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

Issa Omle
Engineering of Mechanical and Electrical (BSc),
Engineering of Mechanical and Manufacturing (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
2024

CONTENTS

CONTENTS .... **I**

SUPERVISOR'S RECOMMENDATIONS .... **III**

LIST OF SYMBOLS AND ABBREVIATIONS .... **IV**

1. INTRODUCTION **5**

1.1. The Importance and the Application of Conduction, Convection, and Radiation Heat Transfer *5*

1.2. Energy Efficiency of Buildings *6*

1.3. The Governing Differential Equation of Heat Conduction, Convection and radiation *8*

1.4. Numerical Methods for Solving the Heat Conduction Equation *11*

1.5. Outline and the Aims of the Thesis *13*

2. EXPLICIT DISCRETIZATION AND EXPLICIT METHODS **14**

2.1. The Equations and Its Discretization *14*

2.2. Some Explicit Methods *17*

2.2.1. The constant neighbor (CNe) method *18*

2.2.2. The UPFD method *18*

2.2.3. The original odd-even hopscotch (OOEH) *19*

2.2.4. The reversed odd-even hopscotch method (ROEH) *19*

2.2.5. The Linear-Neighbour Method LNe *20*

2.2.6. The CpC algorithm *20*

2.2.7. Heun's method *21*

2.2.8. The pseudo-implicit (PI) method *21*

2.2.9. The Dufort–Frankel (DF) algorithm *21*

2.2.10. Rational Runge-Kutta methods *22*

2.2.11. The shifted-hopscotch (SH) *22*

2.2.12. The Leapfrog–Hopscotch method *23*

3. STABLE, EXPLICIT, LEAPFROG-HOPSCOTCH ALGORITHMS FOR THE HEAT EQUATION **24**

3.1. The Results of Leapfrog-Hopscotch Tests *25*

3.1.1. Preliminary Tests *26*

3.1.2. Comparison with Other Methods for a Large, Moderately Stiff System *27*

3.1.3. Comparison with Other Methods for a Large, Very Stiff System *30*

4. A COMPARATIVE STUDY OF EXPLICIT AND STABLE TIME INTEGRATION SCHEMES FOR HEAT CONDUCTION IN AN INSULATED WALL **33**

4.1. The Geometry and Materials *33*

4.2. The Initial and the Boundary Conditions *35*

4.3. Results 36

4.3.1. Verification using the analytical solution *36*

4.4. Brick wall with Insulation *38*

4.5. Summary of this chapter *40*

5. COMPARISON OF THE PERFORMANCE OF TRADITIONAL AND NEW NUMERICAL METHODS FOR LONG-TERM HEAT TRANSFER SIMULATIONS IN WALLS WITH THERMAL BRIDGES **42**

5.1. *The Studied Cases* 43

- 5.1.1. *The Leapfrog–Hopscotch method* 43
- 5.1.2. *The Used Mathematical Equations* 44
- 5.1.3. *Other Stable and Explicit Methods* 45
- 5.1.4. *Professional Solvers Tools Used for Comparison Purposes* 47

5.2. *Numerical Simulation* 48

- 5.2.1. *Geometry and Material Properties* 48
- 5.2.2. *Mesh Construction* 49
- 5.2.3. *Initial and Boundary Conditions* 51
- 5.2.4. *Verification by Analytical Solutions* 52

5.3. *Results: Comparison of Performances by Measuring the Running Times* 54

- 5.3.1. *Comparison with MATLAB Methods and ANSYS Solvers for the Coarse Mesh System* 54
- 5.3.2. *Comparison with MATLAB Methods and ANSYS Solvers for the Moderate Mesh System* ..56
- 5.3.3. *Comparing ANSYS Solvers and MATLAB Methods for the Fine Mesh System* 57

5.4. *Long-Term Simulations* 59

- 5.4.1. *Geometry and Mesh* 59
- 5.4.2. *Initial and Boundary Conditions* 61
- 5.4.3. *Result of the Wall Simulation* 62
- 5.4.4. *Simulation of the Coldest Day in the Month* 67

5.5. *Long-Term Simulation for Different Shapes of Thermal Bridges in Walls* 68

- 5.5.1. *The Geometry* 68
- 5.5.2. *The Material Properties* 69
- 5.5.3. *Numerical Results* 69

5.6. *Summary of this chapter* 73

6. APPLYING RECENT EFFICIENT NUMERICAL METHODS FOR LONG-TERM SIMULATIONS OF HEAT TRANSFER IN WALLS TO OPTIMIZE THERMAL INSULATION74

6.1. *Numerical simulation of the building model* 75

- 6.1.1. *Building properties* 75
- 6.1.2. *Geometry* 76
- 6.1.3. *Material properties* 77
- 6.1.4. *Mesh construction* 77
- 6.1.5. *Initial and boundary conditions* 78

6.2. *Verification using the steady-state case* 80

6.3. *Results of the three external walls simulation* 82

- 6.3.1. *Comparison of the effect of changing the insulator thickness* 82
- 6.3.2. *Comparison of the effect of changing the material of the insulator* 84
- 6.3.3. *Calculation of heat loss through the walls* 85
- 6.3.4. *Calculation of the optimum thicknesses* 88
- 6.3.5. *Calculation of Heat Loss through walls in case of climate change* 91

6.4. *Summary of this chapter* 92

7. THESES – NEW SCIENTIFIC RESULTS 93

REFERENCES 96

LIST OF PUBLICATIONS RELATED TO THE TOPIC OF THE RESEARCH FIELD 101

SUPERVISOR’S RECOMMENDATIONS

Date

Supervisor

LIST OF SYMBOLS AND ABBREVIATIONS

NOMENCLATURE

SYMBOLS			GREEK LETTERS	
LH	*Leapfrog–Hopscotch*		ρ	*Mass density* [kg/m^3]
DF	*Dufort and Frankel*		Δ	*Difference*
S1	*Shifted-hopscotch method (SH)*		α	*Thermal diffusivity* [m^2/s]
S4	*SH-CNe*		Δt	*Time step size* [s]
c	*Specific heat* [J/(kgK)]		σ^*	*Realistic values of the non-black body* [W/(m^2. K^4)]
C	*Heat capacity* [J/K]		σ	*Coefficient of the radiation term* [s^{-1}K^{-3}]
u	*Temperature* [K]		ε	*Emisivity*
L	*Thickness* [m]		δ	*Factor to indicate daily mean temperature*
K	*Convection coefficient* [1/s]		*M*	*The tridiagonal matrix*
k	*Thermal conductivity* [W/(m. K)]		*r*	*Mesh ratio*
h_c	*Heat transfer coefficient* [W/(m^{2}K)]		τ_i	*Characteristic time or time constant*
R	*Thermal resistance* [K/W]		***Subscripts***	
Q	*Heat transfer rate [W]*		*a*	*Ambient air*
q^*	*heat generation* [W/m^2]		*l*	*Left side*
q	*Heat source rate* [K/s]		*r*	*Right side*
U	*Overall heat transfer coefficient* [W/(m^{2}K)]		*b, p,g, ins*	*Brick, Panel, Gypsum and Insulation*
t	*Time* [s]		*c*	*Convection*
S	*Area*			
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. The Importance and the Application of Conduction, Convection, and Radiation Heat Transfer

Heat transfer is frequently used in engineering systems and other facets of daily life. Engineers must be able to compute the amount of energy carried per unit time using the rate equations and understand the physical concepts underlying the various heat transfer modes. Heat transfer analysis is the main basis for the design of heat transfer equipment, including heat exchangers, boilers, condensers, radiators, heaters, furnaces, freezers, and solar collectors [1]. So, to enhance the efficiency of that equipments and understanding the heat transfer processes involved in their functioning can be of interest to a mechanical engineer [2]. Thermal energy analysis is a difficult task that often requires the use of intricate mathematical models and theoretical equations that can forecast a system's behaviour. Given their interconnections with other components of the system under consideration, forced or natural convection, phase change events, and radiation heat transfer are difficult to simplify. To better comprehend all of these processes, the majority of engineers working in the thermal sector rely on test benches and computer models [3].

Nuclear power plants need precise heat transfer data since safe operation is a key component of their design. Heat-exchanging components are also a part of air conditioning and refrigeration systems, and their design must be carefully considered. Electrical engineers are worried about reducing material damage to electric motors, generators, and transformers caused by hot spots formed by poor heat transfer design. Effective methods of removing heat from chips and semi-conductor devices so that they can function at safe temperatures are important to electronic engineers. The cooling requirements of circuit boards are of importance to a computer hardware engineer because of how quickly computing devices are getting smaller. A metallurgical engineer would be interested in learning the rate of heat transfer required for a certain treatment procedure. For instance, the quality of the finished product is greatly influenced by the pace of cooling throughout the casting process. Researchers in the field of chemical engineering are interested in how heat and mass are transferred during various chemical processes. For an agricultural engineer, food processing, grain drying, and preservation are all interesting. Aeronautical engineers are interested in the rate of heat transmission in the heat shields used in rocket nozzles and re-entry vehicles. The civil engineer is aware of the impact of heat transfer on buildings and the development of thermal stresses in structures. An environmental engineer is concerned with how heat affects the way pollutants disperse in the atmosphere, how they travel through soils, lakes, and seas, and how they affect living things. The mechanisms of heat and mass transfer that take place within the human body, such as hypo- and hyperthermia, are often important to bioengineers [4], [5]. These are only a few instances of how heat and mass transfer calculations are used. The solar system and the associated energy transfer from the sun are the primary causes of life on Earth. It is true that in any activity occurring on Earth, it is extremely difficult, if not impossible, to totally avoid heat transmission. By examining the mechanics of heat and mass transfer, several important issues that develop in a variety of engineering equipment may be solved quickly and economically. Developing heat pipes is one example that comes to mind. Even under almost isothermal conditions, the rate at which these pipes may transfer heat is far higher than that of copper or silver rods of the same diameter. Energy-efficient home design is based on minimising heat gain in the summer and heat loss in the winter. Modern gas turbine blades may be developed where the gas temperature is higher than the melting point of the blade material by creating effective cooling systems. I present an additional example of a successful design of heat transfer. Once again, the design of computer chips is a success story in heat transfer, as they undergo heat flux similar to that of re-entry vehicles. This is especially true when the chips' surface temperature is limited to less than 100 °C. There are a lot of effective heat-transfer designs, further study on heat and mass transfer is needed to extend the life and improve the effectiveness of the devices I have previously discussed and maybe serve as an inspiration for the creation of new technologies [1], [5].

1.2. Energy Efficiency of Buildings

A comfortable environment is necessary not only for health, but also for the productivity of the building's inhabitants [6], [7]. Significant progress toward a sustainable economy may be made via improved building energy efficiency since the buildings account for 40% of the primary energy use and 24% of the generation of greenhouse gases [8]. To increase building energy efficiency, many nations employ a variety of policy tools, but far too often these tools are not coordinated and do not work in concert [9]. When compared to other industries, the construction industry has the ability to significantly reduce energy usage and, by extension, greenhouse gas emissions [10]. Heat transfer calculations are widely used in buildings, including determining how much energy is lost or gained through the building’s envelope (heat conduction), and carrying out environmental analysis inside the building [11], [12]. Density, thermal conductivity, and specific heat are the key examples of material qualities that may change greatly in a building’s system. Weather conditions also change over time. These imply that most heat conduction issues are multi-dimensional and transient [13]. As a result, we need to use numerical computer simulation.

EN ISO 13789 [14] is a commonly used standard for calculating heat transfer through a building envelope; which in turn mentions two standards for thermal bridges; EN ISO 10211 [15] provides a structure for thorough calculations of thermal bridges in building construction and EN ISO 14683 [16] provides a simplified approach with default linear thermal transmittance values.

Because thermal bridges have a big effect on heat loss and reduce building energy efficiency, durability, and air quality, requiring integrated thermal and structural design [17], and the thermal bridge models that were investigated should be especially interesting to architects, civil engineers, and people who work in the insulation materials industry. The examination of the effects of thermal bridges is meant to illuminate the feasibility of implementing energy renovation techniques in existing structures [18]. Validation tests on over a thousand real-world examples corroborated the validity of those correlations, allowing for a straightforward and practical analysis of thermal bridges in existing structures and providing practitioners with a resource that accounts for many scenarios [19]. Investments in energy efficiency are more profitable when energy prices (including energy taxes) are higher. The consumer response to higher energy prices in the buildings sector, however, has been shown by studies in several countries to be relatively inelastic in the short term, and less than a 1% increase in household energy prices would result in a decrease in energy demand of between 0.1% and 0.4% [9]. The heat rate moves through walls can be changed by changing the thickness of it or by making the layer of insulation thicker [20]. Transitory wall conduction heat transfer reacts to weather conditions such as temperature swings, sunshine, air movement, etc. Increasing the building envelope's thermal insulation and decreasing the heat loss via walls is one of the most efficient strategies to increase a structure's energy efficiency and decrease its energy consumption [21].

Insulating buildings properly reduces energy consumption, fossil fuel burning, and related pollution, and building insulation lowers space heating costs but increases initial expenditure [22]. The use of insulating materials has greatly expanded during the past few decades, in part because of ever-increasing insulation thickness requirements set by national legislation [23]. An optimum insulation layer delivers the minimum overall cost, considering both insulation costs and energy use over the life of the building [24]. Norm EN 832, which is quite like ISO 9164, has most recently been approved by the European Parliament for use in calculating the heating load, and they are expanding it to include the cooling load in buildings [25].

Although the cost of insulation is almost proportional to the thickness, the rate of energy savings (savings with each additional increase in insulation thickness) decreases as the thickness increases [25]. Therefore, for any application, there is an optimal value of insulating thickness [26]. The ideal thickness will consequently be immediately impacted by the energy cost, as savings are directly tied to reduced heat loss or lower energy costs [27]. The optimum exterior wall insulation thickness, energy cost savings over a ten-year lifetime, and payback times for four distinct wall types are also computed in the city of Elazığ. Depending on the kind of wall, it was found that when the optimum insulation thickness was employed, fuel consumption and emissions of CO_2, SO_2, nitrogen oxides (NO_x), and carbon monoxide (CO) were reduced [28]. On the other hand, a 30-year life-cycle cost study of a structure in Tunisia revealed that the south orientation of the external wall was the most economical option, with a payback period of 3.29 years, an ideal insulating thickness of 10.1 cm, and energy savings of 71.33% [29]. It was found that wall orientation not only had a little impact on the optimum insulation thickness, but also had a larger impact on energy savings, which peaked at a maximum of 23.78 TND/m^2 in the case of an east-facing wall. Additionally, it proved that optimal insulation and energy savings were significantly influenced by economic factors, inflation and discount rates, and building lifetime [29].

Calculations of the optimum insulation thickness and payback period for the different types of walls (brick, light-weight concrete, and stone) in India revealed that, depending on the insulation material, external wall material, and local climate, the optimum insulation thickness ranged between 0.154 m and 0.1703 m and the payback period between 1.17 and 1.53 years [30]. In other studies, [31], [32], the optimal thickness of insulating materials varies from 0.02 to 0.17 m, based on the environment and type. The price of an insulating material depends on its production cost (supply) and how well it regulates temperature (demand). As a result, insulation systems are created with multiple layers of various materials to achieve the best possible economic performance [33], [34]. Expanded polystyrene insulation (EPS) sandwiched concrete panels are cheap, light, load bearing, and insulating building materials that speed up home construction and save money [35].

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

For the Cartesian coordinate system, the differential control volume is defined in Figure 1.1, which is a small rectangular element of width *Δy*, length *Δx*, and height *Δz*. Let us assume that the body density is *ρ* and its specific heat is *c*. The equation for conduction, convection, and radiation is derived using energy balance. During a small time interval *Δt*, the energy balance on this element is expressed as

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

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

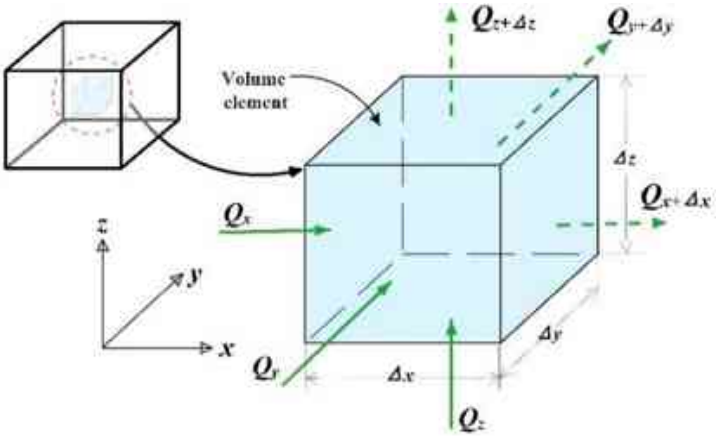

Figure 1.1. Differential control volume, *Δx Δy Δz*, for the heat conduction derivation in Cartesian coordinates.

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

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

$$\Delta m = \rho \Delta x \Delta y \Delta z, \quad \text{and} \quad \Delta V = \Delta x \Delta y \Delta z, \tag{1.2}$$

where ρ is the mass density $\left({kg}/{m^3} \right)$ of the control volume.

The Newton heat convection law:

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

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

Since the ambient temperature u_a does not directly depend on u, the term hSu_a is included in the term of heat generation. The Stefan–Boltzmann law for the outgoing radiation heat:

$$Q_{radiation} = \sigma^* S u^4 \, , \tag{1.4}$$

where $\sigma^* = \varepsilon \cdot \sigma$, i.e., Since that the surface is not a black body, so the universal constant of Stefan-Boltzmann $\sigma = 5.67 \times 10^{-8} \mathrm{W/m^2K^4}$ is multiplied by the suitable emissivity constant ε. The heat source term q contains the incoming heat radiation, which includes direct sunlight, in the same way as the hSu_a term does.

The change of internal energy rate inside the control volume is $\rho c \Delta x \Delta y \Delta z (u_{t+\Delta t} - u_t)$, The expression for the change in the energy content of the element can be written mathematically as follows:

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

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

$$\begin{aligned} & -\frac{1}{\Delta y \Delta z}\frac{Q_{x+\Delta x} - Q_x}{\Delta x} - \frac{1}{\Delta x \Delta z}\frac{Q_{y+\Delta y} - Q_y}{\Delta y} - \frac{1}{\Delta x \Delta y}\frac{Q_{z+\Delta z} - Q_z}{\Delta z} + \frac{Q_{gen}}{\Delta x \Delta y \Delta z} \\ & - \frac{hSu}{\Delta x \Delta y \Delta z} - \frac{\sigma^* S u^4}{\Delta x \Delta y \Delta z} = \rho c \frac{(u_{t+\Delta t} - u_t)}{\Delta t} \end{aligned} \ . \tag{1.6}$$

Taking note of the fact that 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 $\Delta t \ \to 0$.

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

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

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

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

Substituting into Eq. (1.6), I get is 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

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

where the term $\alpha=\frac{k}{\rho c}$ is the material thermal diffusivity, the incoming radiative heat for an unit area is introduced as q^{*}. The direction of heat transfer, both conductive and radiative, is always will be horizontal, thus $S=\Delta y\Delta z$ and $Q_{gen}=\left(q^{*}+hu_a\right)S$.

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

The following is an expression for the temperature equation, which includes the radiation, conduction, convection, and the source of heat generation, where the vector notation is used [36], [37] to express the previous 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 heat transfer convection, 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 $[\mathrm{K}/\mathrm{s}]$.

The fundamental statement of the heat equation is that heat fluxes occur when there are temperature differences, which lead to temperature equalization. If the material properties are non-homogeneous, therefore Eq. (1.10) may be expressed in the general form in more general conditions:

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

In several scientific domains, including biology, chemistry, and physics, mass movement may be modelled using the generalizations of the diffusion equation, such as the advection-diffusion-reaction equation. For example, the charge carriers in semiconductors [38], the atoms in carbon nanotubes [39], and the proteins in developing embryos [40]. Furthermore, a system of equations or very similar equations have been used to simulate the fluid flow through porous media, including moisture, ground water, and crude oil in subterranean reservoirs [41].

1.4. Numerical Methods for Solving the Heat Conduction Equation

Analytical solutions, even new ones [42], exist for spatially homogeneous systems and most numerical methods are developed and tested by mathematicians also for these simple systems. Moreover, some analytical solutions are available for one-dimensional multilayer problems as well, for both steady-state and transient conditions. These solutions are frequently used for heat gain and loss calculations for exterior envelopes and heat storage in interior structures [11]. However, most of the building heat conduction problems are multi-dimensional and transient, while the material properties such as the density and heat conductivity can widely vary in the system [43] (p. 15). Thus, numerical computer simulation cannot be avoided.

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 [42].

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

There are lots of numerical methods to solve the heat conduction equation, such as several finite difference schemes (FDM) [44], [45], finite element methods (FEM) [46], or a combination of these [47]. However, they can be computationally demanding since they require the full spatial discretization of the examined system which converts the Partial Differential Equation (PDEs) into a system of Ordinary Differential Equation (ODEs). After that one can solve the system of ODEs at each time level [48]. If the eigenvalues of the problem have a range of several orders of magnitude due to differences in material properties, then the problem is rather stiff and the so-called Courant–Friedrichs–Lewy (CFL) limit can be very small. When the tolerance is not enough small, instability can occur even in professional commercial adaptive time-step size solvers like MATLAB's ode23 and ode45 [49]. This means that almost all explicit finite difference methods are unstable when the time step size is larger than this small threshold. On the other hand, implicit methods work with whole system matrix, thus they can be extremely slow with huge memory usage when the number of cells is large. Still, these methods are typically used for solving these kinds of equations, see for example [50], [51], [52], [53].

The main issue with the implicit methods is that they cannot be easily parallelized because it requires the solution of an algebraic equation system at each time step. When dealing with a one-dimensional system with a tridiagonal matrix and few nodes, numerical computation can be completed quickly, and it is hard to beat the implicit methods. However, when the implicit methods are employed to handle more complex systems, such reservoir simulation with one trillion cells, the numerical calculations can be time-consuming. On the other hand, parallelism has become more popular in recent years [54], [55], which is advantageous for explicit methods. One can observe that the trend toward increasing parallelism in high-performance computing is reinforced, since unfortunately the CPU clock frequencies nowadays increase much slower than a few decades ago [54], [55]. That is one of the reasons why I believe that the importance of easily parallelizable explicit and unconditionally stable methods is going to increase, even if currently not too many scholars work with them (see [56], [57], [58], [59]).

The second problem with most explicit or implicit methods is that they can produce qualitatively unacceptable solutions, such as unphysical oscillations or negative values for variables that would otherwise be non-negative. The concentrations, densities, or temperatures expressed in Kelvin might be these variables, and the numerical methods should maintain their positives. We explained in our previous investigations [60], [61], [62], [63], that the widely used conventional solvers, either explicit or implicit, have serious difficulties. This information highlights the fact that finding effective numerical methods is still important.

One of the very few easily parallelizable explicit and unconditionally stable methods is the two-stage odd-even hopscotch (OEH) algorithm [60], [64], [65], [66]. In the previous works [60], [61], we showed that this method is robust and powerful for spatially homogeneous grids but, in the case of stiff systems, it can be disastrously inaccurate for large time step sizes. I constructed and tested new hopscotch combinations and found [60], [61], [67] that some of them behave much better, not only for large, but also for medium and small-time step sizes. In this work, I extend my research by further modifying the underlying space and time structure. During the last few years, our research group developed several explicit unconditionally stable methods to calculate heat conduction in arbitrary space dimensions [49], [61], [68], [69]. Unconditional stability here means that the temperature remains finite (i.e., errors are not amplified without bounds) for arbitrary time step size. These new methods either belong to the family of FDMs or are like them. In those of our original papers, we tested the algorithms under general circumstances using discontinuous random parameters and initial conditions and using analytical as well as numerical reference solutions, and I have shown that they can provide quite accurate results, and they are much faster than the professionally optimized MATLAB ‘ode’ routines.

In my research, I worked with my supervisor on investigating and improving families of novel and conventional explicit methods for solving linear and nonlinear heat conduction equations, depending on a new way of thinking. Adapt the most successful methods (especially the Leapfrog and the original hopscotch methods) to cases where there is heat transfer by convection and (Stefan-Boltzmann-type, thus nonlinear) radiation, especially the problems of real-life heat transfer in buildings. In addition, I compare my results and running times with those of the appropriate software, e.g., ANSYS.

1.5. Outline and the Aims of the Thesis

My goal is to elaborate and optimize numerical methods, and then, using these methods, to efficiently investigate heat transfer problems in building walls. These will be the tools by which I can perform optimizations of building envelopes not only from a thermodynamic but also from an economic point of view. In Chapter 2, the fully explicit discretization of the spatial variables is illustrated. The explicit methods, Leapfrog-Hopscotch (LH), Dufort and Frankel (DF) methods, and other methods are also discussed in this chapter. In Chapter 3, new numerical methods to solve linear and nonlinear heat conduction equations, depending on a new way of thinking, are introduced. I increased the efficiency of some methods by combining the hopscotch with the leapfrog technique and performing numerical experiments to investigate the performance of those methods, choosing the best combinations, and testing these algorithms on both small and large systems and for stiff and non-stiff systems. In Chapter 4 the methods are tested on real-life applications to examine how the performance of the individual methods changes and which of them is the best choice under different circumstances. In Chapter 5, I adapt the most successful methods (especially the LH) to cases where there is heat transfer by convection and radiation, especially the problems of real-life heat transfer in walls of buildings with different shapes of thermal bridges. In addition, I compare my results and running times with those of the appropriate software, e.g., ANSYS. In Chapter 6, I reached my goal: the recent efficient numerical methods are applied for long-term simulations of heat transfer in walls to optimize thermal insulation.

2. EXPLICIT DISCRETIZATION AND EXPLICIT METHODS

In this chapter I will provide and illustrate a completely explicit discretization for the space variables in the heat conduction equation which converts the PDE into a system of ODEs. I will later apply the methods we construct to the resulting ODE system. Additionally, I will briefly describe a few novel, traditional and innovative methods in this chapter, which serve as the foundation for all the schemes and applications I will build in the subsequent chapters.

2.1. The Equations and Its Discretization

The phenomenon of the simplest Fourier-type heat conduction within a homogeneous medium with a heat source is described using a linear parabolic PDE as follows:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + q \,. \tag{2.1}$$

The law of Newton's cooling states that the term $K(u_a - u)$ indicates to the convective heat transfer that occurs between a moving fluid and a surface [70]. This process happens as a result of the fluid's motion, which can be either a gas or a liquid and is driven by variations in density and temperature. The rate of heat transfer is influenced by the fluid's physical characteristics, including thermal conductivity and viscosity, the temperature differential between its surface and interior, and the fluid's velocity [71]. This calls for the term Ku_a to be included in the equation q. The Stefan-Boltzmann equation [72] states that the radiative heat loss from a surface may be expressed using the term $-\sigma u^4$, when the surface area and the Stefan-Boltzmann constant, both of which are positive, yield the proportionality constant. Like the Ku_a term stated before, the incoming radiation, which includes direct sunlight, may be included into the source term q. The terms for convection, radiation, and the heat source are added to the heat conduction Eq. (2.1), we obtain 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, I apply to the $\alpha \nabla^2 u$ term the most common central difference equation

$$\frac{\partial^2}{\partial x^2} u(x_i) \approx \frac{\dfrac{u(x_{i+1}) - u(x_i)}{\Delta x} + \dfrac{u(x_{i-1}) - u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \ , \tag{2.2}$$

which is second order in Δx, where $i = 1,...,N$ and N is the overall number of nodes. By doing this, we can derive the spatially discretized version of the heat transfer Eq. (1.10) in one space dimension as follows:

$$\frac{du_i}{dt} = \alpha \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + q - Ku_i - \sigma u_i{}^4 \,. \tag{2.3}$$

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

$$\frac{1}{c(x)\rho(x)}\frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}\right) . \tag{2.4}$$

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

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

Section 5 of the book [73] presents more details about this procedure for the case of underground reservoirs. The nodes are surrounded by cells, and $k_{i,i+1}$ is the heat conductivity between node *i* and its right neighbour. The discretized equation attains the following form:

$$\frac{du_i}{dt}=\frac{1}{c_i\rho_i\Delta x}\left(k_{i,i+1}\frac{u_{i+1}-u_i}{\Delta x}+k_{i-1,i}\frac{u_{i-1}-u_i}{\Delta x}\right)+q-Ku_i-\sigma u_i^4 . \tag{2.5}$$

The dimensions of a cell, measured along its length and across its (typical) cross-section, are represented as Δx and S . Where u_i is the temperature of the cell i , $C_i = c_i\rho_i V$ is the heat capacity of that cell in $\left[\mathrm{J/K}\right]$ units, and $V = S\Delta x$ is the volume of the cell. I introduce two other quantities, the heat source term q ,

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

The thermal resistance between the two neighbouring nodes can be determined as $R_{i,i+1}\approx\Delta x/\left(k_{i,i+1}S\right)$ in $\left(K/W\right)$ units. The distances between the cells center 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, we can write for the resistance between cells *i* and *i*+1 that $Rx_i\approx\left[\Delta x_i/\left(k_iS_i\right)\right]+\left[\Delta x_{i+1}/\left(k_{i+1}S_{i+1}\right)\right]$, and if the cell *j* is below the cell *i*, we have $Rz_i\approx\left[\Delta x_i/\left(k_iS_i\right)\right]+\left[\Delta x_j/\left(k_jS_j\right)\right]$ for the vertical resistance.

Based on these new quantities, one can obtain the following expression for the time derivative of each cell variable:

$$\frac{du_i}{dt}=\frac{u_{i-1}-u_i}{R_{i-1,i}C_i}+\frac{u_{i+1}-u_i}{R_{i+1,i}C_i}+q-Ku_i-\sigma u_i^4 \tag{2.6}$$

It is not hard to generalize Eq. (2.6) even more for the case of arbitrary number of neighbours to obtain the following spatially discretized version of Eq. (1.10):

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} + q - Ku_i - \sigma u_i^{\ 4} \tag{2.7}$$

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

Eq. (2.7) for $K = 0$ and $\sigma = 0$ can be written into a matrix form

$$\frac{d\vec{u}}{dt} = M\vec{u} + \vec{Q}, . \tag{2.8}$$

In the one-dimensional case of Eq. (2.3), the matrix *M* is tridiagonal with the following elements:

$$m_{\mathrm{i,i}} = -\frac{2\alpha}{\Delta x^2} \ (1 < i < N), \ \ m_{\mathrm{i,i-1}} = \frac{\alpha}{\Delta x^2} \ (1 < i \le N), \ \ m_{\mathrm{i,i+1}} = \frac{\alpha}{\Delta x^2} \ (1 \le i < N) \tag{2.9}$$

In the general case of Eq. (2.7), the nonzero elements of the matrix can be given as:

$$m_{\mathrm{i,j}} = \frac{1}{R_{\mathrm{i,j}} C_{\mathrm{i}}}, \quad m_{\mathrm{i,i}} = -\sum_{\mathrm{j \neq i}} m_{\mathrm{i,j}} \tag{2.10}$$

The characteristic time or time constant τ_i of cell *i* can be introduced, which , for the simplest one-dimensional case is always a non-negative quantity:

$$\tau_i = \frac{\Delta x_i^2}{2\alpha}, \ (1 < i < N). \tag{2.11}$$

Only when cells *i* and *j* are adjacent may the off-diagonal $m_{\mathrm{i,j}} = 1/\left(R_{\mathrm{i,j}} C_{\mathrm{i}}\right)$ member of the *M* matrix be nonzero. All summations after this point are over the neighbours of the actual cell, which will be denoted by $\mathrm{j} \in n(\mathrm{i})$. Unless otherwise indicated, I take into account closed (zero Neumann) boundary conditions, meaning that the boundary of the domain under study is thermally isolated with respect to heat transfer of the conductive type. I provide the changeable layout for a 2D system of 18 cells in Figure 2.1 to aid the reader's imagination. I emphasize that the arrangement and shape of the cells are not necessarily regular.

I will consider in Section 3 the system as thermally isolated (zero Neumann boundary conditions), which is implemented by the omission of those terms of the sum in Equation (2.7), which have infinite resistivity in the denominator due to the isolated boundary. Let us denote by λ_{MIN} and λ_{MAX} the eigenvalues of the system matrix *M* with the (nonzero) smallest and the largest absolute values, respectively. Now, $\lambda_{\mathrm{MAX}} / \lambda_{\mathrm{MIN}}$ gives the stiffness ratio of the system, and the maximum possible time step size for the FTCS (explicit Euler) scheme is exactly given by $h_{\mathrm{MAX}}^{\mathrm{FTCS}} = |2 / \lambda_{\mathrm{MAX}}|$ above which the solutions are expected to diverge due to instability. This threshold is frequently called the CFL limit and also holds for the second-order explicit Runge–Kutta (RK) method [74].

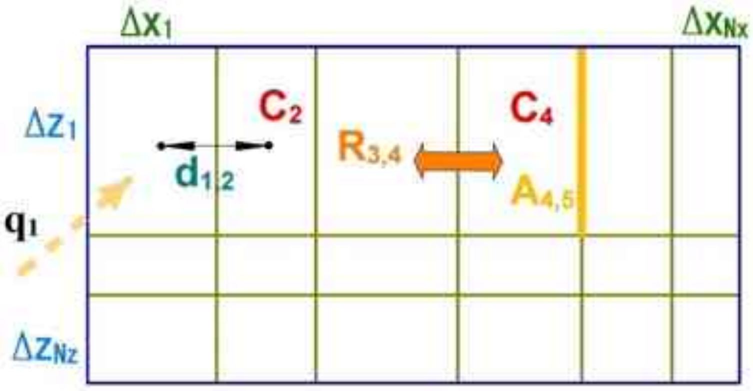

Figure 2.1. The generalized variables visualization for the case when the mesh is not necessarily regular. The orange arrow is for conduction between cells with neighboring through the resistance $R_{3,4}$.

