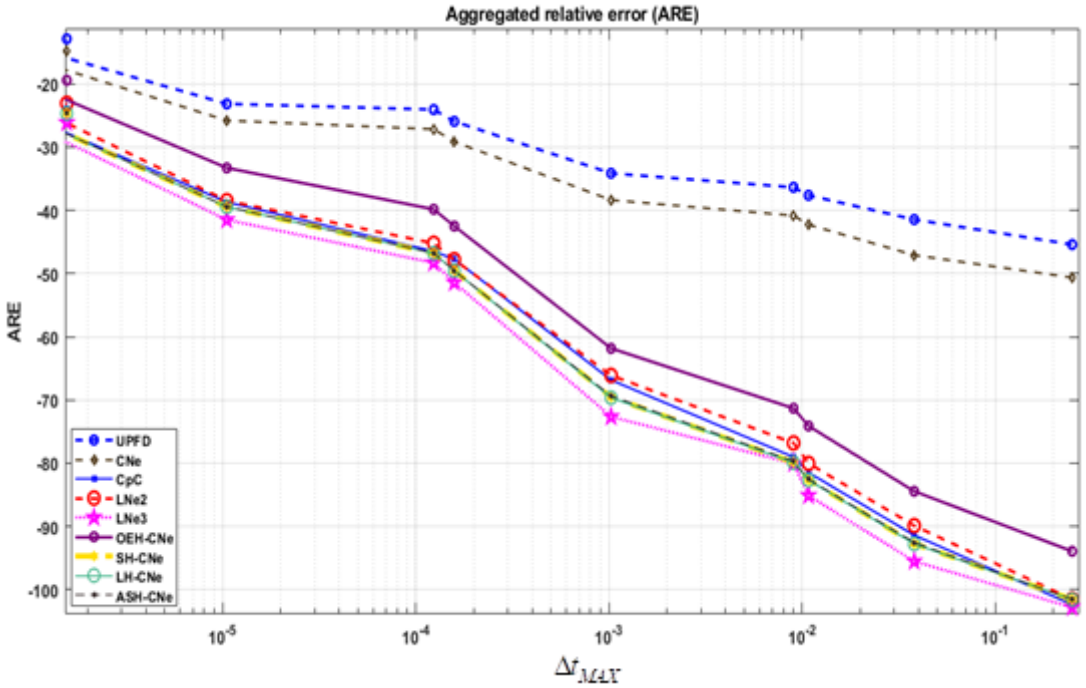


Figure 3.5. The ARE errors as a function of Δt_{MAX} in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

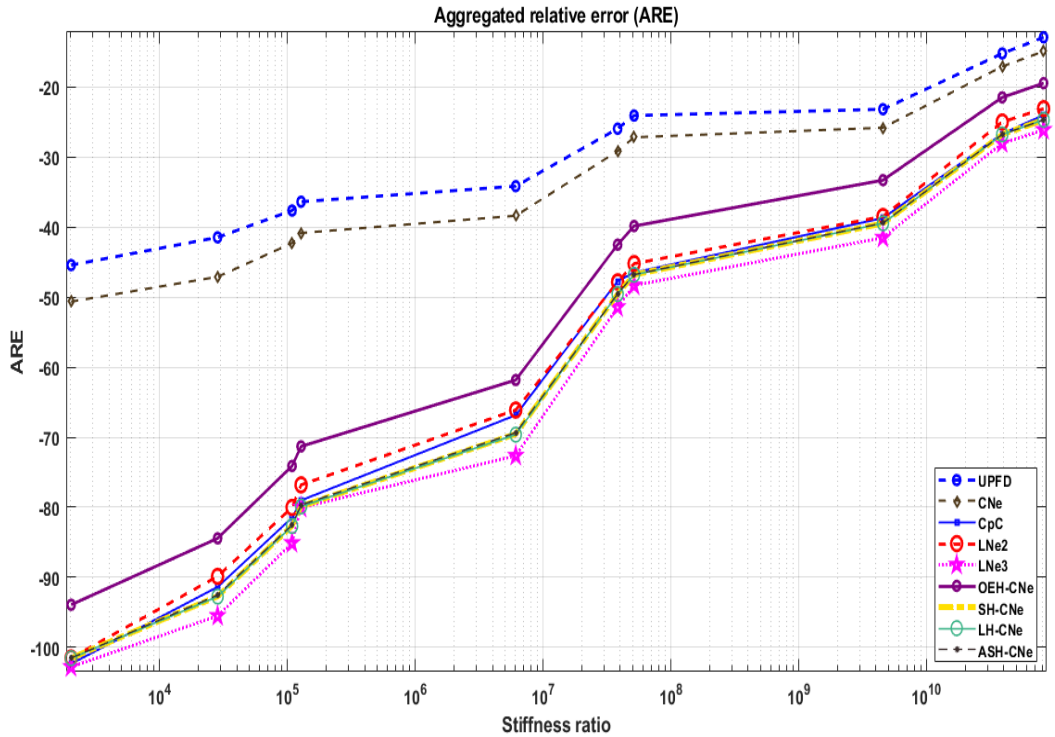


Figure 3.6. The ARE errors as a function of Stiff Ratio in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

I summarize the ARE error quantities, defined in Eq. (3.5), for both case studies in the following Table 3.3:

Table 3.3. ARE (average relative error) quantities of different explicit stable algorithms.

Numerical Method	ARE (Mildly Stiff)	ARE (Very Stiff)
UPFD	–37.4544	–23.1613
CNe	–42.0347	–25.9
CpC	–80.778	–40.07
LNe2	–79.6922	–39.228
LNe3	–84.346	–43.75
OEH-CNe	–72.9442	–35.09
SH-CNe	–81.467	–41.4367
LH-CNe	–81.4812	–41.428
ASH-CNe	–81.376	–41.394

3.5. Comparison the ARE errors between positivity preserving methods as a function of anisotropy coefficient (AC)

First, I solve PDE Eq. (2.2) with $\alpha = 1$, on the unit square $(x,z) \in [0,1] \times [0,1]$. The initial condition is the product of two sine functions:

$$u(x,z,t=0) = \sin(\pi x)\sin(k\pi z) \tag{3.6}$$

where the wave number k is currently fixed to k =1. The simplest zero Dirichlet boundary conditions are used

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

The analytical solution of this problem is obviously

$$u(x,t) = \sin(\pi x)\sin(k\pi z)\,\mathrm{e}^{-\left(1+k^2\right)\pi^2 t} \tag{3.7}$$

I apply an equidistant grid to discretize the space variables first I take $\Delta x = 0.02$, $\Delta z = 0.02$.

The number of cells along the axis x and z are set again to $N_x = 50$ and $N_z = 50$. Thus, I have a grid with total cell number $N = N_x \times N_z = 50 \times 50 = 2500$.

Then I performed systematic experiments by decreasing the dimension of the system as well as the cells in the z direction to introduce anisotropy into the grid. It means Δz is subsequently decreased by a factor of 2, first to $\Delta z = 0.5$, then to $\Delta z = 0.25$, etc. It is convenient to introduce the following anisotropy coefficient:

$$AC = \frac{\Delta x}{\Delta z}$$

Then I examined the aggregated errors as a function of this anisotropy coefficient AC. Of course, the initial condition function in Eq. (3.6) and the exact solution must be also adjusted with recalculating the wave number $k = 2^{AC-1}$. In Figure 3.7, ARE errors are presented as a function of the anisotropy coefficient AC. I note the relative advantage of LNe3 method increased whereas the relative disadvantage of other methods is also increased. In Table 3.4, I give Δt_{MAX} and the stiffness ratio for some AC values.

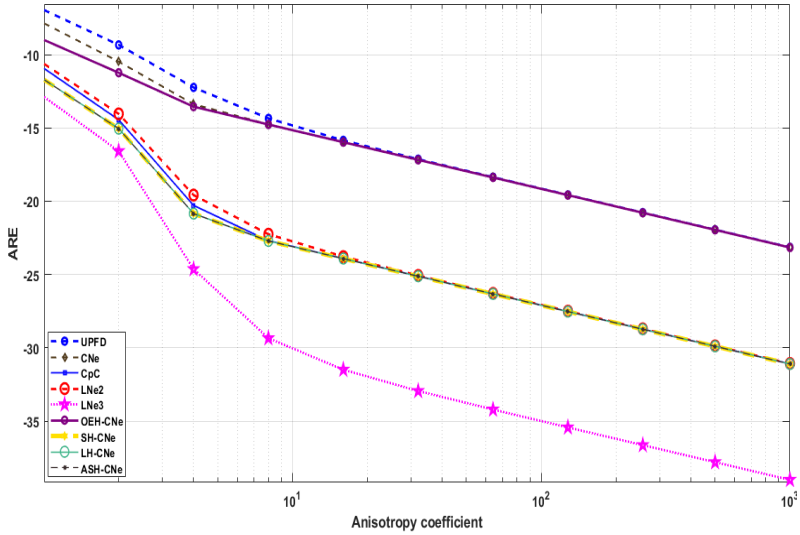


Figure 3.7. The ARE errors as a function of anisotropy coefficient AC in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

Table 3.4. Δt_{MAX} and stiffness ratio quantities of different AC values.

Anisotropy coefficient	Δt_{MAX}	**Stiffness Ratio**
1	1.0423×10^{-4}	$2.025\times10^{+3}$
10	2.0639×10^{-6}	$1.0227\times10^{+5}$
100	2.084×10^{-8}	$1.0127\times10^{+7}$
1000	2.084×10^{-10}	$1.0123\times10^{+9}$

3.6. Summary of this chapter

In this chapter, I conducted systematic tests of nine explicit numerical algorithms which were introduced in previous papers to solve the heat equation. All of the methods preserve the positivity of the solutions, thus stable regardless of the time step size and the stiffness of the system. First, I examined two-dimensional stiff systems. I observed that the 3-stage LNe3 method produced the most accurate results for a given time step size, but the LH-CNe method (and sometimes the SH-CNe method) requires the least CPU time to reach any prescribed accuracy. The increasing stiffness ratio and the decreasing CFL limit decreased the accuracy of methods, and the advantage of the best methods compared to the worst are decreased.

I also examined the performance of the method for different levels of spatial anisotropy. I obtained that if the difference between the horizontal and vertical dimensions of the cells are increasing, the advantages of the LNe3 method and the disadvantage of the first order methods are increasing.

I can conclude that if there is a possibility to construct the OEH structure (because the mesh is rectangular) than the hopscotch-CNe methods, especially the LH-CNe is the most effective among the positivity preserving methods, except when the anisotropy is strong. However, if an unstructured mesh is given, the LNe3 method is the most effective. For a very anisotropic system, for example a thin and wide layer, I propose the LNe3 method. Therefore, the time step size, the shape of the geometry, and the type of materials all affect the stiffness ratio of the system, thus affecting the stability of the methods.

4. EXPLICIT STABLE FINITE DIFFERENCE METHODS FOR DIFFUSION-REACTION TYPE EQUATIONS

In this chapter, I construct a new 2-stage explicit algorithm to solve partial differential equations containing a diffusion term and two reaction terms. One of the reaction terms is linear, which may describe heat convection; the other one is proportional to the fourth power of the variable, which can represent radiation. A member of our research group analytically proved, for the linear case, that the order of accuracy of the method is two, and that it is unconditionally stable. Then large systems with random parameters and discontinuous initial conditions are used to demonstrate that the new method is competitive against several other solvers, even if the nonlinear term is extremely large. Finally, I show that the new method can be adapted to the advection–diffusion-reaction term as well.

4.1. Construction of the New Method

In one space dimension, I take $x_i = i\Delta x,\ i = 0,...,N-1$, which is a common space discretization. Let us fix the time discretization to $t_n = t_0 + n\Delta t,\ n = 0,...,T,\ T = (t_{\text{fin}} - t_0)/\Delta t$. I introduce the mesh-parameter $r = \frac{\alpha \Delta t}{\Delta x^2}$ and $\mu = \frac{a\Delta t}{\Delta x}$. The original UPFD method applies the most common spatial discretization of the diffusion term based on the central difference formula, while it applies the backward difference formula for the advection term. However, the time levels are treated in a tricky way [63], such that the neighbors are taken into account fully at the old time level, where their values are known, and only the actual cell is treated implicitly. It means that for example u_{i-1}^{n} is used instead of u_{i-1}^{n+1}, with which they obtained:

$$\frac{u_i^{\mathrm{n}+1} - u_i^{\mathrm{n}}}{h} = \alpha \frac{u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}+1} + u_{i+1}^{\mathrm{n}}}{\Delta x^2} - a \frac{u_i^{\mathrm{n}+1} - u_{i-1}^{\mathrm{n}}}{\Delta x} - K u_i^{\mathrm{n}+1} \tag{4.1}$$

This can be arranged in a fully explicit form to obtain the following:

Algorithm 1, the original UPFD $$u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) + \mu u_{i-1}^{\mathrm{n}}}{1 + 2r + \mu + Kh} \tag{4.2}$$

Now I adapt this method to Eq. (2.4) where $a = 0$ but $\sigma > 0$. In principle the nonlinear term can be incorporated into this scheme in many different ways. I choose the following treatment: I insert the radiation term at the level of Eq. (4.1) as $u_i^4(t) \approx u_i^{n+1}\left(u_i^n\right)^3$, which again can be expressed in an explicit form, and with this I obtain the following adaptation of the original UPFD algorithm to Eq. (2.4):

Algorithm 2, UPDF for the diffusion-reaction-radiation Eq. (2.4)

$$u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + q_i h}{1 + 2r + K_i h + \sigma h\left(u_i^n\right)^3}$$

If r, q_i, K_i and σ have arbitrary nonnegative values and the values of u at the beginning of the time stapes are nonnegative, then both the numerator and the denominator are nonnegative in this formula. It means that this formula preserves positivity similarly to the original UPFD formula for the strongly nonlinear case as well. As I will see later, its accuracy is not very good, thus I proceed to construct a two-stage method as well.

I am going to combine the UPFD idea with the so called θ-method, which can be applied for the diffusion term in the following way:

$$u_i^{n+1} = u_i^n + r\left[\theta\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) + (1-\theta)\left(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}\right)\right] \tag{4.3}$$

where $\theta \in [0,1]$. If $\theta = 1$, this scheme is the forward-time central-space (FTCS) scheme, which is basically the explicit Euler time integration. For smaller values of θ, this formula is implicit, and for $\theta = 0, ½$ one has the implicit (Euler) and the Crank–Nicolson method, respectively. Using the trick above and incorporating the reaction and the source terms I can write:

$$u_i^{n+1} = u_i^n + r\left[-2\theta u_i^n - 2(1-\theta)u_i^{n+1} + u_{i-1}^n + u_{i+1}^n\right] - \Delta t K_i u_i^{n+1} + \Delta t q_i + \sigma u_i^{n+1}\left(u_i^n\right)^3 \tag{4.4}$$

If one takes $\theta = 0$, the original UPFD treatment is obtained back. The point is that this more general formula can also be easily rearranged to obtain an explicit formula, according to which the new value of the u variable has the following form in the 1D equidistant case:

Algorithm 3, theta-generalization of Algorithm 2

$$u_i^{n+1} = \frac{(1-2r\theta)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t q_i}{1 + 2r(1-\theta) + \Delta t K_i + \sigma \Delta t\left(u_i^n\right)^3} \tag{4.5}$$

Since I formally started from an implicit Eq. (4.3) but made it fully explicit, I started to call these methods pseudo-implicit. The main novelty of this chapter is that I organize Eq. (4.5) into a two-stage method as follows inspired by the well-known predictor-corrector methods [64]. 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 4, 2-stage pseudo-implicit method for the diffusion-reaction-radiation Eq. (2.4)

Stage 1. Take a partial time step $\Delta t_1 = p\Delta t,\ p > 0$ using Eq. (4.5) with parameter θ_1:

$$u_i^{\text{pred}} = \frac{(1-2pr\theta_1)u_i^n + pr\left(u_{i-1}^n + u_{i+1}^n\right) + q_i \Delta t_1 - v_1 K_i \Delta t_1 u_i^n}{1 + 2pr(1-\theta_1) + v_2 K_i \Delta t_1 + \sigma \Delta t_1\left(u_i^n\right)^3} \tag{4.6}$$

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

$$u_i^{\text{pred}} = \lambda u_i^{\text{pred}} + (1-\lambda)u_i^n \tag{4.7}$$

Take a full time step with the Eq. (4.5) with parameter θ_2:

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

where v_1, v_2, w_1, w_2 are real numbers which are considered as free parameters. I must mention that the mathematically correct form of Eq. (4.7) would be $u_i^{\text{lin}} = \lambda u_i^{\text{pred}} + (1-\lambda) u_i^n$, but I immediately put down it in the form which is to be used in a computer code to spare memory. I also note that with this treatment of the nonlinear term I obtain a second-order method with very good stability properties, as I will see later.

In the case when one has a general mesh and the material properties are functions of the space variables, the spatially discretized form of Eq. (2.4) can be generalized as in section 2.1.

Now I can write the modified UPFD and our pseudo-implicit algorithms in the general case as follows:

Algorithm 2G, UPDF for the diffusion-reaction-radiation equation, general mesh-form:

$$u_i^{n+1} = \frac{u_i^n + A_i + q_i \Delta t}{1 + 2r_i + K_i \Delta t + \sigma \Delta t \left(u_i^n\right)^3} \tag{4.8}$$

Algorithm 4G, 2-stage pseudo-implicit method for the diffusion-reaction Eq. (2.4), general-mesh form.

Stage 1. Take a partial time step $\Delta t_1 = \Delta t / 2\lambda$, $\lambda > 0$, with the Eq. (4.8):

$$u_i^{\text{pred}} = \frac{\left(1 + \left(1 - 1/\lambda\right) r_i\right) u_i^n + A_i + q_i \Delta t_1}{1 + r_i + K_i \Delta t_1 + \sigma \Delta t_1 \left(u_i^n\right)^3}$$

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

Take a full time step with the Eq. (4.8):

$$u_i^{n+1} = \frac{(1 - r_i) u_i^n + A_i + K_i \Delta t \left(u_i^{\text{pred}} - u_i^n\right) + q_i \Delta t}{1 + r_i + K_i \Delta t + \sigma \Delta t \left(u_i^{\text{pred}}\right)^2 u_i^n}$$

I emphasize that in Algorithm 4G, $r_i = \dfrac{\Delta t}{2\tau_i}$ in both stages. I stress again that Algorithm 4 is proven to be unconditionally stable only for $\lambda = 1/2$.

4.2. Numerical Results

4.2.1. Comparison with Other Methods for a Large, Extremely Stiff System

In this subsection, I solve Eq. (2.4) in a two space dimensional, topologically rectangle-structured mesh with $N = N_x \times N_z$ cells (see Figure 2.1 for visualization). The size of the system is fixed to $N_x = 100$ and $N_z = 120$, thus the total cell number is 12,000. Randomly generated cell capacities and thermal resistances

$$C_i = 10^{(\alpha_C - \beta_C \times rand)},\ R_{x,i} = 10^{(\alpha_{Rx} - \beta_{Rx} \times rand)},\ R_{z,i} = 10^{(\alpha_{Rz} - \beta_{Rz} \times rand)} \tag{4.9}$$

have been used by a log distribution, where the (pseudo)random number rand is generated by MATLAB for each quantity with a uniform distribution in the unit interval (0, 1). In this subsection, $K = 0$ and $\sigma = 0$, thus I deal with the linear heat equation and $M = M^D$. The exponents have been set to the following values:

$$\alpha_C = \alpha_{Rx} = \alpha_{Rz} = -3, \;\; \beta_C = \beta_{Rx} = \beta_{Rz} = 3,$$

which means that log-uniformly distributed values between 0.001 and 1000 have been given to the capacities and the resistances. Different random values have been generated for the initial conditions $u_i(0) = rand$ and the source term $q_i = 0.2 \times rand - 0.1$ as well. The final time of the simulation has been set to $t_{\text{fin}} = 0.2$.

I consider zero Neumann boundary conditions (isolated system). To implement this, I omit those terms of the sum in Eq. (2.4) which have infinite resistivity in the denominator because of thermal isolation at the boundary. If the (nonzero) smallest and the largest absolute value eigenvalues of the system matrix M, defined in Eq. (2.10) – (2.13), [65], are denoted at the end of section 2.1 In the present case, the stiffness ratio is 2.3×10^{11} and $\Delta t_{\text{MAX}}^{\text{FTCS}} = 1.03 \times 10^{-6}$, respectively. I will see that this implies serious under-performance of the conventional explicit methods, which are only conditionally stable.

In Section 4.2, the reference solution is obtained using the ode15s built-in solver of MATLAB with sufficiently strict error tolerance $\text{'Tol'} = 10^{-12}$ (where $\text{Tol} \doteq \text{'AbsTol'} = \text{'RelTol'}$) and therefore high precision. The equations of error defined in Eq. (3.1) - (3.3).

The performance of the new algorithms was compared with the methods in section 2.2 and 2.3 coded by me.

For the calculations where running times are measured, a desktop computer with an Intel Core i7-9700 CPU, 16.0 GB RAM is used, while the software is the MATLAB R2020b [66]. The total running time of the algorithms is measured by the built-in tic-toc function of that software.

I have examined the L_∞, L_1 and energy errors as a function of the time step size Δt and the running time. In Figure 4.1 I present the L_1 error as a function of h, while in Figure 4.2 one can see the L_1 errors vs. the total running times. Table 4.1 collects some results which have been obtained by the numerical schemes coded by us and the "ode" solvers of MATLAB. I set $\lambda = 1$ as it is explained in Remark 1.

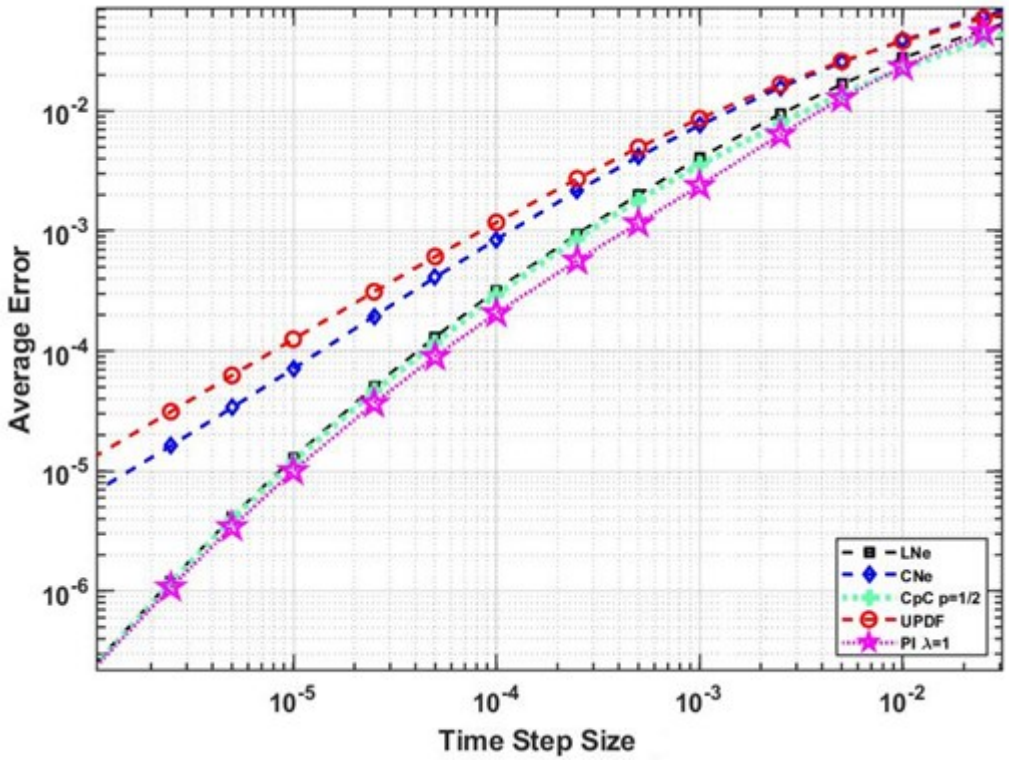


Figure 4.1. L1 (average) errors as a function of the time step size of the new pseudo-implicit (PI) algorithm and some other methods for the first, extremely stiff system with $K = 0$, $\sigma = 0$.

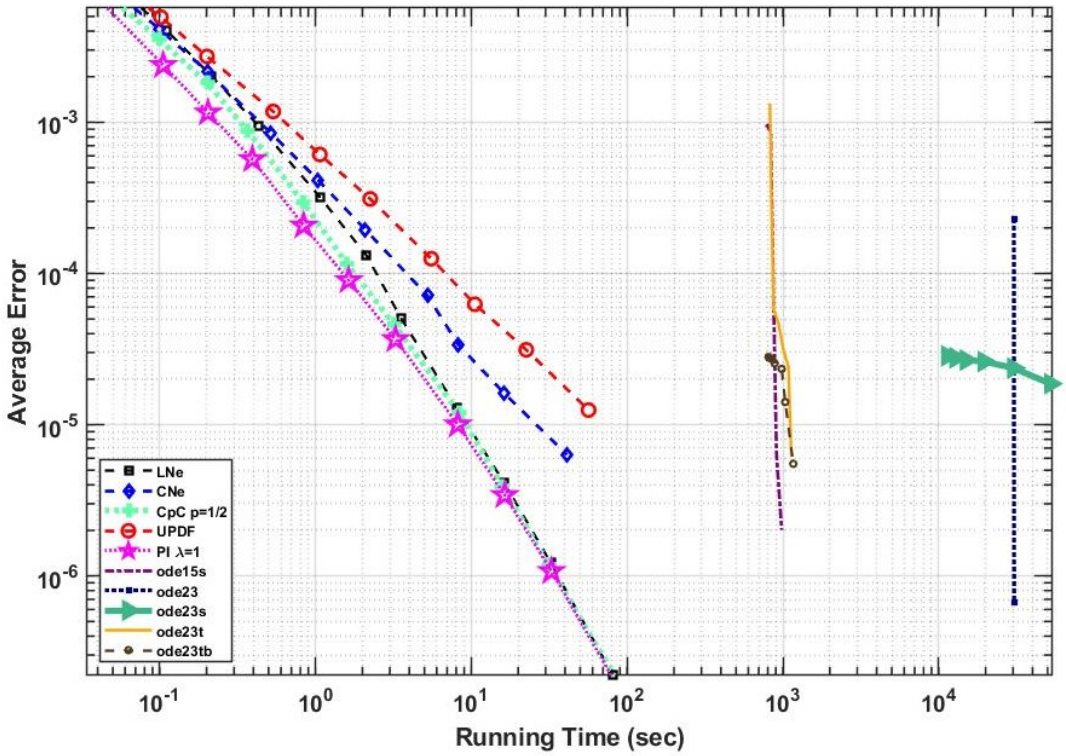


Figure 4.2. L1 (average) errors as a function of the running time for the first (extremely stiff) system, in the case of the algorithms coded by us as well as of the MATLAB routines.

Table 4.1. Comparison of different algorithms for the extremely stiff system of twelve thousand cells.

Numerical Method	Error(L_∞)	Error(L_1)	Energy Error	Running Time (sec)
ode23, Tol = 10^{-1}	7.15×10^{-3}	6.68×10^{-7}	1.80×10^{-5}	3.04×10^{5}
ode15s, Tol = 10^{-2}	1.30×10^{-3}	7.33×10^{-5}	8.79×10^{0}	8.69×10^{2}
ode23s, Tol = 10^{-2}	4.33×10^{-4}	2.37×10^{-5}	2.80×10^{0}	3.02×10^{5}
ode23t, Tol = 10^{-2}	5.71×10^{-4}	3.14×10^{-5}	3.75×10^{0}	1.00×10^{3}
ode23tb, Tol = 10^{-2}	4.28×10^{-4}	2.33×10^{-5}	2.77×10^{0}	9.82×10^{2}
UPFD, $\Delta t = 1 \times 10^{-6}$	2.20×10^{-3}	1.24×10^{-5}	4.86×10^{-1}	5.65×10^{1}
Heun, $\Delta t = 1 \times 10^{-6}$	1.23×10^{-11}	3.79×10^{-13}	4.01×10^{-8}	1.05×10^{2}
CNe, $\Delta t = 5 \times 10^{-6}$	5.85×10^{-3}	3.36×10^{-5}	1.28×10^{0}	8.28×10^{0}
LNe, $\Delta t = 1 \times 10^{-5}$	2.70×10^{-3}	1.28×10^{-5}	3.58×10^{-1}	8.07×10^{0}
CpC p = 1/2, $\Delta t = 2.5 \times 10^{-5}$	1.21×10^{-2}	4.61×10^{-5}	1.08×10^{0}	3.17×10^{0}
PI $\lambda = 1$, $\Delta t = 2.5 \times 10^{-5}$	8.44×10^{-3}	3.66×10^{-5}	8.62×10^{-1}	3.26×10^{0}
PI $\lambda = 1$, $\Delta t = 1 \times 10^{-5}$	2.54×10^{-3}	1.00×10^{-5}	2.41×10^{-1}	8.19×10^{0}
PI $\lambda = 1$, $\Delta t = 5 \times 10^{-6}$	9.25×10^{-4}	3.41×10^{-6}	8.50×10^{-2}	1.63×10^{1}

One can see that the new scheme is slightly more accurate than the LNe and the CpC, and significantly more accurate than the first order UPFD and CNe methods. I note that the Heun method is not present in the figures, because it is convergent only below the CFL limit, which is lower than the time step sizes presented in the case of my methods. The explicit MATLAB solvers ode45 and ode113 were not able to provide any meaningful results and in the case of the ode23, it was hard work to find those tolerances for which the method works, albeit very slowly. The implicit MATLAB routines performed usually much better, but even they are severely outperformed by the explicit and stable algorithms if running times are considered.

4.2.2. Comparison with Other Methods for a Large System with Strong Nonlinearity

In the second case study, I set $K_i = 3 \times rand$, $q_i = 2 \times rand$ and $\sigma = 1000$. The latter coefficient has been chosen so large because I would like to demonstrate the performance of the new method for a strongly nonlinear case, but the values of the variable u are typically between zero and one, thus their fourth power is usually a rather small number. I give new values to the α and β exponents:

$$\alpha_C = 3,\ \beta_C = 6,\ \alpha_{Rx} = \alpha_{Rz} = 3,\ \beta_{Rx} = \beta_{Rz} = 0$$

I calculate the stiffness ratio and the CFL limit in two different ways, both of them without taking into account the nonlinear term. If I use the full M matrix, I obtain that the stiffness ratio is 7.7×10^5, much smaller than in the previous case, while the CFL limit for the standard FTCS was $\Delta t_{MAX}^{EE} = 9.76 \times 10^{-4}$, which, I stress again, holds for the Heun method as well. If I use only the M^D matrix instead of M, the stiffness ratio is 6.8×10^9, while the CFL limit is $\Delta t_{MAX}^{EE} = 9.75 \times 10^{-4}$. The reason behind these numbers is that the eigenvalues close to zero have been significantly increased (in absolute value) by the new reaction term while those with large absolute values remained almost the same. All other parameters and circumstances, such as the size of the system and the range of the initial values are the same as in the previous subsection. I note that I were not able to adapt our previous methods CNe, LNe and CpC for the $K \neq 0,\ \sigma \neq 0$ case, nor when the advection term is present, without losing their order of convergence (that is why I started to develop the current methods), thus they are not presented in this and the next subsection. In Figure 4.3 and

Figure 4.4 the energy and the average errors are presented as a function of the time step size and the total running time, respectively. In Table 4.2. I report the data that belong to this numerical experiment.