The system of ODEs in matrix form for this system may be expressed as

$$\frac{d}{dt}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\end{pmatrix}=\begin{pmatrix}\frac{-1}{C_1R_{12}}+\frac{-1}{C_1R_{13}} & \frac{1}{C_1R_{12}} & \frac{1}{C_1R_{13}} & 0\\ \frac{1}{C_2R_{12}} & \frac{-1}{C_2R_{12}}+\frac{-1}{C_2R_{23}}+\frac{-1}{C_2R_{24}} & \frac{1}{C_2R_{23}} & \frac{1}{C_2R_{24}}\\ \frac{1}{C_3R_{13}} & \frac{1}{C_3R_{23}} & \frac{-1}{C_3R_{13}}+\frac{-1}{C_3R_{23}}+\frac{-1}{C_3R_{34}} & \frac{1}{C_3R_{34}}\\ 0 & \frac{1}{C_4R_{24}} & \frac{1}{C_4R_{34}} & \frac{-1}{C_4R_{24}}+\frac{-1}{C_4R_{34}}\end{pmatrix}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\end{pmatrix}+\begin{pmatrix}q_1\\q_2\\q_3\\q_4\end{pmatrix}.$$

2.2. *Some Explicit Methods*

In this section, I give essential information about the algorithms which some of which already exist and others that I invented. I first present their formula for the simplest case (one dimensional, equidistant mesh, Eq. (2.1)), then immediately for a general, arbitrary mesh as well. The simplest form is useful for comparison purposes, since numerical schemes are given in this form in most textbooks on numerical methods. The more general forms are necessary because I use only them in this work.

I introduce the usual mesh-ratio $r_i=\frac{\alpha\Delta t}{\Delta x^2}$ for the 1D equidistant mesh, e.g. for Eq. (2.1). On the other hand, for the case of the general mesh, I introduce the following notations.

$$r_i=\Delta t\sum_{j\neq i}\frac{1}{C_iR_{ij}}\ \text{and}\ A_i=\Delta t\sum_{j\neq i}\frac{u_j^n}{C_iR_{ij}} \tag{2.12}$$

The first quantity is the generalization of *r* (defined above), while the second one reflects the state and the effect of the neighbours of cell *i* as well.

2.2.1. The constant neighbor (CNe) method

The constant neighbor (CNe) method [62], [68] for Eq. (2.1) is:

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-2r}\right) \tag{2.13}$$

While for general grids it is:

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

To proceed, let us recall that the following general time discretization

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

implies the so-called theta method:

$$u_i^{\mathrm{n}+1} = u_i^{\mathrm{n}} + r\left[\theta\left(u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) + (1-\theta)\left(u_{i-1}^{\mathrm{n}+1} - 2u_i^{\mathrm{n}+1} + u_{i+1}^{\mathrm{n}+1}\right)\right] \tag{2.14}$$

where $\theta \in [0,1]$. For $\theta = 0, \frac{1}{2}$, and 1 one have the (standard) implicit Euler, the Crank–Nicolson, and the explicit Euler (FTCS) schemes, respectively [75]. If $\theta < 1$, the theta method is implicit. It can be modified to be explicit by taking the neighbors into account at the old time level, where their values are already calculated. Thus, one can insert $u_{i\pm1}^n$ into the theta-scheme (2.14) instead of $u_{i\pm1}^n$ to obtain

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

With this modification, the final formula is completely explicit:

$$u_i^{\mathrm{n}+1} = \frac{(1-2r\theta)u_i^n + r\left(u_{i-1}^{n+1} + u_{i+1}^{n+1}\right)}{1 + 2r(1-\theta)} \;. \tag{2.15}$$

2.2.2. The UPFD method

The UPFD (unconditionally positive finite difference) method is the theta-method (2.15) for $\theta = 0$. In the case of Eq. (2.1), it reads as follows:

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

and the general form for Eq. (2.6) or (2.7) is:

$$u_i^{n+1} = \frac{u_i^n + A_i}{1 + 2r_i} \;. \tag{2.17}$$

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

Over half a century has passed since the discovery of the original odd–even hop-scotch (OOEH) algorithm [66]. Its temporal and spatial organization has been described in [76]. It is designed to be a quick, all-purpose algorithm that produces results with little effort from the user or the computer. This completely explicit two-stage approach has, as far as we know, undergone modification and generalization procedures to increase its accuracy, but always in the direction of implicitness. After the first step by the FTCS formula (which is based on explicit Euler time discretization) for the odd cells, the BTCS formula (which is based on implicit Euler time discretization) is used for the even cells. The labels odd and even are interchanged after each time step. If we would like to apply an odd-even hopscotch method, we need a bipartite grid, where all the nearest neighbours of the odd cells are even and vice versa as is shown in Figure 2.2. I modify this method here to include the convection component, which is always considered at the new time level for enhanced stability. The radiation term is handled first explicitly and then implicitly [77]. These are the equations that are being used:

First stage:

$$u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i - \Delta t\sigma\left(u_i^n\right)^4}{1+\Delta tK} \,. \tag{2.18}$$

Second stage:

$$u_i^{n+1} = \frac{u_i^n + A_i^{\text{new}}}{1+r_i + \Delta tK + \Delta t\sigma\left(u_i^n\right)^3} \,, \tag{2.19}$$

where A_i^{new} is calculated in the same way as A_i in Eq. (2.12), but using the new values of the temperatures, which make the implicit formula effectively explicit.

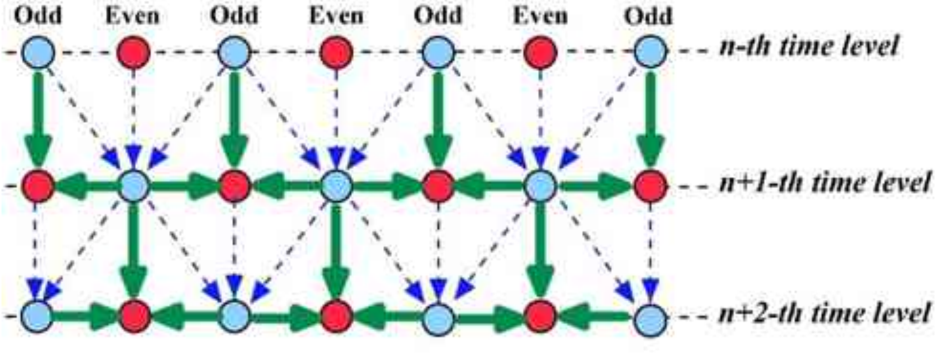

Figure 2.2. The stencil of the original odd-even hopscotch algorithm. Thin blue arrows and thick green arrows indicate operations at the first stage and second stage, respectively.

2.2.4. *The reversed odd-even hopscotch method (ROEH)*

This method applies the formulas of the OOEH method in the opposite order. However, since the new values of the neighbors are not known when first-stage calculations begin, the implicit formula can be applied only with a trick, which is that of the UPFD method. Obtaining the code of this method is easy, since one only needs to change the order of the two formulas in the code of the original OOEH. We showed that this method produces much smaller errors in the case of very stiff systems than the OOEH method [78].

2.2.5. The Linear-Neighbour method LNe

The next method is the 2-stage linear-neighbor (LNe or LNe2) method [79]. It is 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. 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:

$$u_i^{n+1} = u_i^n e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-2r}\right) + s_i \frac{\Delta t^2}{2r}\left(1 - \frac{1 - e^{-2r}}{2r}\right). \tag{2.20}$$

For the general case,

$$A_i^{\mathrm{new}} = \Delta t \sum_{\mathrm{j \neq i}} \frac{u_{\mathrm{j}}^{\mathrm{pred}}}{C_i R_{\mathrm{ij}}} \tag{2.21}$$

with which I can make the corrector step as follows:

$$u_{\mathrm{i}}^{\mathrm{n+1}} = u_{\mathrm{i}}^{\mathrm{n}} e^{-r_{\mathrm{i}}} + \left(A_{\mathrm{i}} - \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}\right)\frac{1 - e^{-r_{\mathrm{i}}}}{r_{\mathrm{i}}} + \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}. \tag{2.22}$$

The values given in Eq. (2.22) can be used to recalculate A_i^{new} again, which makes sense to repeat (2.22) to obtain new results. In this case, we have three stages altogether, thus the method is called the LNe3 method [79]. This algorithm is still second order, but more accurate than LNe2.

2.2.6. The CpC algorithm

The CpC algorithm [80] generally starts with a fractional time step with length $p\Delta t$, but here I take $p = {}^{1}\!/_{2}$, because this version has usually better accuracy than for other values of *p*. So, in the first stage, I calculate new predictor values of the variables with the CNe formula, 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 (2.21) 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

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}}{2}\left(1 - e^{-2r}\right) \quad \text{and} \quad u_i^{n+1} = u_i^n \cdot e^{-r_\text{i}} + \frac{A_i^{\text{new}}}{r_\text{i}}\left(1 - e^{-r_\text{i}}\right) \quad (2.23)$$

2.2.7. *Heun's method*

This method also called explicit trapezoidal rule, may be the most common second-order RK scheme [81]. 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:

$$u_i^{n+1} = u_i^n - r\left(u_i^n + u_i^{\text{pred}}\right) + r\frac{u_{i-1}^n + u_{i-1}^{\text{pred}} + u_{i+1}^n + u_{i+1}^{\text{pred}}}{2} \quad (2.24)$$

and

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

2.2.8. *The pseudo-implicit (PI) method*

In the case of the pseudo-implicit (PI) method, we took Algorithm 5 from [82] in the case of the pure heat equation with parameter setting $\lambda = 1$, which gives the following two-stage algorithm for the special case:

Stage 1: $u_i^{\text{pred}} = \dfrac{u_i^n + {}^{r}\!/_{2}\left(u_{i-1}^n + u_{i+1}^n\right)}{1 + r}$

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

For a general grid, we have

Stage 1: $u_i^{\text{pred}} = \dfrac{u_i^n + A_i}{1 + r_i}$, where $A_i = \dfrac{\Delta t}{2}\sum\limits_{\text{j}\neq\text{i}}\dfrac{u_\text{j}^n}{C_i R_\text{ij}}$

Stage 2: $u_i^{n+1} = \dfrac{(1 - r_i)u_i^n + A_i^{\text{new}}}{1 + r_i}$, where $A_i^{\text{new}} = \Delta t\sum\limits_{\text{j}\neq\text{i}}\dfrac{u_\text{j}^{\text{pred}}}{C_i R_\text{ij}}$.

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

This method can be obtained from the so called leapfrog explicit scheme by a modification [83] (p. 313). It is a known explicit unconditionally stable scheme that has the formula in the special and general case:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1} + 2r\left(u_{i-1}^{n} + u_{i+1}^{n}\right)}{1+2r} \quad \text{and} \quad u_i^{\mathrm{n}+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1+r_i} \qquad (2.25)$$

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).

2.2.10. Rational Runge-Kutta methods

These methods are a family of nonlinear methods, which means that the new u_i^{n+1} values are not the linear combinations of the old u_i^n values. We chose a 2-stage version [84] defined as follows. The first stage is a full step by the explicit Euler (FTCS) to obtain the predicted value:

$$g_i^1 = r\left(u_{i-1}^{\mathrm{n}} - 2u_i^{\mathrm{n}} + u_{i+1}^{\mathrm{n}}\right) \quad \text{and} \quad g_i^1 = -ru_i^n + A_i$$

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

Then the increment of a repeated Euler-step is calculated, using the predictor values obtained above:

$$g_i^2 = r\left(u_{i-1}^{\text{pred}} - 2u_i^{\text{pred}} + u_{i+1}^{\text{pred}}\right) \quad \text{and} \quad g_i^2 = -r_i u_i^{\text{pred}} + A_i^{\text{new}} .$$

If we introduce the scalar products, which are common for the cells, thus they need to be calculated only once in a time step, we can write

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

we obtain the final expression for the new variables:

$$u_i^{n+1} = u_i^n + \frac{2p_1 g_i^1 - 2p_{12} g_i^1 + p_1 g_i^2}{4p_1 - 4p_{12} + p_2} \; . \qquad (2.26)$$

2.2.11. The shifted-hopscotch (SH)

In the shifted-hopscotch (SH) method [61], we have a repeating block consisting of five stages, which corresponds to two half and three full-time steps, which altogether span two time steps for odd as well as even cells, as one can see in Figure 2.3 A. The calculation starts with a half-sized time step for the odd cells which is symbolized by a light blue box with the number 1 in the figure. Then, a full-time step for the even, the odd, and the even cells follows again. Finally, a half-size time step for the odd cells closes the calculation of the values. In our original work [61] we used the θ formula

$$u_i^{\mu+1} = \frac{(1-2r\theta)u_i^{\mu} + r\left(u_{i-1}^{\mu+1/2} + u_{i+1}^{\mu+1/2}\right)}{1+2r(1-\theta)} \tag{2.27}$$

and (in the general case)

$$u_i^{\mu+1} = \frac{(1-r_i\theta)u_i^{\mu} + A_i}{1+r_i(1-\theta)} \quad . \tag{2.28}$$

In this chapter, I use only the combination already proven to be the best (S4 in [61]), which means $\theta = 0$ is used at the first, $\theta = 1$ at the fifth, and $\theta = 1/2$ in all other stages. The upper index containing μ means that always the latest available values are used when the new values of u are calculated, regardless of the size of the time step.

2.2.12. The Leapfrog–Hopscotch method

Finally, in the leapfrog-hopscotch (LH) method [85] 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, then, for the even and odd nodes, full-time steps are taken strictly alternately until the end of the last timestep (orange box in Figure 2.3 B), which should be halved for odd nodes to reach the same final time point as the even nodes.

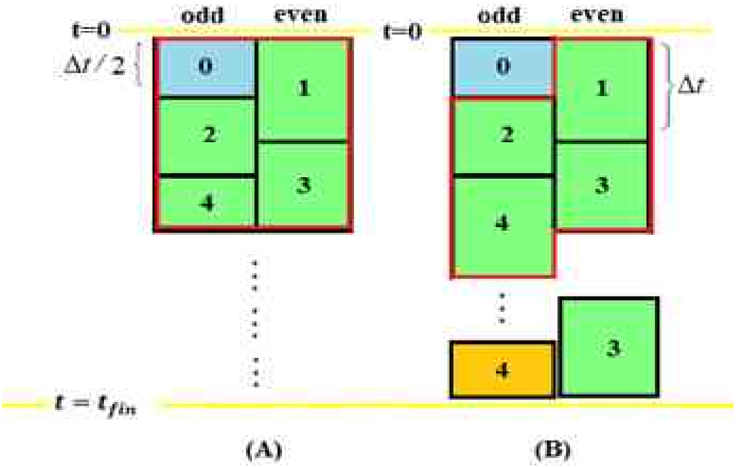

Figure 2.3. (A) The shifted-hopscotch structure. (B) The leapfrog-hopscotch structure.

3. STABLE, EXPLICIT, LEAPFROG-HOPSCOTCH ALGORITHMS FOR THE HEAT EQUATION

In this chapter, I construct novel numerical algorithms to solve the heat or diffusion equation. I combined the hopscotch space structure with leapfrog time integration. By applying the theta method with nine different values of θ and the recently invented CNe method, I constructed 10^5 different leapfrog-hopscotch algorithm combinations and via subsequent numerical investigations, this huge number was decreased by excluding the combinations that underperformed and, finally, only the top five of these were retained [85]. I used two two-dimensional stiff systems containing 10,000 cells with completely discontinuous random parameters and initial conditions, I demonstrate the performance of these top five methods in the case of large systems with random parameters and discontinuous initial conditions, by comparing them with other methods. My current work was inspired by the well-known leapfrog method [86] used by the molecular dynamics community to solve Newton's equations of motion. In their books, Hockney and Eastwood [87] (p. 28) and, later, Frenkel and Smit [88] (p. 75) introduced this method in the following form:

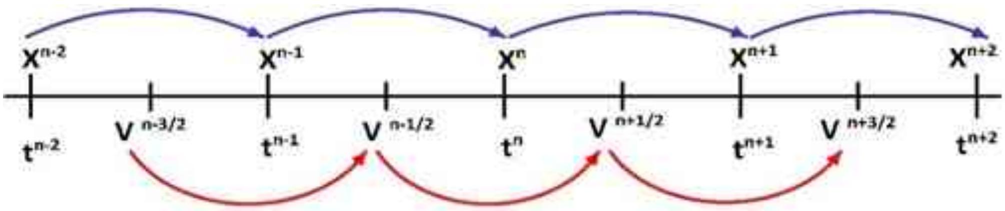

Figure 3.1. Diagram of the original leapfrog algorithm [88].

In Figure 3.1, a diagram of this method is presented, and one can immediately understand why the method is called leapfrog. The leapfrog scheme was successfully adapted and generalized to hyperbolic PDEs [89], but for parabolic equations it is unstable. This can be addressed by a modification to obtain the Dufort–Frankel method already mentioned.

In all of these publications, the nodes of the spatial mesh are treated equivalently, unlike in the OEH method. Verwer and Sommeijer [90] presented a composite method for the advection-diffusion problem, where the leapfrog scheme was used for the horizontal advection part, and the Du Fort Frankel and Crank–Nicolson methods are used for the other parts. It can be easily determined that their method is significantly different from mine. In addition, we have been unable to find any combinations of the leapfrog and hopscotch structures in the literature that are at least a little similar to mine.

In Subsection 3.1.1, first I very briefly present the results of the numerical tests for the first assessment of the 10^5 combinations to obtain a manageable number of methods. Then, in Subsection 3.1.2 and Subsection 3.1.3, two numerical experiments are presented for two space dimensional stiff systems consisting of 10,000 cells. I choose the top five combinations with the highest accuracy and compare them to selected other methods.

3.1. *The Results of Leapfrog-Hopscotch Tests*

In Subsections 3.1.1–3.1.3, two space dimensional, topologically rectangle-structured lattices with $N = N_{\mathrm{x}} \times N_{\mathrm{z}}$ cells are investigated (see Figure 2.1 for visualization). The reference solution is calculated by the ode15s built-in solver of MATLAB, with very strict error tolerance andtherefore high precision. The (global) numerical error is the absolute difference of the numerical solutions $u_{\mathrm{j}}^{\mathrm{num}}$ produced by the examined method and the reference solution $u_{\mathrm{j}}^{\mathrm{ref}}$ at final time t_{fin} . I use these individual errors of the nodes or cells to calculate the maximum error:

$$\mathrm{Error}(L_\infty) = \max_{1\le \mathrm{j}\le N} \left| u_{\mathrm{j}}^{\mathrm{ref}}(t_{\mathrm{fin}}) - u_{\mathrm{j}}^{\mathrm{num}}(t_{\mathrm{fin}}) \right| \tag{3.1}$$

the average error:

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

and the so-called energy error:

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

which, in the case of heat conduction, yields the error in terms of energy, where *C* is the heat capacity. However, for different algorithms, these errors depend on the time step size in different ways. If we want to evaluate the overall performance of the new methods compared to the original OEH method, we consider a fixed value of the final time t_{fin}, and first calculate the solution with a large time step size (typically $t_{\mathrm{fin}} / 4$), then repeat the whole calculation for subsequently halved time step sizes *S* times until *h* reaches a very small value (typically around 2×10^{-6}). Then, the aggregated relative error (ARE) quantities are calculated for each type of error defined above. For instance, in the case of the L_∞ error, it has the following form:

$$\mathrm{ARE}(L_\infty) = \frac{1}{S} \sum_{i=1}^{S} \left(\log\left(\mathrm{Error}(L_\infty)\right)_{\mathrm{OEH}} - \log\left(\mathrm{Error}(L_\infty)\right)_{\mathrm{examined}} \right) . \tag{3.4}$$

We can also take the simple average of these errors,

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

to obtain one single quantity for a method applied on a concrete problem, which will help us to select the best combinations. For example, if ARE = 2, then the examined method produces smaller errors by two orders of magnitude than the OEH method. For the simulations where running times are measured, I used a desktop computer with an Intel Core i5-9400 CPU, 16.0 GB RAM with the MATLAB R2020b software, in which there was a built-in tic-toc function to measure the total running time of the tested algorithms.

3.1.1. Preliminary tests

The following nine values of the parameter theta are substituted into the formula given in Eq. (2.28): $\theta \in \{0, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, 1\}$. Thus, with the CNe Formula, I have 10 different formulas. There are five stages in the leapfrog-hopscotch structure and by inserting these 10 formulas in all possible combinations, I obtain 10^5 = 100,000 different algorithm combinations in total. With one of my colleagues, I wrote a MATLAB code that constructs and tests all of these combinations in a highly automatized manner under the following circumstances.

The generalized Eq. (2.7) is solved in the case of four different small systems with $N_{\mathrm{x}} \times N_{\mathrm{z}} = 2\times 2, 2\times 6, 4\times 4,$ and 3×5, to ensure the total running times are manageable despite the large number of combinations. Here, and also in Subsection 3.1.2 and Subsection 3.1.3, randomly generated cell capacities and thermal resistances following a log-uniform distribution

$$C_{\mathrm{i}} = 10^{(\alpha_C - \beta_C \times rand)},\; R_{\mathrm{x,i}} = 10^{(\alpha_{Rx} - \beta_{Rx} \times rand)},\; R_{\mathrm{z,i}} = 10^{(\alpha_{Rz} - \beta_{Rz} \times rand)} \qquad (3.6)$$

have been used. The parameters α_C, β_C, α_{Rx}, β_{Rx}, α_{Rz}, β_{Rz} of the distribution of the mesh-cells data were chosen to construct test problems with various stiffness ratios and $h_{\mathrm{MAX}}^{\mathrm{FTCS}}$, for example, $\alpha_C = 1, 2,$ or 3, $\beta_C = 2, 4,$ or 6. The (pseudo-) random number, rand, is generated by MATLAB for each quantity with a uniform distribution in the unit interval (0, 1). I also generate different random values for the initial conditions $u_{\mathrm{i}}(0) = rand$. The final times were set to $t_{\mathrm{fin}} = 0.1$ and once to $t_{\mathrm{fin}} = 10$. The purpose of this latter, larger number is to let instabilities manifest themselves in order to exclude the unstable combinations. The computer program calculated the aggregated relative error (ARE) quantities and then sorted the algorithms in descending order according to this quantity to obtain a ranking list of the 10^5 algorithm combinations. Finally, I manually checked the top of these lists for all of the four small systems to select the best 20 combinations, which have the following short form:

$$\begin{aligned}
&(C, C, C, C, C),\ (\tfrac{1}{4}, \tfrac{1}{2}, C, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{4}, \tfrac{1}{2}, C, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{4}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, C),\\
&(\tfrac{1}{4}, \tfrac{1}{2}, C, \tfrac{1}{2}, C),\ (C, \tfrac{1}{2}, C, \tfrac{1}{2}, C),\ (C, \tfrac{1}{4}, \tfrac{1}{2}, C, \tfrac{1}{2}),\ (C, \tfrac{1}{2}, C, \tfrac{1}{2}, \tfrac{1}{2}),\\
&(C, \tfrac{1}{2}, C, C, \tfrac{1}{2}),\ (C, \tfrac{1}{3}, \tfrac{1}{2}, \tfrac{2}{3}, C),\ (\tfrac{1}{4}, C, \tfrac{1}{4}, C, \tfrac{3}{4}),\ (0, C, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\\
&(C, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{4}, C, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}),\ (0, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\ (1, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\\
&(\tfrac{1}{4}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{3}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),\ (\tfrac{1}{5}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}).
\end{aligned} \qquad (3.7)$$

In the next two subsections, I start only with these combinations. I note that there is an "odd one out" in the list above, i.e., the (*C*, *C*, *C*, *C*, *C*) combination, which was typically not at the top of the ranking lists above. I include it into the top 20 and later, in the top five, because it is the best among those combinations which preserve positivity of the solution; more precisely, it always follows the maximum and minimum principles as the true solution [91] (p. 87), which is proven by my supervisor in the analytical section [85]. The only purpose of these preliminary tests on the small systems is the reduction of the large number of combinations by eliminating those which are probably not worthy of more careful investigation.

3.1.2. Comparison with other methods for a large, moderately stiff system

I examine a grid similar to the one in Figure 2.1 with an isolated boundary, but where the sizes are fixed to $N_x = 100$ and $N_z = 100$; thus, the total number of cells is 10,000, and the final time is $t_{\text{fin}} = 0.1$. The exponents introduced above were set to the following values:

$$\alpha_C = 2,\ \beta_{\text{C}} = 4,\ \alpha_{Rx} = \alpha_{Rz} = 1,\ \beta_{Rx} = \beta_{Rz} = 2, \qquad (3.8)$$

which means, e.g., that log-uniformly distributed values between 0.01 and 100 were given to the capacities. The generated system can be characterized by its stiffness ratio and $h_{\text{MAX}}^{\text{FTCS}}$ values, which are 3.1×10^7 and 7.3×10^{-4}, respectively. The performance of the new algorithms is compared with the following, widely used MATLAB solvers: ode15s, ode23s, ode23t, ode23tb, ode45, ode23, and ode113.

In the case of the MATLAB solvers, we could not directly set the time step size, but changed the tolerances instead, starting from a large value, such as $\text{'AbsTol'} = \text{'RelTol'}$ □ $\text{'Tol'} = 10^3$, towards a small minimum value, for example, $\text{'Tol'} = 10^{-5}$. In addition to these solvers, I also used the following methods for comparison purposes; the original UPFD, the CNe, the two- and three-stage linear-neighbour (LNe and LNe3) methods [62], and the Dufort–Frankel (DF) [91] (p. 120) are explicit unconditionally stable schemes. The Heun method, also called the explicit trapezoidal rule, is one of the most common second-order RK scheme [81]. Finally, the widely used Crank–Nicolson scheme (abbreviated here as CrN) is an implicit scheme:

$$\underbrace{\left(I - \frac{h}{2}M\right)}_{A}\vec{u}^{\,n+1} = \underbrace{\left(I + \frac{h}{2}M\right)}_{B}\vec{u}^{\,n},$$

where I is the $N \times N$ unit matrix. The A and B matrixes here are time-independent; thus, it is sufficient to calculate them only once, before the first-time step. We implement this scheme in two different ways: one is to calculate $Y = A^{-1}B$ before the first time step and then each time step is just a simple matrix multiplication $\vec{u}^{\,n+1} = Y\,\vec{u}^{\,n}$, which is denoted by “CrN invert”. The other is the $\vec{u}^{\,n+1} = \text{linsolve}\left(A, B\vec{u}^{\,n}\right)$ command of MATLAB, which we denote by “CrN lins”.

I plotted the L_∞, L_1, and energy errors as a function of the effective time step size h_{EFF}, and based on this (and on similar data that are presented in the next subsection), I selected the following top five combinations from those listed in Eq. (3.7) and discarded the reminder:

$$L1\ \left(C,\ C,\ C,\ C,\ C\right),\ \ L2\ \left(0,\ ½,\ ½,\ ½,\ ½\right),$$
$$L3\ \left(\tfrac{1}{5},\ ½,\ ½,\ ½,\ ½\right),\ \ L4\ \left(¼,\ ½,\ C,\ ½,\ ½\right),$$
$$L5\ \left(\tfrac{1}{5},\ ½,\ C,\ ½,\ ½\right).$$

The L2–3 and L4–5 are highly similar; only the zeroth stage is different. I tried to optimize the value of theta for this zeroth stage by calculating the errors for a few fixed time step sizes for

$\theta = k / N_\theta, k = 0,...,N_\theta$, where $N_\theta = 1000$. In case of the method L2–3 (θ, ½, ½, ½, ½), I found the errors are larger for large values of theta and the smallest when $\theta \approx 0$. In case of the L4–5 (θ, ½, C, ½, ½) combination, I found that there is a nontrivial minimum of the error function at around $\theta \approx 0.2$, but its position slightly depends on the time step size, as shown in Figure 3.2. I can conclude that the L4 combination is usually slightly more accurate than that of L5.

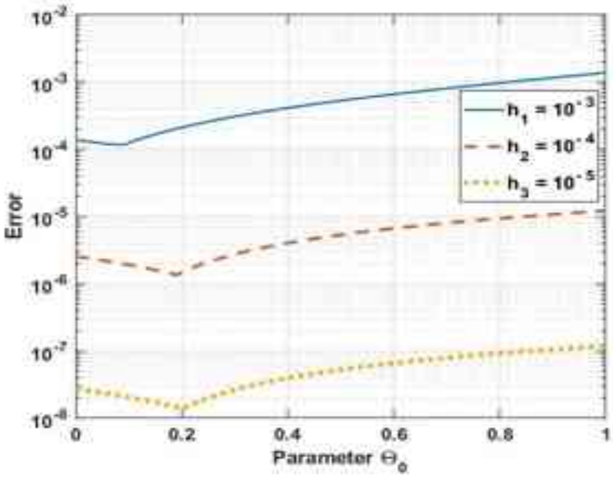

Figure 3.2. Errors as a function of the parameter θ_0 for the first (moderately stiff) system, in the case of the method L4–5 (θ, ½, C, ½, ½).

In Figure 3.3 I present the energy error as a function of the time step size for these top five combinations and other methods, and Figure 3.4 shows the L_1 errors vs. the total running times. Furthermore, Table 3.1 presents some results that were obtained by my numerical schemes and the “ode” routines of MATLAB. One can see that, as expected, the implicit Crank–Nicolson scheme is the most accurate, but the accuracy of the L2 method, in addition to the L5 and S4 methods, approaches it, and is similar to that of the Heun method below the CFL limit.

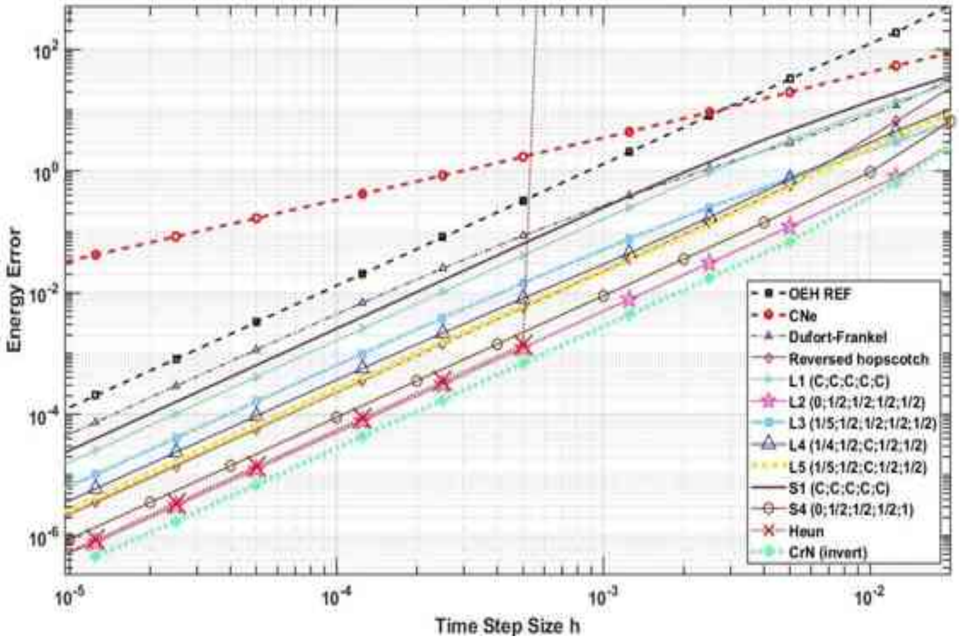

Figure 3.3. Energy errors as a function of the time step size for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), the new algorithms L1–L5, and several other methods.

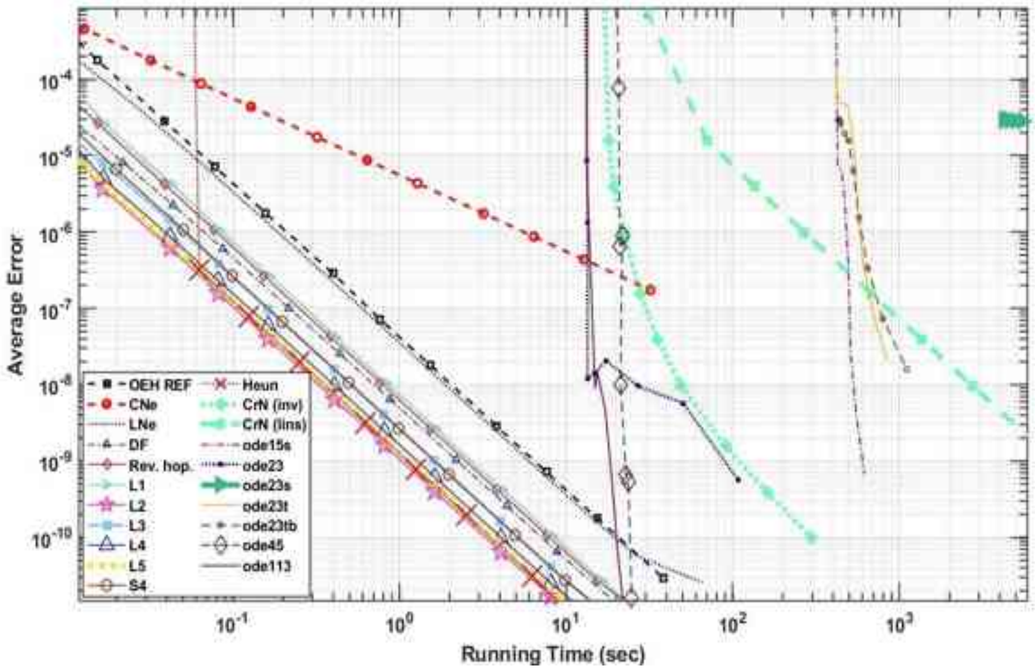

Figure 3.4. The L_1 Errors as a function of the running time for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), the new algorithms L1–L5, and several other methods.

Table 3.1. Comparison of different leapfrog-hopscotch algorithms with other methods for the moderately stiff system of ten thousand cells.