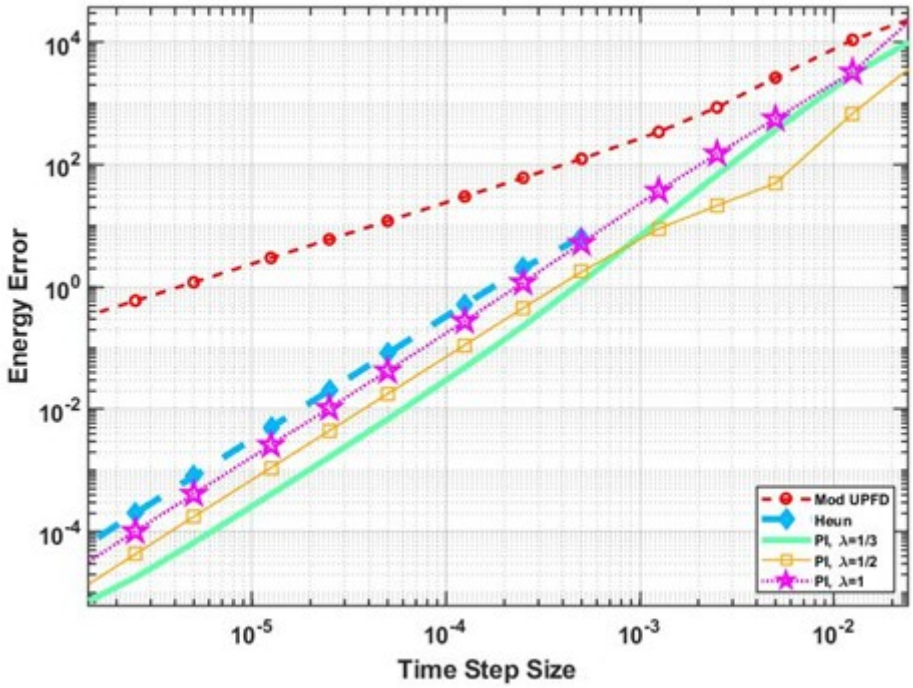


Figure 4.3. Energy errors as a function of the time step size for the second (very stiff) system, in the case of the UPFD Algorithm 2, the Heun method and the new PI algorithms.

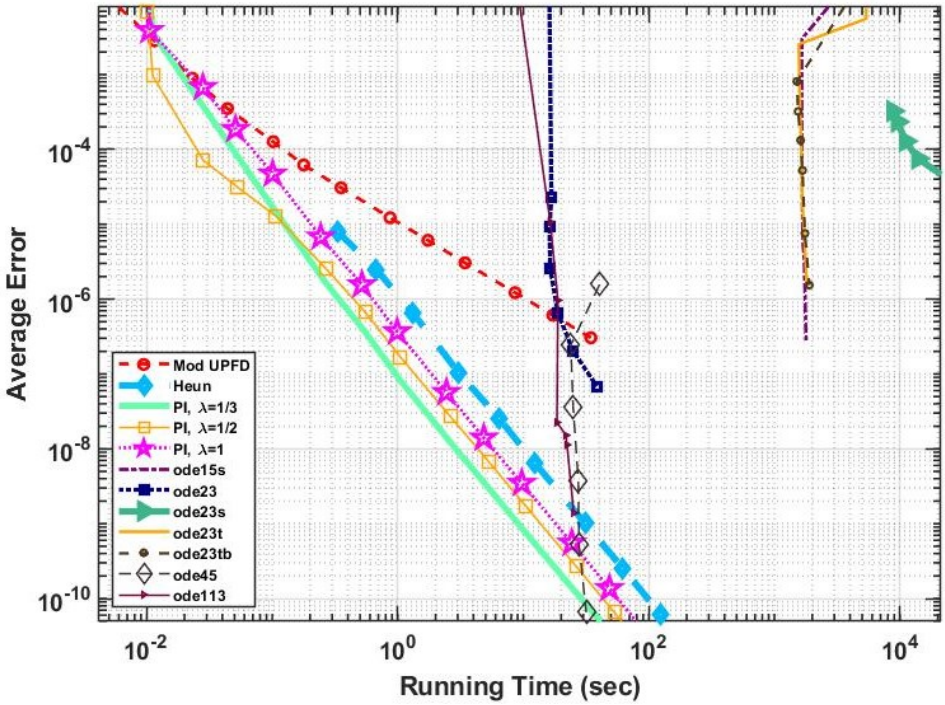


Figure 4.4. Energy (average) errors as a function of the running time for the second (very stiff) system, in the case of the new algorithms and some other methods.

Table 4.2. Comparison of different algorithms for the very stiff system of twelve thousand cells.

Numerical Method	Error(L_∞)	Error(L_1)	Energy Error	Running Time (sec)
ode45, Tol = 10^{-1}	4.49×10^{-3}	1.57×10^{-6}	8.54×10^{-1}	4.05×10^{1}
ode23, Tol = 10^{-1}	7.97×10^{-2}	2.25×10^{-6}	1.17×10^{1}	1.67×10^{1}
ode113, Tol = 10^{-1}	9.23×10^{-2}	1.07×10^{-5}	1.57×10^{0}	1.63×10^{1}
ode15s, Tol = 10^{-3}	3.14×10^{-4}	4.94×10^{-5}	4.20×10^{1}	1.65×10^{3}
ode23s, Tol = 10^{-4}	9.94×10^{-5}	2.31×10^{-5}	1.93×10^{1}	3.84×10^{4}
ode23t, Tol = 10^{-4}	6.78×10^{-5}	1.78×10^{-5}	1.50×10^{1}	1.68×10^{3}
ode23tb, Tol = 10^{-4}	1.41×10^{-4}	5.25×10^{-5}	4.48×10^{1}	1.67×10^{3}
UPFD, $\Delta t = 5 \times 10^{-5}$	1.79×10^{-4}	1.22×10^{-5}	1.19×10^{1}	8.67×10^{-1}
Heun, $\Delta t = 5 \times 10^{-4}$	1.12×10^{-4}	7.85×10^{-6}	6.50×10^{0}	3.33×10^{-1}
PI $\lambda = 1/3$, $\Delta t = 1.25 \times 10^{-3}$	3.92×10^{-4}	1.53×10^{-5}	1.19×10^{1}	1.04×10^{-1}
PI $\lambda = 1/2$, $\Delta t = 1.25 \times 10^{-3}$	3.85×10^{-4}	1.29×10^{-5}	8.97×10^{0}	1.06×10^{-1}
PI $\lambda = 1/2$, $\Delta t = 5 \times 10^{-4}$	7.76×10^{-5}	2.58×10^{-6}	1.76×10^{0}	2.65×10^{-1}
PI $\lambda = 1$, $\Delta t = 1.25 \times 10^{-3}$	4.17×10^{-4}	4.70×10^{-5}	3.71×10^{1}	1.00×10^{-1}

As it is expected, due to the larger CFL limit and weaker stiffness, the conventional explicit methods performed much better than the implicit ones, and especially the ode45 can compete with my methods if high accuracy is required. However, for low and medium accuracy requirements, the new pseudo-implicit method has the best performance.

4.3. Summary of this chapter

In this chapter I reached my goal to construct a fully explicit and stable numerical algorithm to solve the time-dependent diffusion (or heat) equation with linear and nonlinear reaction terms, where the latter represented heat loss due to radiation. Using the UPFD idea, I organized the theta-formula into a two-stage algorithm, where, in each stage, the latest available u values of the neighbors are used to make the originally implicit theta-formula completely explicit. I analytically proved for the linear case that the obtained method is second order in time step size and unconditionally stable.

Then two 2-dimensional stiff systems containing 12,000 cells with discontinuous random parameters and initial conditions were constructed. The performance of the new algorithm as well as several other methods was examined for these systems. According to the numerical results, the new method is quite competitive. It is second order and stable for the non-linear case as well, and it gives quite accurate results orders of magnitude faster than the professionally optimized MATLAB routines and it is more accurate than all other examined explicit and unconditionally stable methods. Although it is not positivity preserving as the original UPFD algorithm, it is stable for relatively large time step sizes as well, even if the nonlinearity is strong. Moreover, it is easy to implement and can be applied for un-structured grids as well. The conclusion is that this new pseudo-implicit algorithm has the most important advantages of the conventional explicit and the implicit methods at the same time.

5. TESTING SOME DIFFERENT IMPLEMENTATIONS OF HEAT CONVECTION AND RADIATION IN THE LEAPFROG-HOPSCOTCH ALGORITHM

Based on many previous experiments the most efficient explicit and stable numerical method to solve heat conduction problems is the leapfrog-hopscotch scheme. In previous chapter I made a successful attempt to solve the nonlinear heat conduction-convection-radiation equation by PI method. Now I implement the convection and radiation terms in several ways to find the optimal implementation. The algorithm-versions are tested by comparing their results to 1D numerical and analytical solutions. Then I perform numerical tests to compare their performance when simulating heat transfer of the two-dimensional surface and cross section of a realistic wall. The latter case contains an insulator layer and a thermal bridge. The stability and convergence properties of the optimal version are analytically proved as well.

5.1. The Examined Numerical Methods

5.1.1. The Leapfrog-Hopscotch Method for the Heat Conduction Equation

To use the leapfrog-hopscotch method, or any other odd–even hopscotch method, the space domain must be discretized using a special, so-called bipartite mesh. This means that the mesh is divided into two disjoint subsets. The nodes or cells belong to the first and second subsets that are labeled as odd and even, respectively. The main requirement is that all the immediate neighbors of the odd cells must be even and vice versa, just like on a checkerboard. I describe it in the case of a 1D interval $x \in [x_0, x_N]$, $L = x_N - x_0$ on which an equidistant grid is constructed with coordinates $x_0, x_1, ..., x_N$ of nodes, so $x_j = x_{j-1} + \Delta x,\ j = 1,...,N,\ \Delta x = L/N$. The time domain is $t \in \left[t^0, t^{\text{fin}}\right]$ and it is discretized as usual: $t^j = t^0 + j\Delta t,\ j = 1,...,T,\ \Delta t T = t^{\text{fin}} - t^0$, where Δt is the time step size. In all stages, the following version of the theta formula (obtained from the central difference formula for the space derivative) is used as a starting point:

$$u_i^{n+1} = u_i^n + r\left[u_{i-1}^n + u_{i+1}^n - 2\theta u_i^n - 2(1-\theta)u_i^{n+1}\right] \tag{5.1}$$

where $\theta \in [0,1]$. The space–time structure of the algorithm is presented in Figure 2.3, where one can see that the neighbors $u_{i\pm1}^n$ are almost always taken into account at the time level half way between the actual old and new time levels. The first stage has the length of a halved time step, and it calculates new values for the odd nodes using $\theta = 0$, thus I have the formula

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i}{1+r} \tag{5.2}$$

symbolized by thick red arrows in the figure. Then a full-time step is made with $\theta = 1/2$ for the even nodes using

$$u_i^1 = \frac{(1-r)u_i^0 + r\left(u_{i-1}^{1/2} + u_{i+1}^{1/2}\right) + \Delta t q_i}{1+r} \tag{5.3}$$

After this, full time steps are taken alternately for the odd and even nodes with Eq. (5.3), symbolized by blue and green arrows in the figure. Finally, a half-length time step (green arrows) must close the calculations with $\theta = 1/2$ for the odd nodes

$$u_i^T = \frac{\left(1-r/2\right)u_i^{T-1/2} + r/2\left(u_{i-1}^T + u_{i+1}^T\right) + \Delta t/2 \cdot q_i}{1+r/2} \tag{5.4}$$

The key point here is that the latest values of the *u* function are always used, which means that the time indices of the node variables have to be set according to this logic, in which the figure can help. For example, when the odd node value $u_i^{1+1/2}$ is calculated, u_{i-1}^1, u_{i+1}^1, and $u_i^{1/2}$ are used, etc.

5.1.2. Implementations of the Convection Term

Until this point, the LH algorithm has been given only for the conduction and the heat source term. Now, the $-Ku$ term is also included, which is done in several ways.

1. Explicit treatment means that one calculates the increment due to the term and simply adds it to the final value of the new *u*, just as it would happen with the explicit Euler method. For example, in the case of the first stage with a halved time step, the increment is $-\frac{\Delta t}{2} K u_i^0$, thus I have

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i}{1+r} - \frac{\Delta t}{2} K u_i^0 \tag{5.5}$$

I note that I exemplify the versions with a first-stage formula, since it is the least nontrivial due to the half-sized time step.

2. Quasi-exact treatment means that I analytically solve the ODE

$$\frac{du}{dt} = -K \cdot u \tag{5.6}$$

and then take the effect of the convection terms into account in a separate calculation at the end of each stage. For example, in the case of the first stage with a halved time step, I have

$$u_i^{1/2,\,\mathrm{temp}} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i}{1+r} \tag{5.7}$$

and

$$u_i^{1/2} = e^{-K\Delta t/2} u_i^{1/2,\,\mathrm{temp}} \tag{5.8}$$

I expect that this version has outstanding stability, since the absolute value of the solution is always smaller than the temporary value $u_i^{1/2,\,\mathrm{temp}}$, where only the conduction and the source terms are taken into account.

3. Pseudo-implicit treatment means that the *u* variable in the convection term is taken into account at the new time level, so Eq. (5.1) is extended as follows:

$$u_i^{n+1} = u_i^n + r\left[u_{i-1}^n + u_{i+1}^n - 2\theta u_i^n - 2(1-\theta)u_i^{n+1}\right] - K\Delta t u_i^{n+1} \tag{5.9}$$

With this, the K term turns up only in the denominator, thus the first stage formula is the following:

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i}{1 + r + \Delta t K / 2} \tag{5.10}$$

Note that in our group research paper [67], only this implementation was proposed.

4. Now, the u variable in the convection term is taken into account at the old time level, so in Eq. (5.9), the last term is changed to $-K\Delta t u_i^n$. This means that the K term turns up only *inside* the numerator, so the first stage formula is the following:

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0 / 2}{1 + r} \tag{5.11}$$

Due to the lack of a better name, I call this version temporarily 'inside'.

5. Mixed treatment means that I make a linear combination of the last two versions (pseudo-implicit and inside) at the level of Eq. (5.9), where the last term is changed to $-pK\Delta t u_i^{n+1} - (1-p)K\Delta t u_i^n$. The real parameter p has a similar role as that which θ has for the conduction case. Now, the formulas are the following:

$$\text{First stage: } u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - (1-p)K\Delta t u_i^0 / 2}{1 + r + pK\Delta t / 2}$$

$$\text{Intermediate stages: } u_i^{n+1} = \frac{(1-r)u_i^n + r\left(u_{i-1}^{n+1/2} + u_{i+1}^{n+1/2}\right) + \Delta t q_i - (1-p)K\Delta t u_i^n}{1 + r + pK\Delta t} \tag{5.12}$$

$$\text{Last stage: } u_i^T = \frac{\left(1 - r/2\right)u_i^{T-1/2} + r/2\left(u_{i-1}^T + u_{i+1}^T\right) + \Delta t/2 \cdot q_i - (1-p)K\Delta t u_i^{T-1/2} / 2}{1 + r/2 + pK\Delta t / 2}$$

I performed tests with several values of p, but here I present them only for three values, namely, $p = 1/3,\ 1/2,\ 2/3$.

5.1.3. Implementations of the Radiation Term

In these sections and next section, it will turn out that the mixed treatment with $p = 1/2$ is the most effective; thus, when defining different treatments of the radiation term, the convection term will be taken into account that way. The radiation term will be implemented in similar ways as the convection term, but now one has more possibilities.

1. Explicit treatment:

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0 / 2}{1 + r + K\Delta t / 4} - \sigma\Delta t\left(u_i^0\right)^4 / 2 \tag{5.13}$$

2. Quasi-exact treatment: The analytical solution of the ODE

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

is

$$u(t) = \left((u_0)^{-3} + 3\sigma t\right)^{-\frac{1}{3}} \tag{5.14}$$

This means that I have the following two sub-stages:

$$\text{Sub-stage 1: } u_i^{1/2,\text{temp}} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0/2}{1 + r + K\Delta t/4} \tag{5.15}$$

$$\text{Sub-stage 2: } u_i^{1/2} = \left(\left(u_i^{1/2,\text{temp}}\right)^{-3} + 3\sigma\Delta t\right)^{-\frac{1}{3}} \tag{5.16}$$

3. Pseudo-implicit treatment: Eq. (2.1) is now modified as

$$u_i^{n+1} = u_i^n + r\left[u_{i-1}^n - 2u_i^n + u_{i+1}^n - 2\theta u_i^n - 2(1-\theta)u_i^{n+1}\right] - K\Delta t u_i^{n+1}/2 - K\Delta t u_i^n/2 - \sigma\Delta t u_i^{n+1}\left(u_i^n\right)^3 \tag{5.17}$$

This yields

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0/2}{1 + r + K\Delta t/4 + \sigma\Delta t\left(u_i^0\right)^3/2} \tag{5.18}$$

4. 'Inside' treatment: The last term of Eq. (5.17) is now written as $-\sigma\Delta t\left(u_i^n\right)^4$, which yields

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0/2 - \sigma\Delta t\left(u_i^0\right)^4/2}{1 + r + K\Delta t/4} \tag{5.19}$$

5. Mixed treatment with equal share of the pseudo-implicit and inside treatments. The last term of Eq. (5.17) is the average of the previous two cases, i.e., it is $-\sigma\Delta t\left(u_i^n\right)^4/2 - \sigma\Delta t u_i^{n+1}\left(u_i^n\right)^3/2$, which

yields

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^0/2 - \sigma\Delta t\left(u_i^0\right)^4/4}{1 + r + K\Delta t/4 + \sigma\Delta t\left(u_i^0\right)^3/4} \tag{5.20}$$

Now, I turn my attention to the nonstandard or nonlocal treatments of the radiation term. Aiming to avoid symmetrical breakings and an extensive increase in the running times, I try three different possibilities. Since the pseudo-implicit version Eq. (5.20) is the most successful among the treatments presented so far, I modify this version, mostly by changing one or two of the u_i-s in the $\left(u_i\right)^3$ product in the denominator of, e.g., Eq. (5.20) in the following three ways.

6. Product treatment (denoted by LH PI NL prod): Instead of $\left(u_i^0\right)^3$ and $\left(u_i^n\right)^3$, I write $u_{i-1}^0 u_i^0 u_{i+1}^0$ and $u_{i-1}^n u_i^n u_{i+1}^n$, respectively.

7. Average treatment (denoted by LH PI NL av): Instead of $\left(u_i^0\right)^3$ and $\left(u_i^n\right)^3$, I write $\frac{u_{i-1}^0 + u_{i+1}^0}{2}\left(u_i^0\right)^2$ and $\frac{u_{i-1}^n + u_{i+1}^n}{2}\left(u_i^n\right)^2$, respectively.

8. In the case of the time-average treatment (denoted by LH PI NL time), there are two sub-stages. First, I calculate the effect of the diffusion and the source terms by (5) – (7) to obtain a temporary value u_i^{temp}. Then the time average $u_i^{\text{timeav}} = \left(u_i^n + u_i^{\text{temp}}\right)/2$ is inserted into Eq. (5.20), as follows:

$$u_i^{1/2} = \frac{u_i^0 + r/2\left(u_{i-1}^0 + u_{i+1}^0\right) + \Delta t/2 \cdot q_i - K\Delta t u_i^{\text{timeav}}/2}{1 + r + K\Delta t/4 + \sigma\Delta t u_i^{\text{timeav}}\left(u_i^0\right)^2/2} \tag{5.21}$$

Due to the $1-r$ factor in the numerators of, e.g., the second equation of equations (5.12), the formulas can give negative temperatures for large *r*. In these cases, large negative values of the term $(u_i^n)^3$ can arise in the denominator, which may cause instability. To avoid this, in some cases, I apply a simple trick with the following conditional statement:

$$\text{if } u_i^{n+1} < 0 \text{ then } u_i^{n+1} = 0 \tag{5.22}$$

5.1.4. Methods Used for Comparison Purposes

I present three explicit methods, which are known to be unconditionally stable for the heat conduction case. However, as far as I know, they have not been applied to the case where convection and radiation are also present.

1. The Dufort–Frankel (DF) method [64] (p. 313) is the textbook example of explicit and unconditionally stable methods. It is a two-step but one-stage algorithm with the following formula, where the convection and the radiation terms are treated in a mixed way:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^n + u_{i+1}^n\right) + 2\Delta t q_i - \Delta t K u_i^n - \Delta t\sigma\left(u_i^n\right)^4}{1 + 2r + \Delta t K + \Delta t\sigma\left(u_i^n\right)^3} \tag{5.23}$$

Since this algorithm is not a self-starter, u_i^1 must be calculated from u_i^0 by another method. I employ the UPFD formula [67] for this purpose:

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

2. The alternating direction explicit (ADE) scheme is a known, but non-conventional method [68], [69]. In a one-dimensional equidistant mesh, one splits the calculation into two directions, first sweeping the mesh from the left to right (using auxiliary variable *a*) and then vice versa (with variable *b*). In the case of Dirichlet boundary conditions at nodes 0 and *N*, one sets

$$a_i^n = u_i^n,\ i = 1,\ldots N,\ a_0^{n+1} = u_0^{n+1} \text{ and } b_i^n = u_i^n,\ i = N, N-1,\ldots 1,\ b_N^{n+1} = u_N^{n+1}$$

Then, in case of pure conduction, the following equations are solved from left to right and from right to left, respectively:

$$\frac{a_i^{n+1} - a_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left(a_{i-1}^{n+1} - a_i^{n+1} - a_i^n + a_{i+1}^n\right)$$

and

$$\frac{b_i^{n+1} - b_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left(b_{i-1}^n - b_i^{n+1} - b_i^n + b_{i+1}^{n+1}\right)$$

Since, on the right hand side of these formulas, both a_i and b_i are taken into account 50–50% in the old and new time level, it is plausible to use the mixed treatment of the convection and radiation term here, too. With this, the explicit expressions are the following:

$$a_i^{n+1} = \frac{(1 - r - \Delta t K/2)a_i^n + r\left(a_{i-1}^{n+1} + a_{i+1}^n\right) + \Delta t q_i - \Delta t\sigma\left(a_i^n\right)^4/2}{1 + r + \Delta t K + \Delta t\sigma\left(a_i^n\right)^3/2}$$

$$b_i^{n+1} = \frac{(1-r-\Delta tK/2)b_i^n + r\left(b_{i-1}^n + b_{i+1}^{n+1}\right) + \Delta t q_i - \Delta t\sigma\left(b_i^n\right)^4/2}{1+r+\Delta tK/2+\Delta t\sigma\left(b_i^n\right)^3/2} \tag{5.24}$$

The final values are the simple averages of the two half-sided terms: $u_i^{n+1} = \left(a_i^{n+1} + b_i^{n+1}\right)/2$. I note that for non-uniform meshes, the ADE method loses its fully explicit character, and matrix calculations would be necessary, so in Section 5.2 it is not used.

3. The original odd–even hopscotch (OOEH) algorithm has been known for half a century [70]. Its time–space structure is presented, e.g., in [71]. It uses the usual FTCS formula (based on explicit Euler time discretization) at the first stage and the backward time central space (BTCS) formula (implicit Euler time discretization) in the second stage. I now adapt it to my case in a way where the convection term is always taken into account at the new time level, while the radiation term is treated first explicitly and then in the pseudo-implicit way. The used formulas are the following:

First stage:
$$u_i^{n+1} = \frac{(1-2r)u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) + \Delta t q_i - \Delta t\sigma\left(u_i^n\right)^4}{1+\Delta tK}$$

Second stage:
$$u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right) + \Delta t q_i}{1+2r+\Delta tK+\Delta t\sigma\left(u_i^n\right)^3}\ ,\ u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right) + \Delta t q_i}{1+2r+\Delta tK+\Delta t\sigma\left(u_i^n\right)^3}$$

5.2. Simulation of a Realistic Wall

5.2.1. The Structure and the Materials of the Wall

In this section, a wall segment is simulated with dimensions 1m in the x and z direction and 0.2m in the y direction. As one can see in Figure 5.1, the following two geometries are considered:

A. A wall's surface is examined, which is entirely built of brick.

B. Cross section of a wall with two layers composed of brick and rigid polyurethane foam insulator with a steel beam thermal bridge.

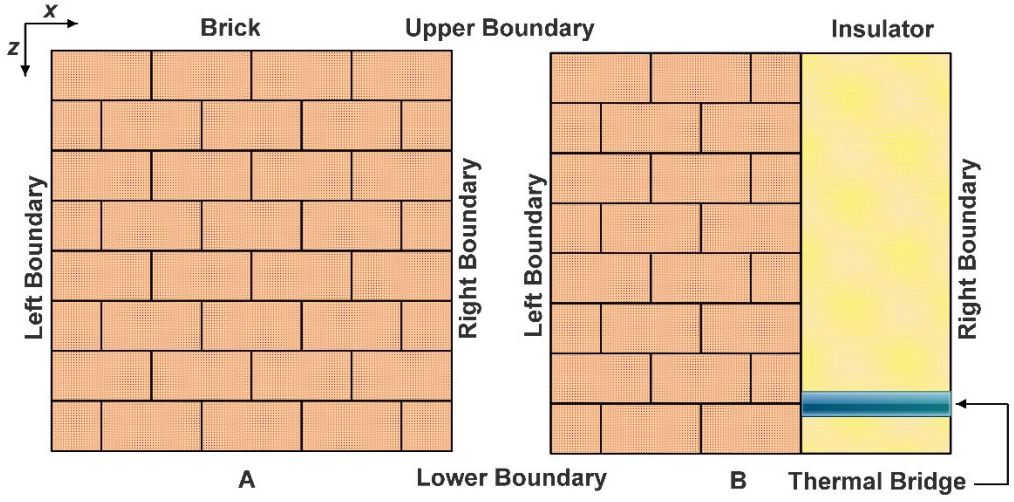


Figure 5.1. (A) surface of the wall, (B) wall with insulator and thermal bridge.

Real material attributes are taken into consideration, which are presented in Table 5.1. One should keep in mind that although these coefficients are constant (that is, they don't change with time, space, or temperature) within a material, they have a sharp discontinuity at the boundaries of different materials.

Table 5.1. Properties of the applied materials [25].

	$\rho\left[\mathrm{kg\cdot m^{-3}}\right]$	$k\ \left[\mathrm{W\cdot m^{-1}\cdot K^{-1}}\right]$	$c\ \left[\mathrm{J\cdot kg^{-1}\cdot K^{-1}}\right]$
Brick	1900	0.73	840
Rigid Polyurethane Foam	320	0.023	1400
Steel Beam	7800	16.2	840

5.2.2. Mesh Construction

In the used approximation, no physical quantities are changing in the *y*-direction, which is perpendicular to the surface of Figure 5.1. I use $\Delta y_i = 0.2$ m in order to get a realistic problem. The other two coordinates are in the unit interval, $(x,z) \in [0,1]\times[0,1]$, thus the total area of the meshes is 1 m^2. An equidistant mesh with square-shaped cells and a non-equidistant mesh with rectangular cells are the two types of meshes that were constructed. The number of cells along the *x* and the *z* axes are $N_x = 100$ and $N_z = 100$, therefore the total number of cells is $N = N_x N_z = 10{,}000$. The non-equidistant mesh contains high cells on the upper side and low ones on the lower side of the wall, as well as wide cells on the left side and narrow ones on the right side of the wall. The width and height are decreasing gradually both in the *x* and *z* directions consecutively, following a geometric series. The sum of the first $n + 1$ terms of a geometric series, up to the term r^n, ($r \neq 1$), is

$$a + ar + ar^2 + ar^3 + \ldots\ldots + ar^n = \sum_{k=0}^{n} ar^k = a\left(\frac{1-r^{n+1}}{1-r}\right) \tag{5.25}$$

Here $n = N_x - 1 = N_z - 1$. The values $r = 0.98$ and $a = 0.0234$ are used, which give $\Delta x_1 = 0.0234$ and $\Delta z_1 = 0.0234$ on the left and the upper sides, respectively, while $\Delta z_{Nz} = \Delta x_{N_x} = 0.98^{99} \cdot \Delta x_1 = 0.00317$ on the right and lower sides. The obtained meshes are shown in (Figure A1-1 in Appendix).

In the case of the wall's surface, I apply only the equidistant grid. However, in the case of the cross section of the wall with an insulator, I apply both kinds of mesh: the equidistant and the non-equidistant. For programming simplicity, in a cross section, bricks always make up the left half of the cells, whereas the insulator (containing the thermal bridge) makes up the right half. This means that, in the equidistant case, the thermal bridge has the same thickness and volume as the insulator, but the thickness of the insulator is smaller (0.269 m) in the non-equidistant case, as shown in Figure 5.1. B. The horizontal position of the thermal bridge is between *x*=0.5 m and *x*=1 m for an equidistant mesh and between *x*=0.735 m and *x*=1m for a non-equidistant mesh. The height of the bridge is two cells (2 cm) in the *z* direction, i.e., 0.02 m, while it is vertically positioned in rows number 20 and 21 from z=0.37 m to z=0.39 m. Resistance and capacitance of heat conduction are defined in section 2.1.

5.2.3. The Initial and the Boundary Conditions

In this section, the final time (the end of the analyzed time span) is $t_{\text{fin}} = 10{,}000$ s. The time step size will be measured in seconds as well. I used zero Neumann boundary conditions in all cases for all boundaries, which prohibits the flow of conductive heat 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 accomplished by setting the necessary resistances to infinity and setting the value zero for the matrix elements describing heat conduction through the boundary.

A) In the case of the surface area simulation, the heat transfer by radiation and convection is happening to the *y* direction, i.e., orthogonal to the plane of Figure 5.1.

A linear function of the *x* variable is applied as initial condition:

$$u(x, z, t = 0) = 300 - 280x$$

I have used values from the literature [25] for the convection heat transfer coefficient *h*, as one can see in Table 5.2. Since the surface is not an ideal black body, I multiplied it by the appropriate emissivity constant to obtain realistic values for σ^*. I estimate the value of q^* for the heat generation that contains a factor of the solar radiation as shown below. The ambient air temperature is taken to be $27°\text{C} \approx 300\text{K}$.

Table 5.2. The convection and radiation parameters, heat source in case of wall surface area [39].

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

Due to the nonzero temperature u_a of the air (in Kelvin), the expression *q* contains also the convective heat gain. I can obtain the value of *q* as follows

$$q = \frac{q^*}{c\rho\Delta y} + \frac{h}{c\rho\Delta y} \cdot u_{\text{a}} .$$

The convective and radiative energy transfer occurs in the *y* direction, perpendicular to the surface. Consequently, they are proportional to the element's free surface area, which is $\Delta x \Delta z$ here. Using this information, I find the following values for the coefficients in Eq. (2.4):

$$K = \frac{h}{c\rho\Delta y}, \ \sigma = \frac{\sigma^*}{c\rho\Delta y} .$$

I assumed that the lower half side of the surface is in the shade, resulting in much less incoming heat there. Specifically, I have

the first portion of N (sunny side): $q = \frac{1}{c\rho\Delta y} \times 800 \frac{W}{m^2} + \frac{h}{c\rho\Delta y} \times 300\text{K}$;

the second portion of N (shaded side): $q = \frac{1}{c\rho\Delta y} \times 300 \frac{W}{m^2} + \frac{h}{c\rho\Delta y} \times 300\text{K}$.

B) In the case of the simulation of the cross-sectional area with the thermal bridge, the interior components cannot absorb or lose heat via convection, radiation and the heat source. According

to Table 5.3, elements on the right and left sides may transfer heat in the *x* direction through radiation and convection.

Table 5.3. The convection, radiation, and heat source characteristics on both sides of the wall components in the case of the cross-sectional area.

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

I obtain the coefficient values in my equations as follows:

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

The ambient air temperature is taken to be $20°\mathrm{C}\approx 293\mathrm{K}$ and $40°\mathrm{C}\approx 313\mathrm{K}$ on the brick and the insulation side (in- and outside of the building), respectively. It gives the following convection and radiation heat sources for left and right elements:

In terms of the left-hand side: $q=\frac{1}{c\rho}\times 500\frac{W}{m^2}+\frac{h}{c\rho\cdot\Delta x}\times 293\mathrm{K}$

In terms of the right-hand side: $q=\frac{1}{c\rho}\times 500\frac{W}{m^2}+\frac{h}{c\rho\cdot\Delta x}\times 313\mathrm{K}$

In this case, the initial condition is a linear function of the *z* variable:

$$u(x,z,t=0)=313-293z$$

Until this point all temperatures were close to room temperature. However, for significantly larger temperatures, the nonlinear radiation term has much stronger effect. Thus, in the following point I try to simulate a case, e.g. a furnace, where the temperature on the left side of the wall is much higher than on the right side.