Numerical Method	Error L_∞	Error (L_1)	Energy Error	Running Time (sec)
ode45, $\mathrm{Tol}=10^{-3}$	5.18×10^{-5}	1.01×10^{-8}	4.26×10^{-5}	2.12×10^{1}
ode23, $\mathrm{Tol}=10^{-5}$	3.85×10^{-6}	2.06×10^{-8}	1.03×10^{-4}	1.73×10^{1}
ode113, $\mathrm{Tol}=10^{-4}$	6.77×10^{-5}	1.49×10^{-8}	6.42×10^{-5}	1.54×10^{1}
ode15s, $\mathrm{Tol}=10^{2}$	1.18×10^{-2}	1.05×10^{-3}	5.27×10^{0}	$4.03\times10^{+2}$
ode23s, $\mathrm{Tol}=10^{2}$	4.11×10^{-4}	2.90×10^{-5}	1.46×10^{-1}	4.23×10^{3}
ode23t, $\mathrm{Tol}=10^{-7}$	1.32×10^{-6}	9.32×10^{-8}	4.68×10^{-4}	6.88×10^{2}
ode23tb, $\mathrm{Tol}=10^{3}$	4.18×10^{-4}	2.95×10^{-5}	1.48×10^{-1}	4.36×10^{2}
CNe, $h=5\times10^{-4}$	3.33×10^{-3}	1.78×10^{-4}	1.71×10^{0}	3.19×10^{-2}
LNe, $h=5\times10^{-5}$	2.99×10^{-6}	8.33×10^{-8}	6.84×10^{-4}	6.52×10^{-1}
LNe3, $h=1.25\times10^{-4}$	8.78×10^{-6}	2.11×10^{-7}	2.50×10^{-3}	2.61×10^{-1}
DF, $h=2.5\times10^{-6}$	1.04×10^{-8}	2.59×10^{-10}	2.98×10^{-6}	6.7×10^{0}
Rev. hop., $h=5\times10^{-6}$	4.23×10^{-8}	4.27×10^{-10}	5.75×10^{-7}	3.82×10^{0}
L1, $h=1.25\times10^{-4}$	9.06×10^{-6}	2.63×10^{-7}	2.56×10^{-3}	1.68×10^{-1}
L2, $h=2.5\times10^{-4}$	1.16×10^{-5}	1.58×10^{-7}	3.11×10^{-4}	8.19×10^{-2}
L3, $h=1.25\times10^{-6}$	5.19×10^{-10}	1.11×10^{-11}	2.99×10^{-7}	1.62×10^{-1}
L4, $h=1.25\times10^{-5}$	3.05×10^{-8}	6.20×10^{-10}	5.98×10^{-6}	1.62×10^{0}
L5, $h=5\times10^{-6}$	3.51×10^{-9}	8.01×10^{-11}	6.80×10^{-7}	4.07×10^{0}
S1, $h=2.5\times10^{-6}$	3.63×10^{-9}	1.05×10^{-10}	1.03×10^{-6}	9.93×10^{0}
S4, $h=1.25\times10^{-5}$	6.61×10^{-8}	6.68×10^{-10}	8.92×10^{-7}	1.97×10^{0}
Heun, $h=1.25\times10^{-4}$	2.79×10^{-7}	1.97×10^{-8}	8.68×10^{-5}	2.5×10^{-1}
CrN (inv), $h=5\times10^{-3}$	3.67×10^{-4}	1.59×10^{-5}	6.93×10^{-2}	1.80×10^{1}
CrN (lins), $h=5\times10^{-3}$	3.67×10^{-4}	1.59×10^{-5}	6.93×10^{-2}	7.03×10^{1}

3.1.3. Comparison with other methods for a large, very stiff system

In the second case study, new values have been set for the α and β exponents:

$$\alpha_C = 3,\ \beta_C = 6,\ \alpha_{Rx} = 3,\ \alpha_{Rz} = 1,\ \beta_{Rx} = \beta_{Rz} = 4\ .$$

This means that the width of the distribution of the capacities and thermal resistances have been increased and a non-negligible anisotropy appeared, since the resistances in the *x* direction are two orders of magnitude larger than in the *z* direction on average. Note that the geometric mean of the *C* and *R* quantities are still one. With this modification I have gained a system with much higher stiffness ratio, 2.5×10^{11}, while the CFL limit for the standard FTCS was $h_{MAX}^{EE} = 1.6\times10^{-6}$, which, we stress again, holds for the Heun method as well. All other parameters and circumstances are the same as in the previous subsection. In Figure 3.5 and Figure 3.6 the energy and the L_∞ errors is presented as a function of the time step size and the total running time, respectively. It is not surprising that the implicit methods gained a slight advantage compared to the previous, less stiff case, but the new L2 method outperforms all other examined method if not only the accuracy, but the speed is taken into account. In Table 3.2 I report data belong to this numerical experiment, while in Table 3.3 I summarize the ARE error quantities, defined in Eq. (3.5), of the explicit stable methods for both case studies.

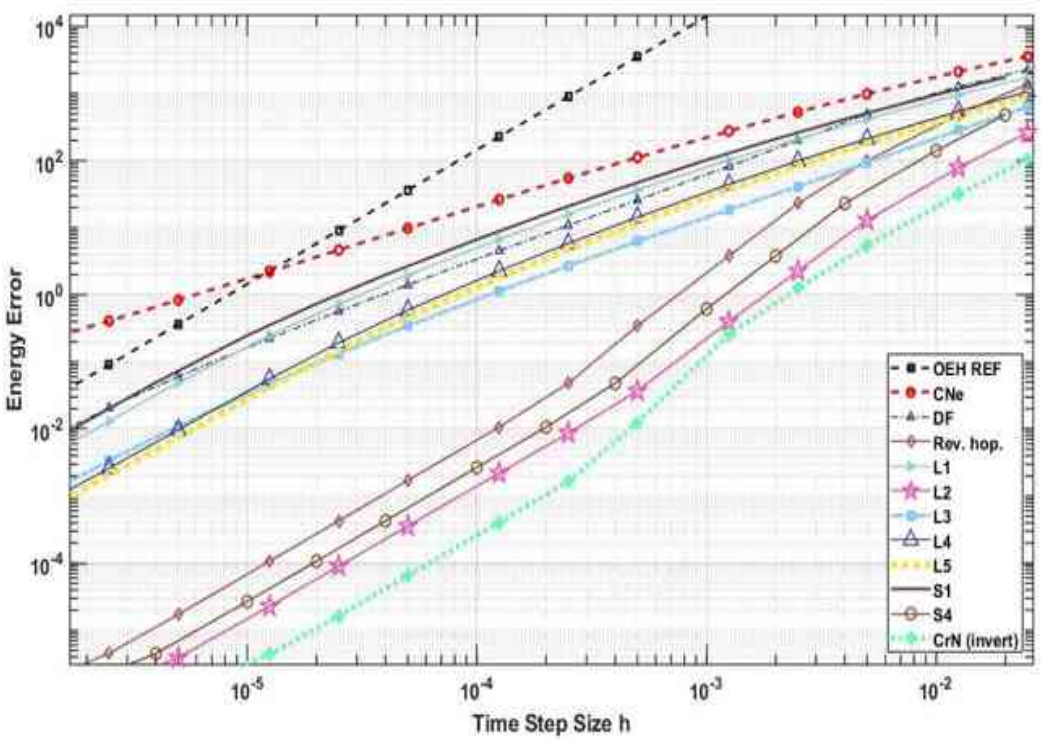

Figure 3.5. Energy errors as a function of the time step size for the second (very stiff) system, in the case of the original OEH method (OEH REF), the new algorithms L1-L5 and some other methods.

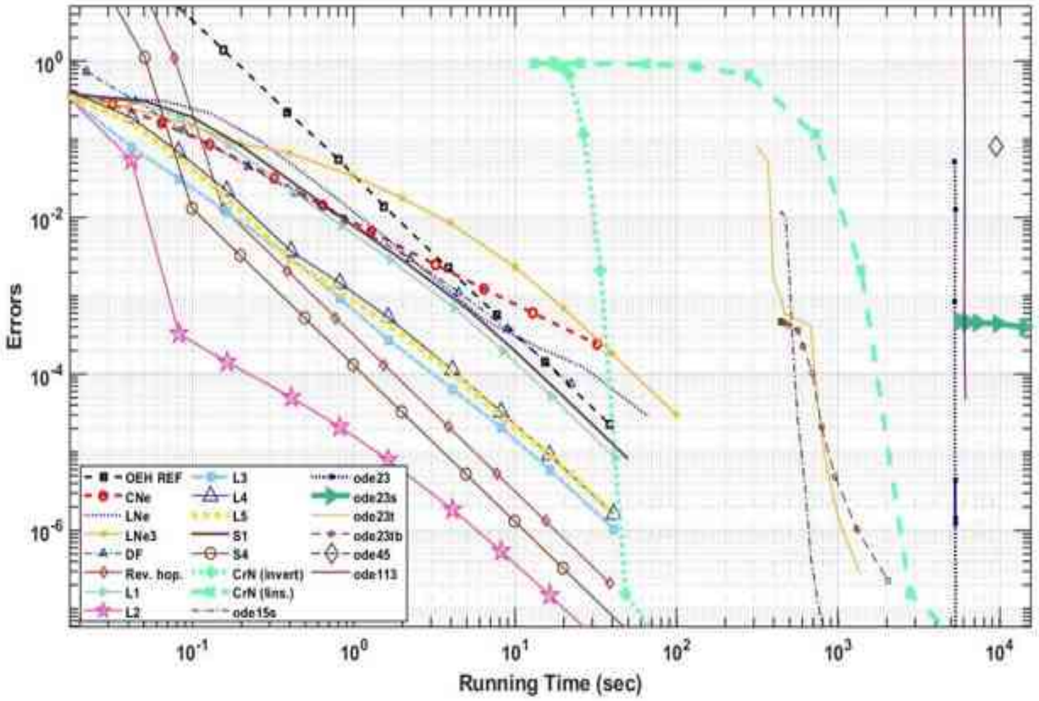

Figure 3.6. The L∞ errors as a function of the running time for the second (very stiff) system, in the case of the original OEH method (OEH REF), one stage CNe method, the new algorithms L1-L5 and a couple of other methods.

Table 3.2. Comparison of different algorithms for the very stiff system of ten thousand cells.

Numerical Method	**Error(L_∞)**	**Error(L_1)**	**Energy Error**	**Running Time (sec)**
ode45, $\mathrm{Tol}=10^{0}$	8.13×10^{-2}	1.52×10^{-5}	7.03×10^{-2}	9.48×10^{3}
ode23, $\mathrm{Tol}=10^{-2}$	1.25×10^{-2}	2.34×10^{-6}	1.08×10^{-2}	5.33×10^{3}
ode113, $\mathrm{Tol}=10^{-4}$	4.07×10^{-5}	8.84×10^{-9}	4.08×10^{-5}	6.15×10^{3}
ode15s, $\mathrm{Tol}=10^{-1}$	1.05×10^{-2}	7.19×10^{-4}	3.65×10^{0}	4.72×10^{2}
ode23s, $\mathrm{Tol}=10^{-3}$	3.12×10^{-4}	1.57×10^{-5}	7.89×10^{-2}	2.48×10^{4}
ode23t, $\mathrm{Tol}=10^{-5}$	2.58×10^{-5}	1.31×10^{-6}	6.55×10^{-3}	7.07×10^{2}
ode23tb, $\mathrm{Tol}=10^{-7}$	1.04×10^{-6}	5.29×10^{-8}	2.65×10^{-4}	1.30×10^{3}
CNe, $h=1.25\times10^{-4}$	8.62×10^{-2}	1.32×10^{-3}	2.64×10^{1}	1.28×10^{-1}
LNe, $h=5\times10^{-5}$	2.35×10^{-2}	1.37×10^{-4}	2.39×10^{0}	6.52×10^{-1}
LNe3, $h=2.5\times10^{-5}$	1.79×10^{-2}	3.81×10^{-5}	7.26×10^{-1}	1.98×10^{0}
DF, $h=1.25\times10^{-5}$	3.74×10^{-4}	3.75×10^{-7}	6.47×10^{-3}	8.9×10^{0}
Rev. hop., $h=1.25\times10^{-5}$	1.31×10^{-4}	3.55×10^{-7}	1.07×10^{-4}	1.54×10^{0}
L1, $h=1.25\times10^{-5}$	2.92×10^{-3}	1.46×10^{-5}	2.48×10^{-1}	1.67×10^{0}
L2, $h=5\times10^{-6}$	1.85×10^{-6}	5.59×10^{-9}	3.89×10^{-6}	4.08×10^{0}
L3, $h=2.5\times10^{-6}$	2.04×10^{-5}	1.52×10^{-7}	3.38×10^{-3}	8.18×10^{0}
L4, $h=1.25\times10^{-3}$	4.36×10^{-1}	3.59×10^{-3}	4.44×10^{1}	1.65×10^{-2}
L5, $h=5\times10^{-7}$	1.61×10^{-6}	6.76×10^{-9}	9.48×10^{-5}	4.08×10^{1}
S1, $h=5.5\times10^{-5}$	2.20×10^{-2}	1.25×10^{-4}	1.99×10^{0}	4.94×10^{-1}
S4, $h=2.5\times10^{-6}$	1.31×10^{-6}	3.55×10^{-9}	1.71×10^{-6}	9.87×10^{0}
Heun, $h=1.25\times10^{-6}$	2.83×10^{-11}	1.39×10^{-12}	7.77×10^{-8}	2.52×10^{1}
CrN (inv), $h=1.25\times10^{-4}$	1.56×10^{-7}	7.91×10^{-9}	4.08×10^{-4}	4.83×10^{1}
CrN (lins), $h=1.25\times10^{-4}$	1.56×10^{-7}	7.91×10^{-9}	4.08×10^{-4}	2.78×10^{3}

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

Numerical Method	ARE (Mod. Stiff)	ARE (Very Stiff)
CNe	−1.242	1.264
LNe	0.699	1.804
LNe3	0.924	1.859
Dufort-Frankel	0.615	1.575
Reversed hopscotch	0.928	2.570
L1 (C, C, C, C, C)	0.985	1.885
L2 (0, ½, ½, ½, ½)	1.745	3.988
L3 (⅕, ½, ½, ½, ½)	1.326	2.451
L4 (¼, ½, C, ½, ½),	1.486	2.315
L5 (⅕, ½, C, ½, ½)	1.573	2.385
S1 (C, C, C, C, C)	0.985	1.885
S4 (0, ½, ½, ½,1)	1.509	3.184

I found that the L2 (0, 1/2, 1/2, 1/2, 1/2) combination is the most competitive. It yields accurate results orders of magnitude faster than the well-optimized MATLAB routines, and it is more accurate than all of the other examined explicit and stable methods. Although, unlike the L1 (C, C, C, C, C) algorithm, L2 is not positivity preserving, it is also surprisingly robust for relatively large time step sizes, which is not that case for the original OEH algorithm. Moreover, this new L2 algorithm is easy to implement and requires an even smaller amount of function evaluation and computer memory than the conventional explicit second order Runge–Kutta methods, such as the Heun method. I can conclude that it combines has the most important advantages of the standard explicit and the implicit methods. Now I present the final formula of the leapfrog-hopscotch method for conduction with the best already proven combination of formulas (L2) as follows:

The first stage has the length of a halved time step, thus we have the following special and general formulas

$$u_i^{\frac{1}{2}} = \frac{u_i^0 + \frac{r}{2}\left(u_{i-1}^0 + u_{i+1}^0\right) + \frac{h}{2}q_i}{1+r}, \quad \text{and} \quad u_i^{\frac{1}{2}} = \frac{u_i^0 + \frac{A_i^0}{2} + \frac{h}{2}q_i}{1+r_i/2} \tag{3.9}$$

Then a full-time step is made for the even nodes using

$$u_i^1 = \frac{(1-r)u_i^0 + r\left(u_{i-1}^{1/2} + u_{i+1}^{1/2}\right) + hq_i}{1+r}, \quad \text{and} \quad u_i^1 = \frac{\left(1-\frac{r_i}{2}\right)u_i^0 + A_i^{1/2} + hq_i}{1+r_i/2} \tag{3.10}$$

After this, full time steps are taken alternately for the odd and even nodes. Finally, a half-length time step must close the calculations for the odd nodes

$$u_i^T = \frac{(1-\frac{r}{2})u_i^{T-1/2} + \frac{r}{2}\left(u_{i-1}^T + u_{i+1}^T\right) + \frac{h}{2}q_i}{1+r/2}, \quad \text{and} \quad u_i^T = \frac{(1-\frac{r_i}{4})u_i^{T-1/2} + \frac{A_i^T}{2} + \frac{h}{2}q_i}{1+r_i/4} \tag{3.11}$$

4. A COMPARATIVE STUDY OF EXPLICIT AND STABLE TIME INTEGRATION SCHEMES FOR HEAT CONDUCTION IN AN INSULATED WALL

In this chapter, I extensively examine 13 numerical methods to solve the linear heat conduction equation in building walls. Eight of the used methods (including the previously examined leapfrog-hopscotch) are recently invented explicit algorithms which are unconditionally stable. First, I performed verification tests in a 2D case by comparing them to analytical solutions using equidistant and non-equidistant grids. Then I tested them on real-life applications in the case of one-layer (brick) and two-layer (brick and insulator) walls to determine how the errors depend on the real properties of the materials, the mesh type, and the time step size. I applied space-dependent boundary conditions on the brick side and time-dependent boundary conditions on the insulation side. The results show that the best algorithm is usually the original odd-even hopscotch method for uniform cases and the leapfrog-hopscotch algorithm for non-uniform cases. So, I perform systematic tests in the building walls by varying some parameters of the system and the mesh to examine how the performance of the individual methods changes and which of them is the best choice under different circumstances. I note that no comparative study has been conducted until my work even about the four already known explicit and stable methods examined in this work, namely the UPFD, odd-even hopscotch, Dufort-Frankel, and rational Runge-Kutta methods.

4.1. The Geometry and Materials

I considered a one-layer wall consisting of brick only and two-layer walls consisting of brick and glass wool insulator as shown in Figure 4.1.

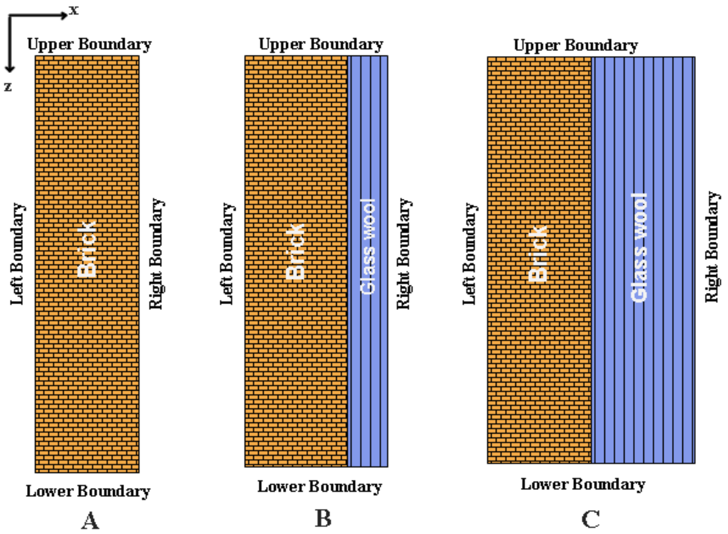

Figure 4.1. (A) One layer wall, (B) and (C) wall with insulator.

In the present chapter, I take real material properties as listed in Table 4.1.

Table 4.1. The properties of the materials used.

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

I apply an equidistant grid and some non-equidistant grids to discretize the space variables in both the one layer and the multilayer cases. The cell number along axis *x* is set to N_x=100. Similarly, the cell number along axis *z* is N_z=100. Thus, we have a grid with a total cell number $N = N_x N_z = 10000$. We have to note that the temperature in the middle of the cell was considered the temperature of the cell. However, we are going to use Dirichlet boundary conditions to reproduce an analytical solution, therefore the boundary of the system should be in the middle of the cells belonging to the boundary. This issue is solved by increasing the size of the cells, so in the case of an equidistant grid and $N_x = N_z = 100$, we have $\Delta x = \Delta z = 0.0101$ instead of just 0.01.

We also wanted to consider wide cells on the left side of the wall and small ones on the right side of the wall. We implemented it in two different ways. In case of abrupt change, we used an equidistant coarse mesh $\Delta x = 0.0105$ at the left 50% of the cells, and an equidistant fine mesh $\Delta x = 0.0097$ at the right side.

For a gradual change we used the following formula: For $r \neq 1$, the sum of the first *n*+1 terms of a geometric series, up to and including the r^n term, is

$$a + ar + ar^2 + ar^3 + ... + ar^n = \sum_{k=0}^{n} ar^k = a\left(\frac{1-r^{n+1}}{1-r}\right) \tag{4.1}$$

We used *r*=0.98, $n = N_x - 1$, and *a*=0.0234. It means that on the left side $\Delta x_1 = 0.0234$ and it is gradually decreased to $\Delta x_{N_x} = 0.98^{99} \cdot \Delta x_1 = 0.00317$. The same abrupt and gradual change can be implemented in the *z*-direction.

In the multilayer case, always the left 50% of the cells were brick and the right 50% were insulator. It implies that if the mesh is equidistant, the volume of the brick and the insulator is the same as in Figure 4.1C. However, if we have abrupt or gradual change in the x-direction, the thickness of the insulator is smaller, similarly to the case in Figure 4.1B.

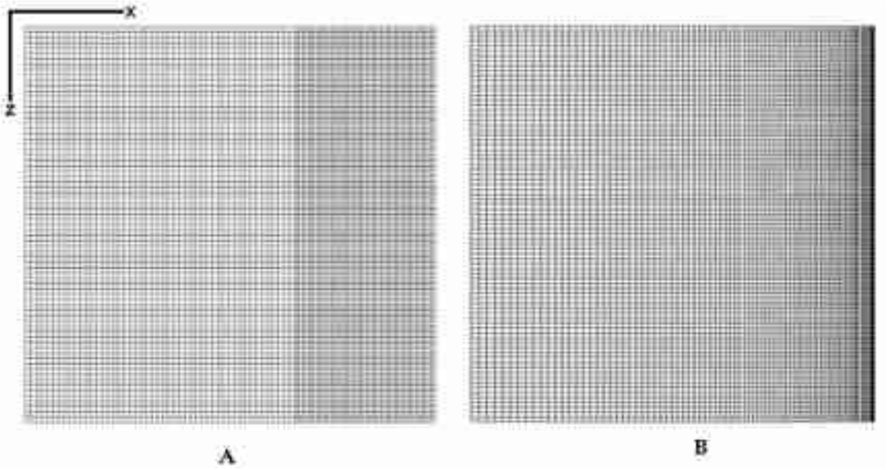

Figure 4.2. (A) Abrupt change. (B) Gradual change in the *x* direction.

We always use equidistant temporal discretization with time step size Δt, and u_i^n denotes the temperature of cell *i* at time moment $n\Delta t$.

4.2. The Initial and the Boundary Conditions

I applied different initial and boundary conditions for both the one layer and the multilayer cases as follows:

I. Sinusoidal initial condition with zero Dirichlet boundary condition.

The initial condition is the product of two sine functions:

$$u(x,z,t=0)=\sin(\pi x)\sin(\pi z) \tag{4.2}$$

The simplest zero Dirichlet boundary conditions are used:

$$u(x=0,z,t)=u(x=1,z,t)=u(x,z=0,t)=u(x,z=1,t)=0 \tag{4.3}$$

Anyone can easily check that the analytical solution to this problem is

$$u(x,z,t)=\sin(\pi x)\sin(\pi y)\,\mathrm{e}^{-2\pi^2 t} \tag{4.4}$$

valid only in homogeneous material, i.e., in a one-layer wall.

II. Linear initial condition with combined boundary conditions.

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

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

Neumann boundary condition at the top and bottom of the wall, meaning thermal isolation:

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

Space-dependent temperature at the left boundary:

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

Time dependent temperature at the right boundary:

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

where λ =0.00004. It means that the temperature at the right boundary is gradually increased from $30°\mathrm{C}$ to $44.75°\mathrm{C}$. I note that we chose so complicated boundary conditions to demonstrate that the methods perform well even in these cases.

4.3. Results

The reference solution is either the analytical solution (4.4) of the PDE or a numerical solution obtained by applying Heun’s method with an extremely small-time step size $\Delta t = 0.002$. I have chosen Heun’s method for reference because this is the most widely tested algorithm among the examined methods.

For the simulations where running times are measured, I used a desktop computer with an Intel Core i7-11700 CPU and 64 GB RAM with the MATLAB R2020b software.

4.3.1. Verification using the analytical solution

I simulated a one-layer brick wall (see Figure 4.1A)As it is written in point I. above, I ap-plied sinusoidal initial temperature distribution (4.2) and zero Dirichlet boundary condition (4.3) using the analytical solution (4.4) at $t_{\mathrm{fin}} = 10000\left(\mathrm{s}\right)$. I made the simulations in all the possible three cases, which are the following:

a) Equidistant mesh.
b) Abrupt change in the *x*-direction, equidistant mesh in the *z*-direction.
c) Abrupt change in both *x* and *z* directions.
d) Gradual changing in *x*-direction, equidistant mesh in *z*-direction.
e) Gradual changing in both *x* and *z* directions.
f) Abrupt change in *x*-direction, gradual changing in *z*-direction.

The obtained results are very similar for all the cases and the residual error (the error for very small-time step sizes due to space discretization) is below 10^{-4}. This means that the code for equidistant and non-equidistant meshes are successfully verified. In Figure 4.3 and Figure 4.4, the errors as a function of the time step sizes are presented in log-log diagrams for cases (a) and (f), respectively. One can see that the UPFD and the CNe methods are first order while the others are second order in the time step size, as it is expected. Note that the hopscotch algorithms, especially the original OOEH, are more accurate than the other algorithms. The Heun’s method are quite accurate once we are below the CFL limit, but above this limit it produces no meaningful results.

In Figure 4.5 and Figure 4.6, the errors as a function of the running times are presented for the same cases. To reduce the effect of the fluctuations in running time measurements, I averaged out the running times of five different runs. As I expected, the differences of the running times for a fixed time step size are mostly caused by the different number of stages, e.g., the LNe3 method consists of three stages and therefore its curve is shifted slightly to the right relative to all other methods in the Figure 4.5 and Figure 4.6.

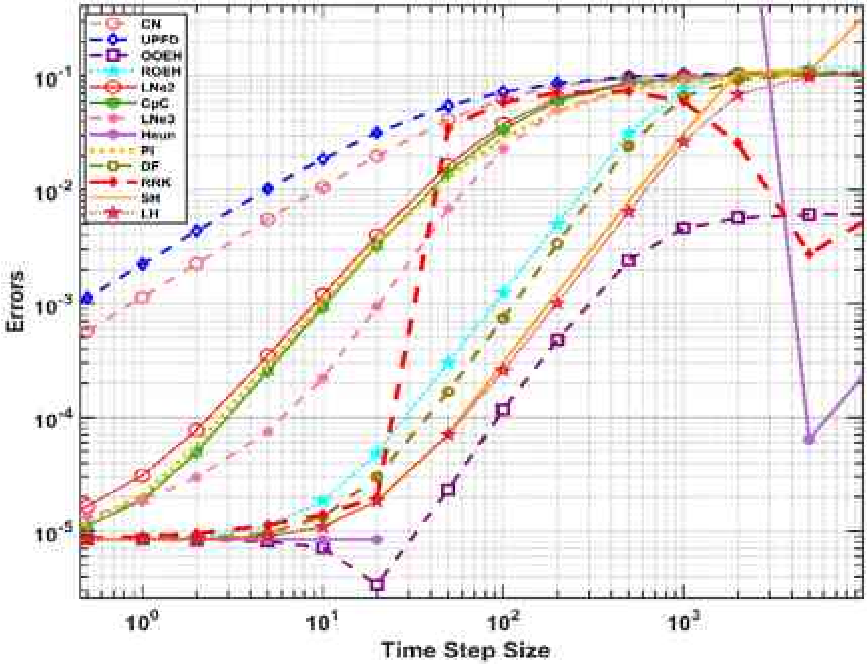

Figure 4.3. The maximum errors as a function of the time step size for the 13 examined methods in the case of an equidistant mesh.

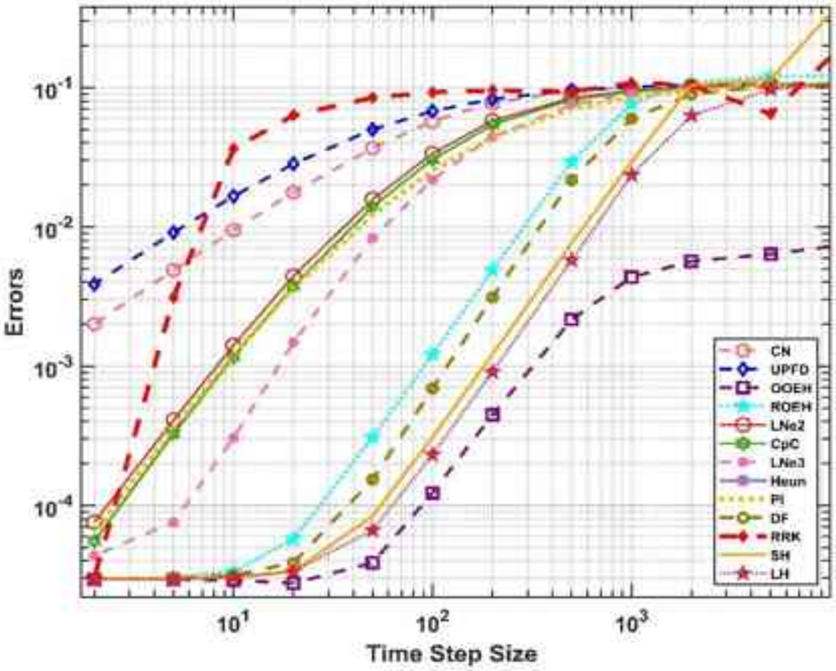

Figure 4.4. The maximum errors as a function of the time step size for the abrupt change in the *x*-direction and gradual change in the *z*-direction.

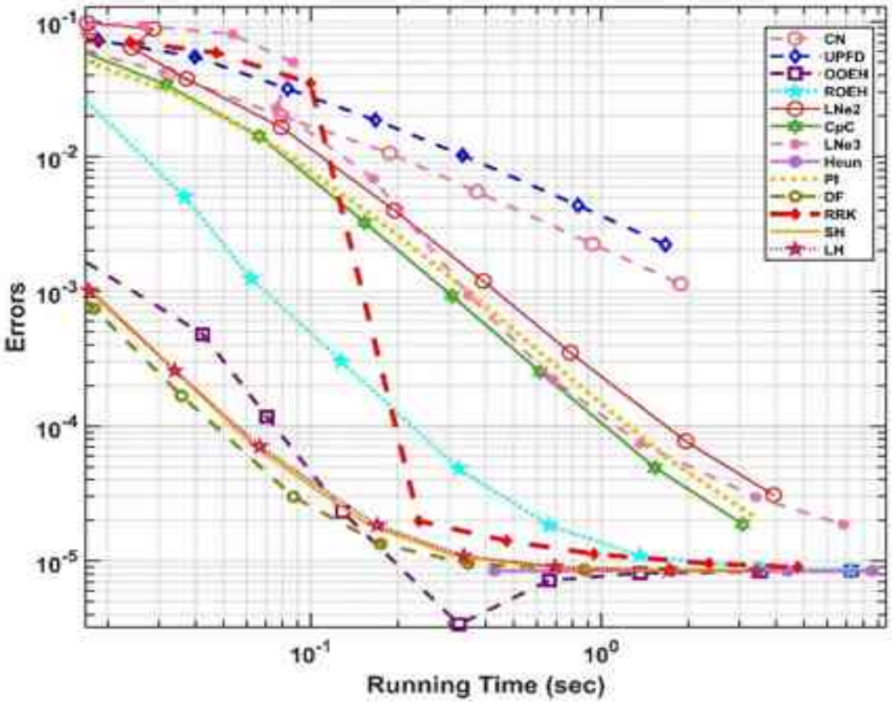

Figure 4.5. The maximum errors as a function of the running time for the 13 examined methods in the case of an equidistant mesh.

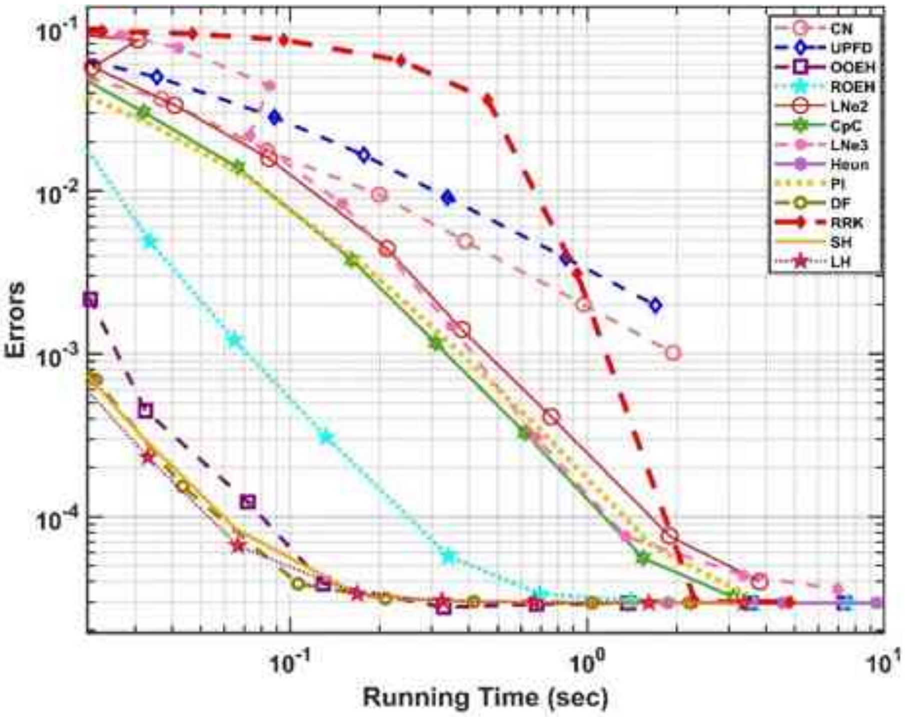

Figure 4.6. The maximum errors as a function of the running time for the abrupt change in the x-direction and gradual change in the z-direction.