C) In the case of the cross-section of the wall with high temperatures, the geometry is similar to the previous case. The concrete values of the constants change according to Table 5.4.

Table 5.4. The heat source, convection, and radiation characteristics are on both sides of the wall components in the case of cross-sectional area.

	$\mathrm{h}\left[\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}}\right]$	$\sigma^*\left[\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}^4}\times 10^{-8}\right]$	$q^*[\mathrm{W}]$
Right Elements	2	5	500
Left Elements	25	4	3500

The "ambient" air temperature inside the furnace is taken to be $227°\mathrm{C}\approx 500\mathrm{K}$. This yields the following convection and radiation heat sources for right and left elements, respectively:

$$q=\frac{1}{c\rho}\times 500\frac{W}{m^2}+\frac{h}{c\rho\cdot\Delta x}\times 303\mathrm{K}\ \text{ and }\ q=\frac{1}{c\rho}\times 3500\frac{W}{m^2}+\frac{h}{c\rho\cdot\Delta x}\times 500\mathrm{K}$$

In this case, the initial condition is again a linear function of the *x* variable:

$$u(x,z,t=0)=500-303x$$

In this section, the ode15s solver has been employed to obtain the reference solution. I have calculated the maximum time step size (CFL limit, above which the explicit Euler time integration becomes unstable) and the stiffness ratio in the usual way [42], [28] considering only the conduction term. Table 5.5 shows the value of these quantities for different cases.

Table 5.5. The CFL limit and the stiffness ratio quantities for the different cases.

		CFL limit	**Stiff. ratio**
surface	equidistant	55.78	8.1×10^3
cross-section with-thermal bridge	equidistant	5.01	5.167×10^5
	non-equidistant	2.28	4.29×10^5

5.2.4. Results for the surface of the wall

Part I: There is only convection, and no radiation, which means σ^* and q^* are set to zero. A one-layer brick wall is examined here (see Figure 5.1 A). The initial and boundary conditions as stated in point A) above have been considered and only the equidistant mesh has been used. The maximum errors as a function of time step size are shown in Figure 5.2 for all methods of Subsection 5.1.2 [72]. For smaller time step sizes, the mixed treatment of the convection terms with 50% weight for the PI and the 'inside' treatment is the most accurate. It clearly has a second order convergence, in accordance with the results of the previous sections. Thus, from this point, the convection term will always be treated in this optimal way. For some large time, step sizes, however, the original hopscotch method is slightly more accurate, which can be due to the low stiffness ratio of the problem.

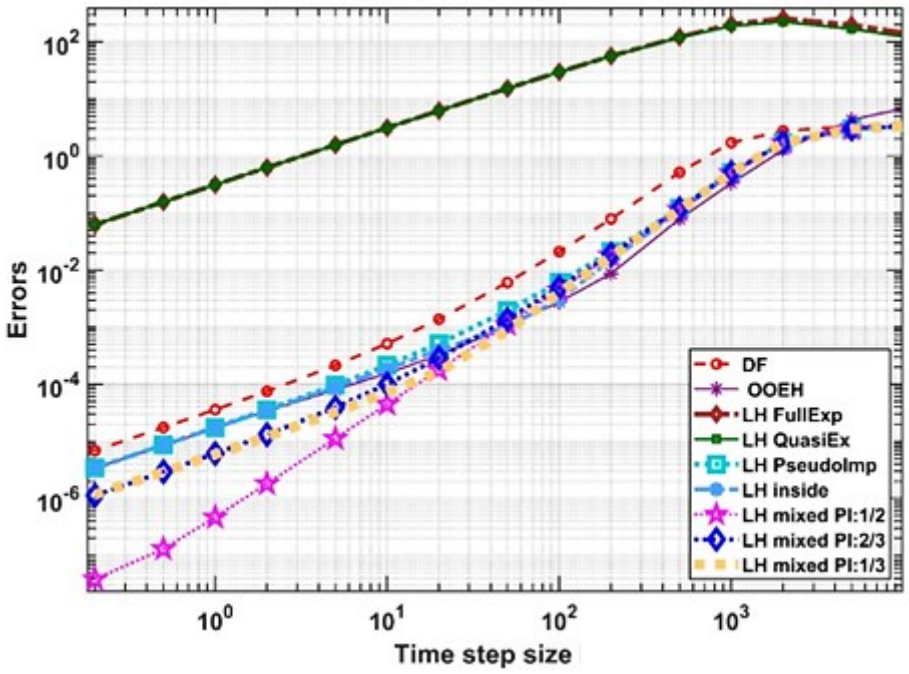


Figure 5.2. The maximum errors as a function of the time step size for the 9 examined methods in the case of a surface area (Section 5.2.4 Part I). for the convection term.

Part II: The surface of the brick wall is simulated with radiation, where conditions in point (A) and the values from Table 5.3 are used. The errors are shown in Figure 5.3, and the temperature distributions are shown in Figure 5.4. The influence of the initial condition, as well as the shadow, on the lower part of the wall is clearly visible. The coordinates in cm units are represented by the numbers on the vertical and horizontal axes of the contours, which are the cell indices as well.

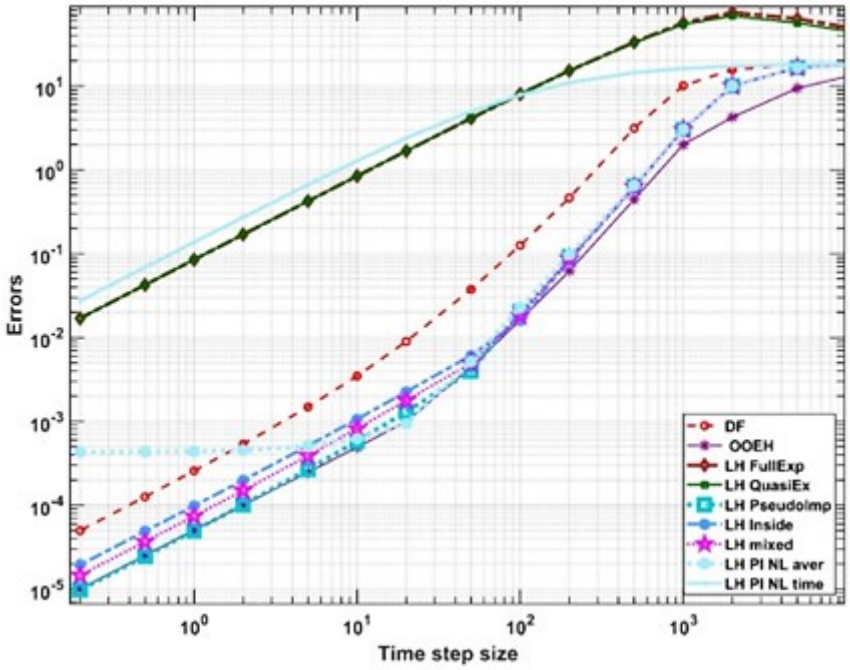


Figure 5.3. The maximum errors as a function of the time step size for the nine examined methods in the case of a surface area (Section 5.2.4 Part II). For the radiation term.

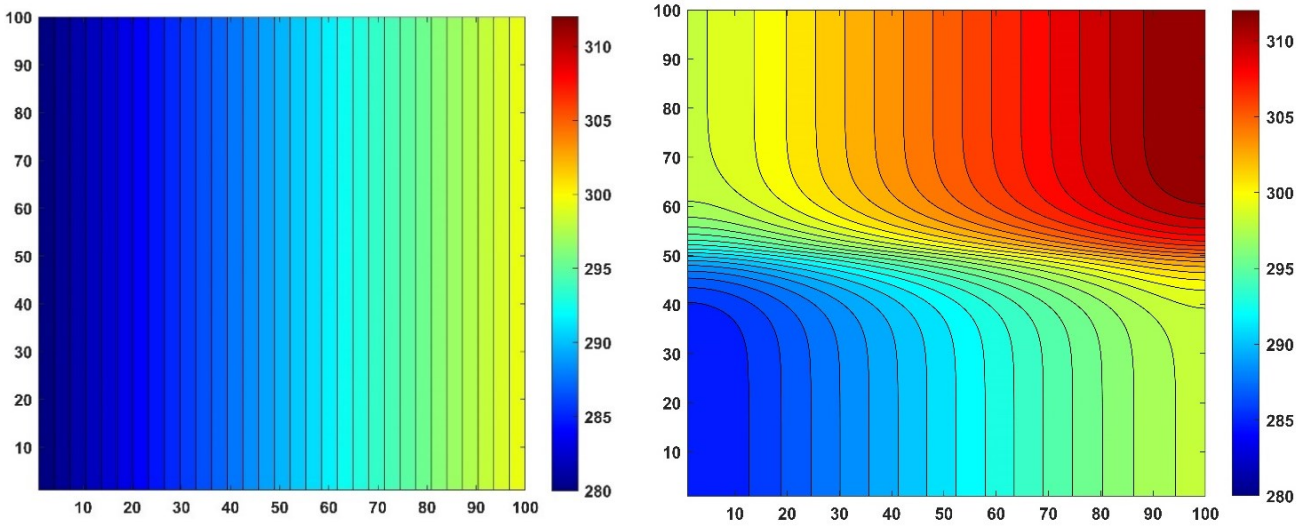


Figure 5.4. The temperature distribution contour in Kelvin in the case of (Section 5.2.4 Part II) at initial (left part) and final time (right part), in the case of a surface area.

5.2.5. Results for the cross-section of the insulated wall with thermal bridging

Part I: The equidistant mesh. Here the initial and boundary conditions of point (B) are applied to the multilayer wall. The maximum errors are plotted in Figure 5.5. The temperature distribution

contour for final time moments is shown in Figure 5.7. The temperature on the right side of the wall is rising due to the higher outside temperature, but the insulator allows this heat to enter the wall at a very slow rate.

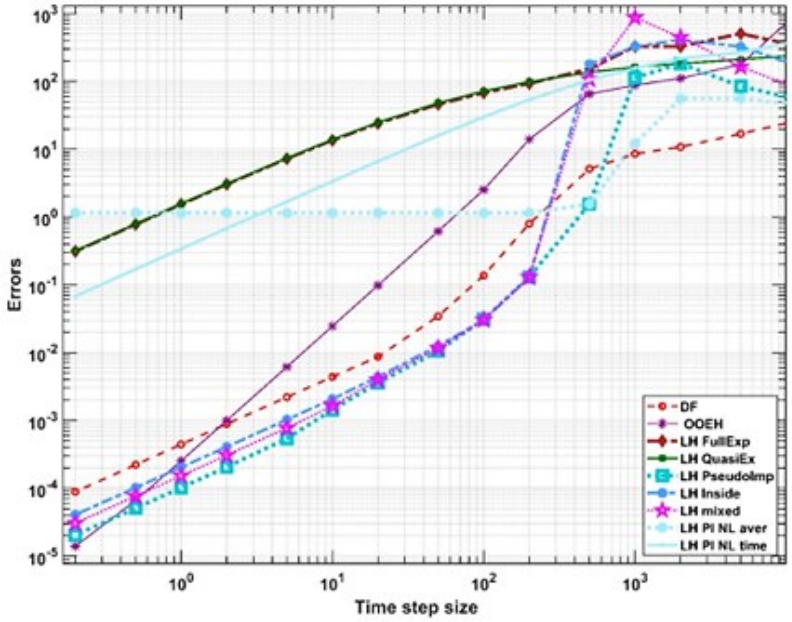


Figure 5.5. The maximum errors as a function of time step size for the equidistant mesh (Section 5.2.5 Part I) in the case of convection and radiation boundary conditions.

Part II: non-equidistant mesh, is the same as in (Section 5.2.5 Part I), but the mesh is replaced with the non-equidistant one presented in [72], (b). The errors are presented in Figure 5.6.

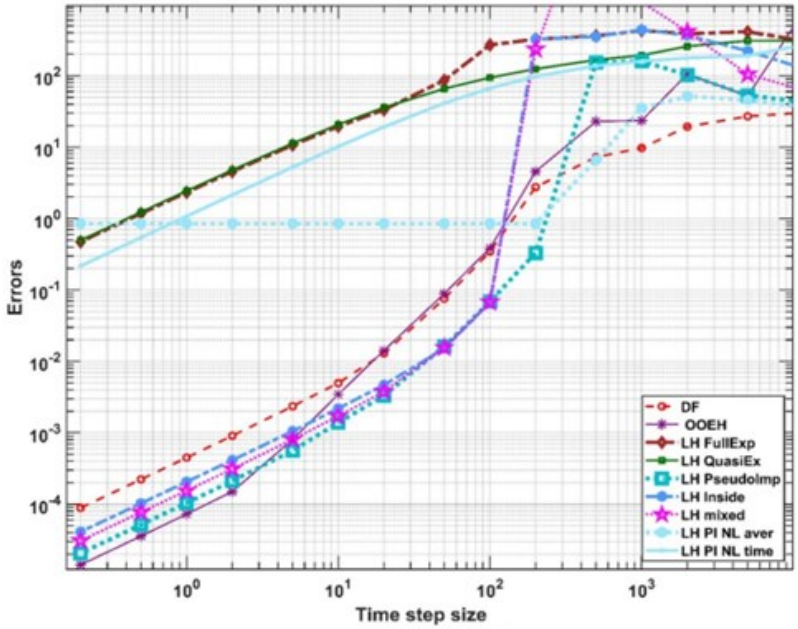


Figure 5.6. The maximum errors vs. the time step size for the non-equidistant mesh (Section 5.2.5 Part II) in the case of convection and radiation boundary conditions.

while the temperature contours are presented in Figure 5.7, the non- equidistant mesh, in case of the multilayer cross-sectional area. The numbers on the vertical and horizontal axes of the contours are the coordinates in cm units.

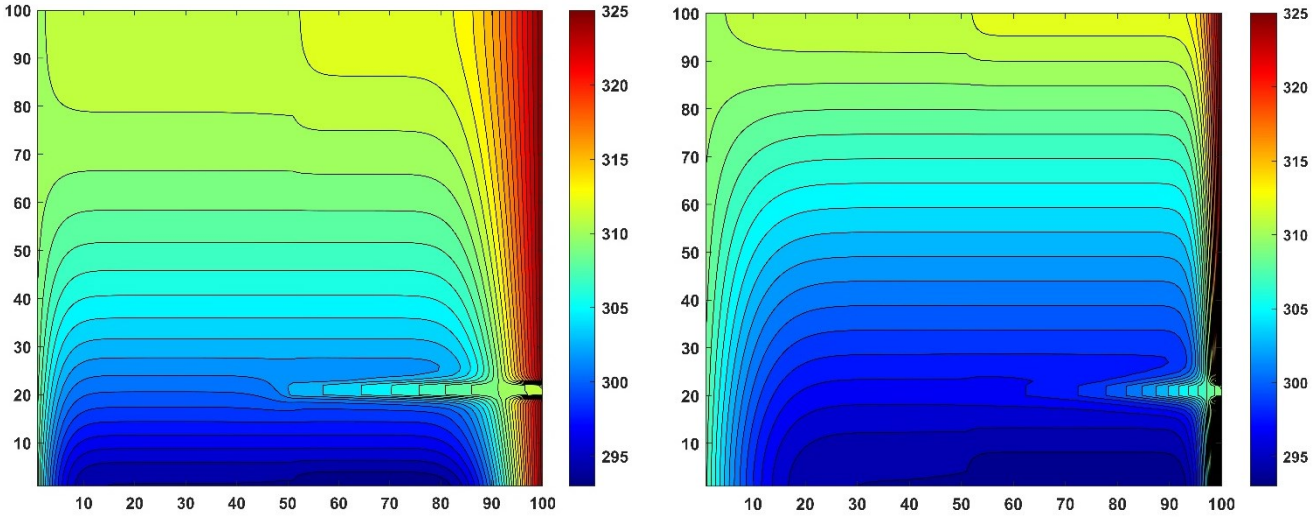


Figure 5.7. The temperature distribution contour in Kelvin for the final time (left) in the case of (Section 5.2.5 Part I) (equidistant mesh). And final time (right) (Section 5.2.5 Part II).

Part III: High-temperature boundary conditions, non-equidistant mesh. Here the linear initial and Neumann boundary conditions of point C) are applied to the multilayer wall. The maximum errors are plotted for non-equidistant meshes in Figure 5.8. One can see that the mixed and inside versions produce rather large errors, and they are clearly outperformed by the PI treatment. The temperature distribution contour for the initial and final time moments are shown in Figure 5.9. The numbers on the axes are the coordinates in meter units.

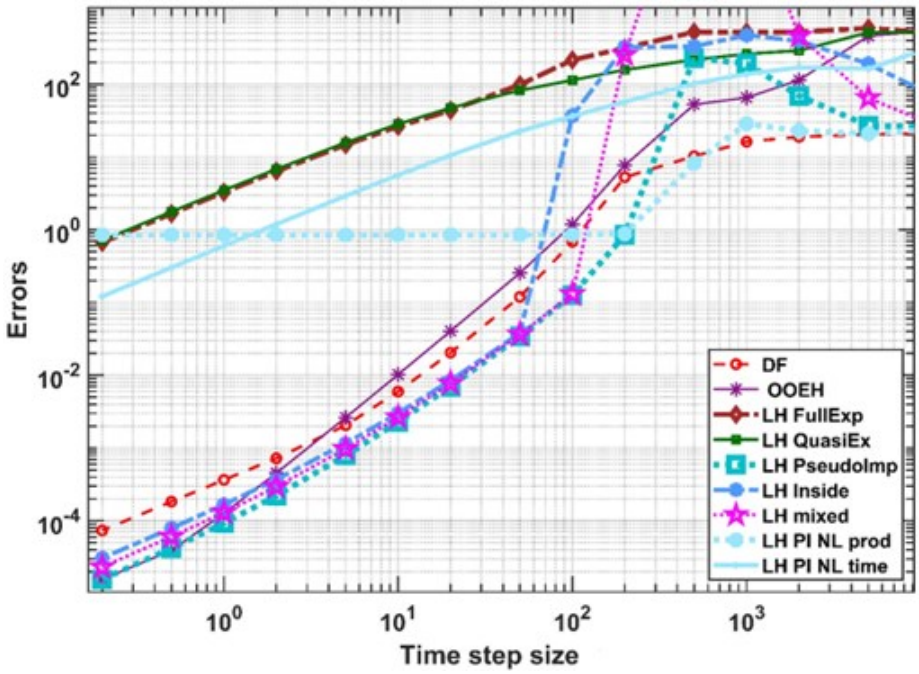


Figure 5.8. The maximum errors as a function of the time step size for (Section 5.2.5 Part III) in the case of convection and radiation with high-temperature boundary conditions.

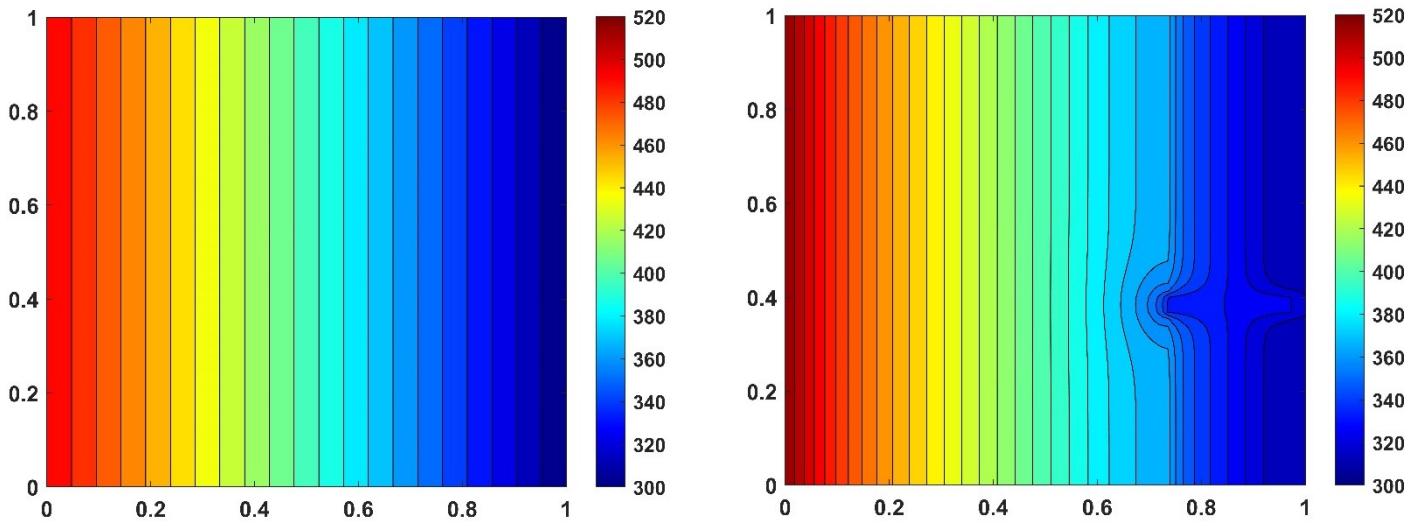


Figure 5.9. The distribution contour of the temperature in Kelvin units for the equidistant mesh at initial and final time (left and right figure, respectively), in the case of (Section 5.2.5 Part III).

5.3. Summary of this chapter

In this chapter, I have studied several implementations of the free convection and radiation terms using the leapfrog-hopscotch method, which had originally been optimized to solve the heat conduction equation. I observed that, usually, the best performance is achieved when the convection term is treated in a mixed way, i.e., taking into account 50% at the old and 50% at the new time level. The order of temporal convergence reaches two only for this optimal version, which was also proven by the calculation of the truncation errors. The unconditional stability of this version was also proven by von Neumann analysis in the linear case (conduction + convection) by a member of our research group.

On the other hand, according to the numerical experiments, the radiation term should be taken into account fully in the pseudo-implicit way. In this case, one of the four powers is taken into account at the new time level, so the term turns up only in the denominator, which ensures very good stability properties. I performed five numerical experiments to simulate heat transfer on a realistic wall. The proposed algorithm performs quite well, even when the CFL limit for the mainstream explicit methods is rather low.

6. NUMERICAL RESULTS FOR THE DIFFUSION-REACTION EQUATION WHEN THE REACTION COEFFICIENT DEPENDS ON SIMULTANEOUSLY THE SPACE AND TIME COORDINATES

It is well known that linear transient diffusion and heat conduction are analogous phenomena: mathematically, both of them are described by the simplest parabolic PDE (partial differential equation). Diffusion means the distribution of the particles is changing [73], while in heat conduction, energy is transported via interacting particles of the material [74].

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

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} - K\,u(x,t) \tag{6.1}$$

where $u:\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R};\ (x,t)\mapsto u(x,t)$ is the unknown function (temperature in case of heat conduction and concentration in case of particle diffusion), and $D\in\mathbb{R}$ is the constant diffusion coefficient. In the case of heat conduction, $D = k/(c\rho)$ is the thermal diffusivity, while c, ρ and k are the specific heat, the density, and the heat conductivity of the material, respectively. The term $-Ku$ is a reaction term, so Eq. (6.1) is a regular diffusion-reaction PDE. Typically, u^0 is the given initial function, while the boundary conditions will be discussed in the concrete analytical and numerical examples.

In this chapter, we convey new results for a modified version of PDE Eq. (6.1), which will have a reaction coefficient that is not a constant, but depends both on time and space, which is the most important novelty of this work. The analytical solution was experimented with the Lorentzian form and was fortunated by a member of our research group to discover solutions that were highly non-trivial. We explored the impact of periodic driving in another intriguing system.

6.1. Analytical Solution

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

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} + F(x,t)u(x,t) \tag{6.2}$$

where F is the known coefficient of the reaction term, which is usually non-positive. To derive physically relevant solutions, we apply the travelling wave Ansatz [50] . It is always interesting to study the influence of a well-localized impulse-like source, therefore we tried the Lorentzian form and luckily found highly non-trivial solutions.

6.1.1. Lorentzian Coefficient of the Reaction Term

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

$$F(x,t)=\frac{a}{1+(x-ct)^2} \tag{6.3}$$

where $a\in\mathbb{R}$ is the strength parameter (amplitude) of the applied pulse. By utilizing the Ansatz Eq. (6.3) the reduced ODE is immediately obtained

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

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

$$f(\eta)=\mathrm{e}^{\frac{-c\eta}{2D}}\left(1+\eta^2\right)\left[c_1\mathrm{HeunC}\left(0,-\frac{1}{2},1,\frac{c^2}{16D^2},\Psi,-\eta^2\right)+c_2\eta\cdot\mathrm{HeunC}\left(0,\frac{1}{2},1,\frac{c^2}{16D^2},\Psi,-\eta^2\right)\right], \tag{6.4}$$

where HeunC is the Heun function [75] and we introduced the abbreviation $\Psi=\frac{4aD+8D^2-c^2}{16D^2}$

It is almost impossible to give a complete parameter study of Eq. (6.4) for the entire range of (c1, c2, c, a, D). We just restrict us for the c1 = 0, c2 = 1, a > 0, c > 0, D > 0 subspace. Considering negative propagation velocity (c < 0) just means reflecting the solution to the y-axis. For negative source strength (a < 0) we get non-decaying solutions at large η, which means either asymptotic saturation or divergence, which we consider unphysical. Diffusion processes where the concentration or the number of particles explode violate energy and matter conservation laws and it is rather counterintuitive that we obtain it typically for negative and not for positive values of a. So albeit the solutions are mathematically valid for arbitrary values of the x and t variables, the solution on the whole real axis cannot always describe a real physical process. Nevertheless, the obtained functions—in principle—can describe real processes in any finite interval if the system boundaries are not closed, and energy or particles can enter from the surrounding space, which is reflected mathematically by the boundary conditions.

Numerous shape functions are presented in Figure 6.1 for different parameter sets. Smaller wave velocity shifts the singularity to the left and causes more oscillations (see the difference between the black and red curves in Figure 6.1). Increasing the diffusion coefficient D against the source strengths parameter a, but still being (a > D) smears out the decay range of the diffusion (compare the blue curve to the black one). Finally, considering that the diffusion parameter is larger than the source strength (D > a), the oscillations disappear, maintaining a very flat local maximum and a very slow decay (compare the blue curve to the green one).

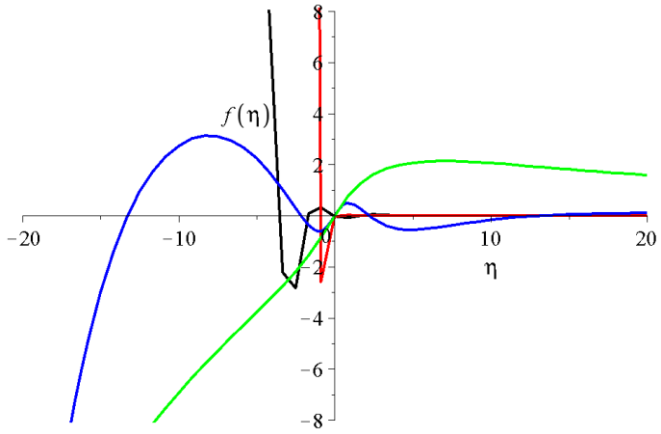


Figure 6.1. The shape function of Eq. (6.4) for the different parameter sets (c1, c2, C, a, D) the black, red, blue and green lines are for (0, 1, 0.5, 7.4, 0.3), (0, 1, 2, 7.4 ,0.3), (0, 1, 0.5, 7.4 1.8) and (0, 1, 0.5, 2.4, 4.1), respectively.

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

$$u(x,t) = e^{\frac{-c(x-ct)}{2D}}\left(1+(x-ct)^2\right)\left[\begin{array}{l} c_1\,\mathrm{HeunC}\left(0,-\frac{1}{2},1,\frac{c^2}{16D^2},\Psi,-(x-ct)^2\right) \\ +\ c_2\,(x-ct)\,\mathrm{HeunC}\left(0,\frac{1}{2},1,\frac{c^2}{16D^2},\Psi,-(x-ct)^2\right)\end{array}\right] \qquad (6.5)$$

We note that since our analytical solution is valid on the whole real axis, boundary conditions need to be specified only when the analytical solution is going to be reproduced by numerical methods. The time development of a solution function $u(x,t)$ of Eq. (6.5) is presented in Figure 6.2 for the same parameter set as the shape function is shown. We checked and found that all the solutions of $u(x,t)$ for all the other three parameter sets given in Figure 6.1 have qualitatively the same shape.

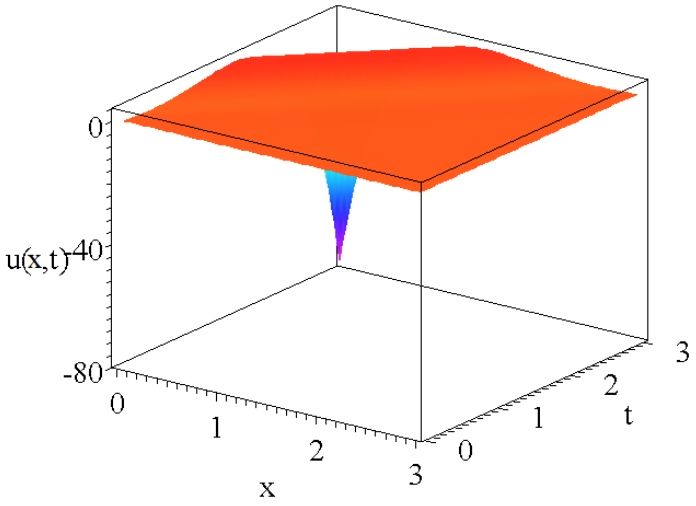


Figure 6.2. The solution function of Eq. (6.5) for the parameters of D = 0.3; a = 7.4; c = 1;. ($\Psi = 5.572222$); c1 = 0; c2 = 1, respectively.

6.2. Numerical Simulation of Surface Subjected to Wind

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

6.2.1. The Structure and the Materials of the Surface

In this section, I model a wall surface with dimensions of 1 m on the x and y axes and 0.1 m on the z axis. In Figure 6.3, a wall's surface is examined. Half of the surface ($0 < x < 0.5\text{m}$) is made of brick and the other half ($0.5\text{m} < x < 1\text{m}$) is made of insulation.

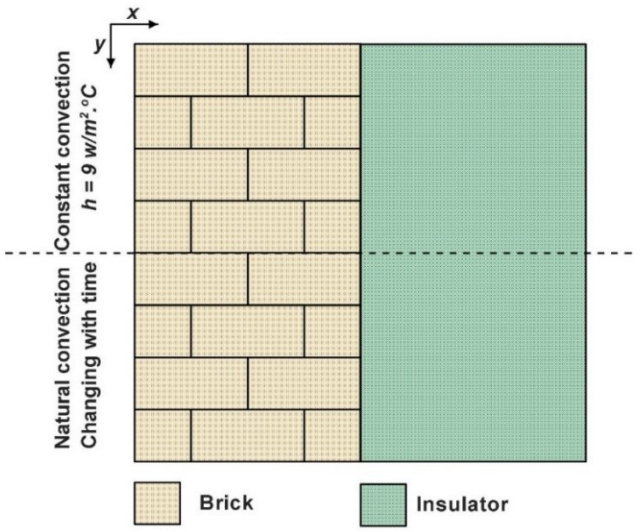


Figure 6.3. The surface of the modelled wall with two layers.

In Table 5.1, I used the same the applied materials [25], of brick and RPF insulation, the material attributes are considered. These coefficients exhibit sharp discontinuity when one material passes into the other. On the other hand, these coefficients are constant inside a material.