4.4. Brick wall with Insulation

I applied the sinusoidal initial and Dirichlet boundary condition of Point I. for the multilayer wall with t_{fin}=10000s. As it was mentioned above, the reference solution was provided by Heun's method. The errors are plotted for equidistant and non-equidistant mesh in Figure 4.7 and Figure

4.8. I also plotted the final temperatures in the middle horizontal line of the wall for the reference solution and for the LH method with a quite large time step size as shown in Figure 4.9. One can now visualize the effect of the insulator (slower decrease of the temperature) and also see that the LH method is accurate for this large time step size.

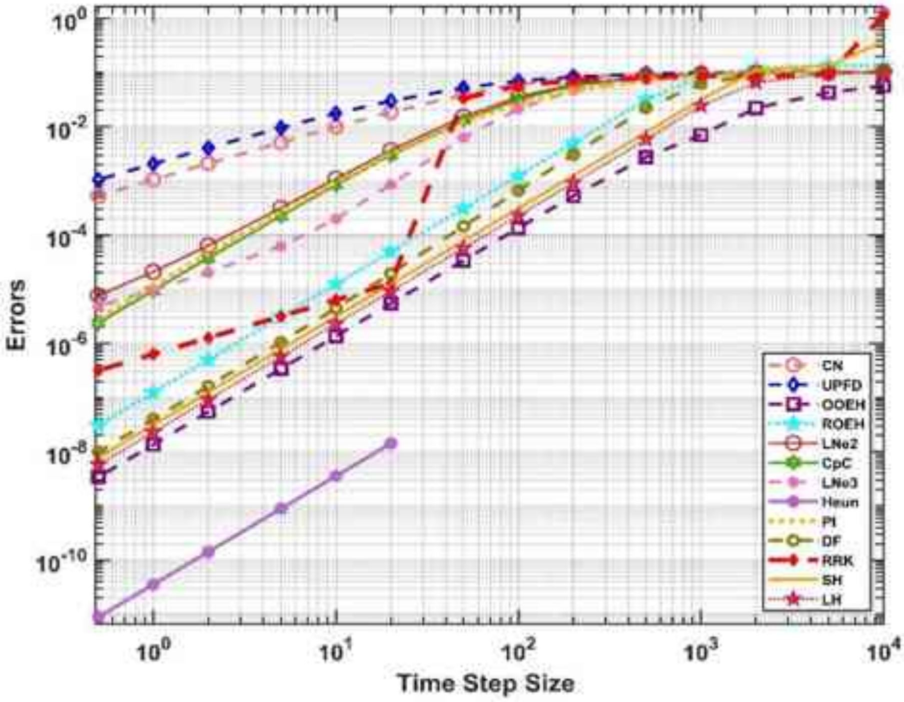

Figure 4.7. The maximum errors as a function of the time step size Δt for equidistant mesh.

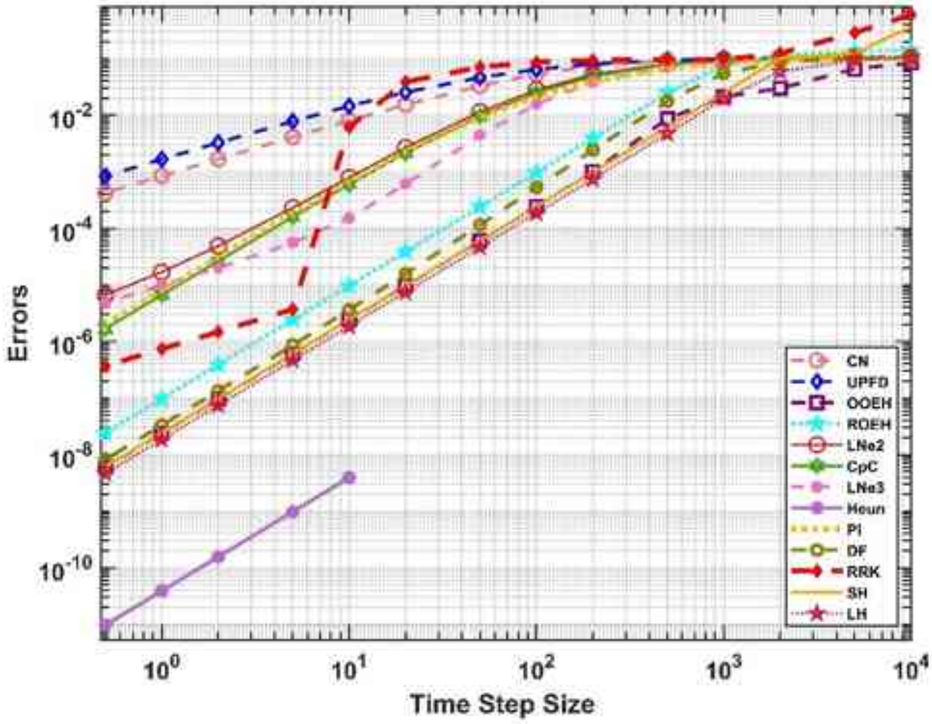

Figure 4.8. The L_∞ errors as a function of the time step size Δt in case of abrupt change in the *x*-direction and gradual change in the *z*-direction.

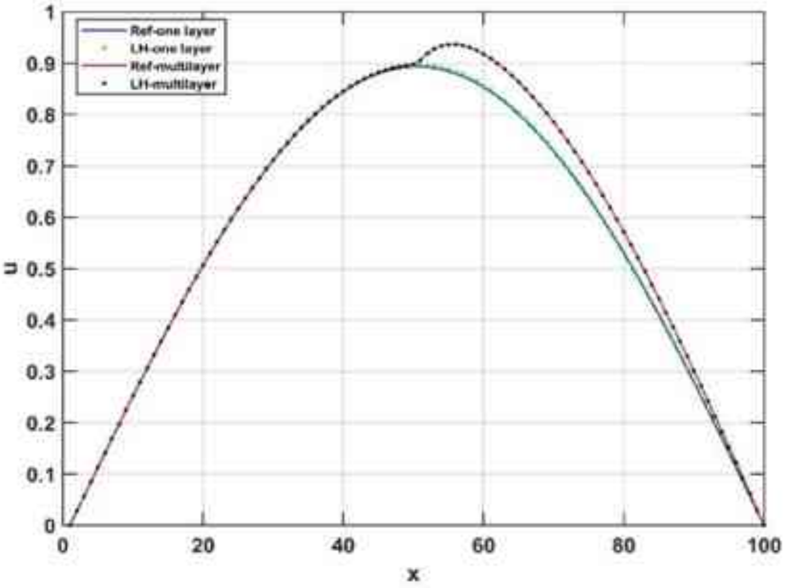

Figure 4.9. The temperature as a function of the cell index in the *x* direction at the middle row $(z \approx 0.5)$ in the case of the reference solution (Ref) and the leapfrog hopscotch (LH) method for $\Delta t = 400$ in the case of the one-layer wall and the insulated wall using an equidistant grid.

One can see that now there are no residual errors. The reason for this is that the reference solution uses the same space discretization as the examined methods, thus this error disappears when the difference of the solutions are calculated as in Eq. (3.1).

I observed that if we apply the insulator or go from equidistant mesh to increasingly non-equidistant meshes (both increase the stiffness), the OOEH method loses its advantage and the LH method will be the most accurate among the unconditionally stable methods, I have other experimental results that were not mentioned in this section but rather were mentioned in the paper [92] with my colleague, which shows that the LH and DF methods were the best among other numerical methods, especially in a realistic case with nontrivial boundary conditions. On the other hand, for very small-time step sizes, Heun's method is extremely accurate. However, this extent of accuracy is redundant in building energetics and in most other fields of engineering. Actually, this is one of the definitions of stiffness: "The step size is dictated by the stability requirements rather than the accuracy requirements" [93]. We stress again that this quoted sentence holds for the mainstream explicit methods, but it is not valid for the unconditionally stable methods, as one can see in the figures.

4.5. Summary of this chapter

All of the numerical methods used are confirmed to be convergent, but their performance is not the same and depends on the circumstances.

The advantages and the disadvantages of the methods are the following:

- The CNe and the UPFD are first order, thus not very accurate, all other methods are second order. Nevertheless, the RRK behaves as a first-order method for large and medium time step sizes.
- In the case of uniform (non-stiff) problems, the OOEH method is the most accurate for large and medium time step sizes. However, if stiffness increases, it can produce larger errors for large time step sizes. On the other hand, the LH always produces acceptable errors, and usually, it is the most accurate for stiff systems.
- Heun's method is only conditionally stable and was divergent for most of the time step sizes used, while all other methods are unconditionally stable.
- The CNe, the UPFD, the LNe2 and LNe3, and the CpC are positivity preserving for arbitrary time step size, all others are not. However, it implies that for medium and small time step sizes they are the least accurate.
- The hopscotch methods (OOEH, ROEH, SH, and LH) need a special bipartite grid. However, they don't require storing another copy of the array for the temperature, even temporarily, so they have minimal memory requirements. Other methods require to store at least one extra array with the same number of elements as the array variable for the temperature.
- The CNe, UPFD, OOEH, ROEH, DF, SH, and LH methods require only one calculation of the new temperature values of any cells in any given time step, so they are the fastest. The LNe2, CpC, Heun, PI, and RRK require two calculations while the LNe3 needs three calculations per cell per time step.
- DF is a two-step method; it needs to be started by another method.

To conclude, we can suggest using the OOEH or maybe the LH method in case of homogeneous material properties and equidistant grid, while in other cases the LH and maybe the SH and the DF algorithms can be proposed. All of them give quite accurate results with orders of magnitude larger time step size thus much faster than the standard explicit methods stricken by instability. However, if unconditional positivity is crucial, the LNe3 method should be used to simulate heat conduction.

5. COMPARISON OF THE PERFORMANCE OF TRADITIONAL AND NEW NUMERICAL METHODS FOR LONG-TERM HEAT TRANSFER SIMULATIONS IN WALLS WITH THERMAL BRIDGES

Some investigations results listed in the previous chapter showed that the leapfrog–hopscotch, the adapted Dufort–Frankel and the OOEH methods are the most efficient among the explicit and stable numerical methods to solve heat transfer problems in building walls [92]. In this chapter, I extensively measure the running times of the most successful methods and compare them to the performance of other available solvers, for example, ANSYS transient thermal analysis and the built-in routines of MATLAB, where three different mesh resolutions are used. I show that the running time of our methods changes linearly with mesh size, unlike in the case of other methods. After that, I make a long-term simulation (one full winter month) of two-dimensional space systems to test the two best versions of the methods. The real-life engineering problem I solve is the examination of thermal bridges with different shapes in buildings to increase energy efficiency.

In the previous chapter, we tested 12 explicit and stable numerical algorithms to simulate heat conduction in building's walls, with and without insulation, using equidistant and non -equidistant meshes. We obtained that the original odd–even hopscotch (OOEH), the leapfrog–hopscotch (LH), and the Dufort–Frankel (DF) are the most effective methods. Then, in [70] we adopted some of the methods to include not only heat conduction but also convection and radiation. According to the results, the LH and the non-standard version of the OOEH are the most accurate if the system is not really stiff. However, the OOEH becomes less accurate when the stiffness is larger, which is the case if the mesh is non-equidistant and/or there are materials in homogeneity. In these cases, the LH takes the lead, but the DF, as well as the shifted-hopscotch (SH) and asymmetric hopscotch (ASH) methods, also perform well. Those methods which are unconditionally stable for the simple conduction case can be used without stability problems with fairly large time step sizes, so they outperform the conventional explicit time integrators. Since usually the LH was the best method, our research group devoted a whole paper [70] on the question of how to implement the convection and radiation term in an optimal way. However, in those papers [70], [92], no running-time measurements were made, and the performance was evaluated only in terms of accuracy versus time step size.

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

In my current work [94], I continue the above-mentioned investigations. I use ANSYS' thermal analysis solutions that help engineers solve the most complex thermal challenges and predict how their designs will perform with temperature changes. However, because simulation by this kind of software takes a long time and requires serious computer resources, I compare my methods with ANSYS to investigate runtime, stability, and other features [95]. Now the goal is to systematically evaluate how the performance of the various solvers (including MATLAB routines and ANSYS) depends on the mesh settings to see which one is optimal for certain accuracy requirements.

Therefore, the work is organized as follows: Section 5.1 presents the examined system with the equations, followed by the numerical methods and implementations of the convection and radiation terms, as well as the methods used for comparison purposes. Subsections 5.2.1, 5.2.2, 5.2.3, display the preliminary conditions for the simulation of the wall: materials, mesh construction, initial, and boundary conditions. In Subsection 5.2.4, I start by verifying my codes in a 2D system for three different cases and comparing the results against simple analytical solutions on a grid that is equidistant. Section 5.3 displays the results of the numerical tests performed to compare differences in errors and running times. Section 5.4 shows the second part of this study, where I chose my best methods to perform a long-term simulation of one month for a real wall in Miskolc city with and without insulation. Two kinds of the thermal bridges are included, and it is calculated how much energy is lost due to the thermal bridges. Section 5.5 shows the third part of this study to make long-term simulation for different shapes of thermal bridges in walls where I also chose my best methods to perform a long-term simulation but for three months and also for a real wall in Miskolc city with three materials, and three types of the thermal bridges, and it is calculated how much energy is lost due to the thermal bridges. Section 5.6 finally concludes with a summary of my results.

5.1. The Studied Cases

5.1.1. The Leapfrog–Hopscotch method

The leapfrog-hopscotch (LH) space-time structure [85] is shown in Subsection 2.2.12. The computation is divided into two half-time phases and numerous full-time steps. Based on the starting values, the calculation begins with a half-sized time step for the odd nodes. A light green half-hexagon in the picture represents this "zeroth" level. Following that, full-sized time steps (shown by light yellow hexagons in the picture) are taken strictly alternately for even and odd nodes until the finish. Finally, for odd nodes to reach the same final time point as even nodes, the last time step (sky blue half-hexagon) should be chopped in half. When a stage calculation is executed for node *i*, the most recently received values of the neighbours $i - 1$ and $i + 1$ must be used.

The leapfrog-hopscotch (LH) formulas are the following:

The ”zeroth” stage equation has the form

$$u_i^{1/2} = \frac{u_i^0 + \frac{A_i^0}{2} - K\Delta t u_i^0 / 4}{1 + \frac{r_i}{2} + \frac{K\Delta t}{4} + \sigma\Delta t \left(u_i^0\right)^3 / 2} \ . \tag{5.1}$$

Then a full time step stages are calculated as

$$u_i^1 = \frac{(1 - \frac{r_i}{2}) u_i^0 + A_i^{1/2} - K\Delta t u_i^0 / 2}{1 + \frac{r_i}{2} + \frac{K\Delta t}{2} + \sigma\Delta t \left(u_i^0\right)^3} \tag{5.2}$$

and this equation is applied at the last stage as well, but with halfed time step size.

5.1.2. *The used mathematical equations*

In this subsection, I give only the formulas applied for Eq. (2.6). Note that the discretization and the formulas for a 1D equidistant mesh can be found in the given references.

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

$$y^{n+1} = y^{n+1} + \Delta t\left[\theta f\left(t^n, y^n\right) + \left(1-\theta\right) f\left(t^{n+1}, y^{n+1}\right)\right] \ , \tag{5.3}$$

where $\theta \in \left[0,1\right]$. If we adapt it for Eq. (2.6), we obtain

$$u_i^{n+1} = u_i^n + \theta\left[A_i^n - r_i u_i^n - \Delta t K u_i^n - \Delta t \sigma\left(u_i^n\right)^4\right] + \left(1-\theta\right)\left[A_i^{n+1} - r_i u_i^{n+1} - \Delta t K u_i^{n+1} - \Delta t \sigma\left(u_i^{n+1}\right)^4\right]. \tag{5.4}$$

The theta method given in Eq. (5.4) is an implicit method if $\theta < 1$. It can be made explicit by the so-called pseudo-implicit trick: the neighbours in the second term at the r. h. s. of (5.4) must be taken at the n-th time level. Furthermore, 3 of the four powers of u_i^{n+1} in the nonlinear term can be replaced by u_i^n. With this, we obtain:

$$u_i^{n+1} = u_i^n + \theta\left[A_i^n - r_i u_i^n - \Delta t K u_i^n - \Delta t \sigma\left(u_i^n\right)^4\right] + \left(1-\theta\right)\left[A_i^n - r_i u_i^{n+1} - \Delta t K u_i^{n+1} - \Delta t \sigma u_i^{n+1}\left(u_i^n\right)^4\right] \tag{5.5}$$

which can be rearranged as

$$u_i^{n+1} = \frac{u_i^n + A_i^n - \theta\left[r_i u_i^n + \Delta t K u_i^n + \Delta t \sigma\left(u_i^n\right)^4\right]}{1 + \left(1-\theta\right)\left[r_i + \Delta t K + \Delta t \sigma\left(u_i^n\right)^3\right]} \ . \tag{5.6}$$

The value $\theta = 1$ yields the explicit Euler method, which has a low CFL limit. If $\theta = 0$ and q is non-negative, Eq. (5.6) preserves the positivity of the temperature for arbitrary time step sizes. In this case, it can be considered as an adaptation of the so-called UPFD (unconditionally positive finite difference) scheme invented a decade ago [76]. Generally, smaller values of θ mean better stability, but it can imply worse accuracy as well. Now we fill the LH structure with this generalized theta-formula as follows.

In the pure conduction case (where $K = 0$ and $\sigma = 0$), the following theta values were obtained [67] during optimization: Stage 0: $\theta = 0$, all other stages: $\theta = \frac{1}{2}$, which will be used everywhere in this work for the term $A_i^n - r_i u_i^n$. However, there is no reason to believe that for the convection and radiation terms, the optimal theta values are the same. In fact, in our last work [70], we analysed the different treatments of these two terms, and according to my experience, only a few versions are competitive. For the convection term, $\theta = \frac{1}{2}$ is always the best choice, since the analytical calculations showed that it preserves second order convergence and unconditional stability at the same time. We currently do not have analytical proofs in the presence of the nonlinear term, but we found that the three theta values are worth examining, namely $\theta = 0, \frac{1}{2}, 1$. We exemplify these by presenting the" zeroth" stage equations as follows.

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

$$u_i^{1/2} = \frac{u_i^0 + A_i^0 / 2 - K\Delta t u_i^0 / 4}{1 + r_i + K\Delta t / 4 + \sigma\Delta t \left(u_i^0\right)^3 / 2} \tag{5.7}$$

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

$$u_i^{1/2} = \frac{u_i^0 + A_i^0 / 2 - K\Delta t u_i^0 / 4 - \sigma\Delta t \left(u_i^0\right)^4 / 2}{1 + r_i + K\Delta t / 4} \tag{5.8}$$

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

$$u_i^{1/2} = \frac{u_i^0 + A_i^0 / 2 - K\Delta t u_i^0 / 4 - \sigma\Delta t \left(u_i^0\right)^4 / 4}{1 + r_i + K\Delta t / 4 + \sigma\Delta t \left(u_i^0\right)^3 / 4} \tag{5.9}$$

5.1.3. Other stable and explicit methods

One well-known example of a system that satisfies the requirements of unconditional stability and explicitness is the Dufort and Frankel (DF) [96] (p 313). Another method must be used to derive u_i^1 from u_i^0. For this, the UPFD calculation technique is used [70] as follows:

$$u_i^1 = \frac{u_i^0 + A_i^0}{1 + r_i + \Delta t K + \Delta t \sigma \left(u_i^0\right)^3} \tag{5.10}$$

Following this first step, the procedure is explained by a simple formula for the heat conduction; however, there are several ways to handling the terms of radiation and convection. Only the most promising treatments are included here.

4. DF-D: only the denominator contains the *K* and Sigma terms:

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i}{1 + r_i + 2\Delta t K + 2\Delta t\sigma\left(u_i^n\right)^3} \tag{5.11}$$

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

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t K u_i^n - \Delta t\sigma\left(u_i^n\right)^4}{1 + r_i + \Delta t K + \Delta t\sigma\left(u_i^n\right)^3} \tag{5.12}$$

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

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t\sigma\left(u_i^n\right)^4}{1 + r_i + 2\Delta t K + \Delta t\sigma\left(u_i^n\right)^3} \tag{5.13}$$

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

$$u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i - \Delta t K u_i^n}{1 + r_i + \Delta t K + 2\Delta t\sigma\left(u_i^n\right)^3} \tag{5.14}$$

8. The original odd-even hop-scotch (OOEH) algorithm:

The equations in use are as follows:

First stage:

$$u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i - \Delta t\sigma\left(u_i^n\right)^4}{1 + \Delta t K} \tag{5.15}$$

Second stage:

$$u_i^{n+1} = \frac{u_i^n + A_i^{\text{new}}}{1 + r_i + \Delta t K + \Delta t\sigma\left(u_i^n\right)^3} \tag{5.16}$$

9. NS-OOEH algorithm:

In addition, we try to alter the first (Explicit Euler) stage of the OOEH to make the terms for convection and radiation appear in the denominator to improve stability. The first stage, instead of (5.15), is as follows:

$$u_i^{n+1} = \frac{(1-r_i)u_i^n + A_i}{1 + \Delta t K + \Delta t\sigma(u_i^n)^3} \tag{5.17}$$

Equation (5.16) is the same for the second stage.

5.1.4. Professional solvers tools used for comparison purposes

The results are compared with the iterative, direct, and programmable solvers provided in student version of ANSYS 2023, R1 Academic software, based on finite element method (FEM).

(1) The direct solver provides the exact solution to the system of equations that defines the finite element (FE) model. The exact solution for the system denoted by equation $[K]\cdot\vec{x}=\vec{b}$ is $\vec{x}=[K]^{-1}\cdot\vec{b}$, where $[K]^{-1}$ is the inverse of the matrix $[K]$. Since calculating the inverse of a matrix is computationally expensive, direct solvers typically use LU decomposition to solve the equation instead of calculating the inverse;

(2) Iterative solvers assume an initial solution and iterate until convergence is achieved to provide an approximate solution within a certain convergence tolerance. So, when the convergence tolerance is set at 0.01%, the solver will iterate until the difference between the current past and current estimates of the solution is below 0.01%.

For complex systems and low-quality grids, direct methods are typically more reliable and robust, although they do demand a relatively significant amount of memory. Because they require significantly less memory, iterative solvers are generally more effective. However, with a well-designed model, they can fail to converge. Thus, the choice of solver type should be guided by the requirements and capabilities. It is optimal to solve the model using an iterative solver, hence that is the preferred method. If there is an trouble with convergence, using a direct solution is recommended. The computer will have to employ an iterative solution if model is very big and its RAM is limited.

(3) The programmable solver uses a combination of direct and iterative solutions and is controlled by a software.

MATLAB solutions have been employed in order to make comparisons, namely ode23, ode23t, ode15s, ode45, ode23tb, and ode113. It is known that odes 23, 45, and 113 use explicit methods, whereas the remaining odes use implicit solutions. Since the MATLAB solvers do not allow for the explicit calculation of time step sizes, the tolerances are defined instead, starting at a big value ($\mathrm{Tol}=10^{-1}$) until an extremely small value ($\mathrm{Tol}=10^{-12}$). There is another solution that is observed, called ode23s, also available in MATLAB as well; however, it is excluded because it proved its extreme slowness for these problems in our earlier work [60], [68]. All the running times are measured and tested using MATLAB R2020b software on a desktop computer equipped with an Intel Core i7-11700F (16 CPUs) and 64 GB of RAM. The program's integrated tic-toc timer is used to record the length of time the algorithms have been running. Table 5.1 lists the names and provides a brief explanation of the techniques employed.

Table 5.1. Names and short descriptions of the methods used.

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

5.2. Numerical Simulation

5.2.1. Geometry and material properties

As shown in Figure 5.1, a wall piece with dimensions of 1 m in the *x* and *z* directions and 0.02 m in the *y* direction is taken into consideration. Two geometries are considered:

(A) Only one brick layer is analysed in case of verification;

(B) Brick and rigid polyurethane foam insulation are layered in two layers and contain a straight thermal bridge from the steel beam in case of measuring running times.

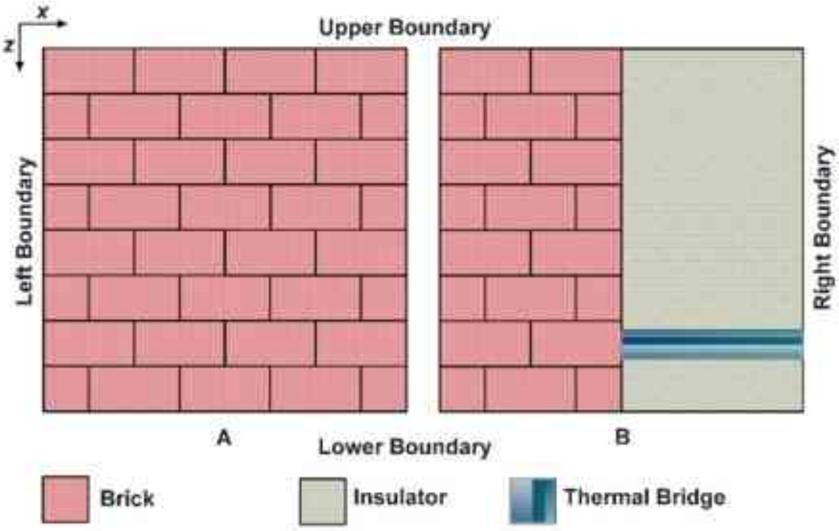

Figure 5.1. (A) One-layer wall. (B) Two-layers wall with brick, insulation and thermal bridge.

Table 5.2 lists the real material properties that are considered in the current investigation. The coefficients exhibit discontinuities at the material's border but are constants inside the material, meaning they are unaffected by changes in time, space, or temperature.

Table 5.2. The materials properties used.

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

5.2.2. *Mesh construction*

When performing the calculations in Section 5.3, a wall dimension of 1 m × 1 m × 0.02 m is taken into account. The approximation that no physical quantities change in that *y*-direction is used, where the *y*-direction is orthogonal to the surface of Figure 5.1A, B, and thus this dimension is irrelevant from a mathematical view. This indicates that $\Delta y_i = 0.02$ m can be used and only have to deal with the cross-section, which is a two-dimensional problem. So, three different sizes of 1 m^2 meshes are created, which means $(x,z)\in[0,1]\times[0,1]$. As seen in Figure 5.2, the cells are square or rectangular in shape.

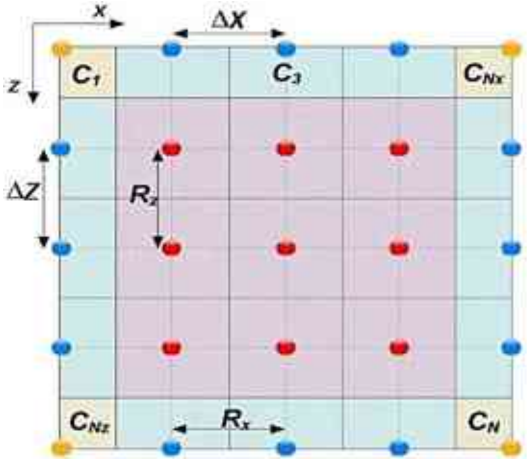

Figure 5.2. The arrangement and visualization of the generalized variables.

In all cases, the space variables is discretized using an equidistant grid. Three different uniform grids were used for the computations: coarse, which was 40 × 40 = 1600; medium, which was 80 × 80 = 6400; and fine, which was 120 × 120 = 14,400. It is important to remember that the temperature at the nodes is determined to be comparable to the finite element method-based ANSYS program, which is why we have three grids. So, these three meshes are combined to form a grid. In the case of a homogeneous material, the heat capacity of the cells surrounding the nodes that are not at the boundary can be expressed as $C_i = c_i\rho_i\Delta x_i\Delta z_i\Delta y_i$. It can be expressed as follows if the material properties are different $C_i = c_i\rho_i\frac{\Delta x\Delta z\Delta y}{2} + c_{i+1}\rho_{i+1}\frac{\Delta x\Delta z\Delta y}{2}$. It is calculated by multiplying the heat capacity by a factor of 1/2 at the lower, upper, right, and left borders, and by a factor of 1/4 at the corners.

In contrast, the approximate equation for the thermal resistance in the *x*-direction between two nodes is $Rx_i \approx \frac{\Delta x}{k_i Sx_i}$, where Sx_i is the surface element orthogonal to *x* and can be expressed as $Sx = \Delta y\Delta z$ in the event that the nodes are not at the boundary. The following equation can be used to calculate resistances in parallel: $\frac{1}{Rx} = \frac{1}{Rx_i} + \frac{1}{Rx_{i+1}} = \frac{k_i S_i}{\Delta x} + \frac{k_{i+1}S_{i+1}}{\Delta x} = \frac{k_i + k_{i+1}}{2}\frac{S}{\Delta x}$.

For a homogeneous material, the horizontal and vertical resistances can be computed as

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

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

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

respectively. The resistance between two nodes, *i* and *i* + 1, can be expressed as follows if the material properties are different:

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

Additionally, since node *i* + *Nx* is situated below node *i*, so

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

for the vertical resistance.

5.2.3. Initial and boundary conditions

In the case of running-time measurements, zero Neumann boundary conditions have been applied to all borders in order to prevent conductive heat from passing through them. To achieve this, the relevant resistances are set to infinity and set zero for the matrix components that describe heat conduction across the border.

Convective and radiative heat transfer have been assumed on the wall's left and right sides. There is no heat source in those parts, heat cannot be lost or gained by radiation or convection in the interior components. The final time $t_{\text{fin}} = 20,000\,\text{s}$ indicates the end of the time interval that is examined; additionally, the time step size is specified in seconds. Table 5.3 shows that elements on the left and right sides may transfer heat through radiation and convection in the *x* direction.

Table 5.3. In the case of a two-layer wall, the heat source, radiation, and convection parameters on both sides of the wall components.

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

As can be seen in Table 5.3, I have used numbers from [97] for the convection heat transfer coefficient h_c. The Stefan–Boltzmann constant is a universal constant for radiation: $5.67 \cdot 10^{-8} \frac{\text{W}}{\text{m}^2\text{K}^4}$. Since the surface is not a perfect black body, we have to multiply the Stefan-Boltzmann constant by an emissivity constant in order to get results that are more in line with realistic values for σ^*. For the definition of "heat source," which includes solar radiation, the value of q^* is an estimate. The left side's ambient air temperature is assumed to be $17\ {}^\circ\text{C} \approx 290\ \text{K}$.

Because of the nonzero air temperature u_a, convectional heat gain is included in the term *q*, this allows us to compute the value of *q* as following. The values that are obtained are used to calculate the values of the coefficients in Eq. (1.10):

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

The following heat sources were assumed to be present in the left and right elements:

On the left side: $q = \frac{1}{c\rho\frac{\Delta x}{2}} \times 360.95 \frac{\mathrm{W}}{\mathrm{m}^2} + \frac{h_c}{c\rho\frac{\Delta x}{2}} \times 290\ \mathrm{K}$

On the right side: $q = \frac{1}{c\rho\frac{\Delta x}{2}} \times 435.39 \frac{\mathrm{W}}{\mathrm{m}^2} + \frac{h_c}{c\rho\frac{\Delta x}{2}} \times 313\ \mathrm{K}$

The constant initial temperature is defined as: $u\left(x,z,t=0\right) = 290\ \mathrm{K}$

5.2.4. Verification by analytical solutions

For verification, three different cases are used. In the first case, a one-layer wall composed of a homogeneous material is considered, as shown in Figure 5.1 A, and a sinusoidal initial condition is applied with a zero Dirichlet boundary condition to investigate the conduction term only.

Then a spatially homogeneous initial temperature is considered in the second and third cases to exclude conduction, resulting in a system with only one temperature. This condition allows us to describe the behavior using an Ordinary Differential Equation (ODE). In the second case, only convection is taken into account, which is orthogonal to the surface and occurs in the y direction. In the third case, only radiation that is perpendicular to that surface is considered. After that, the results in MATLAB and ANSYS are compared with the analytical solution.

a) First verification (only conduction):

- Two sine functions are multiplied to provide the initial condition:

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

- Zero Dirichlet boundary conditions are used:

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

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

$$u(x,z,t) = \sin(k_x \pi x)\sin(k_z \pi y)\,\mathrm{e}^{-\alpha(k_x{}^2+k_z{}^2)\pi^2 t} \tag{5.18}$$

where the wave numbers are fixed to $k_x = 1,\ k_z = 1$, and substituting the physical properties of the brick, we obtained *α*. We used the analytical solution Eq. (5.18) at t_{fin}=2000s as the reference solution.

b) Second verification (for convection and heat generation)

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

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

Its analytical solution is:

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

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

c) Third verification (for radiation)

We have the ODE:

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

with the analytical solution:

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

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

For all systems, the obtained results in the case of MATLAB and ANSYS are very similar to the analytical solution, and the error is below 10^{-5}. This means that the MATLAB code using the ode15s solver, as well as the ANSYS solver, has been verified. The temperature as a function of the *x*-coordinate for *z* = 0.5 m was plotted in the case of ode15s, ANSYS, and the analytical solution for the coarse grid. These lines are so close to one another that they are indistinguishable by the naked eye, as shown in Figure 5.3. For other system sizes, the same behaviour is experienced. In the case of the second and third verification, the maximum error is around 10^{-9} between ode15s and the analytical solution, but it is slightly larger than 10^{-5} between ANSYS and the analytical solution, as shown in Table 5.4. Since the MATLAB ode15s solver is proven to be more accurate in all cases, I choose this as a reference solution when I measure and compare the running times in the next section.