6.2.2. Mesh Construction

I assume that in the z direction there is no heat transfer, and the physical properties do not change, thus I fix $\Delta z = 0.1\text{m}$. From the computational point of view, a two-dimensional issue is studied. The x and y coordinates fall within the unit interval, and uniform square cells are used. I created meshes with four different resolutions with Nx = (50, 70, 90, and 100) and Ny = (50, 70, 90, and 100) for the number of cells in the x-axes and in the y-axes; therefore, $N = N_x N_y = (2500, 4900, 8100,$ and $10000)$ is the global number of cells. The discretization of the problem is explained as in section 2.1.

6.2.3. The Numerical Algorithms Used

The two explicit methods I use here to solve Eq. (6.1) are generalized via the following formulas.

1. The leapfrog-hopscotch (LH) uses the generalized Theta-formula [76], which reads as follows for a full time step size:

$$u_i^{n+1} = \frac{\left(1-\theta\left(r_i - hK_i^n\right)\right)u_i^n + A_i}{1+\left(1-\theta\right)\left(r_i - hK_i^{n+1}\right)},\ \theta \in [0,\,1] \tag{6.6}$$

The length of the stages and the values of the parameter θ are the same as in the special case, see Section 2.2.3.

2. Dufort and Frankel (DF):The initial stage applies the UPFD scheme, which can be obtained by the $\theta = 0$ substitution from Eq.(6.6). In all subsequent time steps I employ the formula

$$u_i^{n+1} = \frac{\left(1-r_i\right)u_i^{n-1} + 2A_i^n}{1+r_i+2hK} \tag{6.7}$$

The two implicit methods are also tested for the simulation of the wall by the PCG and GMRES solvers see section 2.3.

6.2.4. The Initial and the Boundary Conditions

In this section, $t_{\text{fin}} = 22500\ \text{s}$ serves as the final time (the end of the examined time period). The duration of each time step is also expressed in seconds, first with $\Delta t = 900\ \text{s}$ and it is decreased gradually to a small number $\Delta t = 0.01\ \text{s}$. A spatially constant initial temperature $\mathrm{u}\left(\mathrm{x},\mathrm{y},\mathrm{t}=0\right) = 270\ \mathrm{K}$ is set. Neumann boundary conditions with zero temperature-flux are applied to boundaries, preventing any conductive heat transfer:

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

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

Convective heat transfer occurs in the z-direction, which is perpendicular to the plane depicted in Figure 6.3. For the upper part ($0 < y < 0.5\,\text{m}$), the ambient temperature is taken to be constant, and for the lower part ($0.5\,\text{m} < y < 1\,\text{m}$), the convection circumstances are changing with time according to changing weather conditions.

Table 6.1 shows that the z-direction convection coefficients (h) and the temperature in the upper part of the plane in Figure 6.3 has constant values, whereas the lower part of the elements has changing values depending on the environmental conditions. The ambient air temperature and velocity is taken to be $17°\text{C} \approx 290\ \text{K}$ and $v = 0\,\text{m/s}$ for the upper part, and changing according to real weather conditions in Miskolc city from 5 a.m. to 11 p.m. on the second day of January [77] for the lower part, as shown in Table 6.1.

Table 6.1. The temperature, and heat source in case of wall surface area [77].

	Upper Elements	Lower Elements
$h\left[\frac{\mathbf{W}}{\mathbf{m^2K}}\right]$	9	0.6–5.55
T [K]	290	275–280

I obtain the values of the coefficients in my Eq. (6.1) as follows: [78],

$$K = \frac{h}{c\rho\Delta z}, \quad q = \frac{h}{c\rho \cdot \Delta z} \cdot u_{\mathrm{a}} \tag{6.8}$$

I also suppose that the upper elements and lower elements have heat sources as follows:

for the upper elements: $q_{up} = \frac{h_{up}}{c\rho \cdot \Delta z} \times u_{\mathrm{a}}$

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

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

The convection heat transfer coefficient for outside elements as a function of air velocity is estimated as follows [79]

$$h_{low}(t) = 0.6 + 6.64\sqrt{v}(t) \tag{6.9}$$

6.3. Results for the Surface of the Wall

Figure 6.4 shows the temperature distribution contour in Kelvin units for the surface area. The figure shows that in the case of the insulator (right-hand side of the figure), heat can hardly flow from the top of the figure to the bottom, so there are large temperature gradients. Moreover, because the heat capacity of the insulation layer is smaller than that of the brick layer, its temperature increases faster from the original 270 K.

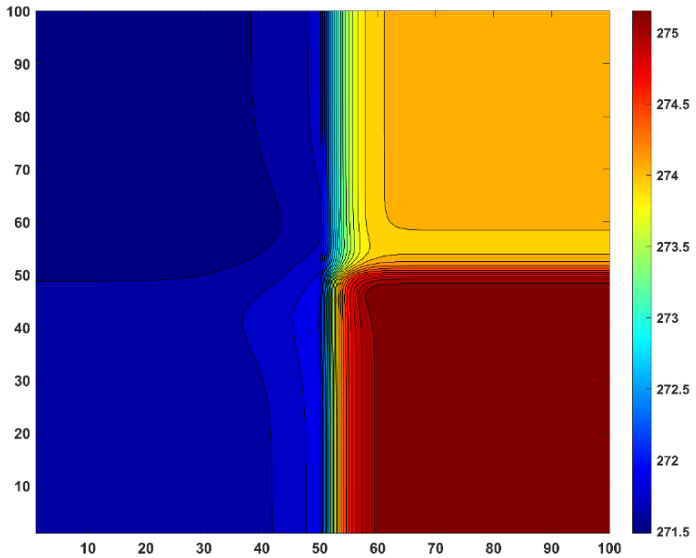


Figure 6.4. The temperature distribution contour in Kelvin units for the surface area (upper half) constant convection and (lower half) the convection changes with time depending on weather data.

The maximum errors of the cell-temperatures at the final time as a function of time step size are shown in Figure 6.5 for the systems 100 by 100. One can see the LH is the most accurate scheme, followed by the DF, compared with the implicit methods with two different tolerances. Figure 6.6 for the system 100 by 100, show the running time with the maximum errors. I use many time steps for the explicit methods (LH and DF) and less for the implicit methods because they are much slower, and I see that LH and DF are faster and more accurate.

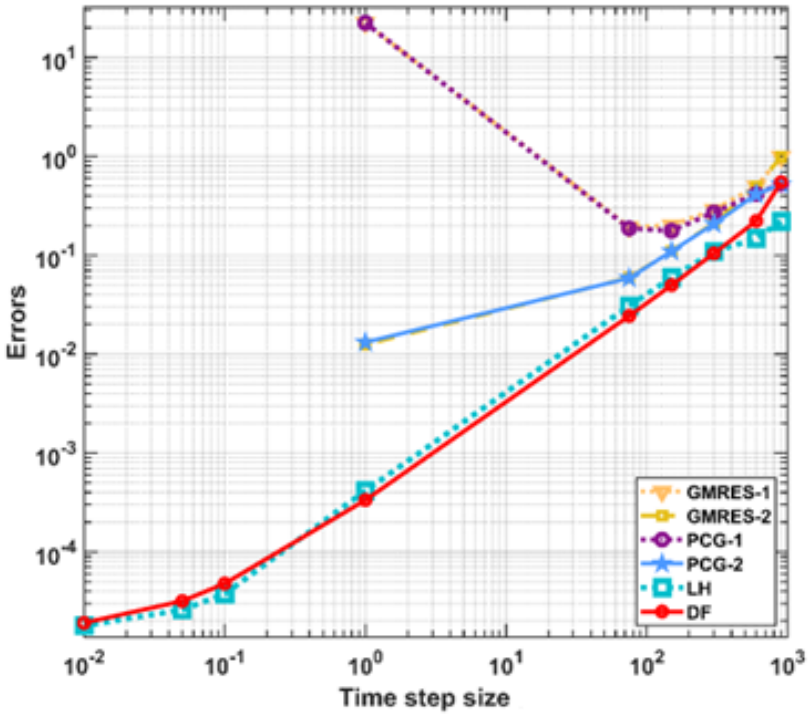


Figure 6.5. The maximum errors as a function of the time step size h for the examined methods for the 100 by 100 system.

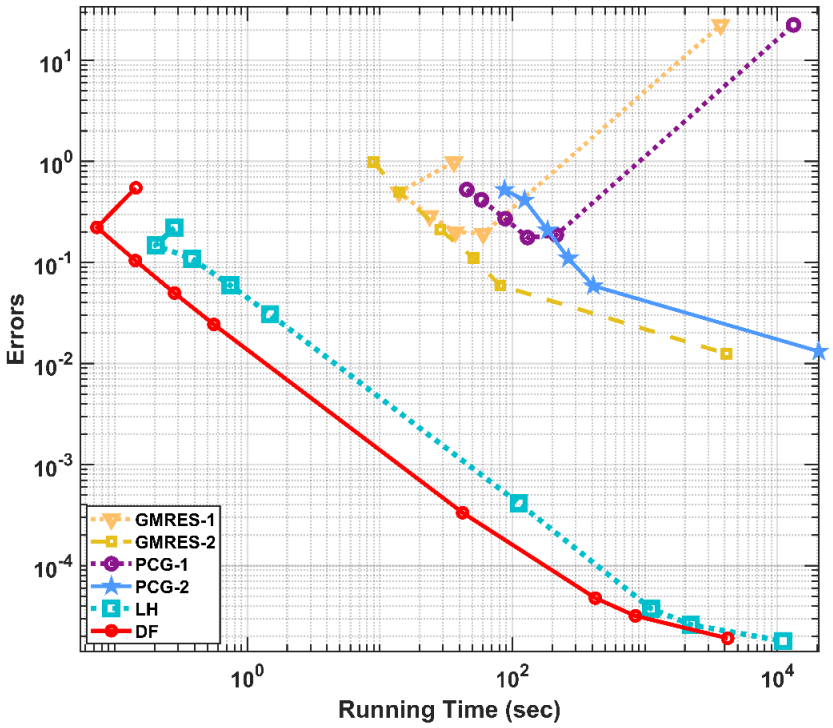


Figure 6.6. The maximum errors as a function of the running time for the tested methods for the 100 by 100 system.

Figure 6.7 shows the running time for time step size $\Delta t = 1\,\mathrm{s}$ with the total number of cells on the horizontal axis. On the left-side axis, it is shown for the LH and DF methods and on the right-side axis for implicit methods. One can see that the running time is a linear function in the case of the explicit methods, but not for the implicit methods. This is expected, since the explicit methods apply loops over the cells, so the running time is directly proportional to N, while the implicit solvers work with the system matrix with size $N \times N$, thus the running time is proportional to N^2. It means that for even larger system sizes, e.g., in 3D, they become even less competitive with the explicit and stable methods.

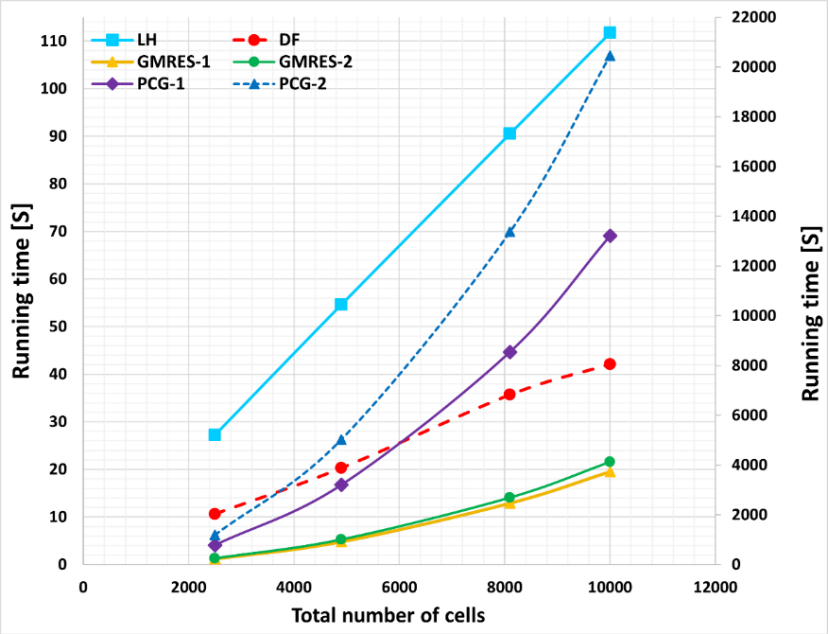


Figure 6.7. The running time with total number of cells for $\Delta t = 1\,\mathrm{s}$. The left-hand side vertical axis refers to the LH and DF methods, while it is on the right-hand side axis for the implicit methods.

For the coarse mesh (50 by 50), I present the spatial distribution of errors in Figure 6.8 as 2D contour plots and in Figure 6.9 as curves for a specific z coordinate for the four methods. The time step sizes are chosen to reach very similar error levels. One can see that with the exception of the DF method, the errors are the largest, where the temperature gradients are largest, i.e., at the boundary of different materials and circumstances. The pattern of odd-even structure is also observable for the LH method, but even with this artificial oscillation, this algorithm is the most accurate. Table 6.2 shows the time step sizes, the errors and the running times for the four methods in the case exemplified in Figure 6.8 and Figure 6.9.

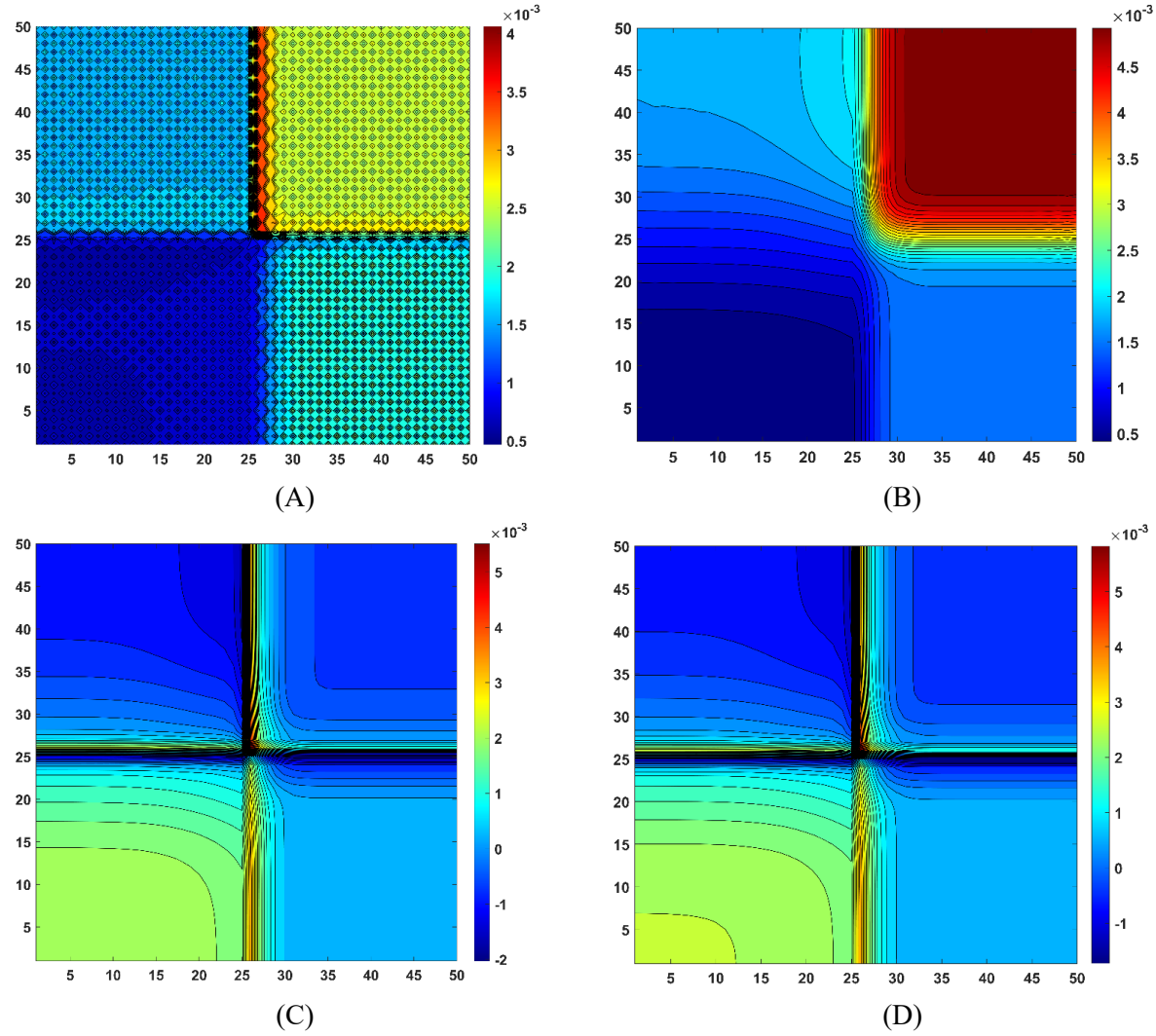


Figure 6.8. The error of temperature distribution contour in Kelvin units (A) LH, (B) DF, (C) GMRES and (D) PCG methods.

Table 6.2. The performance for the numerical methods.

Numerical Method	Time Step Size Δt (Tolerance)	Maximum Error	Running Time [s]
leapfrog-hopscotch (LH)	10	0.0042	1.32
Dufort–Frankel (DF)	10	0.0051	0.97
generalized minimal residual (GMRES)	10−7	0.0058	246.965
preconditioned conjugate gradient (PCG)	10−7	0.0062	1187.404

Figure 6.9 highlights that the LH method achieves the best method of accuracy during all the time of simulation in a 2D wall with a material discontinuity, and the fluctuation is because of the odd-even method, with errors peaking at the brick-insulation interface. The DF method offers a faster but slightly less accurate alternative, while the implicit GMRES and PCG methods lag in both accuracy and speed, underscoring the advantages of explicit, stable schemes in this context.

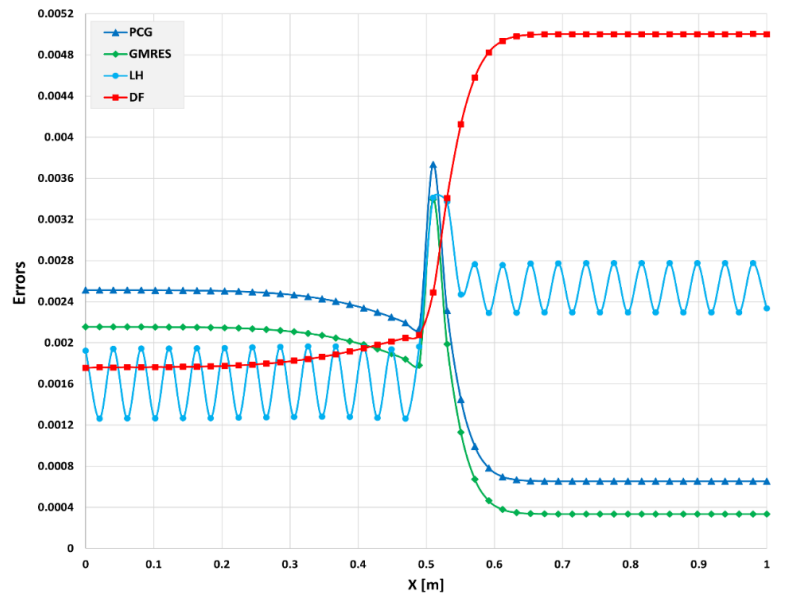


Figure 6.9. The error of temperature distribution at z = 80 [cm] in Kelvin units for LH, DF, GMRES and PCG methods along x-axis.

6.4. Summary of this chapter

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

7. PREDICTION AND OPTIMIZATION OF THERMAL LOADS IN BUILDINGS WITH DIFFERENT SHAPES BY NEURAL NETWORKS AND RECENT FINITE DIFFERENCE METHODS

This chapter uses two neural networks: an MLP and an RB, as well as three algorithms to optimize their parameters: Levenberg–Marquardt (LM), Scaled Conjugate Gradient (SCG), and Radial Basis Function (RB). The building all with standard floor area is 200 m^2 for residential homes to select the best design. By building a network model for these homes, I can estimate the heat load from the external observation design of the house. Also, I used the LH method used in the previous chapter for long-term simulation. The rest of this chapter is organized as follows: it describes the methodology, factors, and algorithms used in this research. and provides examples of the data used to train the neural network, and also evaluates and discusses the results. Furthermore, it presents the second part of this work, where I chose my best model to perform a simulation of the peak load of the months in winter and summer, December and July, and present the optimal model details. For this simulation, I use the recently invented leapfrog–hopscotch algorithm, which is faster and can cope with stiff and non-stiff systems. This algorithm is used for the first time in summer simulations to calculate the heat gained through the wall. Finally, it summarizes the main findings and conclusions of this study.

7.1. Artificial Neural Network (ANN)

Energy optimization of buildings can be computationally intensive, and the artificial neural network technique balances accuracy and computation time. This study uses feature selection to simplify models. Variables such as building orientation, interior and exterior wall and roof, glazing, window shading, and infiltration rate are taken into account. The energy performance of buildings (EPB) takes into account its two primary parameters, cooling and heating, and the ANN (MLP and RB) is trained while taking these values into consideration by Figure 7.1. The parameters of ANN are optimized through the utilization of three distinct optimization techniques. A comparative analysis is conducted between the derived outcomes and empirical data, demonstrating the validity and practicality of the theoretical frameworks.

In the process of creating my database, a selection of six distinct building shapes was made through a randomization procedure. Users may easily generate a 3D model of a structure using the software AutoCAD MEP 24.2 (mechanical, electrical, and plumbing). The thermal loads of the building are determined using the transfer functions method and the ASHRAE Heat Balance procedure [80]. The present study involves the implementation of a database simulation approach utilizing an hourly analysis program (HAP) for the purpose of ascertaining the cooling and heating requirements.

During the process of ANN training, preprocessed data is partitioned randomly into three subsets: a training dataset, which usually constitutes 70% of the data; a test dataset, which constitutes 15% of the data; and a validation dataset, which constitutes 15% of the data [81].

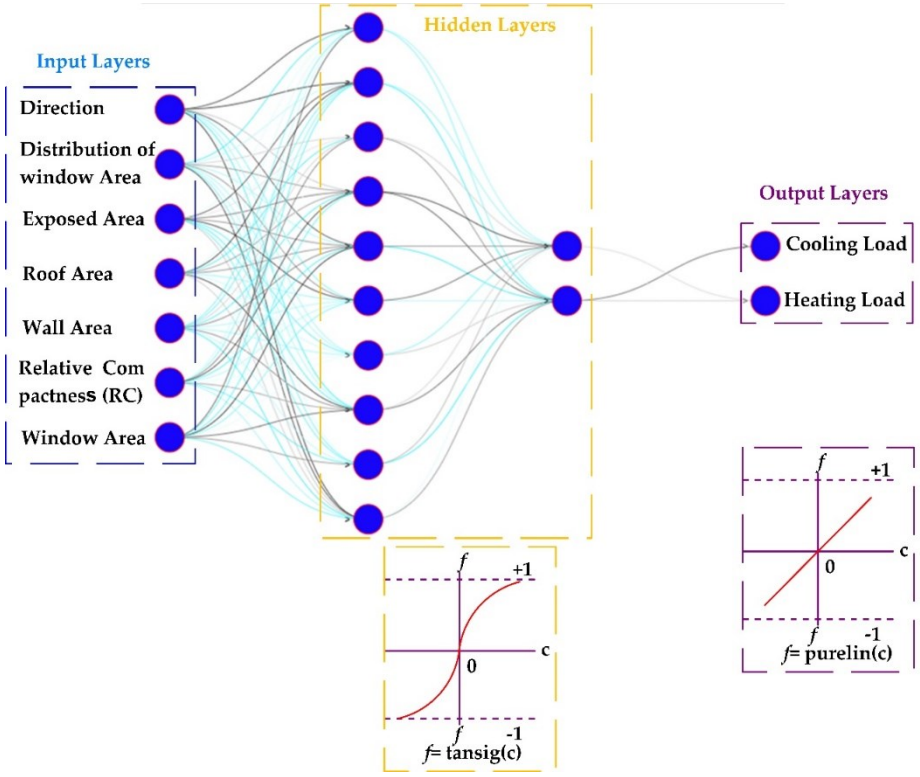


Figure 7.1. A schematic diagram of the ANN and its relation to the input and output parameters.

The evaluation of the optimized ANN predictive performance and precision is conducted through statistical analysis. Energy performance of buildings initially takes into account two primary parameters: cooling and heating. This form of ANN is fed with experimental data and trained using cooling and heating settings.

7.1.1. Multi-Layer Perceptron (MLP)

MLPs are feedforward neural instruments and universal approximates learned via backpropagation [82]. MLPs use neurons to approximate every input–output mapping. Figure 7.1 shows the ANN's input, output, and hidden layers. Neurons in adjacent layers are also connected [24].

7.1.2. Radial Basis Function (RB)

Radial Basis Function is a type of neural network that can learn from data and make decisions for different applications. The output layer uses linear combinations of the hidden units to predict or classify the data. The RB model has two steps of training: first, it clusters the data using some methods, and then it estimates the weights between the hidden and output units. The RB model has a faster learning speed than the Multi-Layer Perceptron (MLP) model because it only needs to train the weights in the second step [83].

7.1.3. The Used Algorithms

This study compares two types of neural networks: Multi-Layer Perceptron and Radial Basis Function (both using MATLAB R2020b and Statistical Package for the Social Sciences (IBM SPSS 20)). The goal is to find the best algorithm that balances accuracy, authenticity, and performance. After testing, I chose three algorithms: Levenberg–Marquardt algorithm (LMA or LM), Scaled Conjugate Gradient algorithm (SCG) and Radial Basis Function (RB).

7.2. Statistical Analysis and the Used Data

The aim of this study is to find the best ANN for higher accuracy, as stated before. The research question is answered by following the systematic approach shown as a flowchart in in Figure A1-2 in Appendix.

7.2.1. The Used Data

Before estimating the cooling load of any building, some basic information is necessary to design an appropriate HVAC system, like building location, orientation, weather conditions, building spacing, building materials, etc. The more exact the information is, the more accurate the load estimated will be. The shape of a building significantly impacts its energy performance and thermal comfort. Its shape depends on factors like the orientation, window-to-wall ratio, climate, and occupancy patterns. Common shapes include rectangle, L-shape, T-shape, U-shape, H-shape, and rectangle with interior-shape see Figure 7.2. In fact, one of the goals of this study is to reduce the time required for thermal load calculations when there are multiple probabilities for the outcome of complex buildings. Several building applications, such as swimming pools, theatres, commercial buildings, stadiums, and residential buildings, impose restrictions. However, with the help of a specialized program called HAP that contains all these applications, it is easy to access accurate information. It can predict thermal loads for both simple and complex buildings. A rectangular building has less heat loss or gain through the walls, which reduces the heating and cooling loads. It also has less natural ventilation and daylight, requiring more mechanical systems. An L-shaped building offers more solar exposure and natural ventilation, increasing thermal comfort but increasing the heating and cooling loads. T-shaped and H-shaped buildings offer more solar exposure but may have disadvantages like increased heat loss, uneven loads, and increased cooling and structural loads. A U-shaped building creates a courtyard, improving thermal comfort but also increasing heating and cooling loads. The six building shapes used in this study were generated using the hourly analysis program (HAP) by Carrier. This is a specialized program for calculating cooling and heating loads and simulating buildings, which uses the transfer functions method and the heat balance method, Figure 7.3 shows the dimensions of buildings. The buildings considered in this study are supposed to be located at 20.4 E longitude and 48.6 N latitude in Miskolc, Hungary at an elevation of about 130 m above mean sea level. In Miskolc, the summers are warm, the winters are cold and sometimes snowy, and it is partly cloudy year round. Over the course of the year, the temperature typically varies from −4 °C to 27 °C and is rarely below −11 °C or above 33 °C. The peak heating load for the winter occurred on December 14, when the

outside temperature was −9 °C. The peak cooling load for the summer occurred on July 17, when the outside temperature was 34 °C. The inside temperature was always considered to be 22 °C.

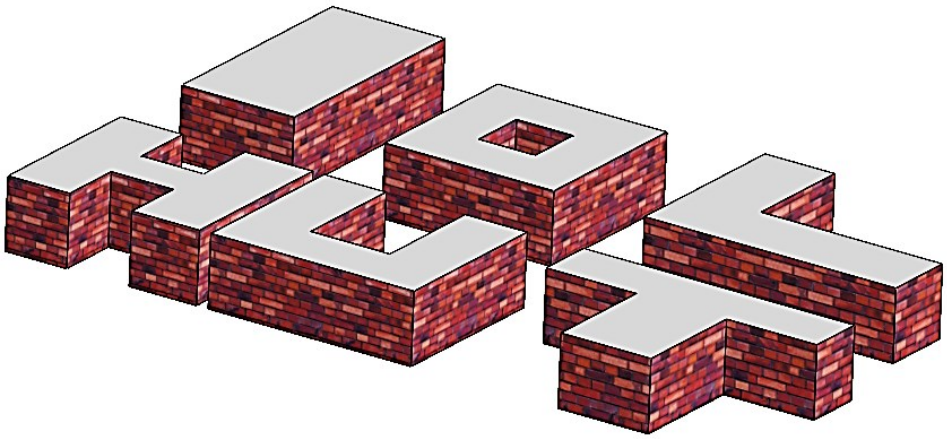


Figure 7.2. Different shapes of the buildings.

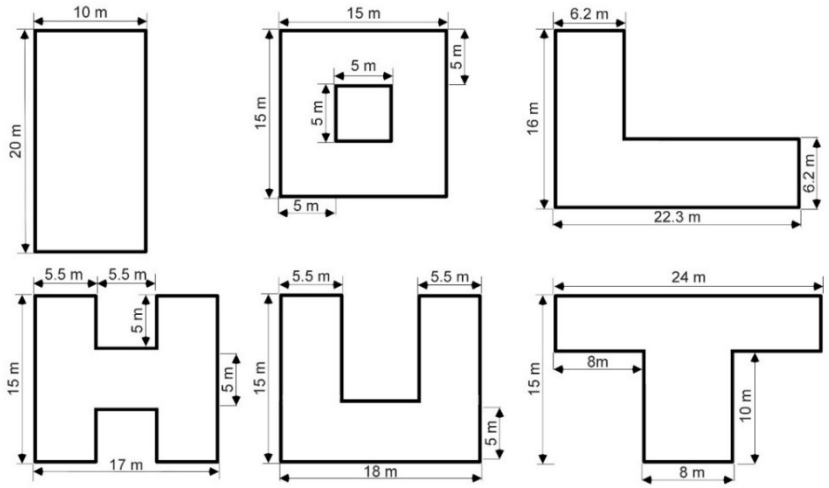


Figure 7.3. Dimensions of the buildings.

All the buildings have the same floor area (200 m^2) and height (6 m), thus the same volume (1200 m^3). However, they have different shapes and therefore different wall surface areas and thus total surface areas. The materials used for each component of a building are the same for all building forms. The newest and most prevalent materials in the building construction sector, as well as those with the lowest U-value in walls (0.637 W/m^2K), roofs (0.513 W/m^2K), windows (3.123 W/m^2K), floors (0.568 W/m^2K), were used to make the sample buildings. These buildings were each simulated as residential dwellings in Miskolc. External heat gains arrive from the transfer of thermal energy from the outside hot medium to the inside space, mostly in summer. Heat transfer takes place via conduction through external walls, the top roof, and the bottom ground. Solar radiation heat travels as electromagnetic waves from the sun and enters the houses through windows and doors. The amount of solar radiation depends on the orientation of the windows and doors, the time of day and year, and the presence of shading devices. The overall shading coefficient is 0.870 [84], an indicator of how well the glass is thermally insulating

(shading) the interior when there is direct sunlight on the window. Solar radiation can be beneficial for heating the house in the winter, but it frequently causes overheating in the summer. Ventilation is the transfer of heat through the movement of air between indoor and outdoor spaces. Ventilation can provide fresh air and improve indoor air quality, but it can also increase heat loss or gain depending on the temperature difference between indoor and outdoor air. The ventilation requirement is 0.3 L/s/m^2 [85] for residential dwelling unit applications. Infiltration is the movement of air through cracks and gaps in the building envelope. Heat can leak out of the building through poorly sealed windows and doors or enter through openings around pipes and wires. Infiltration can cause unwanted heat loss or gain, as well as moisture problems and air quality issues. The enter infiltration is taken for air change per hour (0.5 ACH) [86]. Other sources are internal heat generation, like in residential buildings with a maximum of ten occupants (40 W/m^2), sedentary activities (67.4 W/m^2 sensible heat and 35.2 W/m^2 latent heat), miscellaneous loads (60 W/m^2 sensible heat and 55 W/m^2 latent heat), electric equipment (2.69 W/m^2), and light (10.76 W/m^2). Thermal characteristics were determined using mixed modes with 95% efficiency, a thermostat range of 24–18.3 °C, and 15–20 h of operation on weekdays and 10–20 h on weekends. Each input parameter corresponds to a different attribute of the structure. For example, the relative compactness (RC), which is the ratio of surface area to volume [87], is determined as follows [88]:

$$RC = 6V^{2/3}A^{-1} \tag{7.1}$$

where A and V are the building surface area and volume, respectively. For a cuboid-shaped building, its value is unity, thus for any other rectangular building, it is smaller than 1.

The study compares the structure with and without windows. For the unglazed system, it analyzes six different building types with four orientations, which yields 24 cases. For the glazed system, Table 7.1 summarizes the possibilities used. There are six building shapes, depicted in Figure 7.3. There are four orientations of the building, N, E, S, or W. I examined five glazing areas, namely 5%, 10%, 15%, 20%, or 30% glazing area of the floor area. Five distribution scenarios were used in the study. Finally, I generated five window-distribution scenarios for each building shape, orientation, and glazing area. In each scenario, the windows were distributed into two of the five surfaces (N, E, S, W, and the horizontal roof), with half of the windows.

Table 7.1. The assumptions and conditions used to generate the database.

Features	Values and States
Number of buildings	6
orientations	4
Glazing areas	5 (5%, 10%, 15%, 20%, and 30%) of the floor area.
Distribution scenarios	5
Building volume	1200 m^3

This results in 6 × 5 × 5 × 4 = 600 samples for the glazed system, thus the total number of the examined possibilities is 600 + 24 = 624. The study focuses on the residential buildings' heating

load and cooling load parameters, respectively. These parameters depend on seven factors: RC, exposed area, wall area, roof area, glazing area (which is the total area of the glazing including the frame and sash [89]), orientation, and glazing area distribution.

7.2.2. The Main Equations

To estimate the thermal comfort of residents, this study applies intelligent models. The models' accuracy is measured by three criteria: R^2, MAE, and RMSE. Previous research has extensively validated and confirmed these criteria. Equations below show the formulae for these criteria, which contrast the observed and predicted values of the thermal comfort [90].

$$R^2 = 1 - \left[\sum_{i=1}^{N} (Q_{i_{predicted}} - Q_{i_{observed}})^2 \Big/ \sum_{i=1}^{N} (Q_{i_{observed}} - \overline{Q_{i_{observed}}})^2 \right] \tag{7.2}$$

$$MAE = (1/N) \sum_{i=1}^{N} \left| Q_{i_{observed}} - Q_{i_{predicted}} \right| \tag{7.3}$$

$$RMSE = \sqrt{(1/N) \sum_{i=1}^{N} \left[Q_{i_{observed}} - Q_{i_{predicted}} \right]^2} \tag{7.4}$$

The normalized root-mean-square deviation (NRMSD) depends on RMSE. The equations above show how to compare the actual and predicted values of HL or CL for an efficient building, which are denoted by $Q_{i_{observed}}$ and $Q_{i_{predicted}}$, respectively. The term N represents the total number of samples, and $\overline{Q_{i_{observed}}}$ is the average of the actual values of HL (or CL).

7.3. Results and Discussion

This study applied MLP and RB neural networks with three algorithms (LM, SCG, and RB) to simulate the HL and CL of the residential building. The algorithms were implemented using MATLAB R2020b and IBM SPSS 20 statistics software. The data set used to train the proposed network consisted of 624 samples, with seven independent factors affecting the HL and CL values, as explained earlier. A 70, 15, and 15 split was used to divide the data into training, testing, and validating sets, respectively. The performance of the models was evaluated using R^2, MAE, and RMSE, which are widely used in the literature [91]. To obtain the optimal structure of the models, an extensive trial and error process was conducted. The prediction results of each model for the HL and CL values were presented and discussed in detail.

To validate my results, I compared the predicted data with the experimental data collected by HAP, based on the CL and HL parameters. I used R^2 as a criterion for the predicted data. Figure 7.4 shows the comparison of the predicted data with the experimental data. The results indicate that the LM model has the best prediction performance among all models. The analytical results are also in good agreement with the experimental data, which confirms the reliability of the ANN training process using different algorithms. The RB model, however, has the lowest prediction accuracy and fails to optimize the ANN parameters effectively. The SCG model can be a suitable

alternative to the LM model and provides accurate results and predictions, but it is still less accurate than the LM model. Figure 7.5 presents a graphical comparison between the predicted and actual values of HL and CL for each model. Furthermore, you can see in Figure A1-3 in Appendix the histograms of the errors for each model and parameter. These figures illustrate the prediction capability of each model and the error distribution. The error analysis shows that the LM model has less error in prediction. Figure 7.5 also shows how close the experimental data and the prediction data are for each sample number using the LM, SCG, and RB models for HL and CL parameters. As mentioned before, I used the prediction data to verify the accuracy and validity of my analysis. Therefore, I compared the data of LM, SCG, and RB models for HL and CL parameters in detail and measured their deviation from the target data.

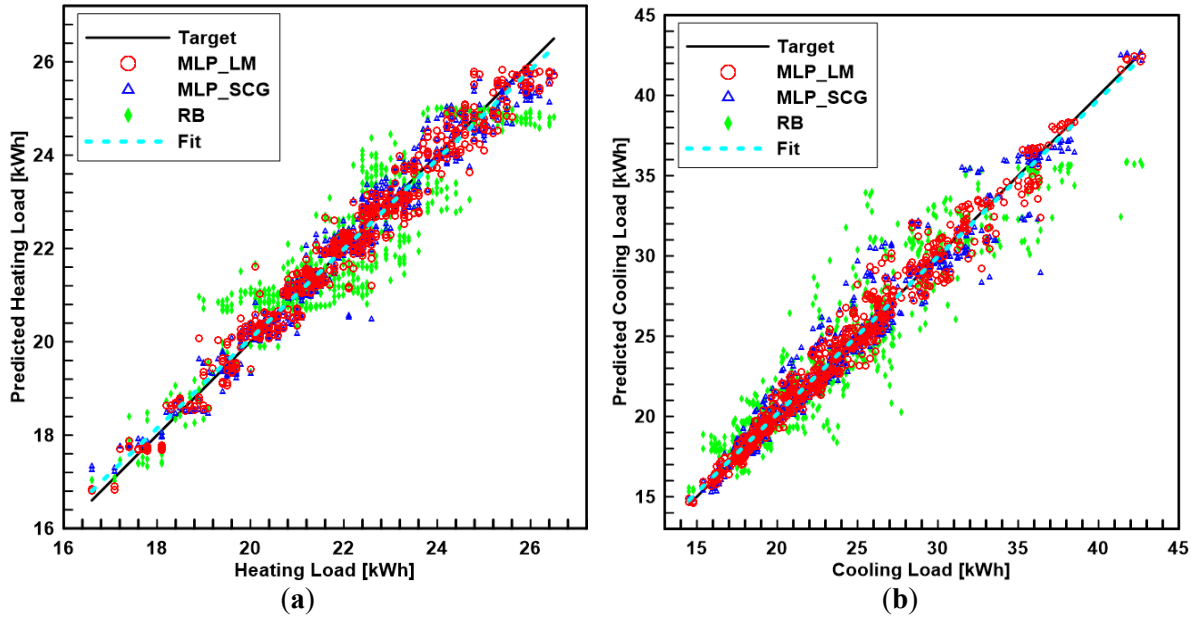
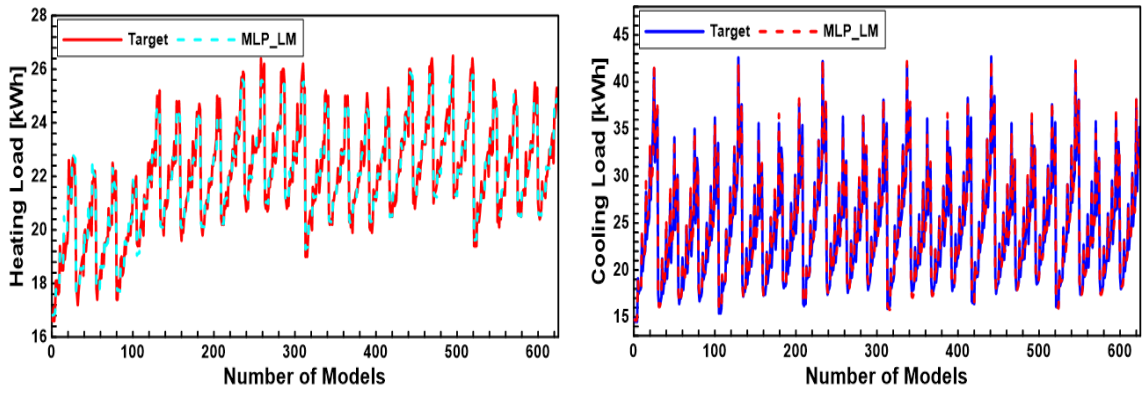


Figure 7.4. The result of R^2 values for the data predicted by three neural networks: MLP_LM, MLP_SCG, and RB (**a**) for the heating load and (**b**) for the cooling load.



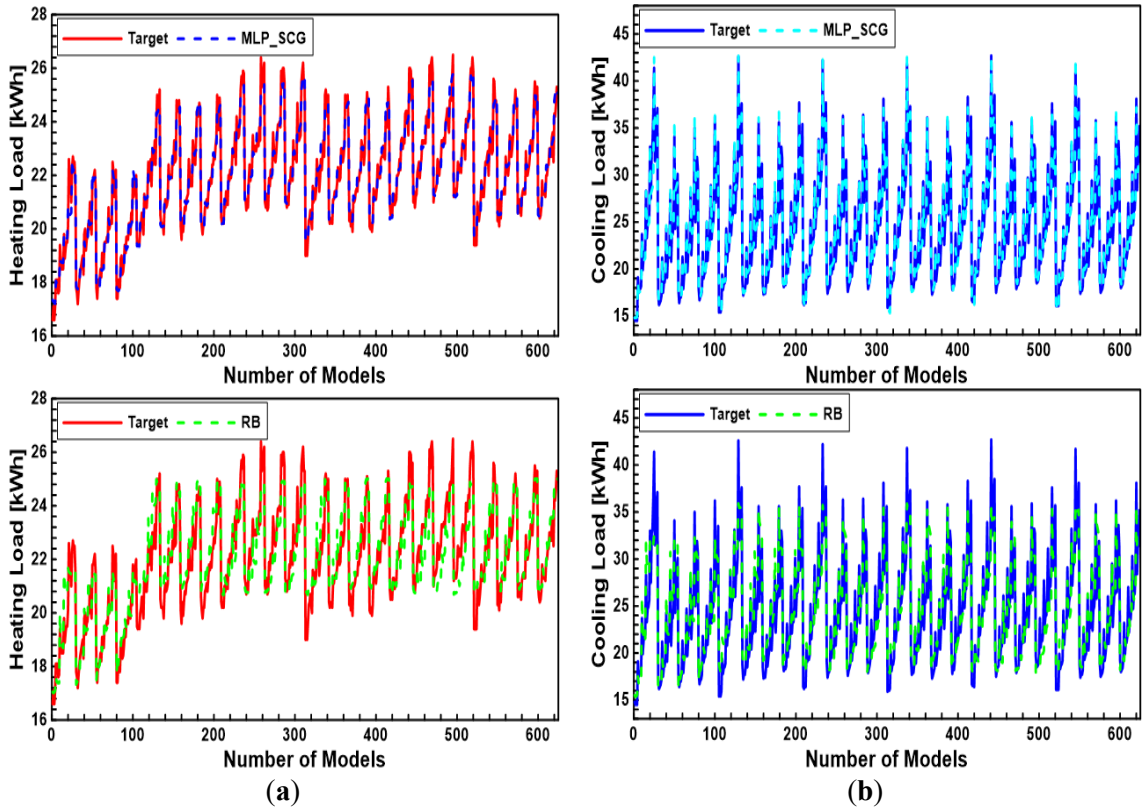


Figure 7.5. The predicted data values per model number for the (**a**) heating load on the left side and (**b**) cooling load on the right side, through MLP_LM, MLP_SCG, and RB with target.

Table 7.2 and Table 7.3 show the values of these criteria for each model and parameter. A lower RMSD value indicates a higher accuracy of the model. The LM model has the lowest RMSD values for both HL and CL parameters (0.348 and 0.947, respectively), which means it has the highest accuracy among all models. The RB model has the highest RMSD values and the lowest accuracy. The NRMSD value is another measure of accuracy, with a lower value indicating a better performance. The LM model has the lowest NRMSD values for both parameters, which confirms its superior performance. The R^2 value is a measure of how well the model fits the data, with a higher value indicating a better fit. Table 7.2 and Table 7.3 show that the LM model has the highest R^2 values for both parameters, which means it has the best fit to the data. The MAE value is a measure of the average error of the model, with a lower value indicating a smaller error. In the tables, the LM model has the lowest MAE values for both parameters (0.273 and 0.682, respectively), which means it has the smallest error among all models. Based on these results, I can conclude that the LM model is the best model for optimizing the ANN parameters and predicting the target data.

Table 7.2. Performance measures of the models for the HL parameter: RMSE, R^2, and MAE.

Models	RMSD	NRMSD	R^2	MAE
LM	0.348	0.0157	0.973	0.273
SCG	0.371	0.0168	0.961	0.287
RB	0.746	0.0338	0.902	0.609

Table 7.3. Performance measures of the models for the CL parameter: RMSE, R^2, and MAE.

Models	RMSD	NRMSD	R^2	MAE
LM	0.947	0.0388	0.93	0.682
SCG	1.226	0.0502	0.787	0.811
RB	2.185	0.0895	0.457	1.615

The thermal energy demand necessary to maintain an indoor comfort zone within a building is commonly referred to as the heating and cooling load. This load is subject to modification based on the architectural characteristics and design of the building. The heating and cooling demand is influenced by various factors, including but not limited to the building envelope, orientation, window-to-wall ratio, climate, and occupancy patterns.

The results of the analysis show that the heating and cooling loads vary significantly depending on the building shape (exposed area for wall, roof, and RC), orientation, glazing area, and glazing distribution. The results also show that there is a trade-off between minimizing the heating load in winter and minimizing the cooling load in summer. The results indicate that some building shapes are more efficient than others in terms of energy consumption and thermal comfort.

My work concludes that the optimal building shape that balances the heating and cooling load close to the mean value is the L-shape shown in Figure 7.6. This is beneficial because it allows for the use of smaller and more efficient HVAC systems compared to all other shapes except the rectangular one. This can lead to lower energy costs and reduced carbon dioxide emissions. Additionally, having balanced heating and cooling load can help maintain a comfortable indoor environment by reducing temperature fluctuations [92]. Also, the best orientation for this shape is the south, which allows maximum solar gain in winter. The design area has a moderately cold climate that requires heating for about half of the year. Furthermore, the best glazing area for this shape is 20%, which provides sufficient daylight and ventilation without causing excessive heat loss or gain. From this point of view, the L-shape is better than the rectangular shape. Finally, the best glazing distribution for this shape is on the east and south sides, which optimizes the solar exposure and shading effects throughout the day. In this case, the building has a heating load of 22.5 kWh and a cooling load of 24.5 kWh for the temperatures given in Section 7.2.1, which are both close to the mean values in Table A2-1 in Appendix, of 22 kWh and 24.4 kWh, respectively. This model's heating and cooling values are the closest to the mean value compared to the other 624 models.

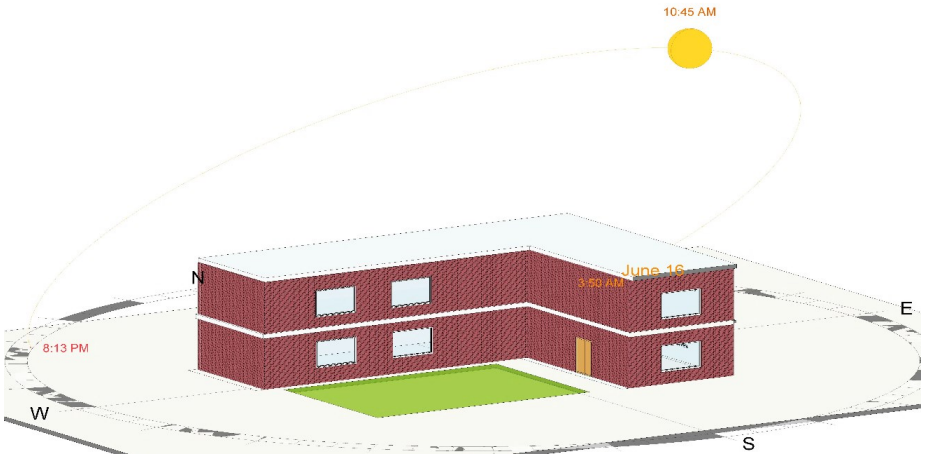


Figure 7.6. The L-shaped building.

7.4. Numerical Simulation

The final part of this research numerically simulates the walls of this building for the winter and summer seasons based on the peak load of each season. This transient simulation does not include other components that affect the heating and cooling load, such as the roof, ventilation, leakage, glass load, and internal loads from people, lighting, etc. These factors are only considered in the steady-state case. The transient simulation uses December and July as representative months. For the Equation and its discretization and the Leapfrog–Hopscotch Structure see section 2.1.

7.4.1. Geometry and Material Properties

A cross-section of a wall is considered with dimensions 42 cm in the x and 100 cm in the y and z direction, as can be seen in Figure 7.7. The four layers consist of gypsum plaster, heavy weight concrete, insulation, and face brick. This wall structure is the same and has the same U value as used in Section 7.2.1.

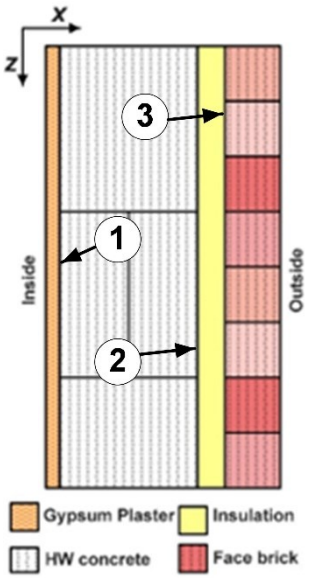


Figure 7.7. Cross section of a four-layer wall and the numbers mean border layer 1,2, and 3.

In the current work, the real material properties listed in Table 7.4 are taken into account. Note that these coefficients are constants inside a material, that is, they do not change with time, space, or temperature, but they have a discontinuity at the border of the materials.

Table 7.4. The properties of the materials used were taken from HAP based on ASHRAE Standard [84].

Layers	Thickness [cm]	$\rho\left[\mathrm{kgm^{-3}}\right]$	$c\left[\mathrm{Jkg^{-1}K^{-1}}\right]$	$k\left[\mathrm{Wm^{-1}K^{-1}}\right]$
Gypsum plaster	2	1601.8	840	0.726
HW concrete	25	2242.6	840	1.73
Insulation	5	91.3	840	0.043
Face brick	10	2082.4	920	1.3

7.4.2. Mesh Construction, Initial and Boundary Conditions

In my one-dimensional simulation, I assumed that the wall thickness is 42 cm. To test the independence of the results on mesh density, nine meshes with an increasing number of cells along the *x*-axis were used for the four layers of wall starting from $\Delta x_i = 1.0$ cm (Nx = 42) until $\Delta x_i = 0.1$ cm (N = 420). The heat loss was calculated according to the number of cells. See Figure A1-4 in Appendix due to that when I increase the number of cells to more than 336, the heat loss remains approximately the same. Therefore, I chose that mesh with $\Delta x_i = 0.125$ cm, where the number of cells in each layer was $N_1 = 16, N_2 = 200, N_3 = 40$, and $N_4 = 80$. The simulation time is one month for the winter season and for the summer, and both contain 31 days, thus $t_{\mathrm{fin}} = 2{,}678{,}400\mathrm{s}$ serves as the final time (the end of the examined time period). The duration of each time step is also expressed in seconds, first with $\Delta t = 100\mathrm{s}$.

Convective and radiation heat transfer occur in the *x*-direction, which is the direction of heat transfer depicted in Figure 7.7. Table 7.5 summarizes the boundary inside and outside condition of heat transfer for winter season convection coefficients (h_c), temperature and emissivity (ε) of the surfaces for the inside constant conditions and for the outside changing with time depending on the ambient conditions. The inside boundary condition is taken to be the constant comfort zone, which is taken from [84]. The outside condition changes with time depending on the changing weather conditions. The ambient air temperature, velocity, and the total solar radiation are taken according to real weather conditions in Miskolc city from 1st December 00.0 a.m. to 31st December 00.0 p.m. [93]. Table 7.6 shows the boundary condition for the summer season and the data are taken according to real weather conditions in Miskolc city from 1st July 00.0 a.m. to 31st July 00.0 p.m. [93].

Table 7.5. Convection, radiation, and heat source parameters are present on both sides of the wall during the winter season [93].

	$h\left[\frac{\mathrm{W}}{\mathrm{m^2K}}\right]$	Temperature [K]	ε	$\sigma^*\left[\frac{\mathrm{W}}{\mathrm{m^2K^4}}\times 10^{-8}\right]$
Inside condition	8.3	295	0.9	5.1
Outside condition	0.6–19.6	264–284	0.9	5.1

Table 7.6. Convection, radiation, and heat source parameters are present on both sides of the wall during the summer season [93].

	$h_c \left[\frac{\mathbf{W}}{\mathbf{m^2K}}\right]$	Temperature [K]	ε	$\sigma^* \left[\frac{\mathbf{W}}{\mathbf{m^2K^4}} \times \mathbf{10^{-8}}\right]$
Inside condition	8.3	295	0.9	5.1
Outside condition	2.7–16.9	284.2–307	0.9	5.1

I obtain the values of the coefficients in my equations as follows: [25],

$$K = \frac{h}{c\rho\Delta x},\ \sigma = \frac{\sigma^*}{c\rho\Delta x},\ \sigma^* = 5.678 * 10^{-9} * \varepsilon,\ q = \frac{q^*}{c\rho\Delta x} + \frac{h}{c\rho \cdot \Delta x} \cdot u_{\mathrm{a}}$$

I also supposed that the inside and outside elements have heat sources as follows:

For the inside elements: $q_{in} = \frac{1}{c\rho} \times q_{in}^* + \frac{h_{in}}{c\rho \cdot \Delta x} \times u_{in}$

For the outside elements: $q_{\mathrm{out}}(t) = \frac{1}{c\rho} \times q_{\mathrm{out}}^*(t) + \frac{h_{\mathrm{out}}(t)}{c\rho \cdot \Delta x} \times u_{\mathrm{out}}(t)$

And $K(t) = \frac{h_{\mathrm{out}}(t)}{c\rho\Delta x},\ \sigma = \frac{\sigma_{\mathrm{out}}{}^*}{c\rho\Delta x},\ q(t) = \frac{q_{\mathrm{out}}{}^*(t)}{c\rho\Delta x} + \frac{h_{\mathrm{out}}(t)}{c\rho \cdot \Delta x} \cdot u_{\mathrm{out}}(t)$

where $q_{in}^* = \varepsilon_{in}\sigma_{in}(u_{in})^4$ and $q_{\mathrm{out}}^*(t) = \alpha_{\mathrm{sun}} G_{\mathrm{cout}}(t) + \alpha_{\mathrm{Low}}\varepsilon_{\mathrm{out}}\sigma_{\mathrm{out}}\left[u_{\mathrm{out}}(t)\right]^4$ [94].

The convection heat transfer coefficient for outside elements as a function of air velocity is like in section 6.2.4

The environment's air temperature is taken to be $22\ ^{\circ}\mathrm{C} \approx 295\ \mathrm{K}$ inside, and changes depending on weather conditions outside.

I calculated the initial temperature inside the wall using the assumption that before the simulation time, a stationary heat flow with constant flux evolved between the given boundary values of the inside and outside air temperatures. For example, in the case of one layer, it yields a linear function of the *x* variable for the initial condition:

$$u(x, t=0) = (u_{\mathrm{out,initial}} - u_{in})x / L_x + u_{in}$$

where $u_{\mathrm{out,initial}} = 264\ \mathrm{K}$ for the winter season, and for the summer season $u_{\mathrm{out,initial}} = 307\ \mathrm{K}$.

For the multi-layer wall case, the assumption of stationary heat conduction with the initial values at the boundaries implies that I have to use piecewise linear functions of the *x* variable for the initial condition:

$$q_{\mathrm{flux}} = (u_{\mathrm{out,initial}} - u_{in}) / \sum_{i=1}^{i=4} \frac{L_i}{K_i} \quad \text{where } u_{\mathrm{initial,Layer},i+1} = u_i - (q_{\mathrm{flux}} L_i / k_i)$$

7.5. Result for the Simulation of the Wall

7.5.1. Winter Simulation

This paragraph presents the results of a simulation of heat transfer and temperature distribution in a four-layered wall during the winter season (December). The simulation shows how the temperature varies across the layers and how much heat is lost from the inside to the outside of the

wall per meter square. Figure 7.8 shows the temperature distribution at the boundary surfaces of the layers when the wall faces North, Figure A1-5- A1-7 in Appendix, provide information for the East, South, and West directions, respectively. The following observations can be made from the results. The inside surface temperature (gypsum plaster) and the first cell of the HW concrete layer are close to the inside air temperature, indicating that these layers are insulated from the outside and the insulation layer's ability to minimize heat transfer from the building's interior to the exterior.

The first cell of the insulation layer and the outside surface temperature (brick) are close to the outside air temperature, indicating that the insulation layer has a low thermal conductivity and reduces heat transfer from the building's interior to the exterior.

The North-facing wall has the largest temperature gradient and the highest heat loss because it does not receive any solar radiation. For the South-facing wall, the opposite is true. The East-facing and West-facing walls have approximately similar temperature gradients and heat losses because they receive approximately similar amounts of solar radiation. They are intermediate between the North-facing and South-facing walls, but the east wall is slightly colder than the west wall.

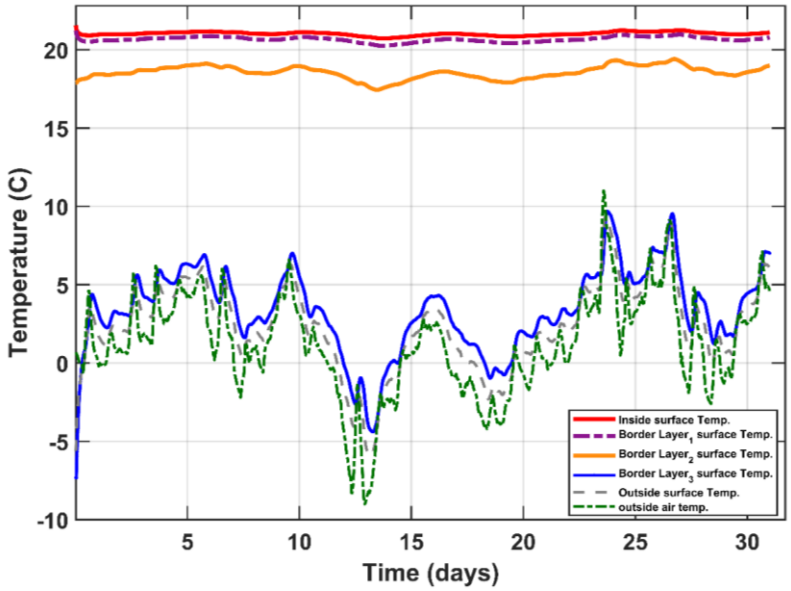


Figure 7.8. The temperature distribution in °C as a function of time in days for the four-layer wall facing North. Border layer means the boundary between two different materials (Border 1 between gypsum plaster and heavy-weight concrete, Border 2 between HWC and insulation, and Border 3 between insulation and face brick.).

Figure 7.9 shows that the heat loss per meter squared is the highest for the wall facing North and lowest for the wall facing South.

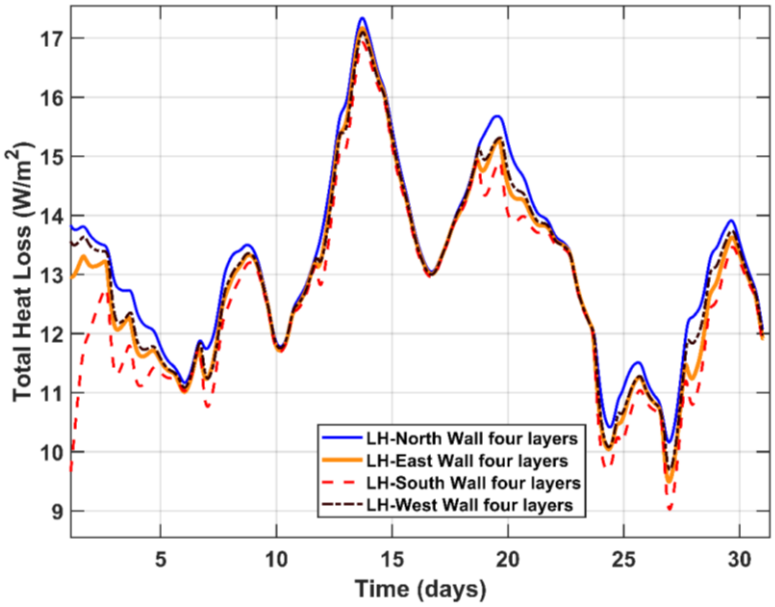


Figure 7.9. Total heat loss distribution in W/m^2 as a function of time in days for the wall simulation facing North, East, South, and West.

7.5.2. Summer Simulation

This section shows the results of a simulation of heat transfer and temperature distribution in the four-layered wall during the summer season (July), depicted in Figure 7.7. The simulation shows how the temperature varies across the layers and how much heat is gained from the outside to the inside of the wall per meter squared. Figure 7.10 shows the temperature distribution at the boundary surfaces of the layers when the wall faces North, Figure A1-8- A1-10 in Appendix, provide information for the East, South, and West directions, respectively. The following observations can be made from the results.

The inside surface temperature (gypsum plaster) and the first cell of the HW concrete layer are close to the inside air temperature, indicating that these layers have a high thermal conductivity, and these layers have a high ability to transfer heat from the building's exterior to the interior.

The first cell of the insulation layer and the outside surface temperature (brick) are higher than the outside air temperature during nights, indicating that the insulation layer has a low thermal conductivity and decreases the heat gain from outside to inside during the daytime. But in the case of the East, West, and especially the South wall, the surface temperature can be higher even during the daytime due to solar radiation and the rather high emissivity.

The North-facing wall has the smallest temperature gradient and the lowest heat gain because it does not receive a significant amount of solar radiation. The opposite is true for the South-facing wall. The East-facing and West-facing walls are between the North-facing and South-facing walls, but surprisingly, the East-facing wall has noticeably higher temperatures and heat gain than the West wall.