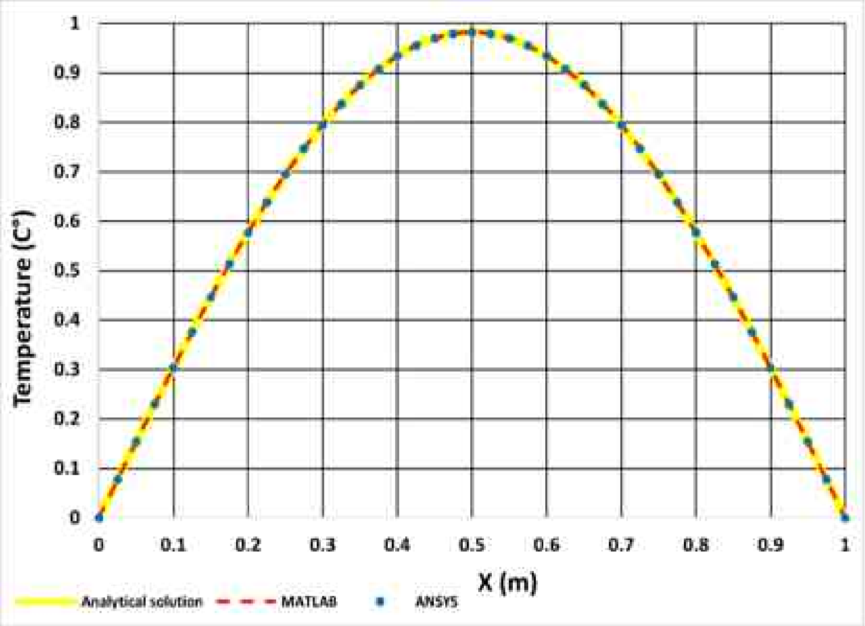

Figure 5.3. The temperature comparison for the sinusoidal initial temperature in the case of the coarse mesh.

Table 5.4. The comparison between MATLAB and ANSYS at different mesh sizes and boundary conditions.

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

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

5.3.1. Comparison with MATLAB methods and ANSYS solvers for the coarse mesh system

The coarse mesh, conditions in Subsection 5.2.3, and values from Table 5.2 are used to simulate the two-layer wall with straight thermal bridge as shown in Figure 5.1 B. Figure 5.4 displays the maximum errors as a function of time step size. Figure 5.5 compares the running times of the three ANSYS solvers with the tested methods, and Figure 5.6 displays the temperature distributions. It is clear that the LH pseudo-implicit treatment of radiation was the most accurate for the coarse mesh; however, two of the DF versions and the three ANSYS solvers are not really accurate. In terms of speed, it is clear that the explicit and stable methods are faster than ANSYS. However, if small or moderate accuracy is needed, they are more effective than the MATLAB routines.

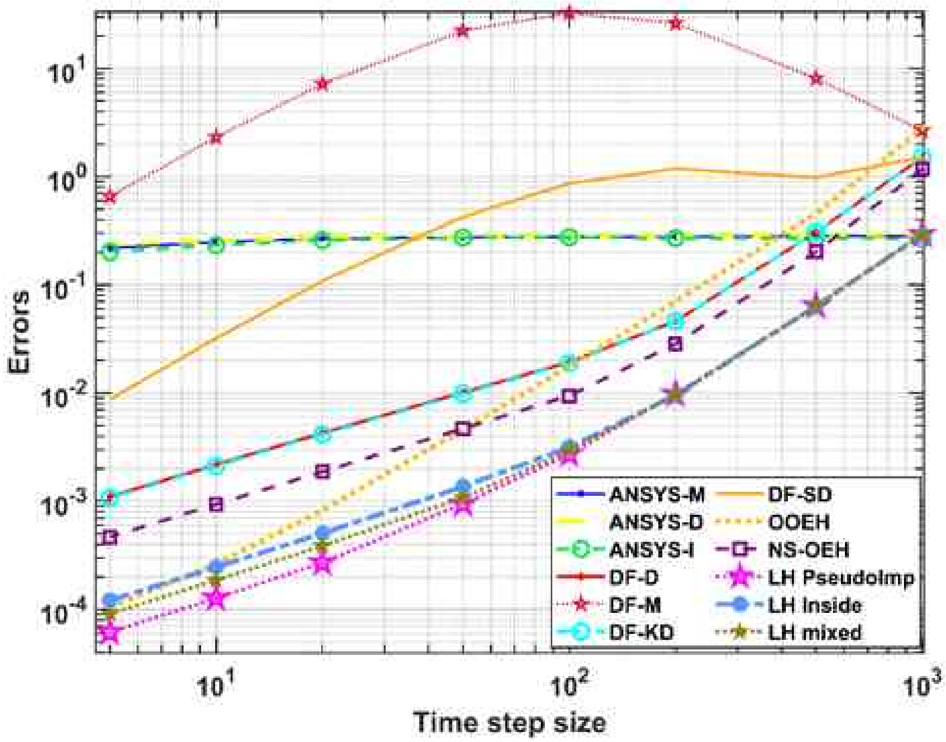

Figure 5.4. The maximum errors for the tested methods in the coarse mesh case as a function of the time step size.

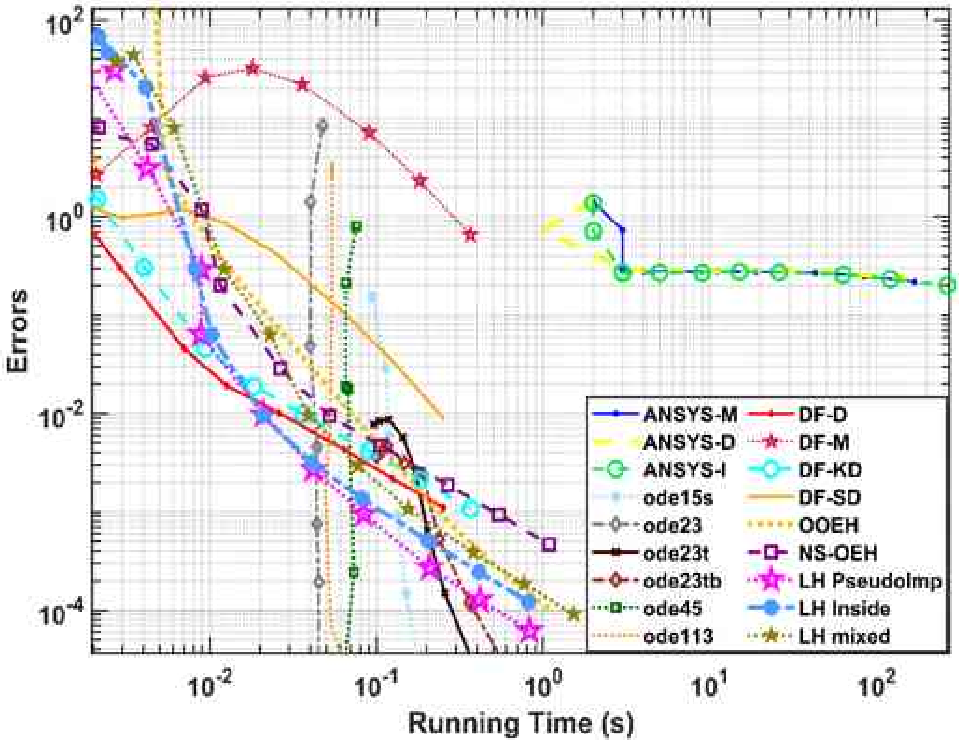

Figure 5.5. The maximum errors for the tested methods in the coarse mesh case as a function of the running time.

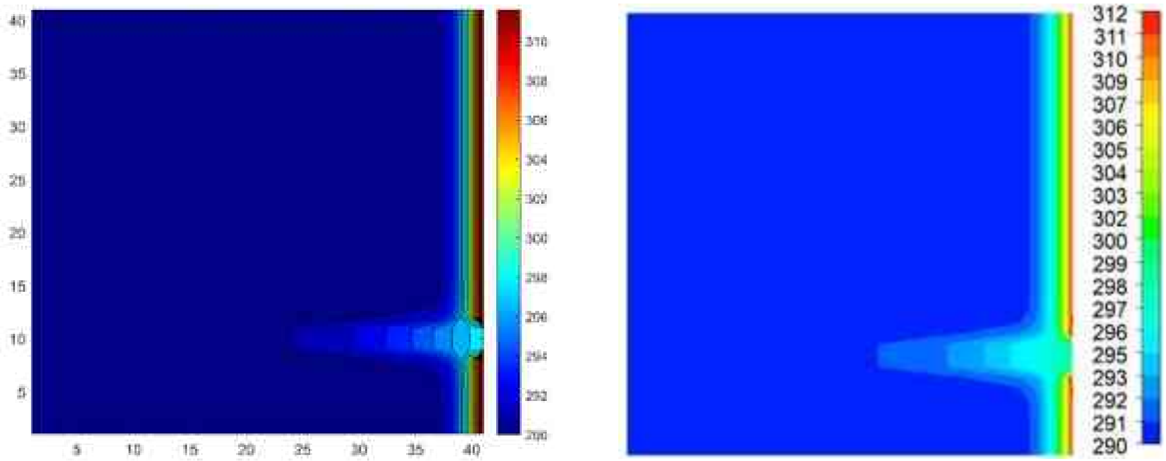

Figure 5.6. The temperature distribution contour in Kelvin unit for the coarse mesh in case of the running time measurements (left) for MATLAB and ANSYS on (right).

5.3.2. *Comparison with MATLAB methods and ANSYS solvers for the moderate mesh system*

The same conditions and values used for the previous coarse grid are applied, this time to a moderately fine resolution mesh. Again, most of the suggested methods, especially the LH-PI is better than MATLAB routines and ANSYS solvers for both accuracy and speed. Figure 5.7 compares the running times of the three ANSYS solvers with the tested methods in MATLAB.

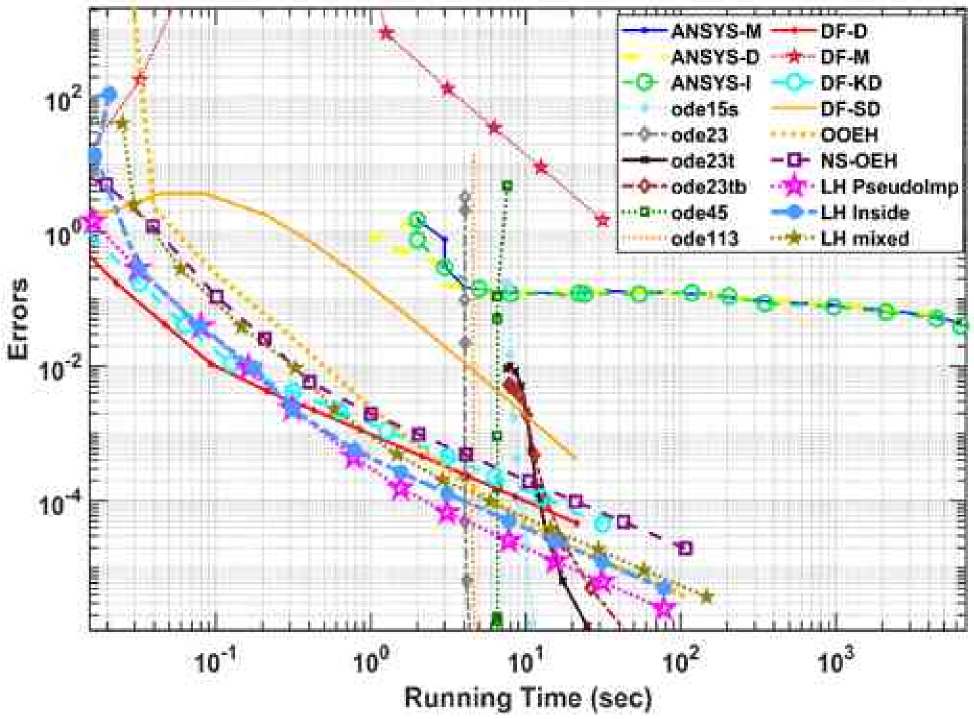

Figure 5.7. The maximum errors for the tested methods in the medium mesh case as a function of the running time.

5.3.3. Comparing ANSYS solvers and MATLAB methods for the fine mesh system

The errors for the fine mesh are represented as a function of the running time in Figure 5.8. It is observed that the mesh smoothing has a positive effect on the ANSYS solvers' accuracy; the errors are decreasing with the time step size. Despite this, some of the suggested explicit methods coded in MATLAB are still the best in both speed and accuracy. If there is a large system, the ANSYS solvers are very slow. However, it is believed that with increasing the system size, the MATLAB built-in routines would be slower at a larger rate, so it is also believed they would be the slowest for even finer mesh. The two best methods are the Dufort-Frankel schemes with the pseudo-implicit treatment of both the convection and the radiation term (DF-D) and the leapfrog-hopscotch with the pseudo-implicit treatment of the radiation term (LH-PseudoImp). In fact, their benefits grow as the size of the system increases. For this reason, these two methods are chosen to create a long-term simulation in the following part.

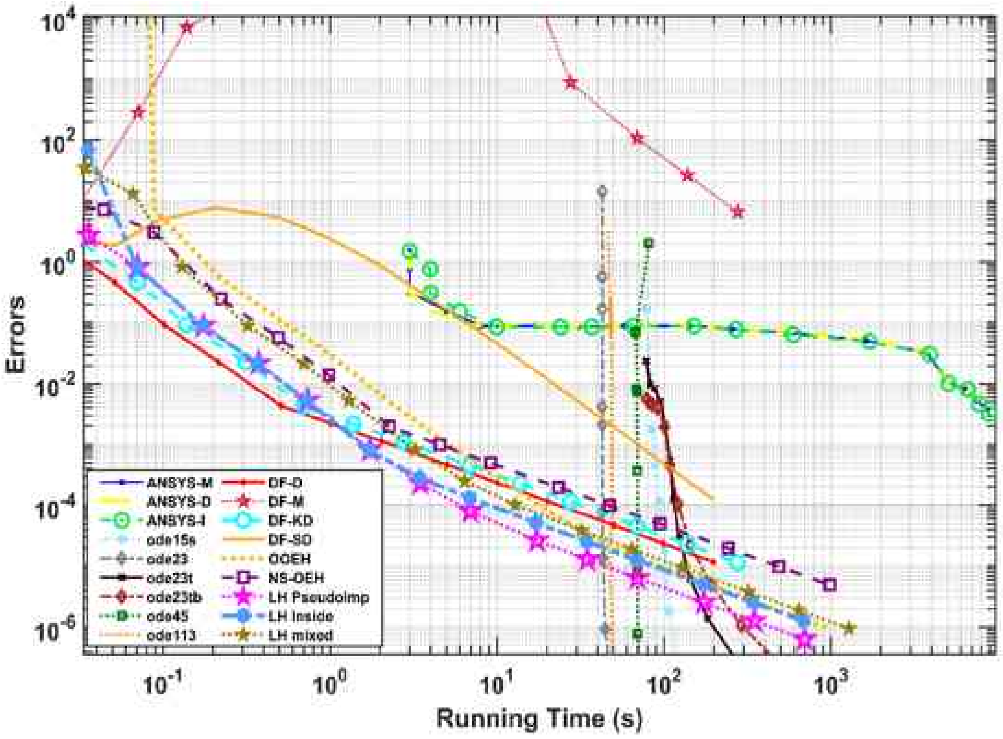

Figure 5.8. The maximum errors for the tested methods in the fine mesh case as a function of the running time.

The temperature of the LH pseudoImp method is plotted for all three meshes along a horizontal line that is taken into consideration for mesh-independence at the middle of the thermal bridge (z = 0.225 m). The medium mesh is adapted in long-term calculations to reduce the computational cost because, as can be shown in Figure 5.9, the results are nearly identical to those of the fine mesh, with the maximum difference being only 0.097 in Kelvin units.

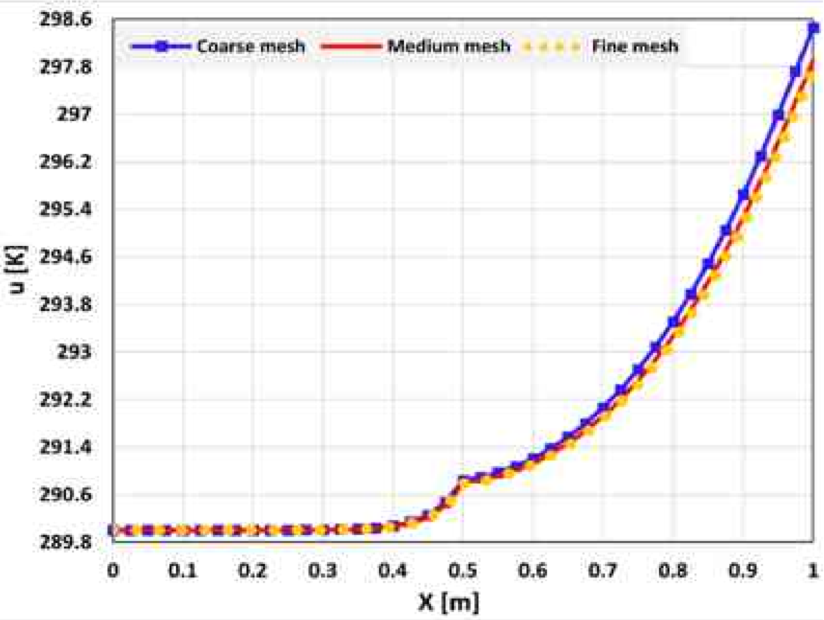

Figure 5.9. Comparison of the three mesh types used for the temperature in the LH-Pseudo-Imp method.

Additionally, a comparison is conducted between three solvers using the medium mesh: iterative ANSYS, the LH-PseudoImp, and ode15s. While ANSYS differs somewhat from LH-Pseudo-Imp and ode15s by 0.2 K, the maximum difference between ode15s and LH-Pseudo-Imp is equal to 7.3×10^{-7} in Kelvin units as shown in Figure 5.10.

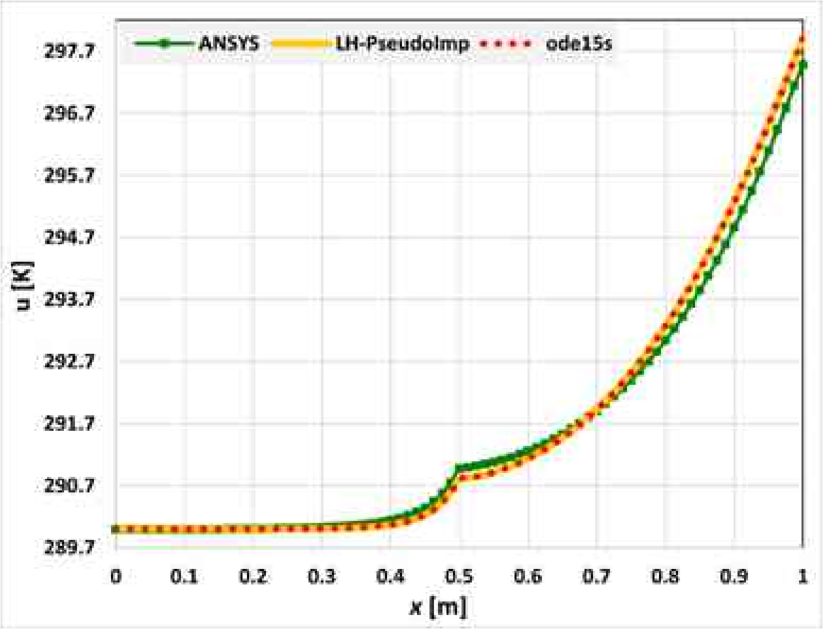

Figure 5.10. Three solvers for the medium mesh are compared.

Finally, how the running times depend on the system size is analysed. It is obtained that the running time of the methods increases linearly with the number of nodes, but it increases more quickly than linearly for other solvers. The actual data for one of the best methods with one of the ANSYS solvers and MATLAB routines ode23t are displayed in Figure 5.11. However, since there are quicker numerical techniques than those used by ANSYS and similar solvers, it is not intended to imply that they are redundant. If someone wants to build their own code, it is tedious work, particularly for complex geometries. In contrast, this simulation program generates the mesh automatically and shows the results graphically. Furthermore, even though simulation software frequently addresses issues like air leakage in building components and routinely handles it, explicit and stable approaches have not been applied to cases requiring the simulation of air motion. However, in some well-established cases, such as those types studied in this work, it is believed that the simulation software is able to incorporate the suggested algorithms as a choice that could be selected by users or an artificial intelligence.

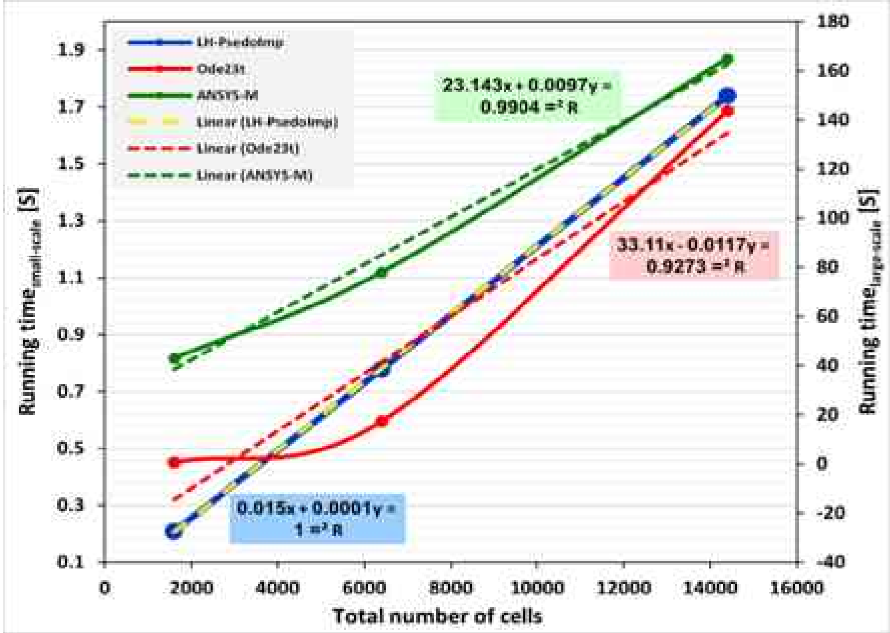

Figure 5.11. Running time for the tested method and other solvers as a function of the cells total number, where the right axis refers to ANSYS and ode23t solver of MATLAB, and the left one refers to LH-Pseudo-Imp.

5.4. *Long-Term Simulations*

5.4.1. Geometry and mesh

The simulations were carried out for a standard two-layers residential wall with different shapes of thermal bridges in Miskolc, Hungary, and they were compared with the wall without a thermal bridge. The data about the convection coefficient, the external temperature, and the solar radiation values for this city, which is used in the simulation, is taken from the website page every 3 hours. Then, the data for every 100 seconds are calculated using a linear interpolation. January is the coldest month. The greatest and lowest recorded temperatures in the comparatively mild winter of 2022–2023 were 11°C and -6°C, respectively. The predominant wind may blow in several directions depending on the topography. The maximum recorded wind speed was 10.8 m/s [98].

The LH-PI and DF-D best methods are applied to determine temperatures and heat loss across the wall. For all transient simulations, with a set and a total calculation period of

T=31×24×3600=2,678,400 s, representing the January month. The total value of time steps was *Time* = 26784, where the time step size is $\Delta t = 100$ s. This time step size was chosen because, from Figure 5.4, where the DF and LH numerical approaches were validated, it was demonstrated that they produce errors around 0.01 °C, satisfying the need for accuracy. It can be emphasized again that the explicit Runge-Kutta method CFL limit is significantly less than this time step size. The results are displayed for the LH method since the variation between the DF and LH was regularly checked during the simulations and found to be very small. The case of two layers with the straight thermal bridge case on the same computer used in the previous section had a running time of 145.5 s for the long-time simulation.

Figure 5.12 displays the different wall models. First, one layer of brick is used, as displayed in Figure 5.12A. The dimensions of the full wall thicknesses, height, and width are (0.45m, 1m, 1 m) in the *x*, *y*, and *z* axis directions, respectively. The second model consists of two layers: as Figure 5.12B illustrates, the brick wall has the same dimensions as the first model, and there is an insulating layer with a thickness of 0.15 m. The third model is identical to the second one but with a straight thermal bridge. As seen in Figure 5.12C, the width of the thermal bridge is the same as that of the insulator. The thermal bridge is positioned horizontally between *x* = 0.45 m and *x* = 0.6 m. The thermal bridge's thickness is 5 cm in the *z*-direction, and the vertical position of the top of it is *z* = 0.75 m, or 25 cm from the wall's base.

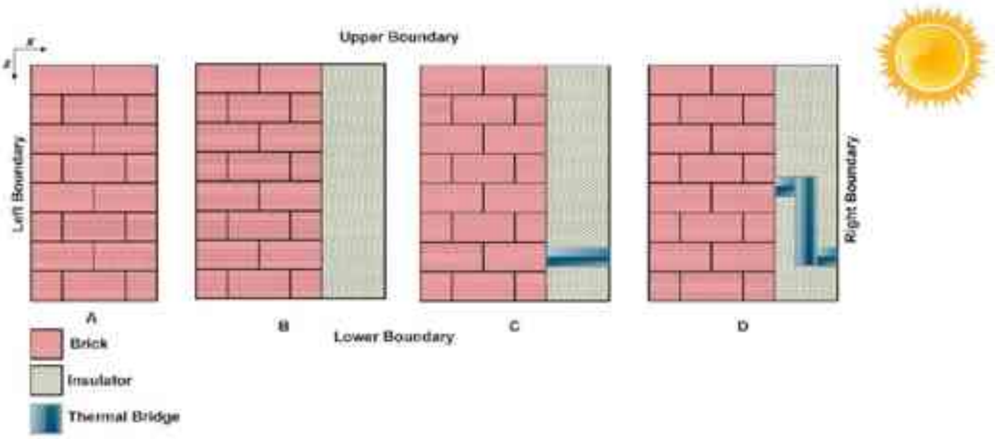

Figure 5.12. The wall types are as follows: (A) One-layer; (B) Two-layers with brick and insulator; (C) Two-layers with straight thermal bridge; (D) Two-layers with bent thermal bridge.

The fourth model contains the same two layers as the second one, but the bar thermal bridge has a curved shape. There are three straight bars in the bending thermal bridge, one of them vertical and the others horizontal. The external side connects to the first horizontal straight bar at *z* = 75 cm (25 cm from the bottom of the right side of the wall), which is going horizontally with a 0.05 m length. It is connected to the vertical bar, with a length of 0.3375 m in the *z*-direction. Then, as illustrated in Figure 5.12D, it is connected to the other horizontal bar, which has the same dimensions as the first horizontal bar, whereas the vertical position of the piece's bottom is 0.5125 m. All cases use the medium mesh; however, in the first, it means $\Delta x = 0.0057$, while in the other cases, the widths of the cells are larger, $\Delta x = 0.0076$, since the wall is thicker.

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

5.4.2. Initial and boundary conditions

The conditions are taken as in Subsection 5.2.3. The convection, radiation, and heat source parameters for left elements are constant, whereas the values for right elements change based on the surrounding external conditions as shown in Table 5.5. In this section, the final time is t_{fin} = 2678400 s, which refers to the simulation of a whole winter month.

Table 5.5. Radiation, convection, and heat source variables are on both sides of all types of walls [97].

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

The following heat sources were also assumed to be present in the left and right components:

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

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

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

where $q_l^* = \varepsilon_l \sigma_l (295)^4$ and $q_r^*(t) = \alpha_{\text{sun}} G_{cr}(t) + \alpha_{\text{Low}} \varepsilon_r \sigma_r \left[u_r(t)\right]^4$ [99].

The convective heat coefficient for external components as a function of air velocity can be calculated using the formula below [100]:

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

$v(t)$: The air velocity is measured every 100 s in January month [m/s].

$u_r(t)$: The outdoor air temperature in January for every 100 s [°C].

u_l : The inside air temperature on the left side [°C].

$G_{cr}(t)$: The solar radiation is taken in January for every 100 s [W/m^2].

α_{sun} : Surface absorptivity to solar radiation, which is 0.95, according to [71].

α_{Low} : Surface absorptivity of the painted surface to low-temperature thermal radiation, which is 0.93, according to [71].

The environment air temperature is taken to be $22\ ^\circ\text{C} \approx 295\ \text{K}$ inside and changing depending on the weather conditions outside.

I calculated the initial temperature inside the wall using the assumption that, before the simulation time, at midnight on December 31st a stationary heat flow with constant flux evolved between the given boundary values of the internal and external air temperatures.

In the case of one-layer, it yields a linear function of the *x* variable for the initial condition:

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

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

For the other three cases, the stationary heat conduction assumption with the initial values at the boundaries implies that two linear functions of the *x* variable must be used for the initial condition:

In the case of the brick part: $u\left(x,z,t=0\right)=(u_{\mathrm{mid}}-u_l)\cdot x/L_b+u_l$

In the case of the insulator part:

$u\left(x,z,t=0\right)=((u_{\mathrm{r,initial}}-u_{\mathrm{mid}})x/L_{\mathrm{ins}})-((u_{\mathrm{r,initial}}-u_{\mathrm{mid}})\cdot L_b/L_{ins})+u_{\mathrm{mid}}$

where $u_{\mathrm{mid}}=u_l-(q_{\mathrm{flux}}L_b/k_b)$

and $q_{\mathrm{flux}}=(u_l-u_{\mathrm{r,initial}})/((L_b/k_b)+(L_{\mathrm{ins}}/k_{\mathrm{ins}}))$

5.4.3. Result of the wall simulation

The results are displayed for the specific points shown in Figure 5.13, and the following information is contained in the subfigures.

(A) For the one-layer wall, three points are indicated: on the inside surface, in the centre of the wall, and on the outside surface;

(B) Two-layer wall consisting of brick and insulation, with five points on the inside surface, the centre of the brick, the boundary between the two layers, the middle of the insulation, and the outside surface;

(C) Two-layer with a straight thermal bridge with six-points: on the inside surface, in the centre of the brick, the boundary between the two layers, the outside surface of the thermal bridge, the centre of the insulation, and the outside surface of the insulation;

(D) Two-layers with the bent thermal bridge with six points: on the inside surface, in the centre of the brick, the boundary of the two layers with bent thermal bridge, the outside surface of the bent thermal bridge, the centre of the insulation, and the outside surface of the insulation.

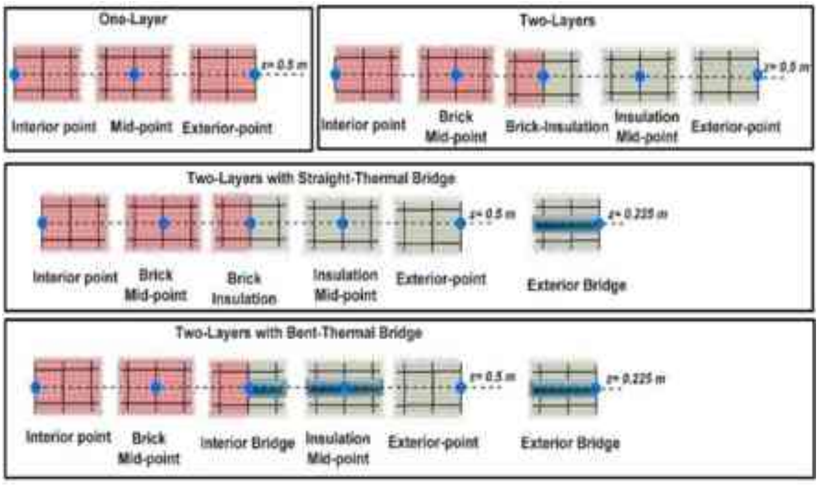

Figure 5.13. Illustration of the grid arrangements and points that show the results of the different cases.

- *One-layer-only brick*

Figure 5.14 displays the temperatures as a function of time. One can see that in the case of a wall without insulation, the inner surface temperature of the wall is initially equal to the temperature of the interior air (22 °C), then decreases due to the cooling effect of the wall and the cold weather outside. The middle of the wall has a low initial temperature that rises due to heat transfer from the inside by conduction, and because heat is flowing to the external surface by conduction, the temperature of the exterior surface is higher than that of the exterior air.

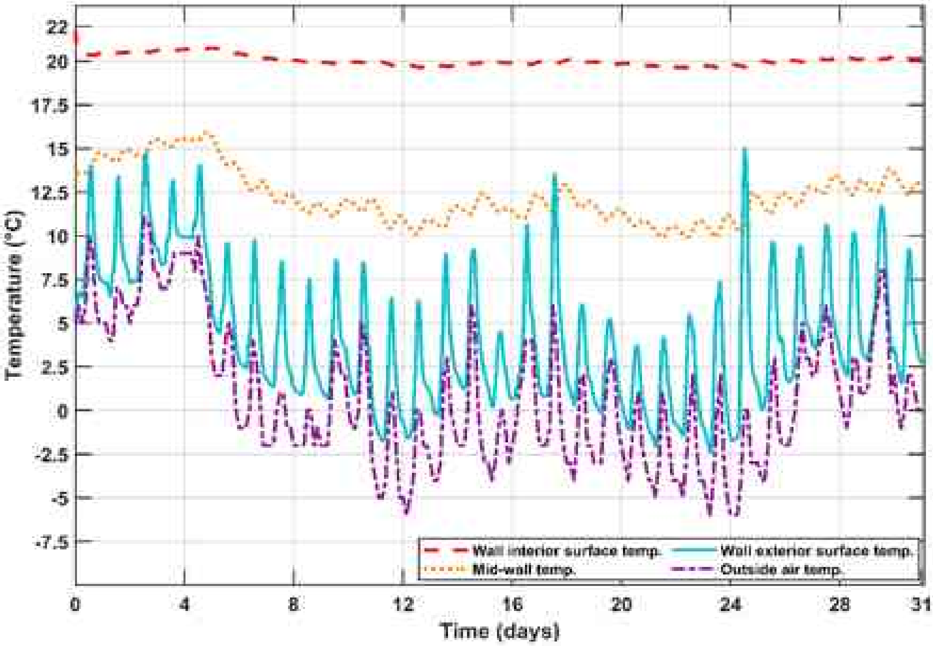

Figure 5.14. The distribution of temperature in °C units as a function of time in days for the long-term simulation of the one-layer case.

- *Brick and insulation in two layers without a thermal bridge*

The temperatures as a function of time are presented in Figure 5.15. One can see that in the case of the presence of insulation, the temperature of any brick point follows the outside air temperature only very slightly. In terms of the temperature of the middle of the insulator, it falls and rises due to the effects of outside conditions, and the minimum values of the external surface temperatures of the wall are slightly less than the outside air temperature because the insulation limits the flow of heat from inside to outside; for the higher values, it is larger owing to the effects of solar radiation. The comparison between the one-layer and two-layer cases of the final temperature is shown in Figure 5.16. The temperature in the insulator part is changing very sharply, but in the brick part, it is changing very slowly.

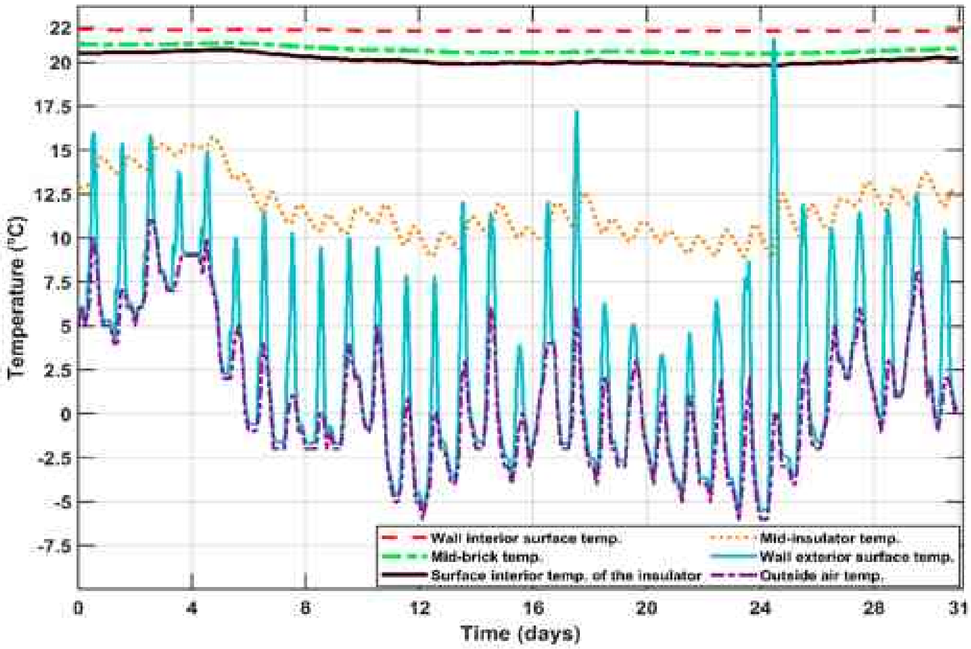

Figure 5.15. The distribution of temperature in °C units as a function of time in days for the long-term simulation of the two-layer case.

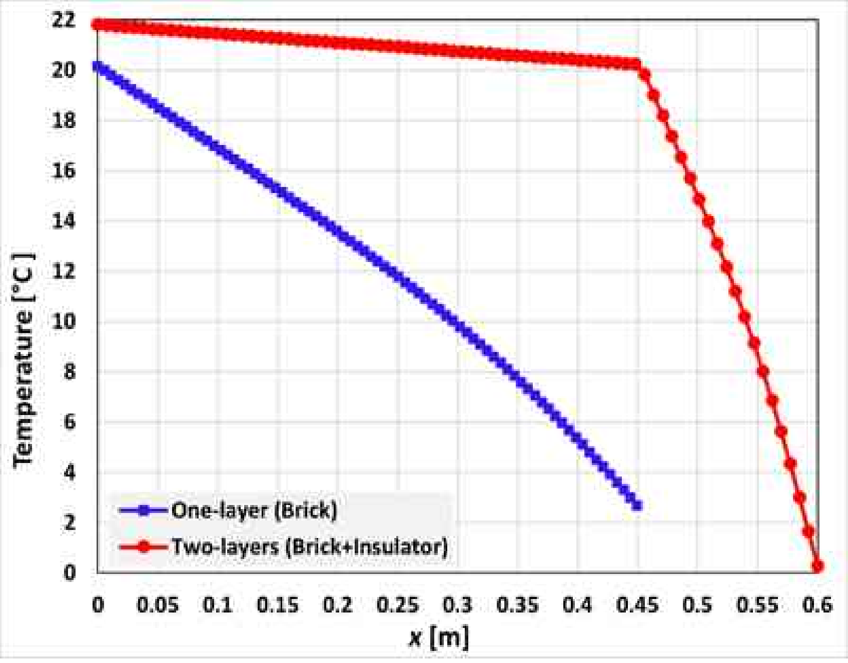

Figure 5.16. The comparison between the one-layer and two-layer wall cases in case of the final temperature at the end of the last day.

- *Two Layers with Straight and Bent Thermal Bridges*

One can see from Figure 5.17 that the temperature of the middle of the brick and the temperature of the internal surface of the insulator decreased slightly compared to the previous case due to the heat loss resulting from the straight thermal bridge. Furthermore, because of the physical properties of the material as well as the flow of heat from inside to outside, the outside temperature of the thermal bridge is higher than the temperature of the outer surface of the insulator. In the case of a bent thermal bridge, the mid-insulator temperature increases due to the bent thermal bridge. The contours of the two types of thermal bridge cases are illustrated in Figure 5.18, where it is noted that the path of the heat in the straight bridge is shorter than in the bent bridge.

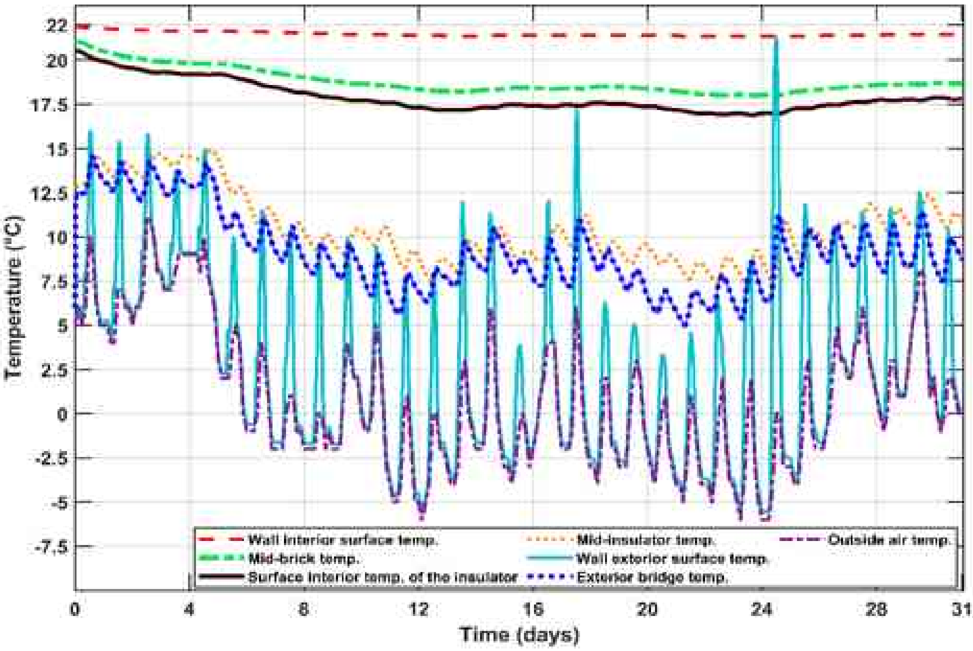

Figure 5.17. The distribution of temperature in °C units as a function of time in days for the long-term simulation of the two-layer with straight thermal bridge.

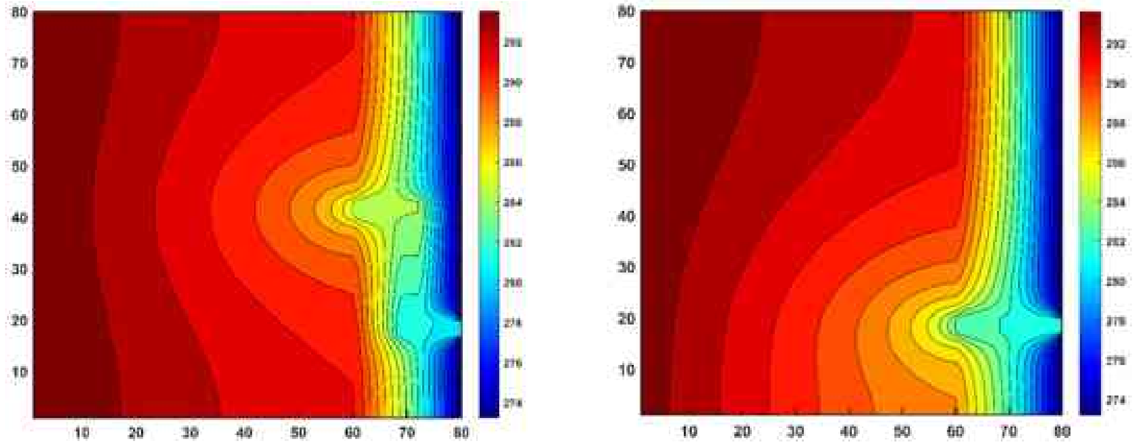

Figure 5.18. The temperature distributions contour in Kelvin units in the case of the medium mesh for the long-term simulations (left) for the bent thermal bridge and (right) for the straight thermal bridge at the end of the last day.

In Figure 5.19, the temperatures of the outer points are compared for both thermal bridges against those of the one- and two-layer cases. It is concluded that the straight thermal bridge provides a quicker flow of heat compared to the bent thermal bridge, as the path for heat transfer is shorter in the straight bridge than in the bent one.

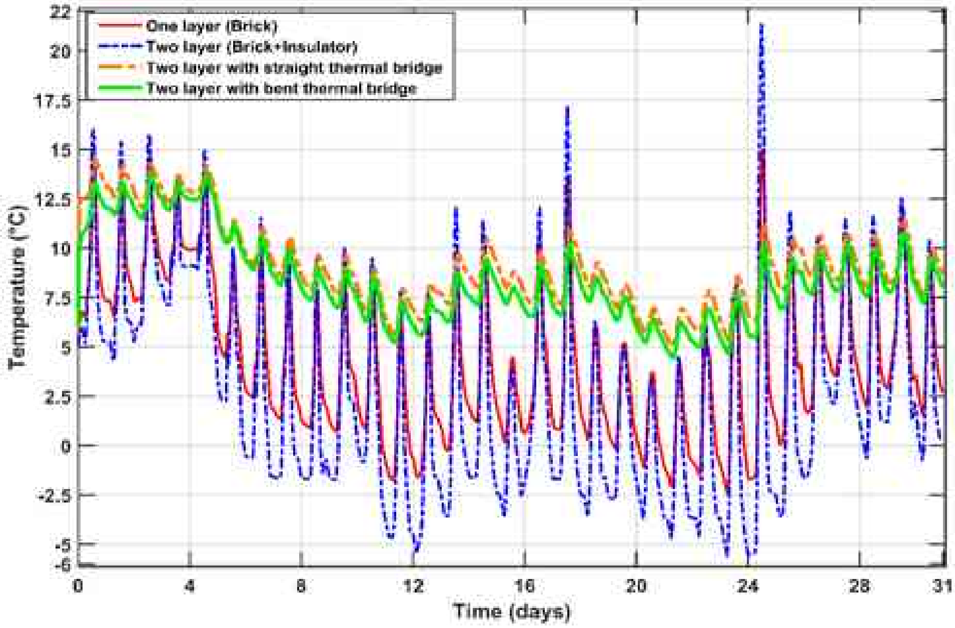

Figure 5.19. The distribution of temperature in °C units as a function of time in days for the long-term simulation for the outside points of straight and bent thermal bridges compared to one-layer and two-layer walls.

- *Heat Loss Calculations*

The total energy requirement Q, measured in kWh units is calculated from Equation (5.19)

$$Q = \sum_{t=1}^{Time} \left(\Delta t \sum_{i=1}^{N} Q_{\text{cell},i}(t) \right) \tag{5.19}$$

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

Now, the cost of the energy used to compensate for the energy loss is calculated by multiplying the quantity of heat losses by the current electricity price used (0.1 USD per kWh). The costs of energy consumption and heat loss are displayed in Table 5.6.

Figure 5.20 illustrates a comparison between the energy losses. The largest thermal loss is shown in the case of a one-layer case, and these losses decrease with the presence of insulation. In the presence of a thermal bridge, the losses are larger than those without, and the losses are slightly larger when the thermal bridge is straight compared to a bent bar.

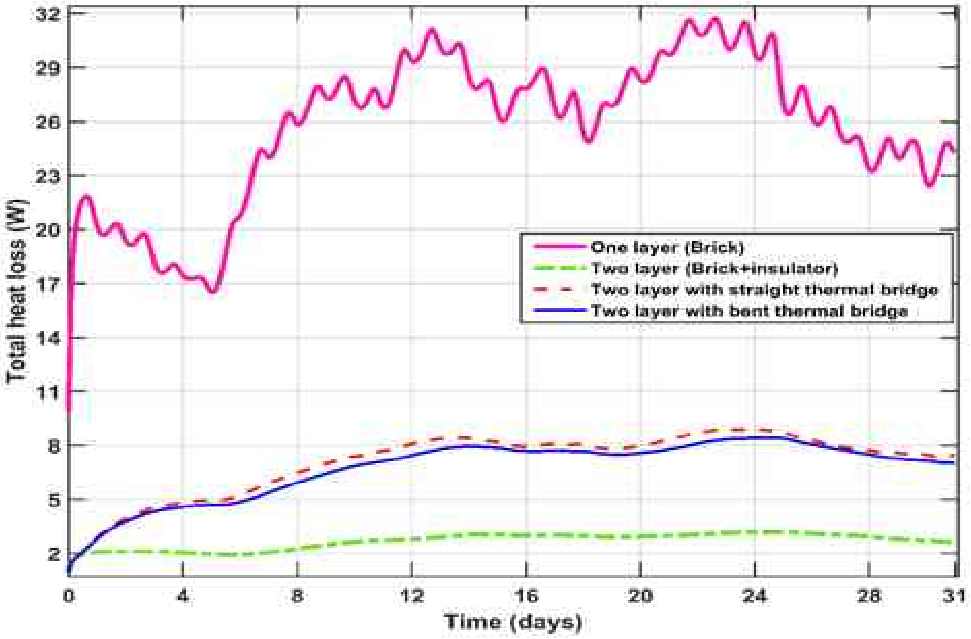

Figure 5.20. The distribution of total heat loss in Watt units as a function of time in days for the long-term simulation of all wall cases.

Table 5.6. Heat loss through the 1 m^2 part of the wall and the energy cost in USD and HUF.

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

5.4.4. Simulation of the coldest Day in the month

The hardest weather conditions are adopted for the heating load design, and the design is depending on them. Therefore, the coldest day in the month is investigated, January 25, when the lowest temperature recorded was -6 °C. The best previous methods are also applied to calculate the temperatures and heat loss through the wall. However, because the overall simulation time T = 24 × 3600 = 86,400 s is so short, a new time step size of $\Delta t = 60$ s is used in this simulation to track and record the temperature change every minute. Table 5.7 shows the cost of consumption on this day. One can see the largest losses in the case of one layer.

Table 5.7. Heat loss through the wall and the energy cost in USD and HUF for the coldest day.

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

5.5. Long-Term Simulation for Different Shapes of Thermal Bridges in Walls

The simulations were carried out for the three-layer typical residential wall with different shapes of thermal bridges in Miskolc, Hungary, and they were compared with the wall without a thermal bridge. The data about the convection coefficient, the external temperature, and the solar radiation values for this city, which is used in the simulation, is taken from the website page every 60 minutes [101]. Then, the data for every 100 seconds are calculated using a linear interpolation. In the relatively mild winter of 2022-2023, the highest temperature recorded was 15.2°C, and the lowest was -9.3°C, both occurred in February. The main wind may blow in several directions, depending on its location. The maximum wind speed recorded was 8.9 m/s [98].

For all transient simulations, with a set and a total calculation period of $\mathrm{T} = 90 \times 24 \times 3600 = 7776000$ s, representing the whole three months in winter (December, January, and February). The total value of time steps was $\mathrm{Time} = 77760$, where the time step size is 100 s. The results are for the LH method since the variation between the DF and LH was regularly checked during the simulations and determined to be very small.

The stiffness ratio was calculated [102], and the maximum (CFL limit, beyond which this causes instability in the explicit Euler time integration) for the time step size while just taking the conduction component into consideration. Table 5.8 presents the values of these quantities for different instances for all cases. The application I used was MATLAB R2020b, and the computer was a desktop which has an Intel Core i3-8100 (4 CPUs) and 24 GB of RAM. The long-time simulation took 805.6 seconds to complete. The duration of the methods is monitored using the built-in tic-toc timer of that application.

Table 5.8. The wall's stiffness ratio and explicit Runge-Kutta CFL limit for cases (A) and (B).

Layers number	Thermal bridge type	CFL limit [s]	Stiffness ratio
Three layers	-	12.4783	8.0266×10^4
	straight	6.6792	3.7396×10^4
	bent	6.69	6.242×10^4
	L-shaped	6.682	7.848×10^4

As can be shown, using our methods, transient heat-transfer calculations are completed quite quickly. Since this type of simulation does not require a supercomputer, it implies that it is accessible to everyone.

5.5.1. The Geometry

Figure 5.21 displays the various wall models. First, three layers of concrete, insulator, and face brick are used, as displayed in Figure 5.21A. The dimensions of the full wall thicknesses, height, and width are $(0.5, 1, 1)$m in the *x*, *y*, and *z* axis directions, respectively, where the thicknesses of the concrete, insulator and face brick layers are (0.25, 0.15, 0.1) m respectively. The other models are the same as the first one in addition to a straight, bent, and L-shaped thermal bridge, respectively, as shown in Figure 5.21B, C, and D. The width of the straight thermal bridge is 30 cm, where the horizontal position of it is between x = 20 cm and y = 50 cm, and the top's vertical position of it is z = 50 cm (50 cm from the wall's bottom), and the thickness of the thermal bridge is 0.05 m in the z-direction. There are three straight bars in the bending thermal bridge, one of them vertical and the other horizontal. The external side connects to the first horizontal straight bar at z = 75 cm (25 cm from the bottom of the right side of the wall), which is going horizontally with a 0.1375 m length. It is connected to the vertical bar, with a length of 0.1625 m in the z-direction. Then, as illustrated in Figure 5.21D, it is connected to the other horizontal bar, which has the same dimensions as the first horizontal bar.

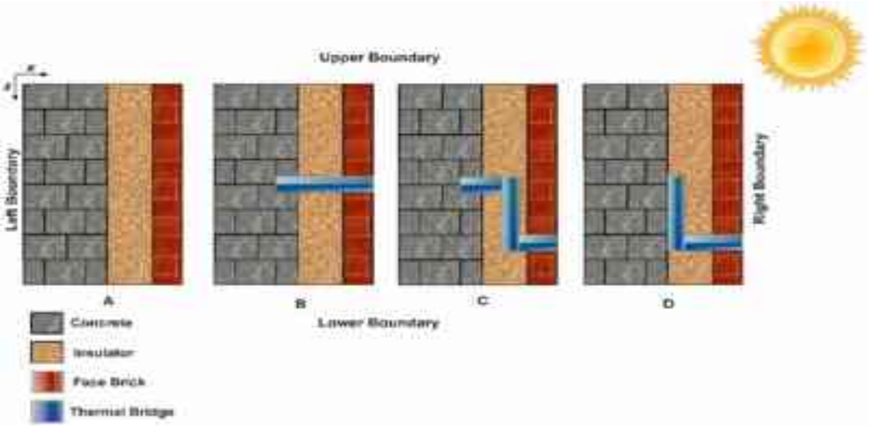

Figure 5.21. The walls A, B, C, and D with different types of thermal bridges.

5.5.2. *The Material properties*

In this investigation, the actual material properties that are mentioned in Table 5.9 are taken into account.

Table 5.9. The materials properties used in the simulations [103].

Material	c $[\mathbf{J.kg^{-1}K^{-1}}]$	ρ $[\mathbf{kg.m^{-3}}]$	k $[\mathbf{W.m^{-1}.K^{-1}}]$
Concrete	840	1601.8	1.73
Insulator (Glass wool)	1400	120	0.039
Face brick	920	2082.4	1.3
Thermal bridge (Steel Beam)	840	7800	16.2

5.5.3. *Numerical results*

The results are calculated for the outside wall with different types of thermal bridges, straight, bent, and L-shaped, and compared with the wall without a thermal bridge.

The results are shown for the particular points shown in Figure 5.22, and the subfigures include the following details. Five points denoted for the three-layer wall (concrete + insulator + face brick) with all thermal bridge cases: on the middle of the interior surface of the wall, the internal surface of the insulator, the external surface of the insulator, the wall's external surface, and the outside air temperature.

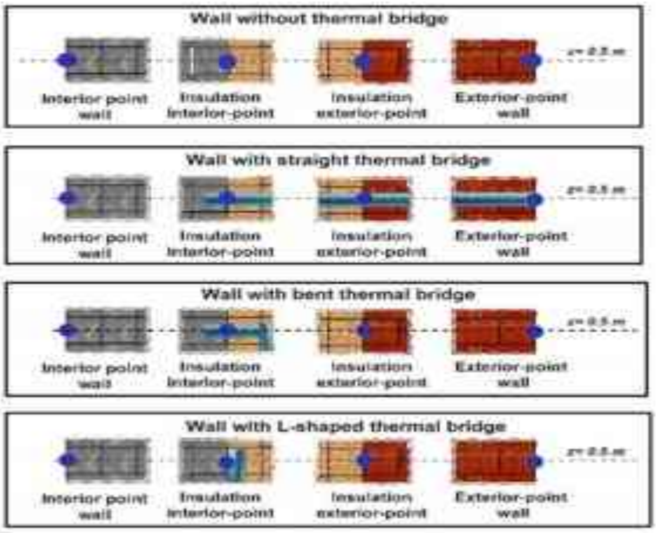
Wall without thermal bridge
z= 0.5 m
Interior point wall
Insulation interior-point
Insulation exterior-point
Exterior-point wall
Wall with straight thermal bridge
z= 0.5 m
Interior point wall
Insulation interior-point
Insulation exterior-point
Exterior-point wall
Wall with bent thermal bridge
z= 0.5 m
Interior point wall
Insulation interior-point
Insulation exterior-point
Exterior-point wall
Wall with L-shaped thermal bridge
z= 0.5 m
Interior point wall
Insulation interior-point
Insulation exterior-point
Exterior-point wall

Figure 5.22. Illustration of the grid and points used to display the results for the various situations.

- *Three layers (concrete + insulator+ face brick) without thermal bridge*

Since the difference between LH and DF methods was continuously evaluated during the simulation process and found to be small, it is enough to show the results for the LH method. The temperatures as a function of the time in days are presented in Figure 5.23. One can see that in the presence of insulation, the temperature of the wall's interior surface first equals that of the inside air temperature (22 °C), then falls due to the wall's cooling effect and the cold weather outside. The minimum values of the wall's exterior surface temperatures are marginally lower than the outside air temperature due to the insulator's restriction of heat transfer from inside to outside. For the higher values, it is bigger due to the solar radiation's impacts. The impacts of external variables cause the temperature of the insulator's internal surface to rise and fall.

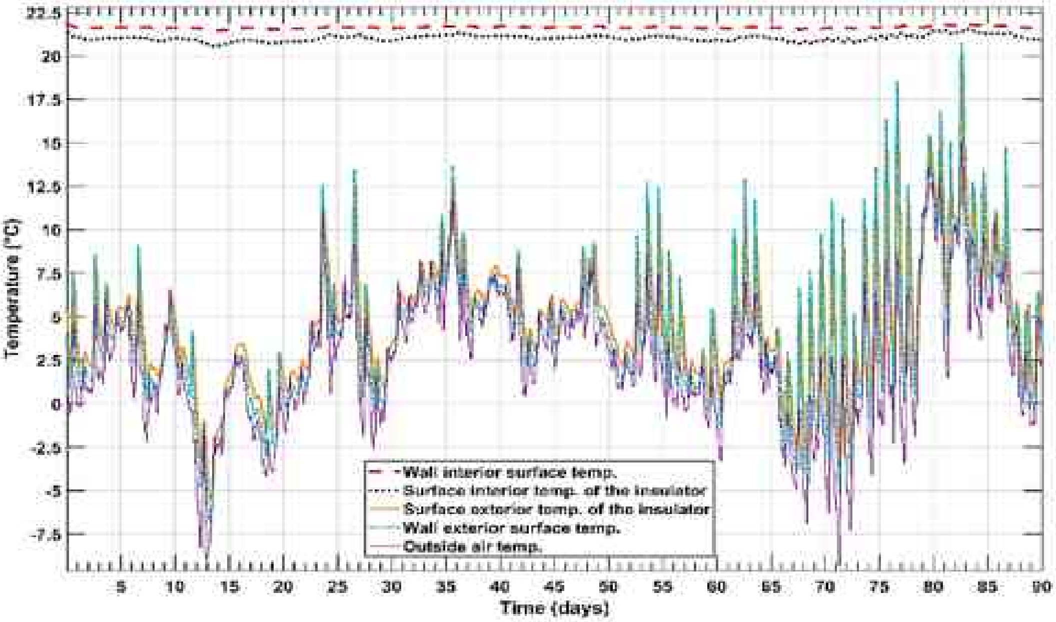

Figure 5.23. The distribution of temperature in °C units versus time in days for the three-layer wall simulation without a thermal bridge.

- *Three layers (concrete + insulator+ face brick) with straight thermal bridge*

Figure 5.24 clearly illustrates how the temperature of the insulator declined more than in the previous case, and the internal surface of the wall decreased slightly as a result of the heat loss that the straight thermal bridge provides. Furthermore, because of the heat transfer from the inside to the outside of the thermal bridge, the outside surface temperatures of the thermal bridge are higher than the air temperatures outside.

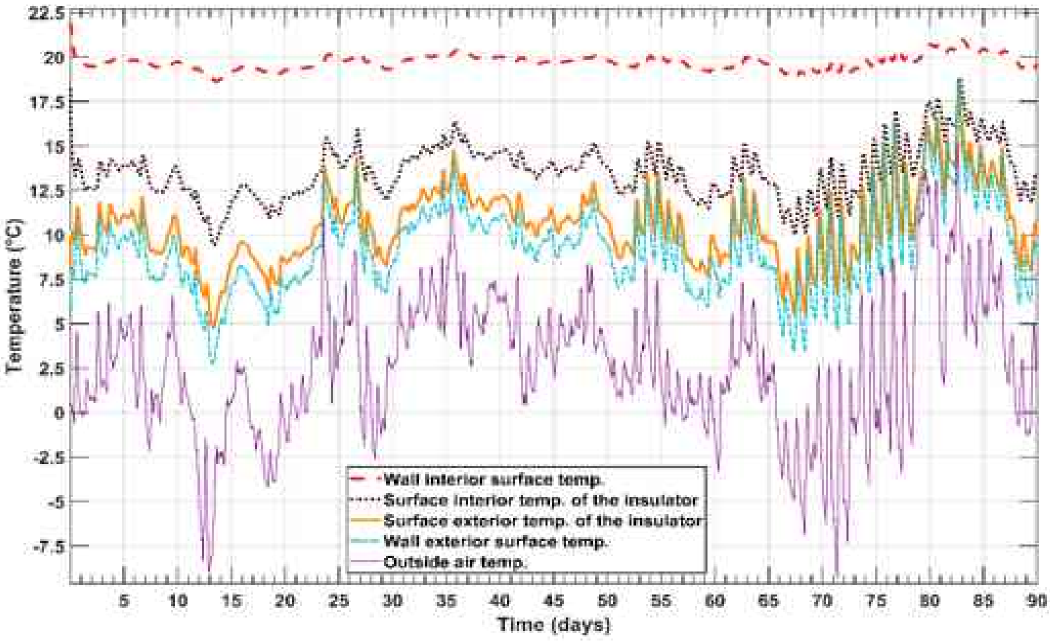

Figure 5.24. The distribution of temperature in °C units versus time in days for the three-layer wall simulation with straight thermal bridge.

- *Three layers (concrete + insulator+ face brick) with bent and L-shaped thermal bridges*

In this case, from Figure 5.25, one can see that the temperature of the insulator's inner surface rises compared to the two cases of the wall with straight and bent thermal bridges and that the temperature of the insulator's outer surface is smaller than them. The effect of the bent thermal bridge is smaller because the heat path in it is longer than in the straight thermal bridge case. Figure 5.26 shows the temperature contours of both walls with bent thermal bridge and L-shaped thermal bridge case.

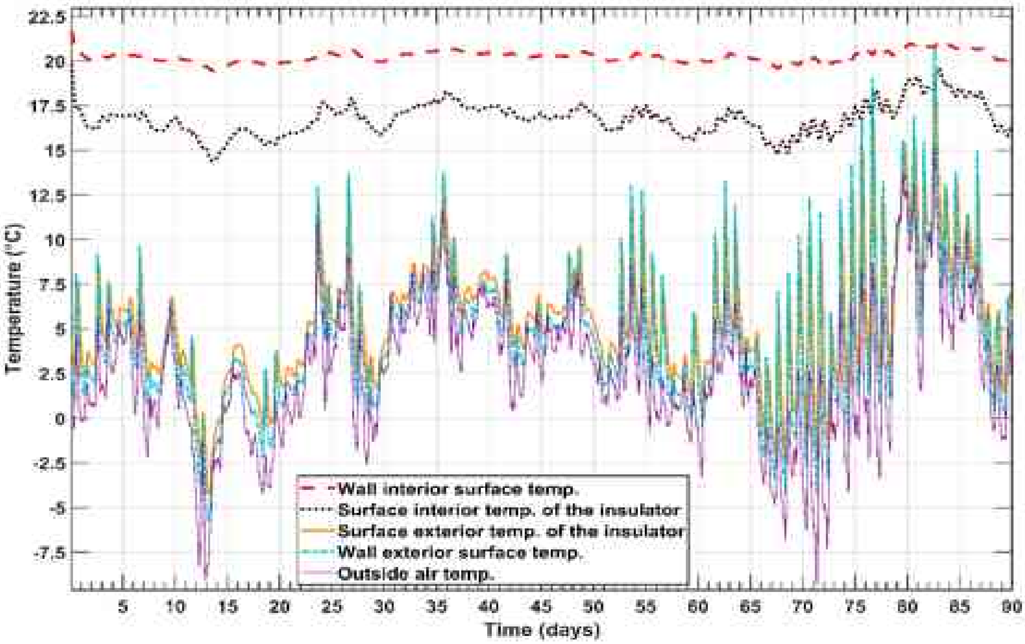

Figure 5.25. The distribution of temperature in °C units versus time in days for the three-layer wall simulation with L-shaped thermal bridge.

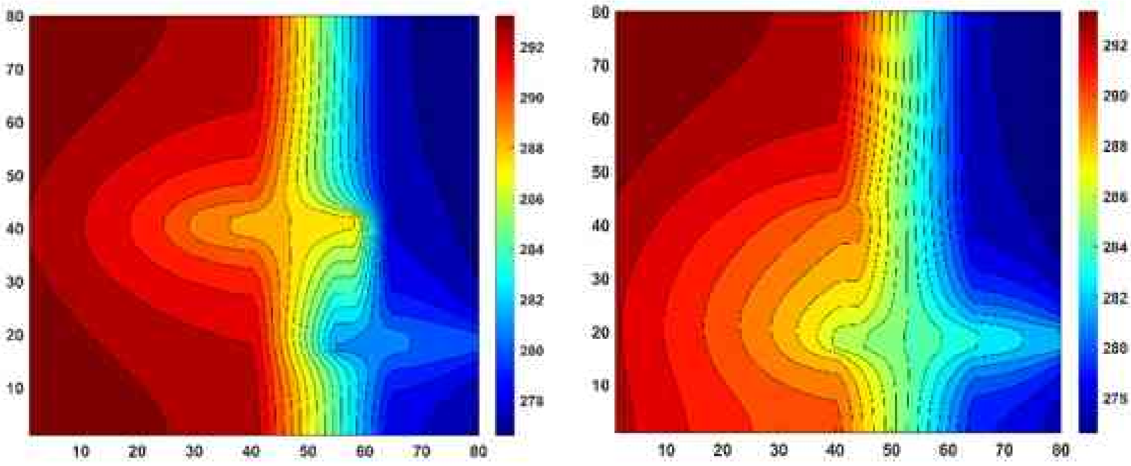

Figure 5.26. At the end of the last day, the temperature distributions contour in Kelvin units for the long-term simulations for the walls with a bent thermal bridge (left) and L-shaped thermal bridge (right).

- *Heat loss*

Figure 5.27 displays a comparison between the loss in energy. The heat loss and energy consumption expenses in USD and HUF are displayed in Table 5.10 below. These losses are higher in the presence of a thermal bridge than in the absence of one, and the largest thermal loss occurs in the case of straight and L-shaped thermal bridges. When I designed the research, I had the hypothesis that the L-shaped bridge causes less loss than the straight and bent bridge since it does not penetrate the concrete, unlike those bridges. Surprisingly, the L-shaped thermal bridge is almost as bad as the straight one and much worse than the bent bridge. The reason for this is probably that the L-shaped metal rod touches and cools down a relatively large part of the surface of the concrete wall.