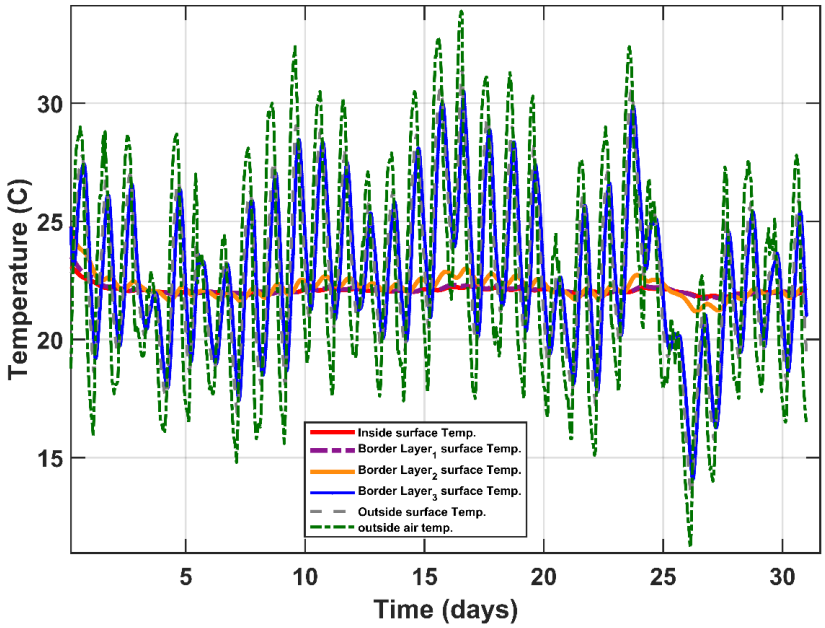


Figure 7.10. The temperature distribution in °C as a function of time in days for the four-layer wall facing North.

Figure 7.11 illustrates that the heat gain per meter squared is lowest for the wall facing North and highest for the wall facing South.

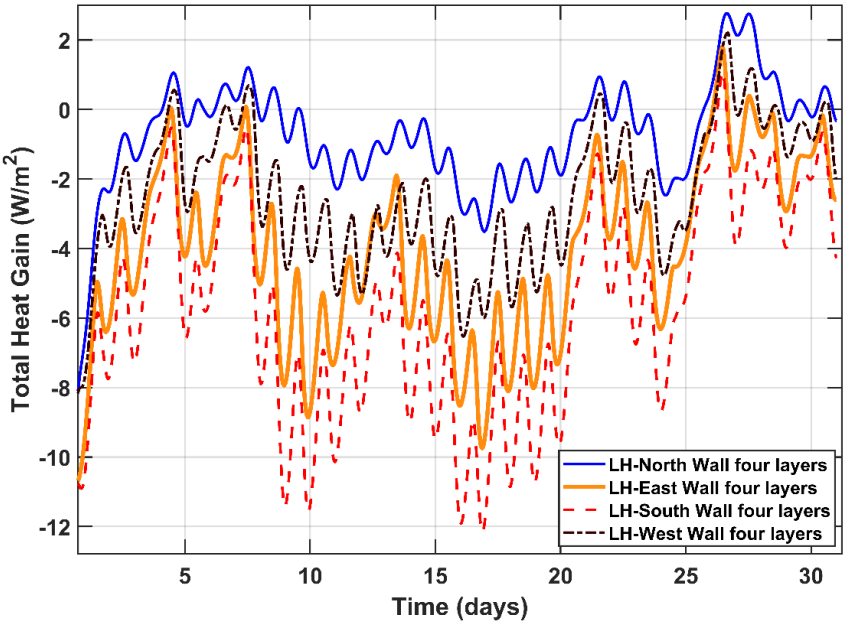


Figure 7.11. Total heat gain distribution in W/m^2 as a function of time in days for the wall simulation facing North, East, South, and West.

The wall simulation presents how the temperature varies in °C on the South-facing side of the building. Figure 7.12 displays the contour of the temperature distribution for two different seasons: winter (left) and summer (right). The simulation results are based on the last midnight of December and July, respectively. One can notice the heat stored by the wall during the summer night.

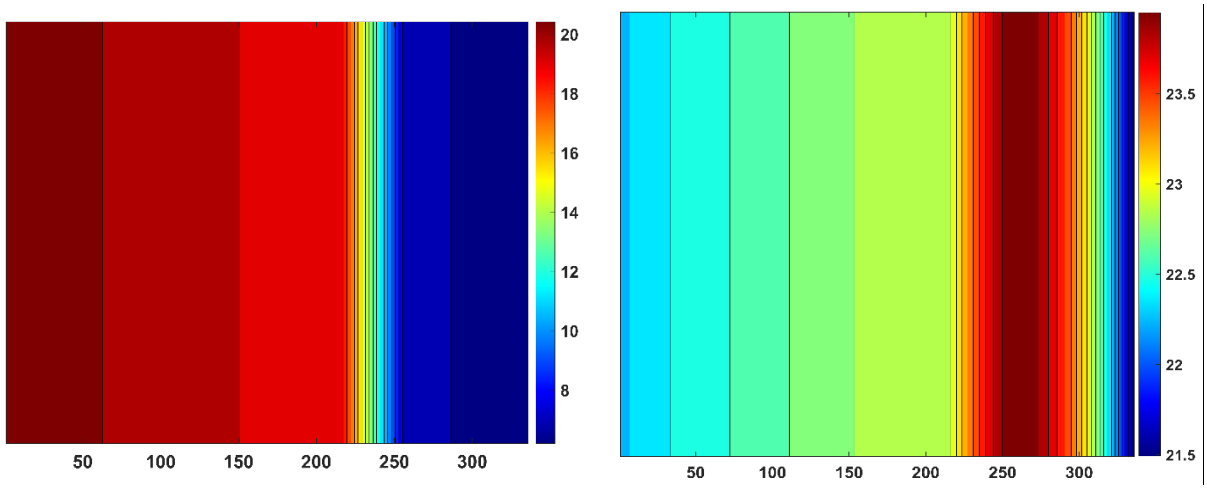


Figure 7.12. Contour of the temperature distribution in °C on the South-facing side of the building for two different seasons: winter (left) and summer (right).

Table 7.7 summarizes the heat loss and gain for the wall based on the simulation and HAP peak load, i.e., for the coldest and hottest days, respectively. The heat loss is the highest in winter on December 14, and the heat gain is the highest in summer on July 17.

Table 7.7. The heat loss and heat gain in transient method by LH and steady-state method by HAP.

Methods	**Winter**				**Summer**			
	Heat Loss [kW/m^2]				**Heat Gain [kW/m^2]**			
Direction	**N**	**E**	**S**	**W**	**N**	**E**	**S**	**W**
LH	17.34	17.16	16.92	17.08	−3.45	−9.745	−12.09	−5.99
HAP	19.35	19.44	19.35	19.43	−4.7	−7.65	−7.8	−7.43
percentage difference	2.7%	3.1%	3.3%	3.2%	7.6%	6.0%	10.7%	5.3%

The transient simulation by LH is more realistic and accurate because it uses real data from the weather website [93] for 2022 and 2023. The steady-state method by HAP is based on the location, design temperature for winter and summer, and estimated factors that increase the load to avoid under design. The steady-state values are always higher in winter than the transient simulation values.

7.6. Summary of this chapter

This study applied and compared MLP and RB neural networks with three algorithms (LM, SCG, and RB) to estimate the HL and CL of a residential building. The buildings were located in Miskolc, Hungary. The walls of the buildings consisted of gypsum plaster, HW concrete, insulation, and brick layers. The models were developed and assessed using 624 samples with seven independent factors. The accuracy and error of the models were evaluated by R^2, MAE, RMSE, RMSD, and NRMSD. The results revealed that the LM model outperformed the other models in terms of prediction accuracy and error minimization. The LM model also had the best fit to the experimental data; furthermore, the error is approaching zero compared to the other algorithms, which verified the validity of the MLP training process. This study demonstrated that

the MLP neural network with the LM algorithm can be an effective method for simulating the HL and CL of residential buildings.

In the simulation part, I investigated the thermal behavior of the same four-layered wall with different orientations in winter and summer by using the finite difference leapfrog–hopscotch algorithm coded in MATLAB. The simulation results showed the variation of temperature and heat transfer across the layers and the effect of solar radiation on the wall performance. The results revealed that the insulation layer played a key role in reducing the heat loss in winter and decreasing the heat gain in summer. The results also indicated that wall orientation had a significant influence on the thermal performance, with the North-facing wall being the most efficient in summer and the South-facing wall being the most efficient in winter. The West- and East-oriented walls are intermediate between the North and the South, but surprisingly, the East wall has noticeably higher temperatures and heat gain in summer than the West wall. The simulation process confirmed that the steady-state calculations for the most extreme day of the winter overestimate the heat loss. On the other hand, on the hottest day of summer, the steady-state method can under- or overestimate the heat gain depending on the orientation of the wall.

8. THE IMPACT OF ROOF ANGLE AND LOCATION ON THERMAL PERFORMANCE OF BUILDINGS

This chapter explores the impact of roof inclination on heat loss and gain within interior spaces. The investigation involves varying the roof angle in multiple ways, with a particular focus on the southern direction during winter to maximize daytime heat gain, and the northern direction in summer to minimize heat gain. After determining the optimal angle for both moderate (Miskolc, Hungary) and hot (Baghdad, Iraq) geo-graphical locations, I simulated the most effective models. I will perform steady-state simulations for several cases using the HAP program to find the optimal roof. Then I will conduct a transient simulation using the ANSYS program, incorporating additional scenarios to enhance the optimal roof. My simulations revealed that the optimal roof angle gives the minimum heat gain and loss. Then, to improve their performance, I implemented Trombe roofs. Additionally, I apply glass wool insulation to the north side during the winter and the south side during the summer.

8.1. The Materials and Studied Cases

8.1.1. Materials and location

In this study, the effects of the angle of the roof facing to the North as well as to the South with a constant horizontal length of the roof with four roof configurations are examined. Data collection was performed under the winter weather conditions of Miskolc, Hungary (48° 06' 9.00″ N and 20° 47' 17.99" E), and during a summer day in Baghdad, Iraq (33°19′ 15.60″ N and 44°24′00.00″ E). The urban climate in Baghdad is characterized by continuous solar radiation that is strong in sunlight throughout the year. In the Iraqi summer, for the design day, the relative humidity is around 21% and the air temperature under the shade ranges between 36 – 49 °C [93]. In addition, during the Hungarian winter, the outside air temperature ranges between -5 °C and 1 °C on the design day. Figure 8.1 demonstrates this, and Table 8.1 and Table 8.2 present the properties of the attic, roof and ceiling, respectively.

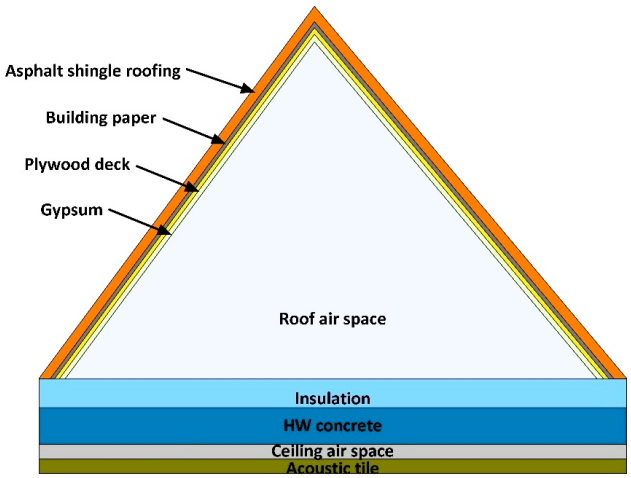


Figure 8.1. The materials used in the roof and ceiling.

For other required values, sample values given in studies in the literature were used [95] [96]. The aim of examining different designs and applying thermal insulation materials is to hinder the thermal energy transfer to or from the structure.

Table 8.1. The properties and thickness of the roof layers.

	Asphalt shingle	**Building paper**	**Plywood deck**	**Gypsum**
***k* [W/(m K)]**	0.79	0.04	0.0554	0.16
ρ [kg/m^3]	1700	930	288	800
C_p [J/(kg K)]	1000	1300	1298	837
Thickness [mm]	40	5	13	13

Table 8.2. The properties and thickness of the ceiling layers [60].

	Acoustic tile	**Ceiling air space**	**HW concrete**	**insulation**
***k* [W/(m K)]**	0.2877	0.4116	1.3	0.1017
ρ[kg/m^3]	480.6	1.1	977.1	8.0
Cp [J/(kg K)]	840	1.007	840	840
Thickness [mm]	30	20	150	100

8.1.2. Mathematical Model, Geometric Configurations, and ANSYS Setup

The methodology section of this research paper explores the impact of roof design on thermal performance across diverse climatic zones. I analyzed a roof model of a 3 m by 3 m section of a building. The inclined surface can be oriented in the north or south direction, and the horizontal surface will be level. The inclined surface can have four proportions for the larger section (1/2, 2/3, 5/6, and 1) of the roof length, and the remainder is the smaller section. In the winter case in Miskolc, the larger section faces south, and the smaller section faces north to maximize heat gain and minimize heat loss. In the Iraqi summer, the opposite occurs: the larger section faces north, and the smaller section faces south to reduce heat gain. Figure 8.2 illustrates the winter cases studied for Miskolc, Hungary, and Figure 8.3 shows the summer cases for Baghdad, Iraq. The larger section of the roof has Θ angle, and the smaller section has Φ angle. The angle selection depends on the latitude of the city, where the maximum radiation occurs close to the latitude angle [97]. I add and subtract the latitude angle by a certain number of degrees to emphasize the best angle by testing more possibilities. Figure 8.2 and Figure 8.3 show this process in detail. For Φ it depends on completing the triangle angles.

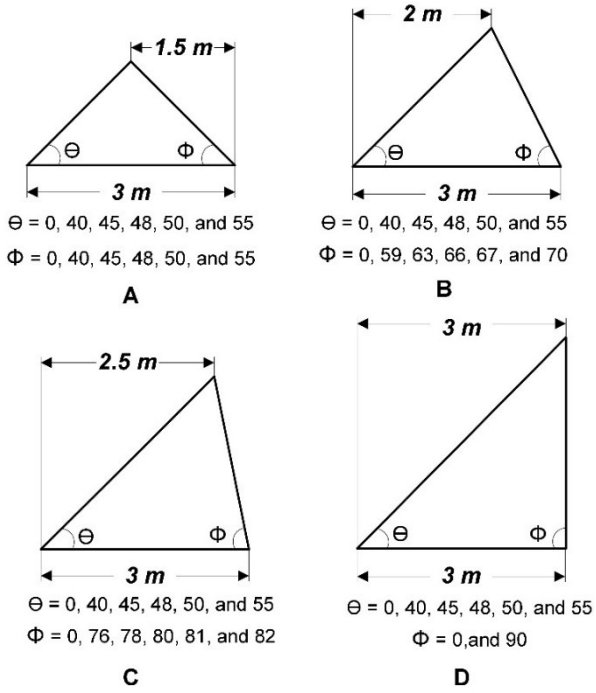


Figure 8.2. Roof dimensions and angles for the winter case (A) case (1/2) of roof length, (B) case (2/3) of roof length, (C) case (5/6) of roof length, and (D) case (1) of roof length.

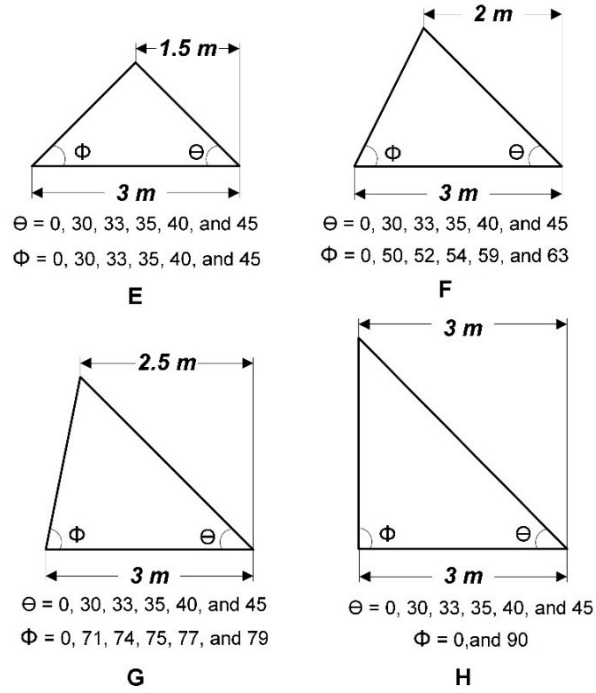


Figure 8.3. Roof dimensions and angles for the summer case (E) case (1/2) of roof length, (F) case (2/3) of roof length, (G) case (5/6) of roof length, and (H) case (1) of roof length.

Heat transfer through attics: The thermal calculation of the attic depends on whether it is ventilated or not. In the summer, the temperatures inside can approach the outside temperature (having a well-vented attic), but it is still necessary to account for the heat radiation transferred to the ceiling from the roof [98]. For unventilated attics, heat transfer occurs through the closed system composed of the ceiling, the attic space and the roof.

In order to calculate the R-value of the roof ceiling system, it is necessary to combine the effects of the R-value of the ceiling, the R-value of the roof, and the thermal resistance of the space. A practical approach for this is calculating first the R-value of the ceiling and roof separately, using resistance networks, and considering the still-air case for the attic surfaces. After that, the following relation expresses the overall R-value of the ceiling-roof combination per unit area of the ceiling.

$$R = R_{ceiling} + R_{roof}\left(\frac{A_{ceiling}}{A_{roof}}\right) \tag{8.1}$$

Data were collected using the HAP program to calculate heat transfer for each steady-state model specifically focusing on selected working hours (00:00, 06:00, 09:00, 12:00, and 15:00) [99]. These models were created by varying angles and lengths, as illustrated in Figure 8.2 and Figure 8.3. Additionally, I analyzed heat loss and gain based on weather conditions in Miskolc during winter and Baghdad during summer.

Upon analyzing the cases presented in Figure 8.2 and Figure 8.3, I will identify the roof that minimizes heat losses and the roof that minimizes heat gain. Subsequently, I will numerically simulate these models using the Fluent program and explore additional solutions to enhance and compare them. This study bases its mathematical explanation of the heat transmission mechanism on the following assumptions:

Because of its low flow velocity, the air is an incompressible fluid, Newton's laws apply to air. The flow field is described as two-dimensional steady-state flow.

The two-dimensional governing equations for air can be written as follows [100].

Continuity equation :
$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = 0 \tag{8.2}$$

Momentum equations:
$$\rho_a\left(u_x\frac{\partial u_x}{\partial x} + \frac{\partial u_x}{\partial y}\right) = \frac{\partial p}{\partial x} + \mu_a\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2}\right) \tag{8.3}$$

$$\rho_a\left(u_x\frac{\partial u_y}{\partial x} + \frac{\partial u_y}{\partial y}\right) = \frac{\partial p}{\partial y} + \mu_a\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2}\right) \tag{8.4}$$

Energy equation:
$$\left(\rho_a C_{pa}\frac{\partial T_a}{\partial t}\right)\rho_a = k_a\left(\frac{\partial^2 T_a}{\partial x^2} + \frac{\partial^2 T_a}{\partial y^2}\right) \tag{8.5}$$

where ρ_a, C_{pa}, k_a, u, T are density, specific heat, thermal conductivity, velocity and temperature of air, respectively.

The transient 2D model of the roof has been established using the Fluent program to study the effect of the range of angles of the roof, the roof with Trombe and with insulation on the effective side in reducing heat gains for residential buildings. I positioned the Trombe along the southern roof surface in winter and along both northern and southern roof surfaces in summer. Trombe roofs have a channel width of 0.05 m and a vertical extension of 0.25 m at 90 degrees parallel to the

effective roof surface. The material is glass with a thickness of 1 cm for the winter case and from the same material as the last layer in summer. The 2D roof model with 300 cm height consists of four layers, and the dimensions of this roof are listed in Figure 8.1. This model was used to study the effect of incorporating Trombe in a building's external roof for heat gain reduction. The produced finite volume mesh resulted in 24500 cells. The time step used in this study was equal to 150 s, and the number of time steps is 576, so the total time is 24 hours. After cautious verification of the independence of the effects on these variables, the time step and the grid size were chosen.

8.1.3. Boundary Conditions and Mesh Independence

The internal boundary conditions are determined based on the comfort zones during the coldest weather in Hungary, and the hottest weather in Iraq. The external conditions, on the other hand, the HAP and ANSYS simulations rely on the meteorological data outlined in Table 8.3 and Table 8.4. I approximated the initial temperature using the assumption that, prior to the simulation period, at midnight on the previous day of the simulation day, a stationary heat flow with constant flux had developed between the specified boundary values of the external and internal air temperatures. The values utilized for these boundary conditions are detailed in Table 8.5 and Table 8.6 [101].

Table 8.3. Weather data on winter [11].

Date: 11/01/2024	location: Miskolc Hungary		
	Temperature [ºC]	Velocity [m/s]	h_c [W/(m^2 K)]
0:00	-4	1.8	22.11
6:00	-5	1.67	19.12
9:00	-3	1.8	22.11
12:00	1	1.67	19.12
15:00	0	1.38	13.25

Table 8.4. Weather data on summer [93].

Date: 16/08/2023	location: Baghdad Iraq		
	Temperature [ºC]	Velocity [m/s]	h_c [W/(m^2 K)]
0:00	39	1.67	19.12
6:00	36	0.83	5.174
9:00	44	3.3	29.88
12:00	47	3.3	29.88
15:00	49	1.8	22.11

Table 8.5. The boundary of convection, radiation, and temperature on both sides of the roof.

	h_c [W/(m^2 K)]	**[ε]**	**[T °C]**
Outer surface	13.25 - 22.11	0.9	0 - -5
Inner surface	8.3	0.9	22

Table 8.6. The boundary of convection, radiation, and temperature on both sides of the roof.

	h_c [W/(m^2 K)]	**[ε]**	**[T °C]**
Outer surface	5.174 -29.88	0.65	39 - 49
Inner surface	8.3	0.65	24

It is unequivocal that enhancing the count of computational cells can lead to more precise outcomes, albeit at a greater computational cost. The results exhibit negligible variation when the cell count surpasses a specific threshold, signifying the achievement of mesh independence. To corroborate the independence of the results from the mesh density, I examined twelve meshes with an escalating total count of elements for roofs. The heat loss was computed based on the element count. I present the best model, the heat loss remains constant when the element count exceeds 24500. Consequently, I established an element size of 0.015 m, resulting in cells shaped as rectangular cuboids with a total element count of 24500.

8.2. Results

8.2.1. HAP simulation

Now I present the results of the Heat transfer (HAP) model, which illustrates the heat loss of roof (in W/m^2) for all scenarios shown in Figure 8.2 and Figure 8.3 over time. I tested the heat loss and gained as a function of time for different angles in winter cases for the north-facing roof cases and the south-facing roof for the same cases. Therefore, the angle Θ of the roof facing north changes from 40 to 55 degrees for selected angles. However, for the roof facing south, Φ changes from 40 to 55 degrees for selected angles. Similarly, I investigated the heat loss and gained for these sides based on roof inclination angles during various time periods. As observed in Figure 8.4, I can achieve maximum heat gain and minimum heat loss; thus, I will emulate this scenario as the optimal model for winter.

In the summer cases I tested, the angle of the south-facing roof varied from 33 to 45 degrees for specific angles. However, for the roof facing north, Φ is changing from 30 to 45 degrees for selected angles, where the north-facing roof and the south-facing roof are on the left side of the previous cases. As one can see in Figure 8.4, I can attain a minimum heat gain. Hence, I will simulate this scenario as the ideal model for summer. I present the optimal roof case in Hungary faced to south and the summer optimal roof case in Iraq faced to north in Figure 8.4. the first case is on the left and the second case is on the right, respectively.

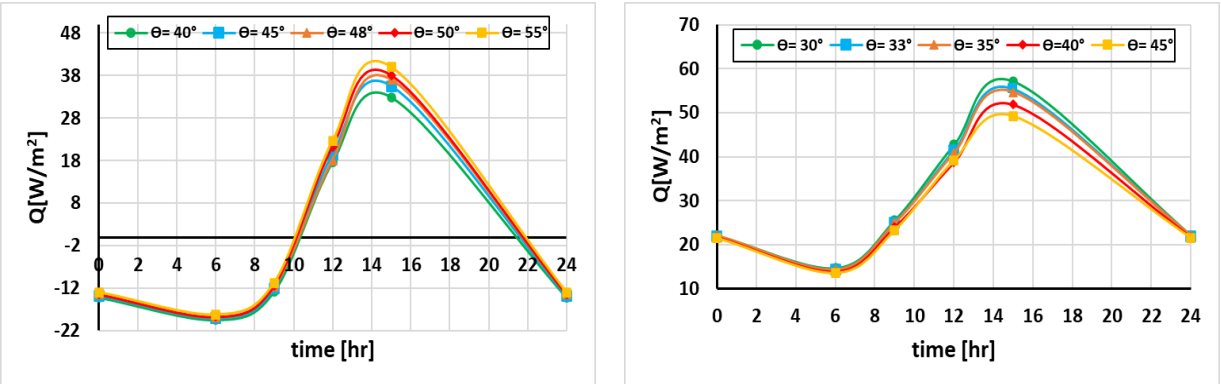
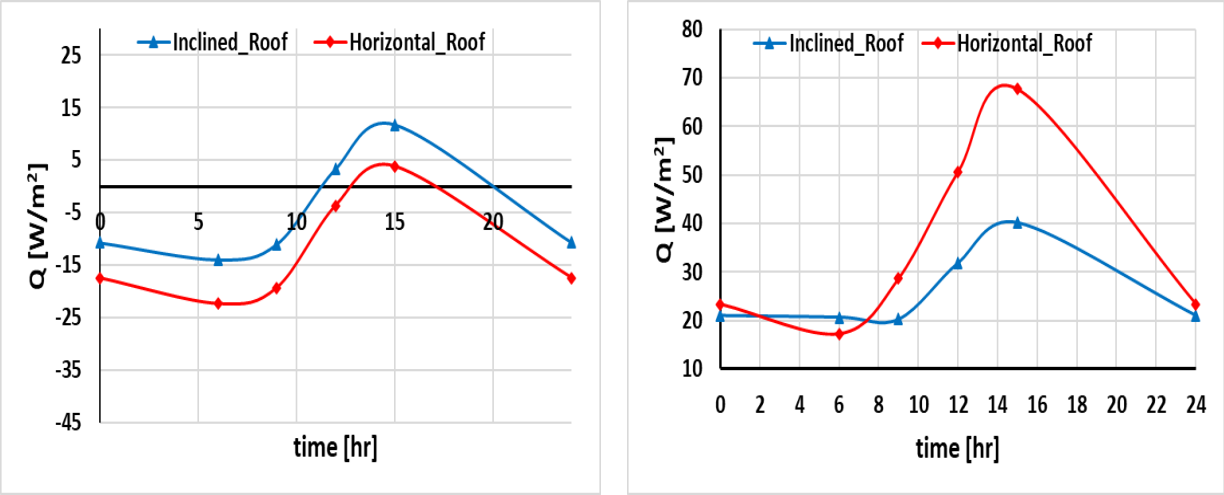


Figure 8.4. Roof heat gain as a function of the time for the winter optimal case in Hungary faced to south right side and the summer optimal case in Iraq faced to north left side.

In Figure 8.5, I conduct a comparative analysis between the optimal models and the horizontal model. Specifically, I take into account the orientations facing east and west, integrating them into the load calculations for the north and south directions. The resulting combined load is subsequently compared to the load associated with the horizontal configuration. Although an inclined roof has a larger surface area than a horizontal roof, it has reduced heat loss and gain. This is attributed to a shorter duration of exposure to beam solar radiation, resulting in lower heat gain, and to the consistent thermal convection on the sides of the inclined roof, additionally, when exposed to solar radiation, it stores energy and re-radiates it, which minimizes heat loss compared to a horizontal roof. Consequently, the inclined design offers improved thermal performance despite its increased surface area.



winter **summer**

Figure 8.5. Roof heat loss and gain with time on left side winter season and right-side summer season.

8.2.2. ANSYS simulation

In my ANSYS simulation, I investigated three scenarios. Initially, I analyzed the optimal model, revealing that heat transfer from the roof to the ceiling amounted to 29.939 W/m^2 of the ceiling area during summer, with a corresponding 24.43 W/m^2 heat loss during winter. Subsequently, I explored enhancements to the roof. My second case involved the addition of a Trombe roof, which effectively cooled the roof's outer surface in summer, mitigating direct sun exposure. As a result, summer heat gain decreased to 16.234 W/m^2, while in winter, the strategy was different by making holes with diameter 5 cm in the bottom and top of the effective roof facing to the south in order to heat the circulated air coming from roof space in exposed Trombe roof to sunshine in the daytime, then heat loss decreased to 21.77 W/m^2. In my final improvement, I installed insulators of glass wool [94] on the south side for summer and on the north side for winter. This adjustment led to a reduction in summer heat gain to 23.997 W/m^2 and a reduction in winter heat loss to 20.91 W/m^2. So Figure 8.6 shows the layout of the three cases in winter and summer and Figure 8.7 shows the contour of temperature of all cases.

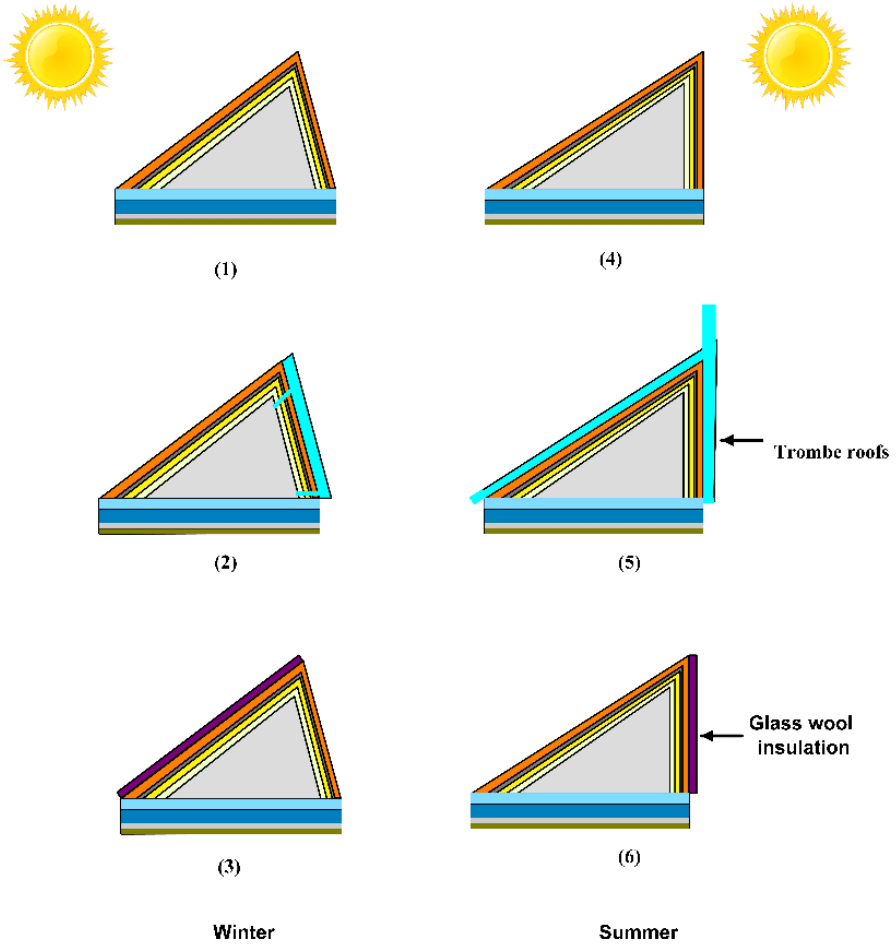
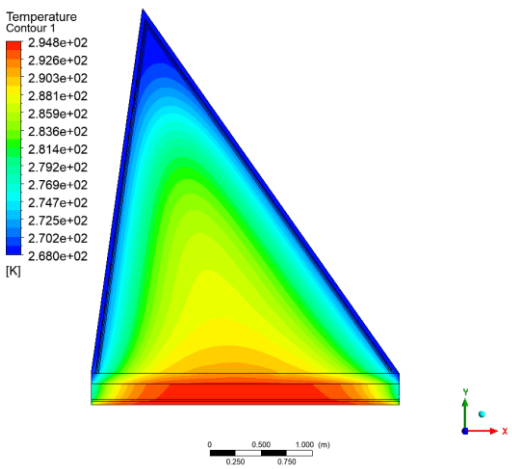
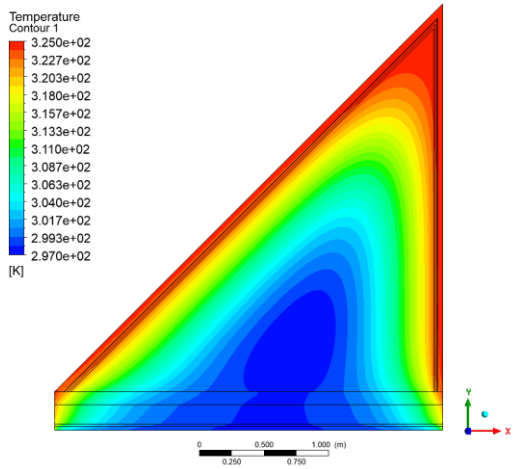


Figure 8.6. Illustration of the cases I simulated, including optimal roof, roof with Trombe, and glass wool insulation on one side of roof. The roofs on left side winter season and right side summer season.



1



4

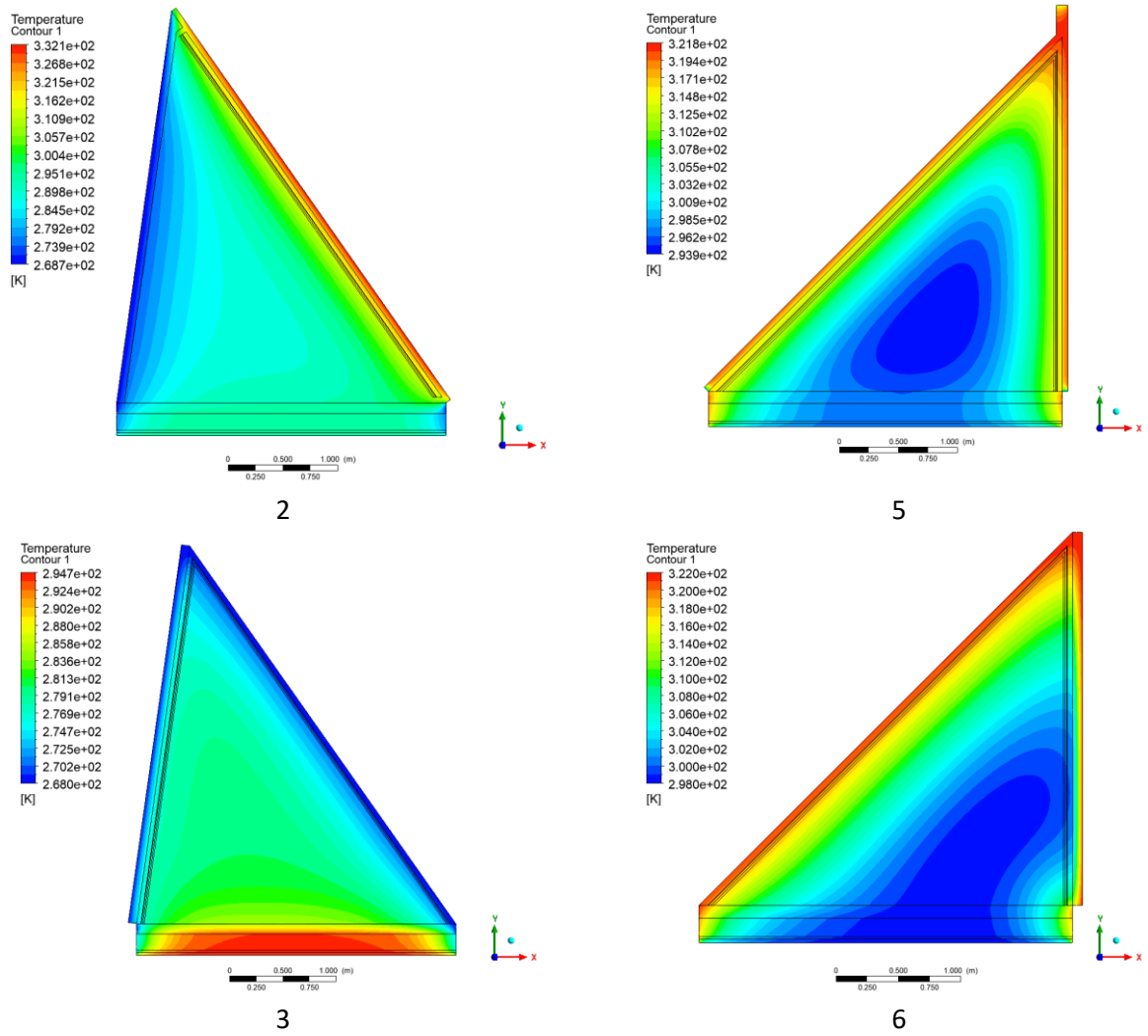


Figure 8.7. Contour of temperature for the optimal roof cases, Trombe and with insulation on effective side on left side winter season Hungary, Miskolc and right-side summer season Baghdad, Iraq.

Figure 8.8 presents a comparative analysis between the HAP program and the ANSYS program concerning the optimal model. The simulations of steady-state behavior in HAP and transitional-state behavior in ANSYS Fluent exhibit convergence, yielding highly similar results.

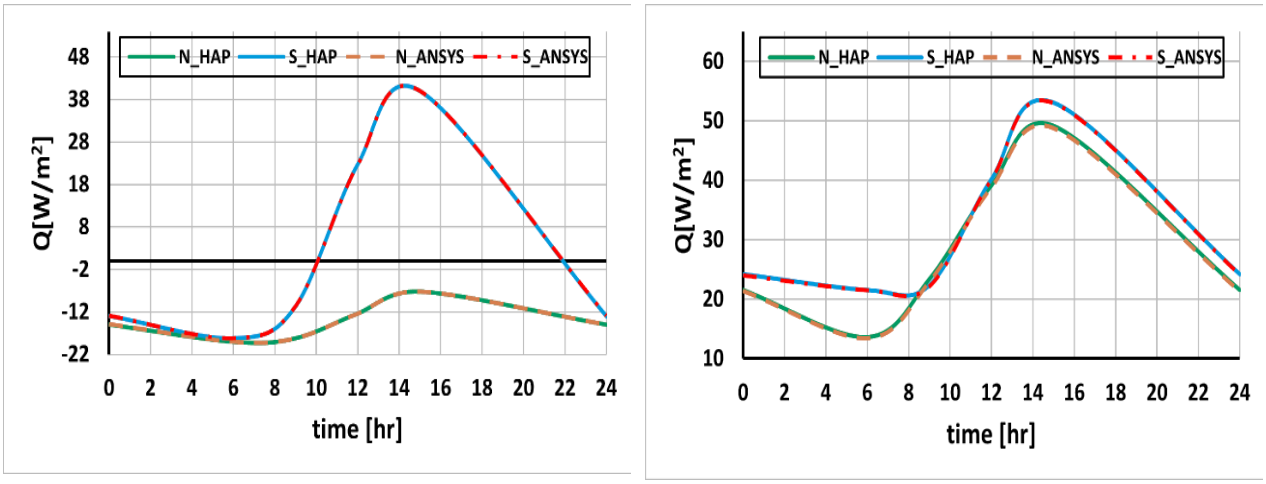


Figure 8.8. Comparison for the optimal roof, based on HAP and ANSYS (N-north side roof, and S-south side roof) result. Left side: winter in Hungary, Miskolc. Right-side: summer in Baghdad, Iraq.

Table 8.7 shows the heat flow across the ceiling to the inside of the building for all cases of simulation in winter and summer

Table 8.7. The properties of the materials used.

Winter cases	$Q\left[\mathrm{W/m^2}\right]$	**Summer cases**	$Q\left[\mathrm{W/m^2}\right]$
Normal roof	24.43	Normal roof	29.939
Trombe roof	21.77	Trombe roof	16.234
Insulator on effective side roof	20.91	Insulator on effective side roof	23.997

8.3. Summary of this chapter

In my study, I investigated the impact of roof inclination angles on energy consumption in buildings. Specifically, I focused on optimizing these angles for different climatic conditions. My research was conducted in two distinct cities: Baghdad, characterized by hot summers, and Miskolc, known for relatively cold winters.

For buildings in Baghdad's scorching summers, I determined that a roof inclination Φ angle of 90 degrees toward the south was most effective in minimizing heat gain. Conversely, a 45-degree inclination Θ angle toward the north provided optimal results. In Miskolc's cold winters, I found that a roof inclination Φ angle of 82 degrees toward the south minimized heat loss, while a 55-degree Θ angle toward the north was ideal.

To further enhance energy efficiency, I explored additional modifications. Notably, the Trombe wall system significantly reduced heat gain to the inside of the building, lowering it from 29.939 to 16.234 W/m^2. Additionally, insulating the active side of the roof reduced heat gain to 23.997 W/m^2 for summer cases 4, 5 and 6 respectively in Figure 8.6 and, for heat loss from inside of building to the environment of roof was 24.43, 21.77, 20.91 W/m^2 for winter cases 1, 2 and 3 respectively in Figure 8.6, which are normal roof Trombe and insulating the active side of the roof, respectively. These findings underscore the importance of thoughtful design and strategic adjustments in achieving sustainable energy use.

9. THESES – NEW SCIENTIFIC RESULTS

T1. I analyzed nine numerical algorithms for solving the heat equation, focusing on positivity-preserving methods stable across time step sizes and system stiffness. The study tested 2500-cell, two-dimensional stiff systems with random, discontinuous parameters. I compared accuracy and CPU efficiency, finding the 3-stage LNe3 and LH-CNe methods most accurate. I examined how increasing stiffness ratios, decreasing CFL limits, and varying spatial anisotropy affected accuracy. The study assessed performance with growing horizontal-vertical cell dimension differences. I recommended optimal methods for scenarios like OEH structure, unstructured meshes, and highly anisotropic systems, aiming to guide effective positivity-preserving method selection. [61].

T2. I developed a novel, fully explicit, stable numerical algorithm for time-dependent diffusion equations with linear and nonlinear reaction terms. Based on the UPFD idea and theta-formula, it's second order in time step size and unconditionally stable for linear cases. It outperforms other methods and MATLAB routines in accuracy and stability for nonlinear cases, though not positivity-preserving. Stable for large time steps even with strong nonlinearity, it's easy to implement and suitable for unstructured grids. This pseudo-implicit algorithm combines key advantages of explicit and implicit methods.[42].

T3. I optimized the leapfrog-hopscotch method for the heat conduction equation, focusing on free convection and radiation terms. Best results treat convection 50% at old- and new-time levels, ensuring stability and second-order temporal convergence. The radiation term is best handled pseudo-implicitly for excellent stability. The algorithm performs well under low CFL limits. [72].

T4. I studied a diffusion-reaction PDE with a linear reaction term and space-time-dependent nonlinear coefficients. Nine numerical algorithms reproduced these, with Dufort-Frankel and leapfrog-hopscotch explicit schemes outperforming standard explicit and implicit methods. In a 2D case simulating wall surface temperature with wind-driven forced convection and rapidly varying material properties, explicit stable methods proved more efficient than implicit ones, with efficiency expected to grow with system size. [102].

T5. I compared MLP and RB neural networks using LM, SCG, and RB algorithms to predict heating and cooling loads in Miskolc, Hungary residences. The MLP with LM algorithm excelled in accuracy and error reduction. I also studied a four-layered wall's thermal behavior across orientations in winter and summer using the leapfrog-hopscotch finite difference algorithm. Insulation and orientation significantly affect thermal performance, with North-facing walls optimal in summer and South-facing in winter. Steady-state calculations overestimate winter heat loss and variably estimate summer heat gain. [99].

T6. I studied heat loss and gain in inclined roofs in Miskolc (cold winters) and Baghdad (hot summers). In Miskolc, optimal roof angles of 82° south and 55° north minimized heat loss. In Baghdad, 90° south and 45° north reduced heat gain. Trombe wall systems cut heat gain from 29.939 to 16.234 W/m², and insulating the active roof side lowered it to 23.997 W/m² in summer. Winter heat loss in Miskolc was 24.43, 21.77, and 20.91 W/m². [103] [104].

ACKNOWLEDGEMENTS

First and foremost, I owe a tremendous debt of gratitude to Dr. Endre Kovács, who has been instrumental in my success over the past four years and introduced me to this fascinating research field. I feel incredibly fortunate to have had the guidance and mentorship of such a highly respected academic. I am deeply grateful for your excellent counsel, support, and persistent, understanding assistance. Beyond our academic interactions, I am grateful to have met someone with such grace and friendliness. I highly valued our bi-daily meetings, which not only served as crucial checkpoints to keep me on track academically but also provided me with ample encouragement. I am profoundly grateful for the immeasurable contributions you have made to my development.

I would also like to thank my co-supervisor, Dr. Betti Bolló. Your exceptional academic expertise and insights have greatly enriched the quality of my work. Your constant support, guidance, and encouragement have been invaluable throughout this entire process.

I extend my heartfelt thanks to my friend Issa Omle, who has been my biggest cheerleader throughout this journey. I cannot thank you enough for your love and support.

Thanks to my teammates and colleagues, Ádám Nagy, Pszota Gabor, Tamás Jenyó, Mahmood Saleh, Saif Ali Kadhim, Salah Hadi Mshehid and Noora Ghader, who have helped me in many ways.

I want to express my deepest gratitude to my family, whose belief in my abilities and unwavering support have played an integral role in my accomplishments. I am deeply grateful to my mother Nooriya, my brother Sajad, and my sisters Hanan, Ghufran, and Duaa, along with the unwavering spirit of my father, who has been with me since the beginning of my studies.

I am incredibly grateful for the opportunity to be a part of an exceptional research university that provides deep insights into different sciences. Completing a PhD is a monumental achievement and a significant milestone in my professional journey.

Lastly, my greatest aspiration is that my research proves beneficial for everyone. The drive to advance knowledge in this field continues to fuel my motivation and determination. I am deeply grateful to all the individuals mentioned, as none of this would have been possible without their guidance and encouragement. I am excited to see what the future holds. It is amazing how life leads us to unexpected places and introduces us to remarkable people when we least expect it.

Thank you for everything. I appreciate everyone's help.

REFERENCES

[1] B. Chegari, M. Tabaa, E. Simeu, F. Moutaouakkil, and H. Medromi, “Multi-objective optimization of building energy performance and indoor thermal comfort by combining artificial neural networks and metaheuristic algorithms,” *Energy Build*, vol. 239, p. 110839, 2021, doi: 10.1016/j.enbuild.2021.110839.

[2] I. International Energy Agency, “Energy Policy Review Hungary 2022,” 2022.

[3] F. Feng, Y. Fu, Z. Yang, and Z. O’Neill, “Enhancement of phase change material hysteresis model: A case study of modeling building envelope in EnergyPlus,” *Energy Build*, vol. 276, p. 112511, 2022, doi: 10.1016/j.enbuild.2022.112511.

[4] F. C. Winkelmann and S. Selkowitz, “Daylighting simulation in the DOE-2 building energy analysis program,” *Energy Build*, vol. 8, no. 4, pp. 271–286, 1985, doi: 10.1016/0378-7788(85)90033-7.

[5] I. Beausoleil-Morrison, M. Kummert, F. MacDonald, R. Jost, T. McDowell, and A. Ferguson, “Demonstration of the new ESP-r and TRNSYS co-simulator for modelling solar buildings,” *Energy Procedia*, vol. 30, pp. 505–514, 2012, doi: 10.1016/j.egypro.2012.11.060.

[6] L. Chujie , Li, Sihui , Penak, Santhan Reddy, Olofsson, Thomas, “Automated machine learning-based framework of heating and cooling load prediction for quick residential building design,” *Energy*, p. 126689, 2023, doi: 10.1016/j.energy.2023.127334.

[7] J. E. Christian and J. Kosny, “Thermal performance and wall ratings,” *ASHRAE J*, vol. 38, no. 3, Mar. 1996.

[8] D. Aelenei and F. M. A. Henriques, “Analysis of the condensation risk on exterior surface of building envelopes,” *Energy Build*, vol. 40, no. 10, pp. 1866–1871, 2008, doi: 10.1016/J.ENBUILD.2008.04.003.

[9] J. Ji, C. Luo, W. Sun, H. Yu, W. He, and G. Pei, “An improved approach for the application of Trombe wall system to building construction with selective thermo-insulation façades,” *Chinese Science Bulletin*, vol. 54, no. 11, pp. 1949–1956, Jun. 2009, doi: 10.1007/S11434-009-0353-6.

[10] S. S. Moody and D. J. Sailor, “Development and Application of a Building Energy Performance Metric for Green Roof Systems,” 2016.

[11] “(3) Passive cooling techniques in buildings: Past and present a review.” Accessed: May 10, 2024. [Online]. Available: https://www.researchgate.net/publication/285901724_Passive_cooling_techniques_in_buildings_Past_and_present_a_review

[12] M. Z. M. Ashhar and C. H. Lim, “Numerical simulation of heat transfer in a roof assembly with reflective insulation and radiant barrier,” *Build Simul*, vol. 13, no. 4, pp. 897–911, 2020, doi: 10.1007/s12273-020-0624-3.

[13] M. Rawat and R. N. Singh, “A study on the comparative review of cool roof thermal performance in various regions,” *Energy and Built Environment*, vol. 3, no. 3, pp. 327–347, 2022, doi: 10.1016/j.enbenv.2021.03.001.

[14] Q. Mao and M. Yang, “Experimental and numerical investigation on heat transfer performance of a solar double-slope hollow glazed roof,” *Appl Therm Eng*, vol. 180, no. August, p. 115832, 2020, doi: 10.1016/j.applthermaleng.2020.115832.

[15] J. Han, L. Lu, and H. Yang, “Investigation on the thermal performance of different lightweight roofing structures and its effect on space cooling load,” *Appl Therm Eng*, vol. 29, no. 11–12, pp. 2491–2499, Aug. 2009, doi: 10.1016/J.APPLTHERMALENG.2008.12.024.

[16] A. Gustavsen, D. Arasteh, B. P. Jelle, C. Curcija, and C. Kohler, “Developing low-conductance window frames: Capabilities and limitations of current window heat transfer

design tools - State-of-the-art review," *J Build Phys*, vol. 32, no. 2, pp. 131–153, Oct. 2008, doi: 10.1177/1744259108097672.

[17] D. W. Hahn, *Heat conduction.* Wiley, 2012.

[18] H.-C. Huang and A. S. Usmani, *Finite Element Analysis for Heat Transfer*. Springer London, 1994. doi: 10.1007/978-1-4471-2091-9.

[19] H. Yu *et al.*, "The Moisture Diffusion Equation for Moisture Absorption of Multiphase Symmetrical Sandwich Structures," *Mathematics*, vol. 10, no. 15, p. 2669, Jul. 2022, doi: 10.3390/math10152669.

[20] L. Mátyás and I. F. Barna, "General Self-Similar Solutions of Diffusion Equation and Related Constructions," *Romanian Journal of Physics*, vol. 67, p. 101, 2022.

[21] T. Kusuda, "Fundamentals of building heat transfer," *J Res Natl Bur Stand (1934)*, vol. 82, no. 2, p. 97, 1977.

[22] J. H. Lienhard, "A Heat Transfer Textbook." [Online]. Available: http://ahtt.mit.edu

[23] L. Mátyás and I. F. Barna, "General Self-Similar Solutions of Diffusion Equation and Related Constructions," *Romanian Journal of Physics*, vol. 67, p. 101, 2022.

[24] G. V. Ochoa, W. Escobar Sánchez, S. De, and L. H. Truyoll, "Experimental and Theoretical Study on Free and Force Convection Heat Transfer," *Contemporary Engineering Sciences*, vol. 10, no. 23, p. 1143, 2017, doi: 10.12988/ces.2017.79124.

[25] J. P. Holman, "Heat Transfer, 10th ed," *McGraw-Hill Education: New York, NY, USA*, 2010.

[26] S. Savović and A. Djordjevich, "Numerical solution of the diffusion equation describing the flow of radon through concrete SEQ CHAPTER," *Appl Radiat Isot*, vol. 66, no. 4, pp. 552–555, Apr. 2008, doi: 10.1016/J.APRADISO.2007.08.018.

[27] F. Suárez-Carreño and L. Rosales-Romero, "Convergency and Stability of Explicit and Implicit Schemes in the Simulation of the Heat Equation," *Applied Sciences 2021, Vol. 11, Page 4468*, vol. 11, no. 10, p. 4468, May 2021, doi: 10.3390/APP11104468.

[28] L. Alberti, A. Angelotti, M. Antelmi, and I. La Licata, "Borehole Heat Exchangers in aquifers: simulation of the grout material impact," *RENDICONTI ONLINE DELLA SOCIETÀ GEOLOGICA ITALIANA*, vol. 41/2016, pp. 268–271, Nov. 2016, doi: 10.3301/ROL.2016.145.

[29] M. K. Singh, S. Rajput, and R. K. Singh, "Study of 2D contaminant transport with depth varying input source in a groundwater reservoir," *Water Sci Technol Water Supply*, vol. 21, no. 4, pp. 1464–1480, Jun. 2021, doi: 10.2166/WS.2021.010/831257/WS2021010.PDF.

[30] Y. Ji, H. Zhang, and Y. Xing, "New Insights into a Three-Sub-Step Composite Method and Its Performance on Multibody Systems," *Mathematics 2022, Vol. 10, Page 2375*, vol. 10, no. 14, p. 2375, Jul. 2022, doi: 10.3390/MATH10142375.

[31] A. Iserles, "A First Course in the Numerical Analysis of Differential Equations (Cambridge Texts in Applied Mathematics)," p. 477, 2008, Accessed: May 18, 2024. [Online]. Available: https://books.google.com/books/about/A_First_Course_in_the_Numerical_Analysis.html?id=M0tkw4oUucoC

[32] A. Saghyan, D. P. Lewis, J. Hrabe, and S. Hrabetova, "Extracellular diffusion in laminar brain structures exemplified by hippocampus," *J Neurosci Methods*, vol. 205, no. 1, pp. 110–118, Mar. 2012, doi: 10.1016/J.JNEUMETH.2011.12.008.

[33] M. H. Jacobs, "Diffusion Processes," 1967, doi: 10.1007/978-3-642-86414-8.

[34] J. H. Lienhard, "A Heat Transfer Textbook", Accessed: May 19, 2024. [Online]. Available: http://ahtt.mit.edu

[35] B. M. Chen-Charpentier and H. V. Kojouharov, "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.

[36] A. R. Appadu, "Performance of UPFD scheme under some different regimes of advection, diffusion and reaction," *Int J Numer Methods Heat Fluid Flow*, vol. 27, no. 7, pp. 1412–1429, 2017, doi: 10.1108/HFF-01-2016-0038.

[37] S. Savović, B. Drljača, and A. Djordjevich, “A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes,” *Ricerche di Matematica*, vol. 71, pp. 245–252, 2022, doi: 10.1007/s11587-021-00665-2.

[38] S. Pourghanbar, J. Manafian, M. Ranjbar, A. Aliyeva, and Y. S. Gasimov, “An efficient alternating direction explicit method for solving a nonlinear partial differential equation,” *Math Probl Eng*, vol. 2020, no. November, pp. 1–12, 2020, doi: 10.1155/2020/9647416.

[39] A. Al-Bayati, S. Manaa, and A. Al-Rozbayani, “Comparison of Finite Difference Solution Methods for Reaction Diffusion System in Two Dimensions,” *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 8, no. 1, pp. 21–36, 2011, doi: 10.33899/csmj.2011.163605.

[40] A. H. Askar, I. Omle, E. Kovács, and J. Majár, “Testing Some Different Implementations of Heat Convection and Radiation in the Leapfrog-Hopscotch Algorithm,” *Algorithms*, vol. 15, no. 11, p. 400, 2022.

[41] I. Omle, A. H. Askar, E. Kovács, and B. Bolló, “Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges,” *Energies (Basel)*, vol. 16, no. 12, p. 4604, 2023.

[42] 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, p. 3308, 2021.

[43] J. Muñoz-Matute, V. M. Calo, D. Pardo, E. Alberdi, and K. G. van der Zee, “Explicit-in-time goal-oriented adaptivity,” *Comput Methods Appl Mech Eng*, vol. 347, pp. 176–200, Apr. 2019, doi: 10.1016/j.cma.2018.12.028.

[44] M. H. Holmes, *Introduction to Numerical Methods in Differential Equations*. New York: Springer, 2007. doi: 10.1007/978-0-387-68121-4.

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

[46] 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.

[47] 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.

[48] B. M. Chen-Charpentier and H. V Kojouharov, “An unconditionally positivity preserving scheme for advection--diffusion reaction equations,” *Math Comput Model*, vol. 57, no. 9–10, pp. 2177–2185, 2013.

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

[50] A. Ali Habeeb, Nagy, Ádám, Barna, Imre Ferenc and Kovács Endre, “Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates,” 2023.

[51] E. Kovács, “New stable, explicit, first order method to solve the heat conduction equation,” *arXiv preprint arXiv:1908.09500*, 2019.

[52] E. Kovács, “A class of new stable, explicit methods to solve the non-stationary heat equation,” *Numer Methods Partial Differ Equ*, vol. 37, no. 3, pp. 2469–2489, 2021.

[53] E. Kovács, “A class of new stable, explicit methods to solve the non-stationary heat equation,” *Numer Methods Partial Differ Equ*, vol. 37, no. 3, pp. 2469–2489, 2020, doi: 10.1002/num.22730.

[54] E. Kovács, Á. Nagy, and M. Saleh, “A set of new stable, explicit, second order schemes for the non-stationary heat conduction equation,” *Mathematics*, vol. 9, no. 18, p. 2284, Sep. 2021, doi: 10.3390/math9182284.

[55] "Heun's method - Wikipedia." Accessed: Jul. 30, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Heun%27s_method

[56] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *J Res Natl Bur Stand (1934)*, vol. 49, no. 6, 1952.

[57] R. Barrett *et al.*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994. doi: 10.1137/1.9781611971538.

[58] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, 1986, doi: https://doi.org/10.1137/0907058.

[59] I. Omle, A. H. Askar, E. Kovács, and B. Bolló, "Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges," *Energies (Basel)*, vol. 16, no. 12, p. 4604, 2023.

[60] A. A. Habeeb, K. Endre, and B. Betti, "Multi objective optimization for house roof using artificial neural network model," *Multidiszciplináris Tudományok*, vol. 13, no. 2, pp. 11–25, Dec. 2023, doi: 10.35925/J.MULTI.2023.2.2.

[61] I. Omle, A. H. Askar, and E. Kovács, "SYSTEMATIC TESTING OF EXPLICIT POSITIVITY PRESERVING ALGORITHMS FOR THE HEAT-EQUATION," 2022, doi: 10.28919/jmcs/7407.

[62] M. Saleh, E. Kovács, and Á. Nagy, "New stable, explicit, second order hopscotch methods for diffusion-type problems," *Math Comput Simul*, vol. 208, pp. 301–325, Jun. 2023, doi: 10.1016/J.MATCOM.2023.01.029.

[63] B. Drljača and S. Savović, "Unconditionally positive finite difference and standard explicit finite difference schemes for power flow equation," *The University Thought-Publication in Natural Sciences*, vol. 9, no. 2, pp. 75–78, 2019.

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

[65] N. A. Mbroh and J. B. Munyakazi, "A robust numerical scheme for singularly perturbed parabolic reaction-diffusion problems via the method of lines," *Int J Comput Math*, vol. 99, no. 6, pp. 1139–1158, Jun. 2022, doi: 10.1080/00207160.2021.1954621.

[66] B. M. Chen-Charpentier and H. V Kojouharov, "An unconditionally positivity preserving scheme for advection--diffusion reaction equations," *Math Comput Model*, vol. 57, no. 9–10, pp. 2177–2185, 2013.

[67] 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 2022, Vol. 12, Page 1365*, vol. 12, no. 9, p. 1365, Sep. 2022, doi: 10.3390/BUILDINGS12091365.

[68] H. Liu and S. Leung, "An Alternating Direction Explicit Method for Time Evolution Equations with Applications to Fractional Differential Equations".

[69] H. Z. Barakat and J. A. Clark, "On the solution of the diffusion equations by numerical methods," 1966.

[70] 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.

[71] M. Saleh, Á. Nagy, and E. Kovács, "Part 1: Construction and investigation of new numerical algorithms for the heat equation," *Multidiszciplináris tudományok*, 2020, doi: 10.35925/j.multi.2020.4.36.

[72] A. H. Askar, I. Omle, E. Kovács, and J. Majár, "Testing Some Different Implementations of Heat Convection and Radiation in the Leapfrog-Hopscotch Algorithm," *Algorithms*, vol. 15, no. 11, p. 400, 2022.

[73] M. H. Jacobs, *Diffusion Processes*. Springer Berlin Heidelberg, 1935. doi: https://doi.org/10.1007/978-3-642-86414-8.

[74] J. H. Lienhard IV and J. H. Lienhard V, “A HEAT TRANSFER TEXTBOOK, fourth edition,” *Phlogiston Press*, p. 766, 2017, [Online]. Available: http://web.mit.edu/lienhard/www/ahttv211.pdf

[75] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, and C.W. Clark, *NIST Handbook of Mathematical Functions*, vol. 66, no. 1. New York: Cambridge Univ. Press, 2011. doi: 10.11316/butsuri.66.1_67_2.

[76] Y. Ji and Y. Xing, “Highly Accurate and Efficient Time Integration Methods with Unconditional Stability and Flexible Numerical Dissipation,” *Mathematics 2023, Vol. 11, Page 593*, vol. 11, no. 3, p. 593, Jan. 2023, doi: 10.3390/MATH11030593.

[77] G. Settanni and I. Sgura, “Devising efficient numerical methods for oscillating patterns in reaction-diffusion systems,” *J Comput Appl Math*, vol. 292, pp. 674–693, Jan. 2016, doi: 10.1016/j.cam.2015.04.044.

[78] L. Beuken, O. Cheffert, A. Tutueva, D. Butusov, and V. Legat, “Numerical Stability and Performance of Semi-Explicit and Semi-Implicit Predictor–Corrector Methods,” *Mathematics 2022, Vol. 10, Page 2015*, vol. 10, no. 12, p. 2015, Jun. 2022, doi: 10.3390/MATH10122015.

[79] V. S. Yadav, A. Singh, V. Maurya, and M. K. Rajpoot, “New RK type time-integration methods for stiff convection–diffusion–reaction systems,” *Comput Fluids*, vol. 257, p. 105865, May 2023, doi: 10.1016/J.COMPFLUID.2023.105865.

[80] V. Khakre, A. Wankhade, and M. A. Ali, “Cooling load estimation by CLTD method and hap 4.5 for an evaporative cooling system,” *International Research Journal of Engineering and Technology*, vol. 4, no. 1, pp. 1457–1460, 2017.

[81] A. H. Askar, E. Kovács, and B. Bolló, “ANN Modeling for Thermal Load Estimation in a Cabin Vehicle,” in *Vehicle and Automotive Engineering 4: Select Proceedings of the 4th VAE2022, Miskolc, Hungary*, Springer, 2022, pp. 357–373.

[82] T. Zarei and R. Behyad, “Predicting the water production of a solar seawater greenhouse desalination unit using multi-layer perceptron model,” *Solar Energy*, vol. 177, no. November 2018, pp. 595–603, 2019, doi: 10.1016/j.solener.2018.11.059.

[83] J. De Jesús, D. Garcia, H. Sossa, I. Garcia, A. Zacarias, and D. Mujica-vargas, “Energy processes prediction by a convolutional radial basis function network,” *Energy*, vol. 284, no. March, p. 128470, 2023, doi: 10.1016/j.energy.2023.128470.