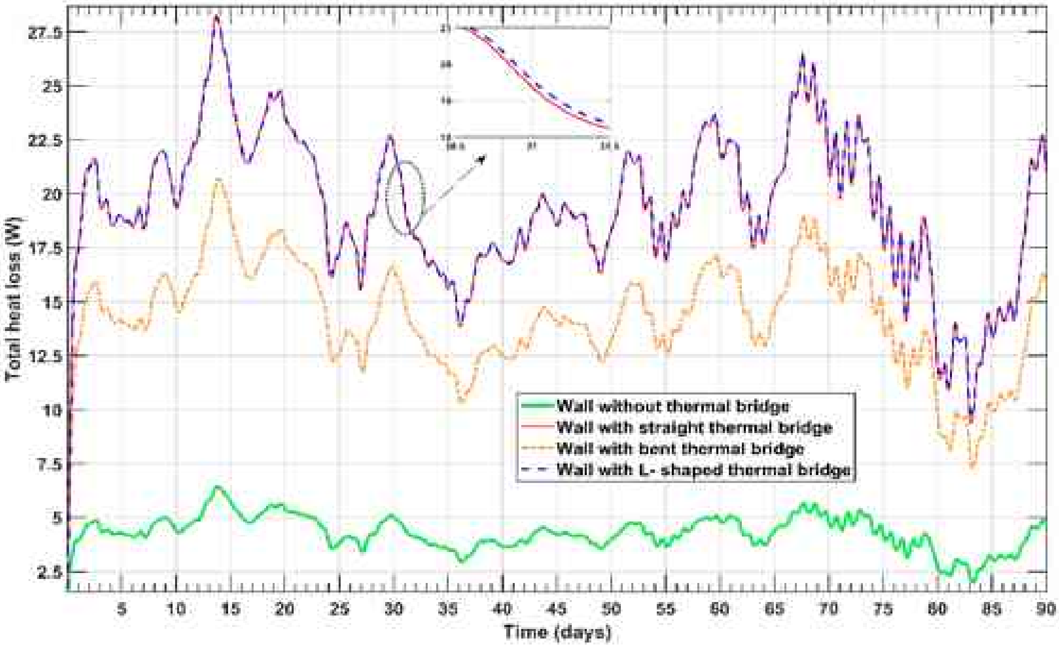

Figure 5.27. Total heat loss distributions in Watt versus time for the four cases.

Table 5.10. Heat loss through the 1 m^2 part of a wall and the corresponding energy cost.

Thermal bridge type	Without	Straight	Bent	L-shape
Heat loss (in three months, kWh)	9.28	41.67	30.8	41.86
The cost in USD	0.93	4.167	3.08	4.186
The cost in HUF	347.564	1561	1154	1568.2

5.6. Summary of this chapter

The analysis suggest that my methods are better than all ANSYS solvers and MATLAB routines, whereas ANSYS was less accurate and slower, and it was observed that the best performance was achieved by the leapfrog–hopscotch and the Dufort–Frankel algorithms with the pseudo-implicit treatment of the nonlinear radiation term. Therefore, these two methods were applied to real problems, and a long-term simulation of four cases was performed. The temperature distribution and total heat losses of all cases were calculated. I found the straight thermal bridge to be energetically worse than others, and the total heat loss during the month (one-layer, two-layer, two-layer with a straight thermal bridge, and two-layer with a bent thermal bridge) was, respectively, 19.14, 1.99, 5.29, and 5.01 kWh for a 1 m^2 wall surface. I can conclude that the numerical simulation methodology is established in this work.

6. APPLYING RECENT EFFICIENT NUMERICAL METHODS FOR LONG-TERM SIMULATIONS OF HEAT TRANSFER IN WALLS TO OPTIMIZE THERMAL INSULATION

Since transient simulations need a lot of resources, the heat loss through the walls of buildings in winter is often estimated by a simple steady-state calculation based on methods like the Degree-days, which is frequently rather inaccurate. In this chapter, I carried out transient simulations using the new leapfrog-hopscotch and the modified Dufort-Frankel algorithms, which are the most efficient, stable, and explicit numerical methods to deal with heat transfer problems, according to previous investigations. The optimum thickness of insulation, energy savings, and payback time are determined using an economic model that considers the orientation of the external walls, solar radiation, the cost of insulation materials, the present cost of energy consumption, and the cost over the 25-year lifetime of a building in Miskolc City, and a case is analysed in the cold season. Three materials and a range of thicknesses are investigated: Expanded Polystyrene (EPS), glass-wool, and rock-wool. I found the transient way to calculate heat loss to be quick and accurate.

I use an ab-initio approach to compute transient heat transport utilizing fundamental physical rules. As a result, it is anticipated that these findings will be substantially more accurate than those based on the common (ISO) standards [14], [15], [16], which use steady-state calculations without solving the transient PDE and cannot, for example, correctly account for the envelope's heat capacity. To make transient simulations more accessible owing to decreased processing cost and programming difficulties, my long-term aim is to revolutionize these simulations.

In the previous chapters, I found that among explicit and stable numerical methods, the modified Dufort-Frankel and the leapfrog-hopscotch methods are the most effective for solving heat transfer problems in building walls. Under transient conditions, numerical methods were used to calculate the winter heating loss according to wall orientations. Additionally, it was looked at how well the walls conducted heat under optimal conditions. Comparing this study to others of a similar type, one of its unique characteristics was the use of less expensive local materials, to optimize investment on insulation.

In light of this, this section is structured as follows: Section 6.1 shows the preliminary conditions: building, materials, geometry, mesh construction, initial, and boundary conditions. In Section 6.2, I start verifying by using a steady-state case by using two ways to calculate the heat loss through the wall. Section 6.3 displays the results of the numerical simulation of all walls and the calculation of heat loss. Section 6.4 finally concludes with a summary of my results.

6.1. Numerical simulation of the building model

6.1.1. Building properties

The studied problem is the insulation of a residential building in Miskolc, Hungary, as shown in Figure 6.1. This is an eleven-floor building, and the length, height, and width of each floor are 24, 3, and 12 m, respectively. The simulations were performed for three outside walls of a flat as shown in Figure 6.2, where different thicknesses and materials of insulators [104] were used and the overall heat transfer coefficient calculated by the HAP as shown in Table 6.1, and they were compared with the wall that contains two layers without insulation. The outside air temperature, solar radiation, and convection coefficient values for Miskolc were used in the simulations, where the data was taken from the website [98] every one hour. The data for every 50 s were then calculated using linear interpolation.

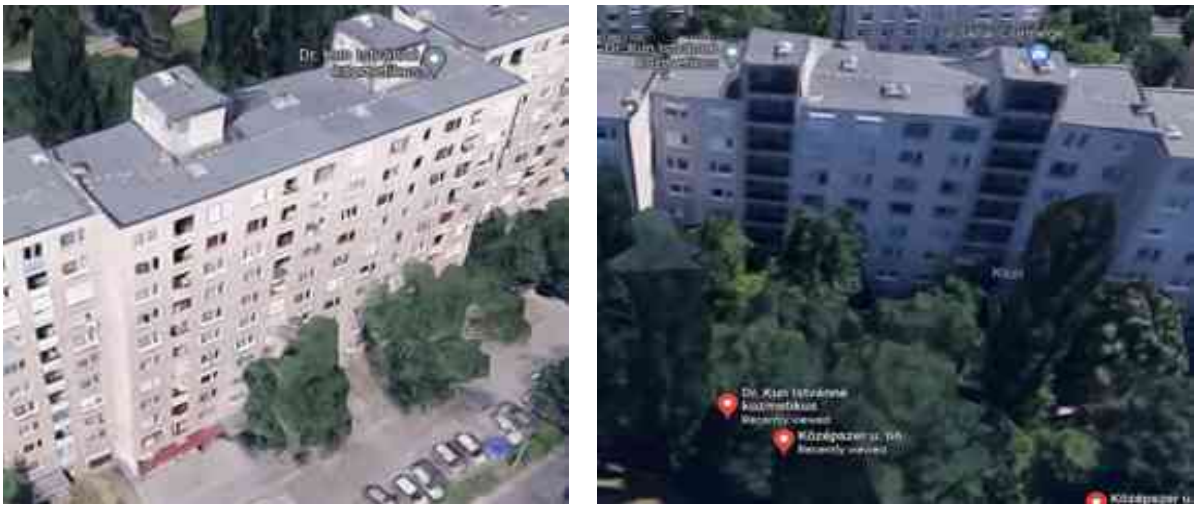

Figure 6.1. The photo of the building [98].

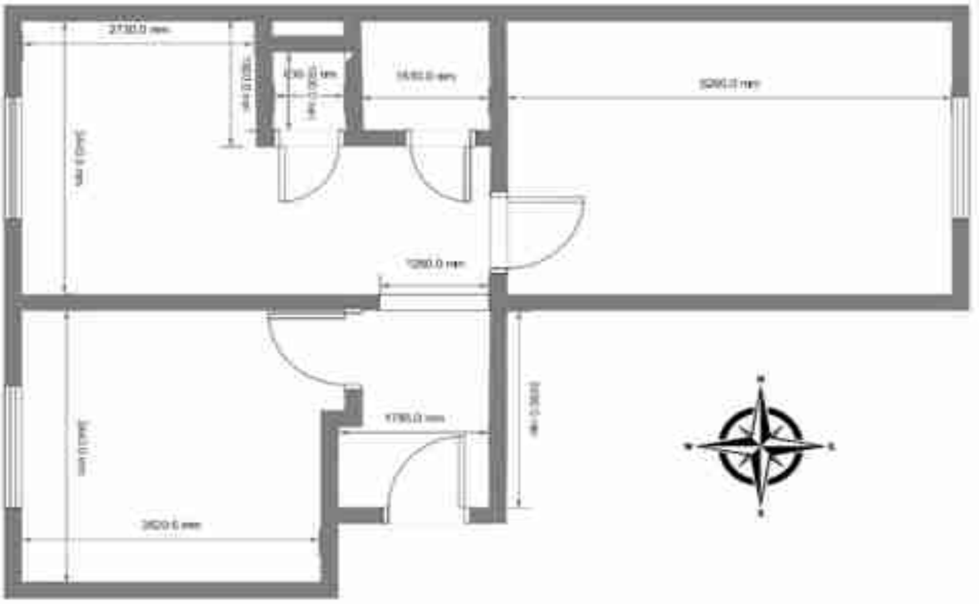

Figure 6.2. The apartment scheme in the residential building.

Table 6.1. Components and total heat transfer coefficient of enclosure structure of the building.

Name	Components of enclosure Structure	The insulation	The thickness	Heat transfer coefficient, U $\left[\mathrm{Wm^{-2}K^{-1}}\right]$
External wall	1 cm Gypsum plaster + 25 cm Panels Miskolc 1971-75	Without insulation	-	2.69
		EPS	10 cm	1.6
			15 cm	1.344
			20 cm	1.158
		Rock wool	10 cm	1.449
			15 cm	1.188
			20 cm	1.007
		Glass wool	10 cm	1.388
			15 cm	1.127
			20 cm	0.949
Roof	Ceiling air space+20 cm HW concrete+1 cm felt & membrane+1.3 cm stone or slag	-	-	1.68
Door	Sample Door Assembly	-	-	1.703
Window	Double clear window	-	-	3.2

I emphasize that I simulated heat transfer through the walls for a fixed internal ambient temperature, thus the effect of the roofs, windows, ventilation etc, was not considered.

6.1.2. Geometry

The different wall models are seen in Figure 6.3. First, two layers of panel and gypsum plaster were used, as shown in Figure 6.3A. The dimensions of the full wall thickness, height and width are (0.26, 1, 1 m) in the *x*, *y*, and *z* directions, respectively. The second model has three layers: the panel and gypsum plaster layers have the same dimensions as the first model, in addition to an insulating layer from EPS with three thicknesses of 0.10, 0.15, and 0.2 m, as illustrated in Figure 6.3B. The third and fourth types are identical to the second but have an insulating layer from glass wool and rock wool, as shown in Figure 6.3C and D, respectively.

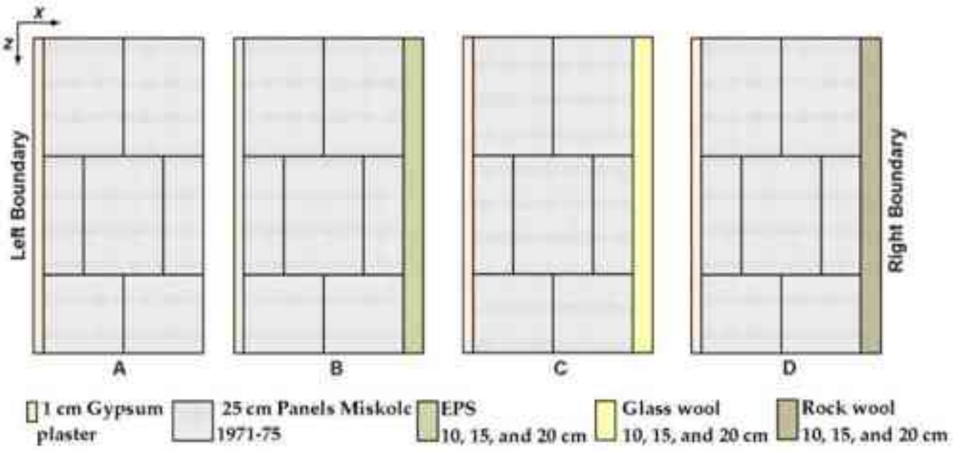
x
z
Left Boundary
Right Boundary
A
B
C
D
1 cm Gypsum plaster
25 cm Panels Miskolc 1971-75
EPS 10, 15, and 20 cm
Glass wool 10, 15, and 20 cm
Rock wool 10, 15, and 20 cm

Figure 6.3. Shows the wall A, B, C, and D with different insulation materials and thicknesses.

6.1.3. Material properties

The real material parameters stated in Table 6.2 are taken into consideration in the current study.

Table 6.2. Thermo-physical properties of the materials used in the experiments [103].

Material	$c\ [\mathbf{J.kg^{-1}K^{-1}}]$	$\rho\ [\mathbf{kg.m^{-3}}]$	$k\ [\mathbf{W.m^{-1}.K^{-1}}]$
Gypsum plaster	977	805	0.29
Panels Miskolc 1971-75	800	1600	0.743
EPS	1400	20	0.038
Glass wool	700	120	0.039
Rock wool	660	28	0.036

6.1.4. Mesh construction

In the calculations, a wall dimension of 0.26 m×1 m ×1 m was considered for wall type A. Since the *y*-direction is orthogonal to the surfaces of Figure 6.3A, B, C, and D, and I assume that no physical values change in that direction, and in the *z*-direction, these dimensions are irrelevant from a mathematical standpoint. In mathematical terms, this indicates that we can deal with a one-dimensional problem; thus, we can use $\Delta y \;\; = 1$ m and $\Delta z \;\; = 1$ m. So, I created several different 1 m^2 mesh versions, which means $(y,z) \in [0,1] \times [0,1]$, thus the cells are oblate square cuboids in form. Undoubtedly, using a larger number of cells can produce a more accurate answer, but I must keep in mind that this will require more significant computational expense. Above a certain number of cells, results do not change as the number of cells increases, suggesting that I have reached the so-called mesh independence. Therefore, eleven meshes with an increasing number of cells along the *x* axis (and therefore decreasing mesh spacing Δx) for the two layers of wall are used to test the independence of the results on the mesh density. The heat loss according to the number of cells is calculated. One can see from the Figure 6.4 that when I increase the number of cells to more than 416, the heat loss remains the same.

The cell numbers along axis *x* for all wall cases A, B, C, and D are N_x= 416, 576, 656 and 736 for 0, 10, 15 and 20 cm thickness, respectively. In each case, the spatial variable *x* is discretized using an equidistant grid and $\Delta x = 0.000626\,\mathrm{m}$, which is fixed. It's important to remember that the cell's middle temperature was regarded as its overall temperature.

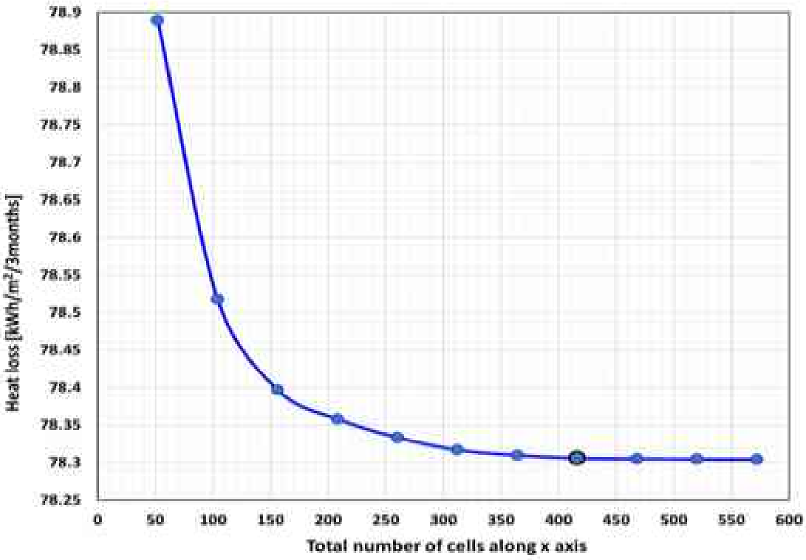

Figure 6.4. Heat loss as a function of the total number of cells along the *x* axis in the case of two layers at the north wall.

The approximate formula for the thermal resistance in the *x*-direction is $Rx_i \approx \frac{\Delta x}{k_i S}$.

If the sizes or the material properties of the two neighbouring cells are different, the resistance between cells *i* and *i*+1 can be written as follows:

$$Rx_i \approx \frac{\Delta x}{2k_i \Delta z \Delta y} + \frac{\Delta x}{2k_{i+1} \Delta z \Delta y}.$$

6.1.5. *Initial and boundary conditions*

I used the same initial and boundary conditions that were mentioned in Subsection 5.4.2, but I changed some values in the table. Table 6.3 demonstrates how items on the left and right sides may undergo convectional and radiative heat transfer in the *x* direction.

Table 6.3. For all the types of walls previously discussed, the convection, radiation, and heat source factors are on both sides of the wall [105].

	$h_c \left[\frac{\text{W}}{\text{m}^2\text{K}}\right]$	$\sigma^* \left[\frac{\mathbf{W}}{\mathbf{m^2K^4}} \times \mathbf{10^{-8}}\right]$	**ε**
Left Elements (inside)	8	3.97	0.7
Right Elements (outside)	0.6 – 20.41	4.82	0.85

$v(t)$: The air velocity is measured according to website data [101] for each 50 s in December, January, and February [m/s], where I neglected the relative orientation of the wind and wall.

$u_r(t)$: The outside air temperature changes according to weather conditions [101] for each 50 s in December, January, and February months [°C].

u_l : The inside ambient air temperature $22\ ^\circ\text{C} \approx 295\ \text{K}$ on the left side.

$G_{cr}(t)$: The solar radiation is measured according to website data [101] for each 50 s in December, January, and February [W/m^2].

α_{sun} : The solar radiation absorption by the surface, which is taken as 0.95 [97].

α_{Low} : Surface absorptivity to low-temperature thermal radiation, which is 0.93 for the painted surface [97].

Using the presumption that a stationary heat flow with constant flux had developed between the specified boundary values of the internal and external air temperatures before the simulation period at midnight before the first day of December, I estimated the initial temperature inside the wall. In case of two layers, two linear functions of the *x* variable are used for the initial condition:

For the gypsum plaster part: $u(x,t=0)=(u_{\mathrm{mid}}-u_l)\cdot x/L_g+u_l$

For the panel part:

$$u(x,t=0)=\left[(u_{\mathrm{r,initial}}-u_{\mathrm{mid}})\cdot x/L_p\right]-\left[(u_{\mathrm{r,initial}}-u_{\mathrm{mid}})\cdot L_g/L_p\right]+u_{\mathrm{mid}}$$

where $u_{\mathrm{mid}}=u_l-(q_{\mathrm{flux}}L_g/k_g)$

and $q_{\mathrm{flux}}=(u_l-u_{\mathrm{r,initial}})/\left[(L_g/k_g)+(L_p/k_p)\right]$,

where $u_{\mathrm{r,initial}}=282\ \mathrm{K}$, and the subscripts *g* and *p* refer to the gypsum and panel layers, respectively, so for example, L_g is the thickness of the gypsum plaster.

For the three layers, I must use three linear functions of the x variable for the initial condition since I am assuming stationary heat conduction with initial values at the boundary.

For the gypsum plaster part: $u(x,t=0)=(u_{\mathrm{mid,1}}-u_l)\cdot x/L_g+u_l$,

For the panel part:

$$u(x,t=0)=\left[(u_{\mathrm{mid,2}}-u_{\mathrm{mid,1}})\cdot x/L_p\right]-\left[(u_{\mathrm{mid,2}}-u_{\mathrm{mid,1}})\cdot L_g/L_p\right]+u_{\mathrm{mid,1}}$$

where $u_{\mathrm{mid,1}}=u_l-(q_{\mathrm{flux}}L_g/k_g)$

$$u_{\mathrm{mid,2}}=u_{r,initial}+(q_{\mathrm{flux}}L_{ins}/k_{ins})$$

and $q_{\mathrm{flux}}=(u_l-u_{\mathrm{r,initial}})/\left[(L_g/k_g)+(L_p/k_p)+(L_{ins}/k_{ins})\right]$

For the insulation part:

$$u(x,t=0)=\left[(u_{\mathrm{r,initial}}-u_{\mathrm{mid,2}})\cdot x/L_{ins}\right]-\left[(u_{\mathrm{r,initial}}-u_{\mathrm{mid,2}})\cdot(L_g+L_p)/L_{ins}\right]+u_{\mathrm{mid,2}}$$

where the subscript *ins* refers to the insulation layer.

I choose a fixed time step size of Δ*t*= 50 s for all transient simulations.

The total duration of the simulation is *T*=90×24×3600=7776000 s, which corresponds to the full winter season (December, January, and February). The total number of time steps is always

$Time = T / \Delta t = 155520\,\text{s}$. I determined the stiffness ratio in the usual way [70] and the maximum time step size (CFL limit, over which the explicit Euler time integration becomes unstable), considering only the conduction term. The values of these quantities for various cases are displayed in Table 6.4.

The running times for the simulations for the north wall are shown in Table 6.4, and they were measured on an old laptop with an Intel Core i3-5005U (4 CPUs) and 8 GB of RAM, whereas the program I used is MATLAB R2020b.

Table 6.4. The stiffness ratio and CFL limit quantities for wall (A) and (B).

Wall direction	Wall type	Thickness [cm]	CFL limit [s]	Stiffness ratio	Running time [s]
North wall	Wall (A)	-	0.6	3.83 × 10^4	15.89
	Wall (B), EPS	10	0.1444	1.627 ×]	26.17
		15	0.14436	1.67 × 10^5	29.94
		20	0.1443	1.93 × 10^5	37.24

One can see that the transient heat-transfer calculations with my methods are very short. It means it is available for everyone, since one does not need a supercomputer to make this kind of simulation.

6.2. Verification using the steady-state case

To estimate how much Heat loss through the two layers of the wall without insulation in the winter, I use the Heating Degree-Day method. It means that the heating system is turned on when the daily average of the external temperature falls below a baseline temperature, which is typically T_b=16°C. The inside temperature of a building is determined by factors such as the insulation, the ventilation, the quantity of the sunshine (especially which enters the rooms via the windows and trapped due to the greenhouse effect), the internal heat sources (electric equipment, humans), etc. Due to these factors, the Heating Degree-Day approach assumes that an average building has a comfortable internal temperature without heating if the external temperature is above the base temperature. The inside temperature is 22°C from ASHREA standard [106]. Heat loss by conduction through the two layers of the wall without insulation during the winter season is calculated by using the Heating Degree-Day method as follows:

$$q = U * HDD \tag{6.1}$$

where q is the heat loss [W/m^2], and U is the overall heat transfer coefficient [W/(m^{2}K)] which is given as

$$U = \frac{1}{\frac{1}{h_{in}} + \sum_{i=1}^{i=N} \frac{x_i}{k_i} + \frac{1}{h_{out}}}$$

The values of U, h_{in} , k, h_{out} are given in Table 6.5. HDD is Heating Degree-Day in [K] which is given as:

$$HDD = \sum_{i=1}^{N_{DH}} (T_b - T_d)\delta$$

where T_b: Base temperature in Hungary 16 C° [107], T_d: Daily mean temperature of the day, N_{DH}: Total number of days in the heating system.

$$\delta = 1 \quad \text{if } T_b \geq T_d$$
$$\delta = 0 \quad \text{if } T_b \leq T_d$$

The Heating Degree-Day is calculated in two ways: either its value is taken directly from the website [108], as shown in Figure 6.5, or it can be estimated based on real temperature data from the weather website [98] by calculating the daily mean temperature. Table 6.6 shows the heat loss calculations in two ways: steady-state and transient simulations.

Table 6.5. Real wall data was calculated for a residential wall in the city of Miskolc [98].

Wall type	$k\ \left[\mathbf{Wm^{-1}K^{-1}}\right]$	$h_{in}\ \left[\mathbf{Wm^{-2}K^{-1}}\right]$	$h_{out}\ \left[\mathbf{Wm^{-2}K^{-1}}\right]$	$U\ \left[\mathbf{Wm^{-2}K^{-1}}\right]$
Gypsum plaster	0.29	8	23	2.69
Panels Miskolc 1971-75	0.743			

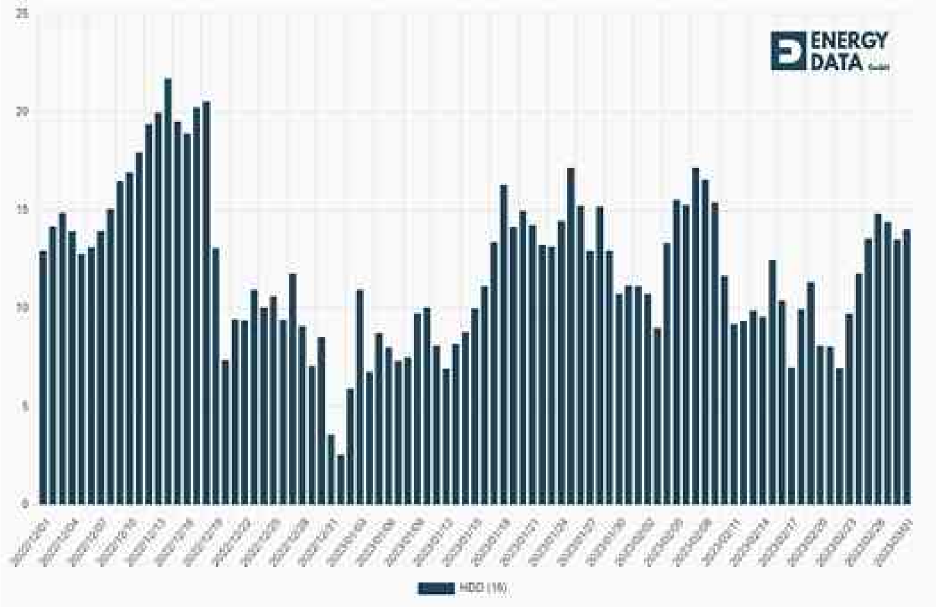

Figure 6.5. Heating Degree Days with a base temperature of 16 C° based in Miskolc city [108].

Table 6.6. Heat loss through the 1 m^2 of the wall (A) in the steady-state and transient cases.

Simulation type	Calculation method of *HDD*	$U\left[\dfrac{\mathbf{W}}{\mathbf{m^2K}}\right]$	*HDD* [K]	Heat loss $\left[\text{kWh}/\text{m}^2/\text{3months}\right]$
Steady state	Directly from website	2.69	1194	77.1
	From weather data	2.69	1207.66	77.96
Transient	-	-	-	78.3

One can see that my transient simulation yields a very similar heat loss value to the simple steady-state calculation. I believe that the former one is more accurate since it considers more physical effect, most importantly the heat capacity of the wall and the nonlinear radiation loss. That latter one can be the reason why steady-state calculations are prone to underestimating the heat loss.

6.3. Results of the three external walls simulation

The results are shown and calculated for the three external walls, North, West, and East, as well as calculating a hypothetical south wall, and three materials and thicknesses are used for them.

6.3.1. Comparison of the effect of changing the insulator thickness

❖ *Two layers (gypsum plaster + panel) without insulation*

The difference between the DF and LH was continuously evaluated during the simulation and found to be extremely small where the difference is below 0.5% as shown in Figure 6.6.

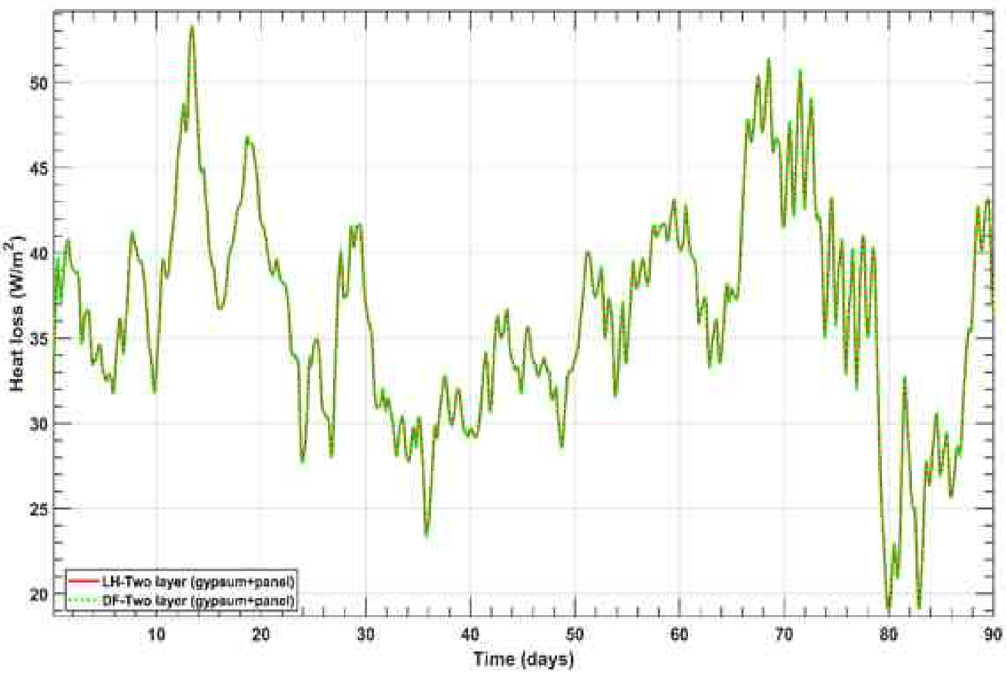

Figure 6.6. The heat loss in W/m^2 units as a function of time in days for the simulation of the north wall without insulation for the two methods LH and DF.

❖ *Three layers (gypsum plaster + panel) with EPS insulation*

Figure 6.7 displays the heat loss over time. It should be noted that the presence of insulation leads to a reduction in heat loss through the wall, and this decrease continues as the thickness of the insulation increases. The simulations are repeated for three layers, employing both glass-wool and rock-wool insulations, and the resulting curves are very similar.

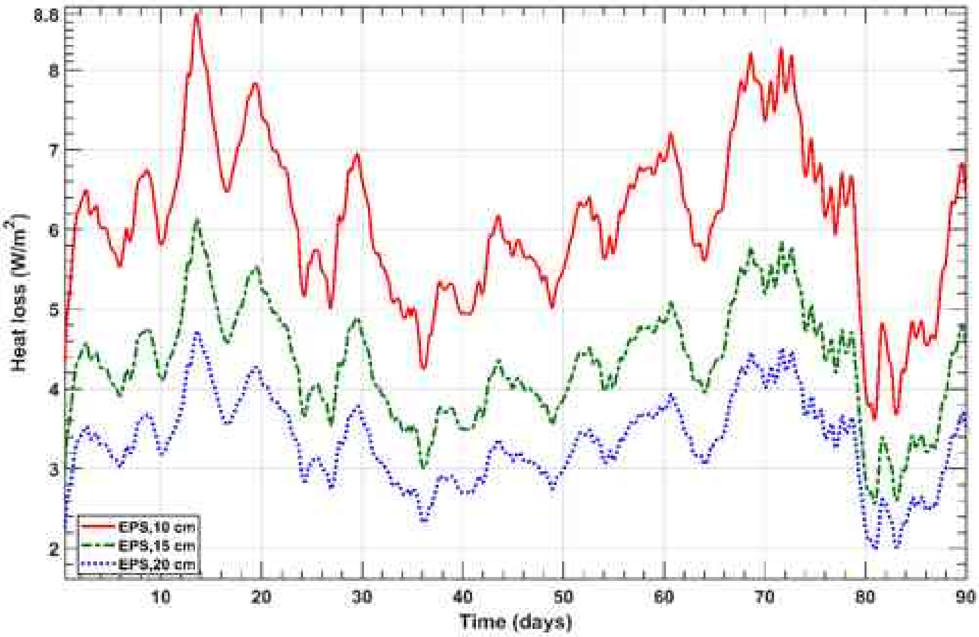

Figure 6.7. The heat loss in W/m^2 units as a function of time in days for the simulation of the north wall with the EPS insulation in case of three thicknesses.

In Figure 6.8 I compared the temperatures of the internal points of the wall without insulation and the wall with different thicknesses of EPS insulation; I concluded that the wall without insulation allows the heat to flow faster than the insulated wall.