[84] J. D. Spitler, *Load Calculation Applications Manual*, SI Edition. Printed in the United States ofAmerica, 2009. [Online]. Available: https://www.ashrae.org/

[85] S. Liaison *et al.*, “Ventilation for Acceptable Indoor Air Quality,” vol. 8400, 2012.

[86] M. Sherman, “Infiltration in ASHRAE ’ s Residential Ventilation Standards Infiltration in ASHRAE ’ s Residential Ventilation Standards,” no. October, 2008.

[87] R. Ouarghi and M. Krarti, “Building Shape Optimization Using Neural,” vol. 112, pp. 484–492, 2006.

[88] Y. Xu, F. Li, and A. Asgari, “Prediction and optimization of heating and cooling loads in a residential building based on multi-layer perceptron neural network and different optimization algorithms,” *Energy*, vol. 240, p. 122692, 2022, doi: 10.1016/j.energy.2021.122692.

[89] Y. Huang, J. Niu, and T. Chung, “Comprehensive analysis on thermal and daylighting performance of glazing and shading designs on office building envelope in cooling-dominant climates,” *Appl Energy*, vol. 134, pp. 215–228, 2014, doi: 10.1016/j.apenergy.2014.07.100.

[90] Y. Wu, P. Xie, and A. Dahlak, “Utilization of Radial Basis Function Neural Network model for Water production forecasting in Seawater Greenhouse units,” *Energy Reports*, vol. 7, pp. 6658–6676, 2021, doi: 10.1016/j.egyr.2021.09.165.

[91] M. Mosallanezhad and H. Moayedi, “Developing hybrid artificial neural network model for predicting uplift resistance of screw piles,” 2017.

[92] A. Burdick, “Strategy Guideline : Accurate Heating and Cooling Load Calculations,” in *Building America Building Technologies Program Office of Energy Efficiency and Renewable Energy U.S. Department of Energy*, no. June, 2011, pp. 1–47.

[93] “Weather-Online Website.,” *https://open-meteo.com, data access in 2022-2023.*, [Online]. Available: https://open-meteo.com

[94] I. Omle, E. Kovács, and B. Bolló, “Applying recent efficient numerical methods for long-term simulations of heat transfer in walls to optimize thermal insulation,” *Results in Engineering*, vol. 20, p. 101476, Dec. 2023, doi: 10.1016/J.RINENG.2023.101476.

[95] P. Bevilacqua, R. Bruno, S. Gallo, and J. Szyszka, “A validated multi-physic model for the optimization of an innovative Trombe Wall for winter use,” *JOURNAL OF BUILDING ENGINEERING*, vol. 86, Jun. 2024, doi: 10.1016/J.JOBE.2024.108915.

[96] M. Dhahri, N. Khalilpoor, H. Aouinet, A. Issakhov, H. Sammouda, and S. Emani, “Numerical Study on the Thermal Performance of Trombe Wall for Passive Solar Building in Semiarid Climate,” *International Journal of Photoenergy*, vol. 2021, 2021, doi: 10.1155/2021/6630140.

[97] S. A. Kadhim, M. K. S. Al-Ghezi, and W. Y. Shehab, “Optimum Orientation of Non-Tracking Solar Applications in Baghdad City.,” *International Journal of Heat \& Technology*, vol. 41, no. 1, 2023.

[98] S. Tong and H. Li, “An efficient model development and experimental study for the heat transfer in naturally ventilated inclined roofs,” *Build Environ*, vol. 81, pp. 296–308, Nov. 2014, doi: 10.1016/J.BUILDENV.2014.07.009.

[99] A. H. Askar, E. Kovács, and B. Bolló, “Prediction and Optimization of Thermal Loads in Buildings with Different Shapes by Neural Networks and Recent Finite Difference Methods,” *Buildings*, vol. 13, no. 11, p. 2862, 2023.

[100] E. Osterman, V. Butala, and U. Stritih, “PCM thermal storage system for ‘free’ heating and cooling of buildings,” *Energy Build*, vol. 106, pp. 125–133, Nov. 2015, doi: 10.1016/J.ENBUILD.2015.04.012.

[101] I. Omle, A. H. Askar, and B. B. Kov, Endre, “Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges,” 2023.

[102] A. H. Askar, Á. Nagy, I. F. Barna, and E. Kovács, “Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates,” *Computation*, vol. 11, no. 7, p. 127, 2023.

[103] A. H. Askar, I. Omle, and E. Kovács, “The role of roof angle and geographic location on the thermal performance of buildings,” *International Journal of Thermofluids*, vol. 27, p. 101192, May 2025, doi: 10.1016/J.IJFT.2025.101192.

[104] A. A. Habeeb, K. Endre, and B. Betti, “Multi objective optimization for house roof using artificial neural network model,” *Multidiszciplináris Tudományok*, vol. 13, no. 2, pp. 11–25, 2023.

LIST OF PUBLICATIONS RELATED TO THE TOPIC OF THE RESEARCH FIELD

(1) A. H. Askar, E. Kovács, and B. Bolló, “Prediction and Optimization of Thermal Loads in Buildings with Different Shapes by Neural Networks and Recent Finite Difference Methods,” Buildings, vol. 13, no. 11, p. 2862, 2023.

(2) A. Ali Habeeb, Nagy, Ádám, Barna, Imre Ferenc and Kovács Endre, “Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates,” 2023.

(3) Askar, A.H.; Omle, I.; Kovács, E.; Majár, J. Testing Some Different Implementations of Heat Convection and Radiation in the Leapfrog-Hopscotch Algorithm. Algorithms 2022.https://doi.org/10.3390/a15110400.

(4) 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, p. 3308, 2021.

(5) Omle, I.; Askar, A.H.; Kovács, E.; Bolló, B. Comparison of the Performance of New and Traditional Numerical Methods for Long-Term Simulations of Heat Transfer in Walls with Thermal Bridges. Energies 2023, 16, 4604. https://doi.org/10.3390/en16124604.

(6) A. H. Askar, E. Kovács, and B. Bolló, “ANN Modeling for Thermal Load Estimation in a Cabin Vehicle,” in Vehicle and Automotive Engineering 4: Select Proceedings of the 4th VAE2022, Miskolc, Hungary, Springer, 2022, pp. 357–373.

(7) Omle, I.; Askar, A.H.; Kovács, E. “Systematic testing of explicit positivity preserving algorithms for the heat-equation”. https://doi.org/10.28919/jmcs/7407.

(8) Omle, Issa, Ali Habeeb Askar, and Endre Kovács. "Optimizing the Design of Container House Walls Using Argon and Recycled Plastic Materials." *Buildings* 14, no. 12 (2024): 3944.

(9) A. A. Habeeb, K. Endre, and B. Betti, “Multi objective optimization for house roof using artificial neural network model,” Multidiszciplináris Tudományok, vol. 13, no. 2, pp. 11–25, Dec. 2023, doi: 10.35925/J.MULTI.2023.2.2.

(10) Abed, H. H., Omle, I., Askar, A. H., & Kovács, E. (2023). Experimental Study of the Stability and Thermophysical Properties for Different Particle Size of Al2o3-H2o Nanofluid. Journal of Engineering Science and Technology, 18(6), 2793-2808.

(11) Omle, I.; Askar, A.H.; Kovács, E.; "Impact of Wall Roughness Elements Type and Height on Heat Transfer Inside a Cavity". Pollack Periodica (2024). DOI:10.1556/606.2024.00986

(12) Habeeb, A. A., Hazim, A., Endre, K., & Károly, J. (2021). A new method to predict temperature distribution on a tube at constant heat flux. Multidiszciplináris Tudományok, 11(5), 363–372. https://doi.org/10.35925/J.MULTI.2021.5.40.

(13) Hazim, A., Habeeb, A. A., Károly, J., & Endre, K. (2021). Interpolated spline method for a thermal distribution of a pipe with a turbulent heat flow. Multidiszciplináris Tudományok, 11(5), 353–362. https://doi.org/10.35925/J.MULTI.2021.5.39.

(14) Jalghaf, H. K., Askar, A. H., Albedran, H., Kovács, E., & Jármai, K. (2023). Comparative study of different meta-heuristics on optimal design of a heat exchanger. Pollack Periodica, 18(2), 119–124. https://doi.org/10.1556/606.2022.00543.

(15) A. H. Askar, I. Omle, and E. Kovács, “The role of roof angle and geographic location on the thermal performance of buildings,” International Journal of Thermofluids, vol. 27, p. 101192, May 2025.

APPENDICES

A1

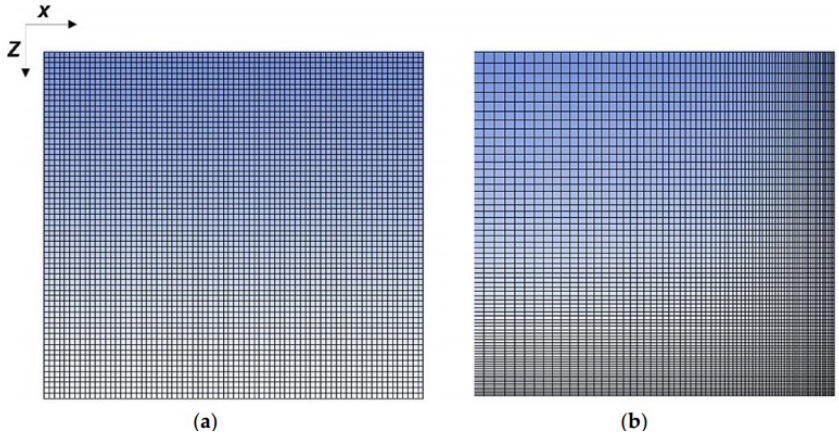


Figure A1-1. (a) Equidistant mesh. (b) Gradual change in the x and z directions.

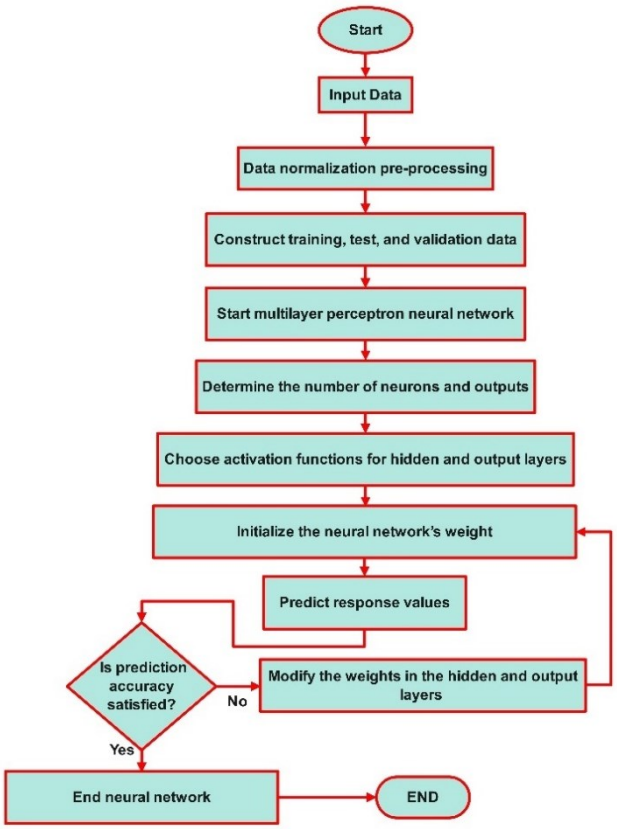
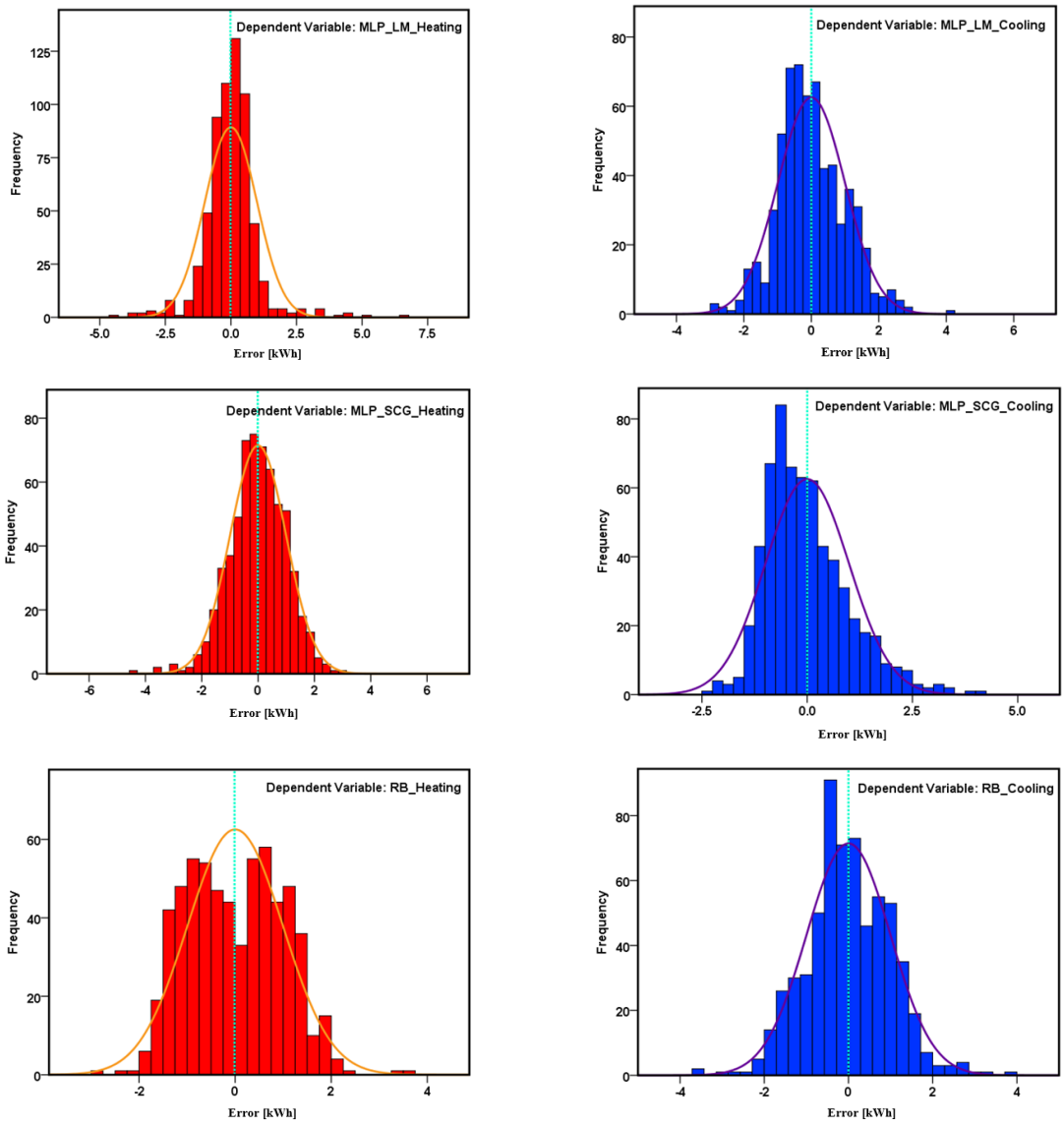


Figure A1-2. The flowchart presents the process used to determine the prediction.



(a) (b)

Figure A1-3. Histogram error for (a) heating load and (b) cooling load prediction through MLP_LM, MLP_SCG, and RB.

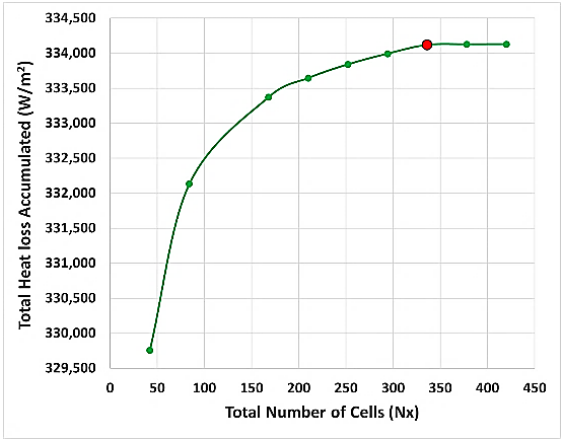


Figure A1-4. Illustrates the accumulated heat loss for December as a function of the total number of cells along the x-axis, in the case of four layers at the south wall.

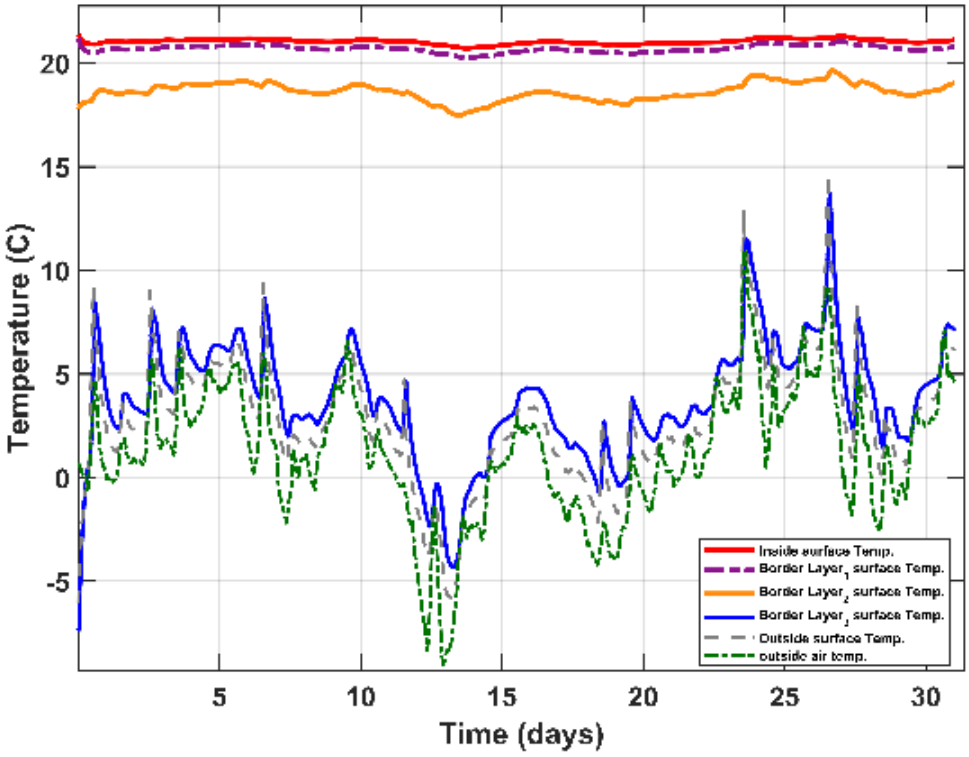


Figure A1-5. The temperature distribution in °C as a function of time in days for the four-layer wall facing East.

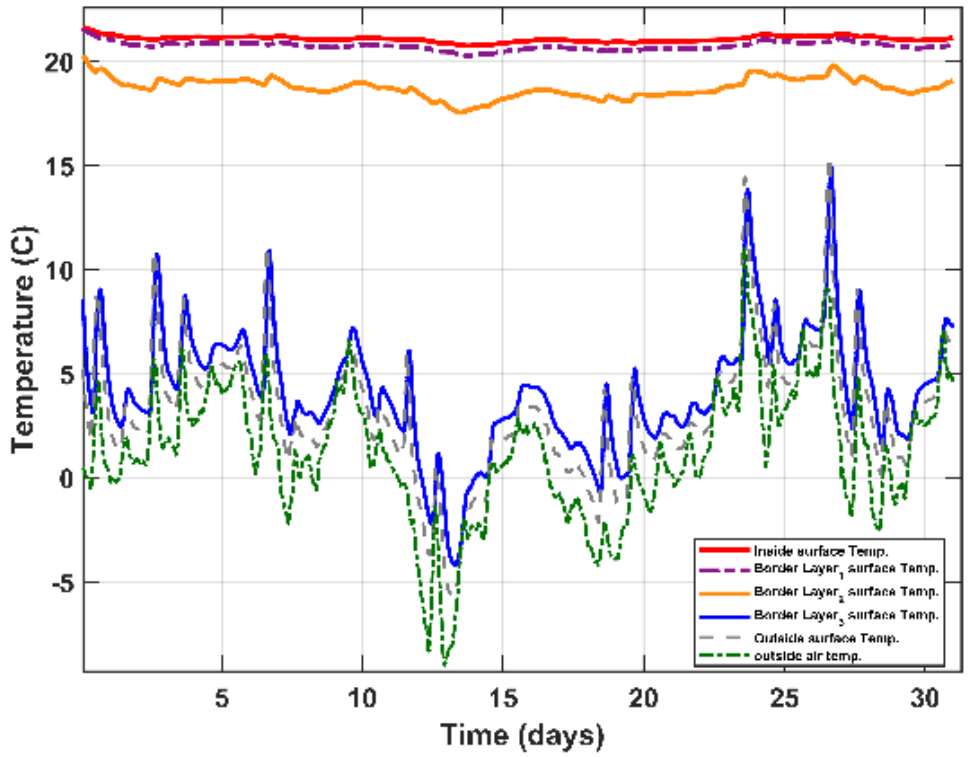


Figure A1-6. The temperature distribution in °C as a function of time in days for the four-layer wall facing South.

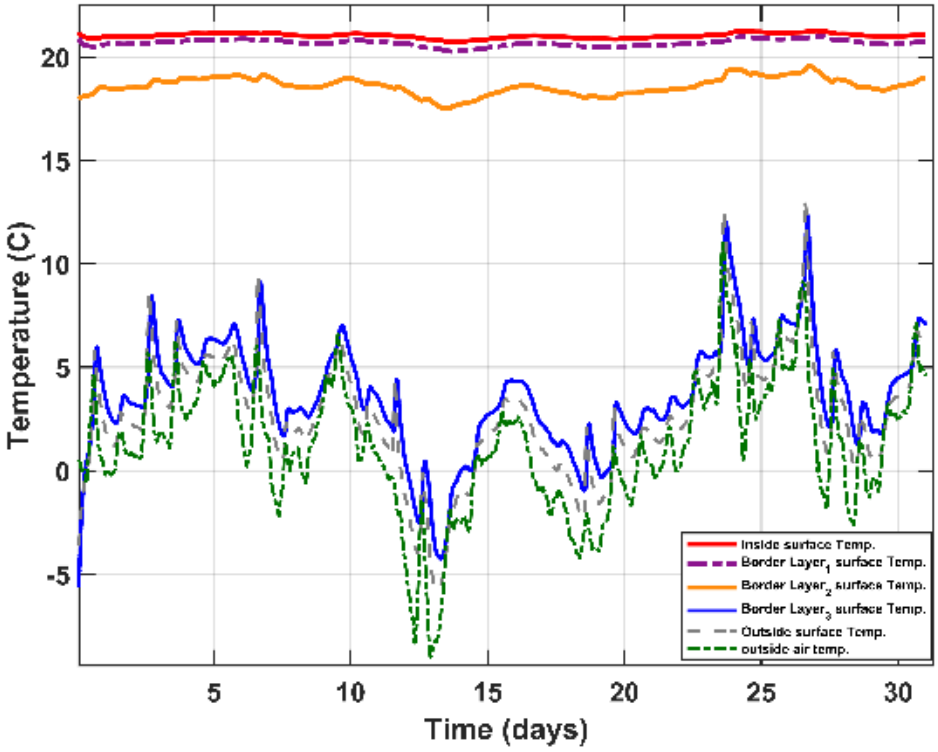


Figure A1-7. The temperature distribution in °C as a function of time in days for the four-layer wall facing West.

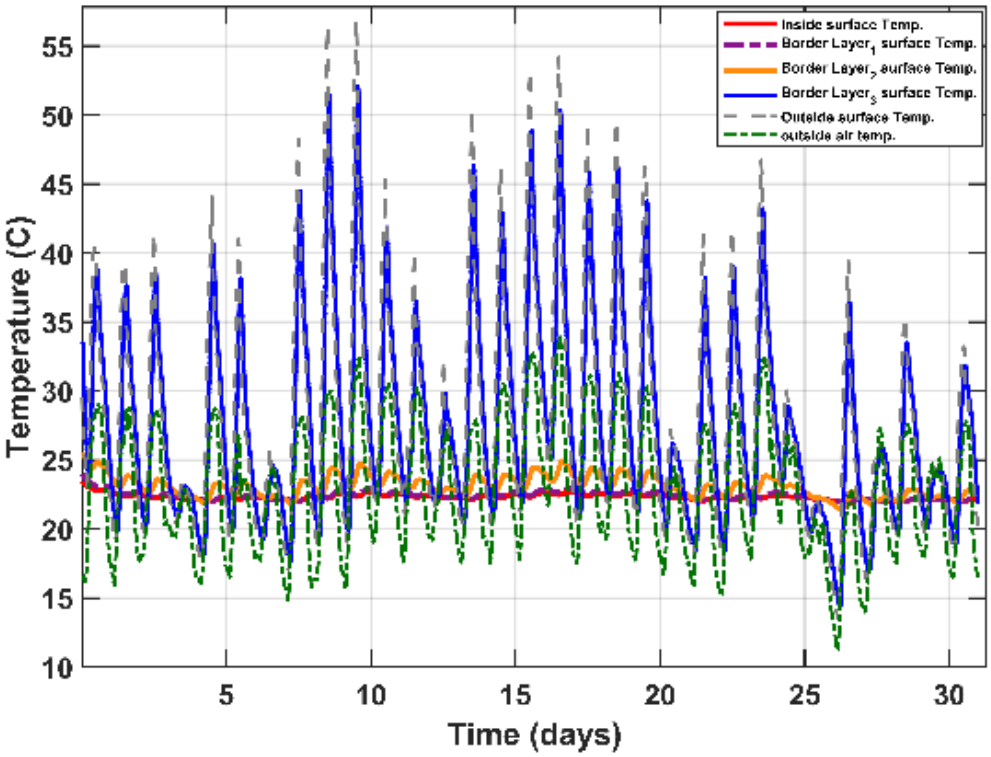


Figure A1-8. The temperature distribution in °C as a function of time in days for the four-layer wall facing East.

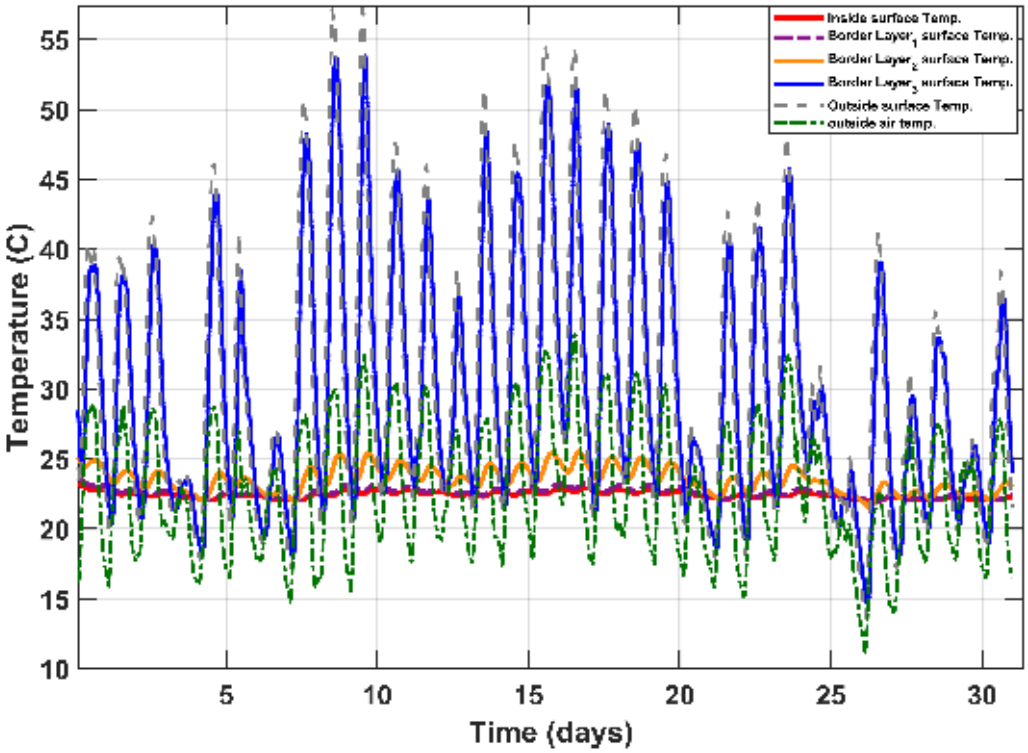


Figure A1-9. The temperature distribution in °C as a function of time in days for the four-layer wall facing South.

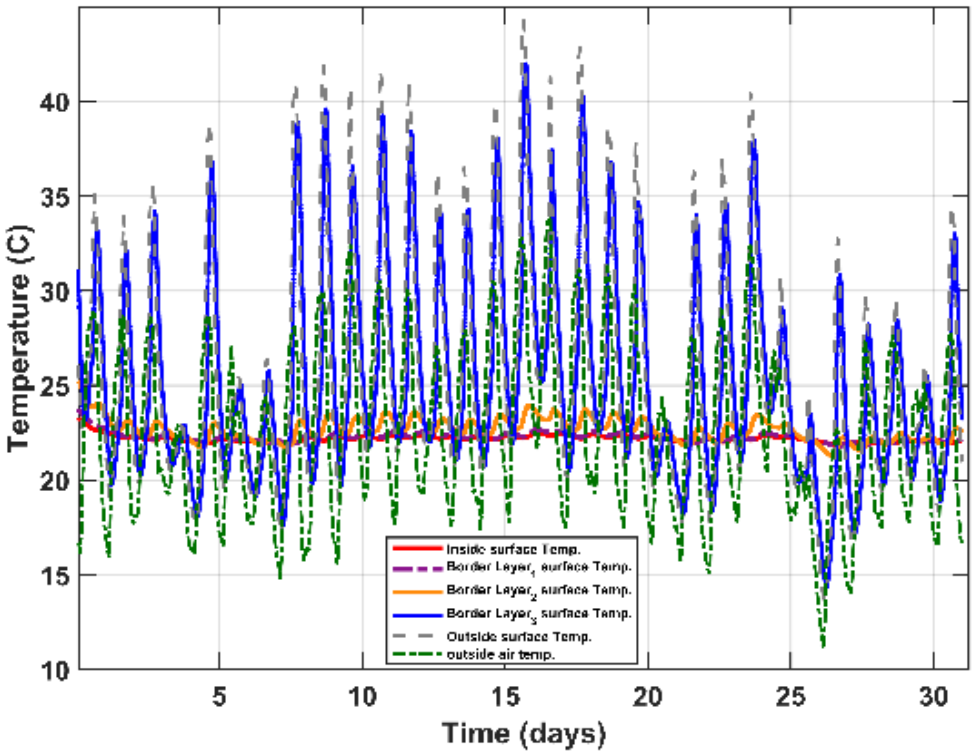


Figure A1-10. The temperature distribution in °C as a function of time in days for the four-layer wall facing West.

A2

Table A2-1. The statistical analysis of energy-efficient design on eight features of efficient building.

Characteristics	Mean	Median	Min.	Max.	Std. Deviation	Standard Error	Variance	Skewness
Direction	2.5	2.5	1	4	1.119	0.045	1.252	0.0
Dist. of AW	7.02	7	0.0	14	3.981	0.159	15.852	0.065
Exposed Area	633.73	648	500	716	53.818	2.154	2896.377	−1.021
Roof Area	193.08	200	140	200	11.260	0.451	126.793	−1.926
RC	0.678	0.658	0.5961	0.8537	0.0639	0.00256	0.004	1.316
Wall Area	440.65	453	300	516	53.645	2.148	2877.819	−1.044
Window Area	30.77	30	0	60	17.972	0.719	323.003	0.324
Cooling Load	24.4	23.5	14.5	42.7	5.7738	0.231	33.33	0.717
Heating Load	22	22	16.6	26.5	1.9871	0.0795	3.949	−0.058