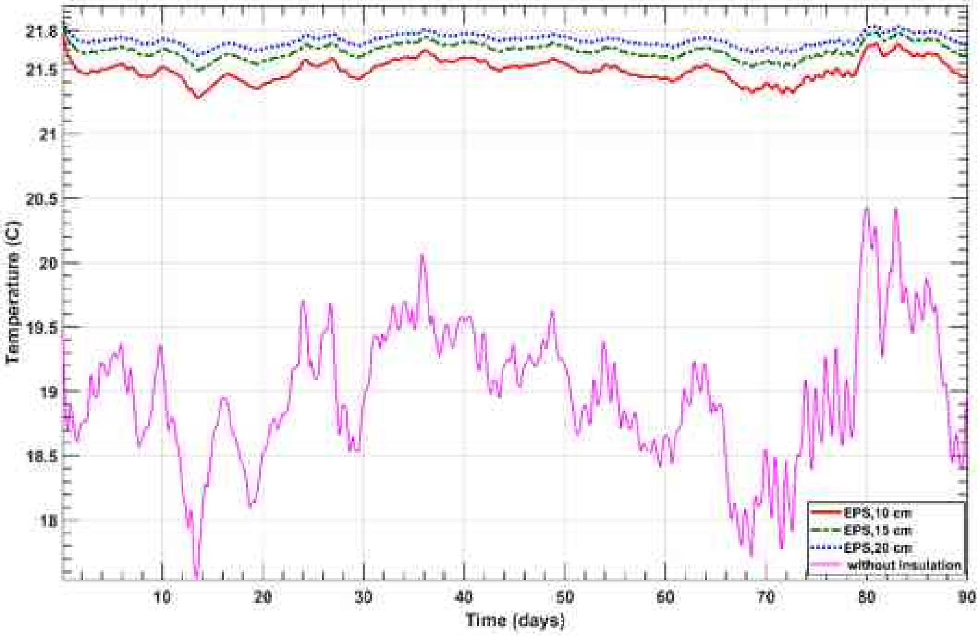

Figure 6.8. The change of temperature [°C] as a function of days of the wall simulation for inside points of the north wall with EPS insulation in case of three thicknesses compared to the wall without insulation.

The comparison between the one-layer, two-layer, and three-layer wall cases as a function of space in terms of the final temperature on the last day of February is shown in Figure 6.9.

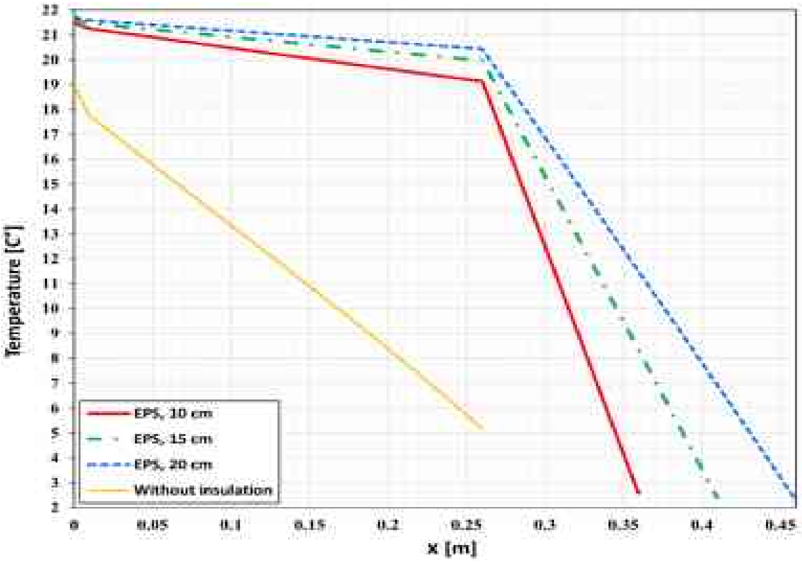

Figure 6.9. The temperature [°C] for the simulation of the north wall with the EPS insulation in case of three thicknesses for three layers at the end of the last day of February.

6.3.2. Comparison of the effect of changing the material of the insulator

Figure 6.10 shows a comparison between the losses in energy in the case of the three material insulations at thickness 10 cm. One can see the largest thermal losses in the case of the glass-wool insulator, and these losses are less in the presence of the rock-wool insulation. However, as shown in Table 6.8, the difference between the three materials (EPS, glass wool, and rock wool) with the same thickness is smaller than the difference between the different thicknesses. For example, for the north direction, the heat losses for EPS, glass wool, and rock wool are 9.3, 9.5, and 8.87 W/m^2 for 15 cm thickness, and 7.188, 7.355, and 6.844 W/m^2 for 20 cm thickness, respectively.

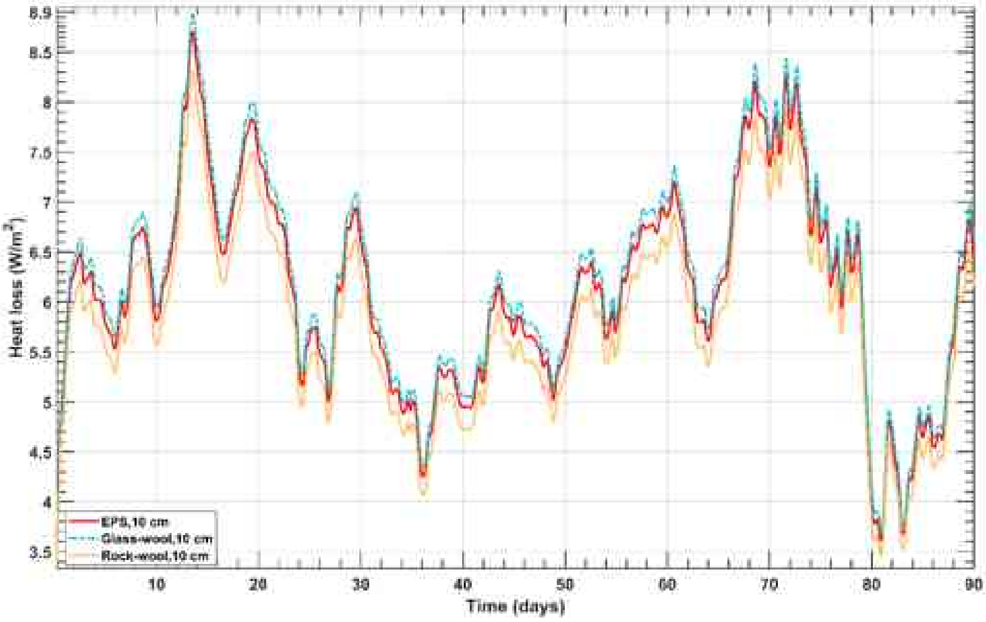

Figure 6.10. The heat loss in W/m^2 units as a function of time in days for the simulation of the north wall in the case of 10 cm thickness for the three insulations.

Figure 6.11 presents a comparison between the energy losses in the case of the four directions of the walls. One can see that the difference between the curves in December and the first part of January is smaller because of the small amount of sunshine, but for the second part of January and February, it is larger because of the large amount of sunshine. So, the smallest thermal losses are in the case of the south wall, because it receives the greatest amount of solar radiation in the winter, and these losses are the largest in the case of the north wall because it does not receive any amount of solar radiation.

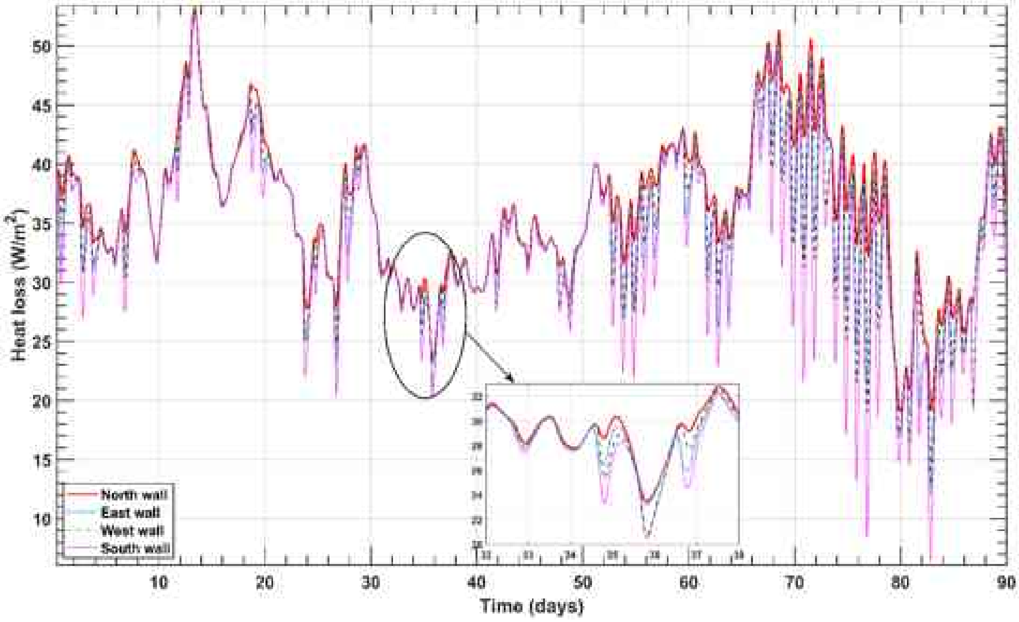

Figure 6.11. The heat loss in W/m^2 units as a function of time in days for the simulation of the north, east, and west walls in case of the wall without insulation.

6.3.3. Calculation of heat loss through the walls

The total energy requirement Q was calculated from Eq. (5.19) in Subsection 5.4.3.

Now I can calculate the cost of the energy used to make up the energy loss by the multiplication of the current price of electricity (per kWh) used by the amount of heat loss. The heat loss and energy consumption expenses are displayed in Table 6.7 below, and the heat loss for full flat is shown in Table 6.8.

The following equations are used in the calculations:

Total initial cost = cost of equipment (insulation) + cost of installation.

Where cost of installation is the estimated average installation cost of the three insulators based on thickness 15 cm, for example in wall (B) with 10 cm thickness the total initial cost = the cost of insulation (9.71 USD) + the estimated cost of installation of 15 cm insulation (14.56 USD) = 24.27 USD.

Cost of energy = price of kWh × heat loss.

Total annual energy cost = cost of energy + cost of maintenance. For the full winter season and the cost of maintenance is taken to 0 during the lifetime.

System lifetime = 25 years.

Annual savings = Total annual energy cost without insulation - Total annual energy cost with insulation.

Payback time = Total initial cost / Annual savings.

Total saved energy cost in 25 years = Annual saved energy cost × 25.

Total Life cycle saving = Total saved energy cost in 25 years - Total initial cost.

Table 6.7. Heat loss through the 1 m^2 of the four types of walls in the transient case and the energy cost in USD in addition to the payback time in years.

Wall orientation	Wall type	Insulator thickness (cm)	Cost of insulation (USD)	Total Initial cost (USD)	Heat loss kWh/m^2/3 months	Total Annual energy cost (USD)	Payback time (Years)	Annual saving energy cost (USD)	Total saving energy cost in 25 years (USD)	Total Life cycle saving (USD)
North wall	Wall (A)	-	-	-	78.3	7.83	-	-	-	-
	Wall (B), EPS	10	9.71	25.71	13.18	1.318	3.95	6.51	162.8	137.1
		15	14.56	30.56	9.3	0.93	4.43	6.9	172.5	141.94
		20	19.7	35.7	7.2	0.72	5	7.11	177.7	142
	Wall (C), Glass wool	10	4.46	20.46	13.46	1.346	3.15	6.48	162.1	141.64
		15	6.69	22.69	9.51	0.95	3.29	6.88	172	149.3
		20	8.92	24.92	7.35	0.735	3.51	7.1	177.4	152.4
	Wall (D), Rock wool	10	17.19	33.19	12.6	1.26	5	6.57	164.25	131
		15	25.78	41.78	8.87	0.887	6	6.94	173.6	131.8
		20	34.37	50.37	6.84	0.684	7	7.15	178.6	128.3
East wall	Wall (A)	-	-	-	75	7.5	-		-	-
	Wall (C), Glass wool	10	4.46	20.46	12.9	1.29	3.29	6.2	155.25	134.8
		15	6.69	22.69	9.1	0.91	3.4	6.6	164.7	142
		20	8.92	24.92	7.05	0.705	3.6	6.8	169.88	144.9
West wall	Wall (A)	-		-	75.44	7.544	-	-	-	-
	Wall (C), Glass wool	10	4.46	20.46	13	1.3	3.27	6.24	156.1	135.64
		15	6.69	22.69	9.18	0.918	3.42	6.62	165.63	142.94
		20	8.9	24.92	7.1	0.71	6.64	6.83	170.85	145.9
South wall	Wall (A)	-	-	-	72.23	7.223	-		-	-
	Wall (C), Glass wool	10	4.46	20.46	12.45	1.245	3.42	5.98	149.45	128.98
		15	6.69	22.69	8.8	0.88	3.57	6.34	158.57	135.88
		20	8.9	24.92	6.8	0.68	3.8	6.54	163.57	138.65

Table 6.8. Heat loss calculation of the full flat in the transient case.

Wall type	Orientation	Insulator thickness (cm)	Heat loss kWh/m^2/3months	The area of the three walls m^2	Heat loss of the three walls kWh/3months	Total Heat loss of the flat kWh/3months
Wall (A)	North	-	78.3	15.78	1235.574	2792.628
	East		75	10.35	776.25	
	West		75.44	10.35	780.804	
Wall (B), EPS	North	10 cm	13.18	15.78	207.98	470.145
	East		12.63	10.35	130.72	
	West		12.7	10.35	131.445	
	North	15 cm	9.3	15.78	146.754	331.812
	East		8.9	10.35	92.115	
	West		8.98	10.35	92.943	
	North	20 cm	7.188	15.78	113.42	256.65
	East		6.89	10.35	71.3	
	West		6.95	10.35	71.93	
Wall (C), Glass wool	North	10 cm	13.46	15.78	212.4	480.45
	East		12.9	10.35	133.5	
	West		13	10.35	134.55	
	North	15 cm	9.5	15.78	149.91	339.53
	East		9.12	10.35	94.4	
	West		9.2	10.35	95.22	
	North	20 cm	7.355	15.78	116	262.48
	East		7.049	10.35	73	
	West		7.1	10.35	73.48	
Wall (D), Rock wool	North	10 cm	12.6	15.78	198.8	449.7
	East		12.07	10.35	125	
	West		12.165	10.35	125.9	
	North	15 cm	8.87	15.78	139.97	316.61
	East		8.499	10.35	88	
	West		8.564	10.35	88.64	
	North	20 cm	6.844	15.78	108	245.14
	East		6.56	10.35	67.9	
	West		6.69	10.35	69.24	

6.3.4. Calculation of the optimum thicknesses

Figure 6.12 shows a comparison between the walls in terms of the total energy saving cost in 25 years and total initial cost as a function of insulation thickness for the north wall. Based on these, Figure 6.13 displays the optimum insulation thicknesses for the three walls according to total life cycle saving and payback time. The optimum thicknesses can be seen in Table 6.9.

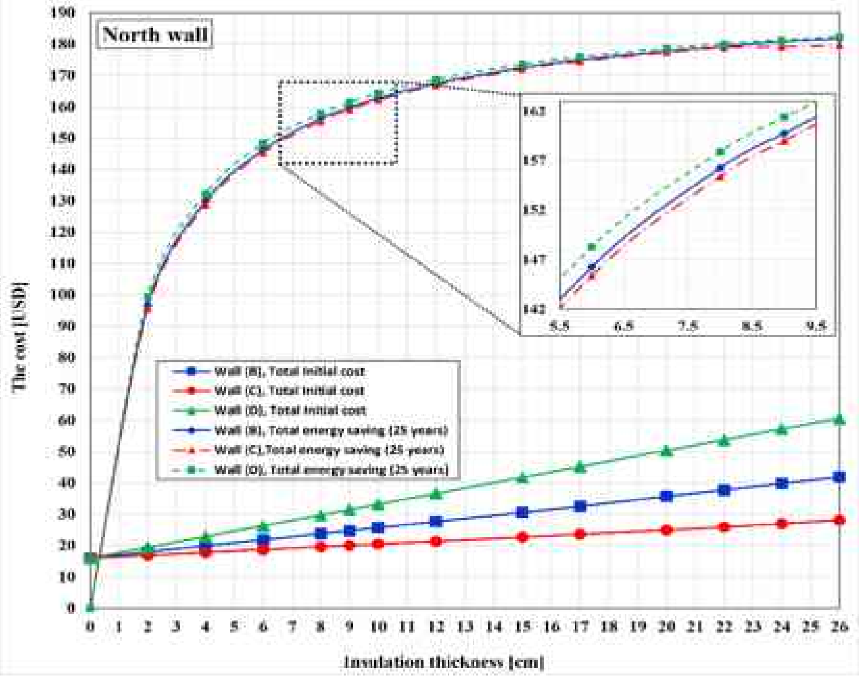

Figure 6.12. The total saved energy cost in 25 years and the total initial cost as a function of insulation thickness for the three walls facing north.

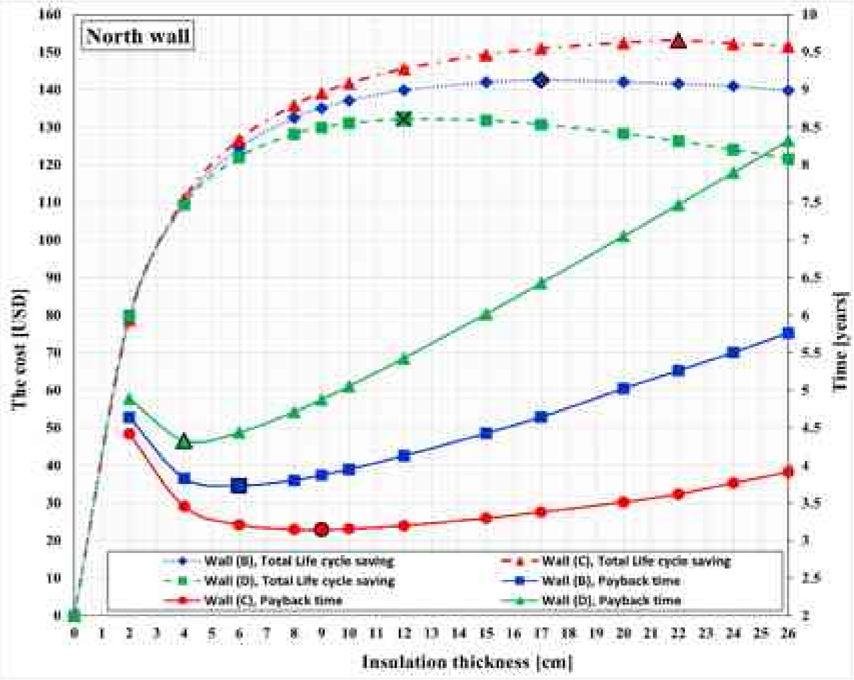

Figure 6.13. The optimum insulation thickness for the three walls according to total life cycle saving and payback time, where the right axis refers to time in years , and the left one refers to the cost in USD.

Table 6.9. Optimum insulation thickness for various wall types in the case of north orientation.

The direction	Wall type	Insulator type	Optimum insulation thickness (cm)	Total Life cycle saving (USD)	Payback time (years)
North	Wall (B)	EPS	17	142.5	3.73
	Wall (C)	Glass wool	22	153	3.14
	Wall (D)	Rock wool	12	132.12	4.33

I chose the best insulation, which is in Wall (C), in terms of the best total life cycle saving and payback time, and I compared the four directions of this wall as shown in Figure 6.14 and Figure 6.15, and the optimal thicknesses is shown in the Table 6.10.

Table 6.10. Optimum insulation thickness, life cycle saving, and payback time: effect of wall orientation.

Wall type	**Wall (C), Glass-wool insulator**			
Orientation	North	East	West	South
Optimum insulation thickness (cm)	22	21.85	21.8	21.7
Total Life cycle saving (USD)	153	145.6	146.2	139.2
Payback time (years)	3.14	3.28	3.26	3.4

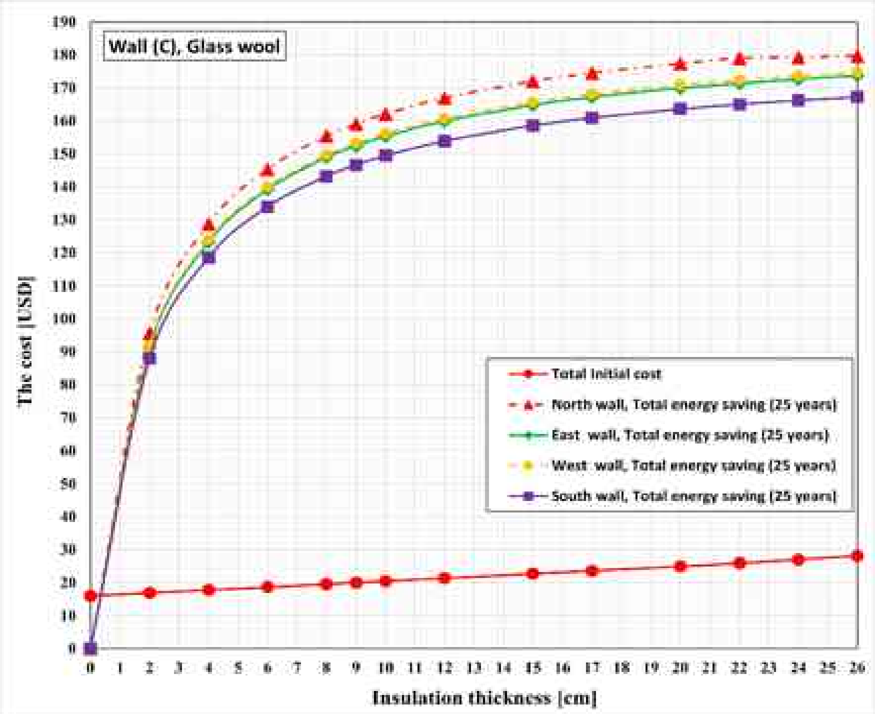

Figure 6.14. Total saved energy cost in 25 years as function of insulation thickness of wall (C).

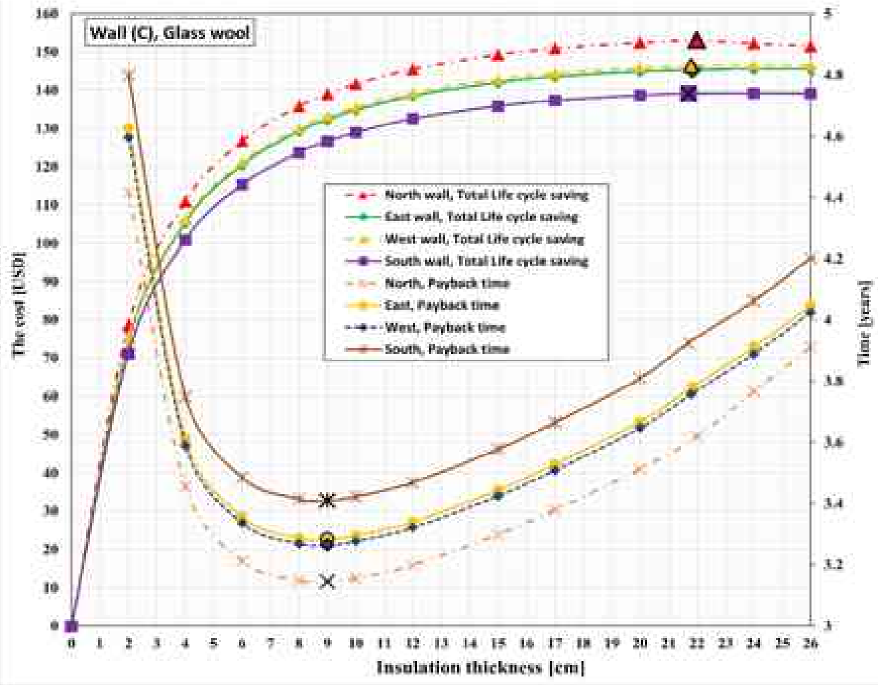

Figure 6.15. The optimum insulation thickness for the four orientations of the wall (C) according to total life cycle saving and payback time, where the right axis refers to time in years , and the left one refers to the cost in USD.

According to these data, the optimal insulator thickness is the largest on the north wall and smallest on the south wall, as it is expected.

Until this point, I used the current energy prices for my calculations since they are uncertain for the future. Now I repeated some calculations with an inflation rate which can be considered as the average rate in Hungary. The following equation were used to compute the change in cost using the inflation rate:

$$F = V * (1 + i)^n \tag{6.2}$$

where, *F*: Future value, *V*: present value, *i*: inflation rate, n: number of years. The average inflation rate in Hungary as per World Bank data is 5% [108]. My analysis considers duration of 25 years starting date 2023 to 2048. The new results are shown in Figure 6.16, where one can see that inflation affected the optimal insulation thickness values. The optimal thickness is 22 cm without inflation rate and 24 cm when I take 5% inflation into account.

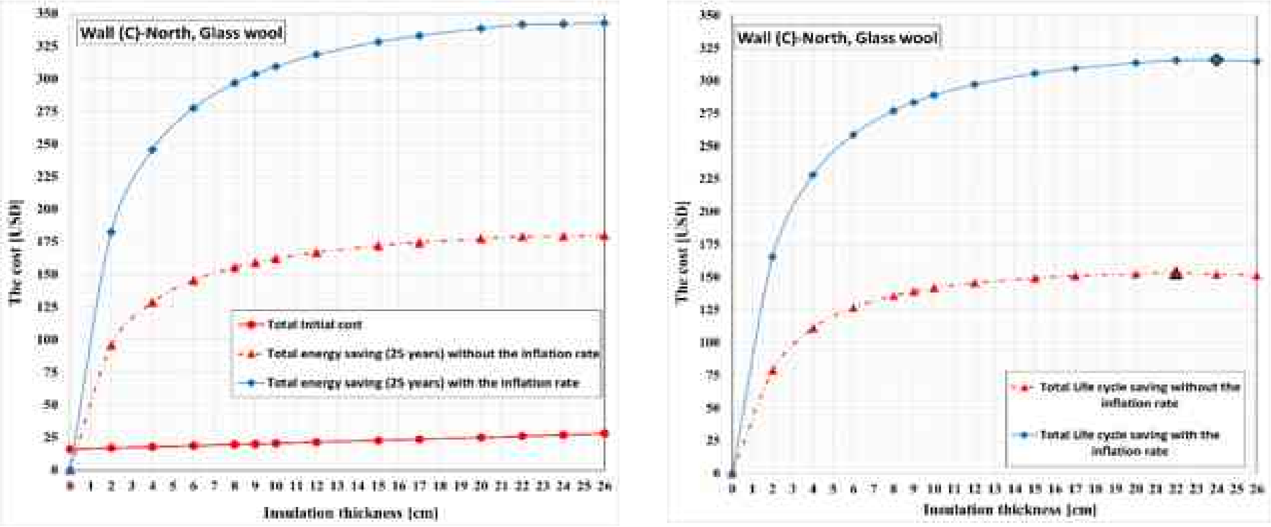

Figure 6.16. The optimum insulation thicknesses for the north direction of the wall (C) with cost according to the total energy saving on the left-side and total life cycle saving on the right-side.

6.3.5. Calculation of Heat Loss through walls in case of climate change

In the previous part, the temperature was calculated for the winter of 2022-2023. The question immediately arises how much the obtained results depend on the concrete temperature (and wind speed, etc.) functions. Moreover, due to the accelerating climate change, the fluctuations in temperature will be more extreme. To give a first, tentative answer to these questions, the temperature of the five warmest days is increased by 5 C° and the temperatures of the five coldest days are decreased by 5 C° as shown in Figure 6.17. The average temperature as well as all other variables and circumstances remain the same and the heat loss of wall (C) is calculated again for this case for all the four orientations, as shown in Table 6.11.

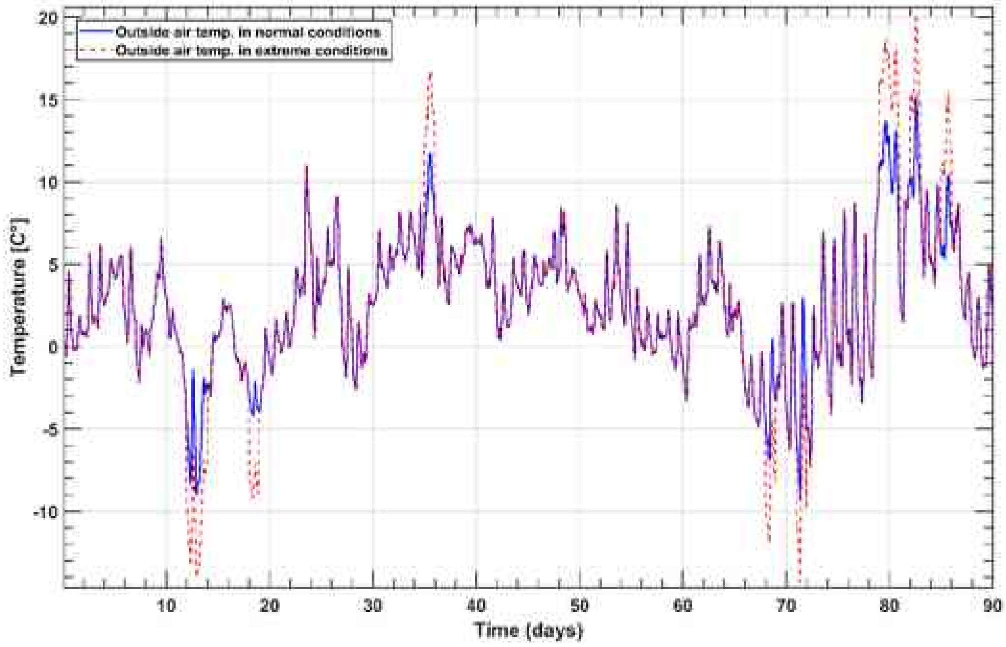

Figure 6.17. The outside air temperature as a function of time in the normal and extreme conditions.

Table 6.11. Heat loss through a wall of 1 m^2 of each of the four kinds in the transitory situation, along with the energy cost in USD and the payback period in years under extreme circumstances.

Wall type	Orientation	Insulator thickness (cm)	Heat loss kWh/m^2/3months		Difference %
			Extreme conditions	Normal conditions	
Wall (A)	North	-	78	78.3	0.384
	East		74.7	75	0.4
	West		75	75.44	0.58
	South		72	72.23	0.32
Wall (C), Glass wool	North	10 cm	13.22	13.46	1.8
	East		12.65	12.9	1.95
	West		12.72	13	2.18
	South		12.26	12.45	1.53
	North	15 cm	9.34	9.5	1.7
	East		8.89	9.12	2.55
	West		8.9	9.2	3.31
	South		8.59	8.8	2.4
	North	20 cm	7.17	7.355	2.54
	East		6.8	7.049	3.6
	West		6.91	7.1	2.71
	South		6.61	6.8	2.83

The difference between the data for the original and the extreme weather is quite small and always the latter one is smaller. The reason for this is that I considered linear heat conduction and convection, (only the radiation is nonlinear), thus the average temperature is the most important single factor. The extreme weather yields smaller heat loss probably due to the convexity of the radiative heat gain function $\alpha_{\mathrm{Low}}\varepsilon_r\sigma_r\left[u_r(t)\right]^4$. For example, $270^4 < \left(260^4 + 280^4\right)/2$.

6.4. Summary of this chapter

I found the total heat loss (without insulation, with a thickness of 10 cm EPS, glass-wool, and Rock-wool) was, respectively, 2792.628, 470.145, 480.45, and 449.7 kWh for a full flat. The optimum insulation thicknesses for the north-facing wall are 17, 22, and 12 cm, and the life cycle energy savings are 142.5, 153, and 132.12 kWh/m^2 for the EPS, glass-wool, and rock-wool, respectively, and the payback times are 3.73, 3.14, and 4.33 years. The optimal insulating properties can be achieved with 22 cm of thick glass wool, which only slightly depends on the wall orientation, according to the life cycle analysis. During optimization, it was assumed that the insulation cost increases linearly with insulation thickness, whereas the energy cost is found to decrease as insulation thickness increases, while the principle of diminishing returns is fulfilled. The optimum thickness varies based on the substance and how it interacts with the outdoor environment. The current conclusions and results, and particularly the relative performance of glass wool, depend on the specific values of the parameters employed in the economic and thermal analysis.

7. THESES – NEW SCIENTIFIC RESULTS

T1. I tested 10^5 different leapfrog-hopscotch algorithm combinations to solve the heat or diffusion equation by combining the hopscotch space structure with leapfrog time integration. I applied the well-known theta method with nine different values of θ and the recently invented CNe method. Then I generated preliminary numerical results, and using these as a basis, I selected the five most effective methods for more research. The effectiveness of the chosen methods was evaluated for two 2-dimensional systems containing 10000 cells in the case of large systems with completely discontinuous random parameters and initial conditions. I showed the competitiveness of the suggested methods by demonstrating that they can provide results with acceptable accuracy orders of magnitude quicker than the well-optimized MATLAB routines [85].

T2. I extensively investigated 13 numerical methods to solve the linear heat conduction equation in building walls. Eight of the used methods are recently invented explicit algorithms including those mentioned in T1 which are unconditionally stable. First, I performed verification tests in a 2D case by comparing them to analytical solutions, using non-equidistant grids. Then I tested them on real-life applications in the case of one-layer (brick) and two-layer (brick and insulator) walls to determine how the errors depend on the real properties of the materials, the mesh type, and the time step size. I applied zero Dirichlet boundary on the walls. The results show that the best algorithm is usually the original odd-even hopscotch method for uniform cases and the leapfrog-hopscotch and the Dufort-Frankel algorithms for non-uniform cases [92].

T3. I extensively measured the running times of the most successful methods and compared them to the performance of other available solvers, for example, ANSYS transient thermal analysis and the built-in routines of MATLAB. I systematically evaluated how the performance of the various solvers (including MATLAB routines and ANSYS) depends on the mesh settings to see which one is optimal for certain accuracy requirements. I used three mesh sizes: 40 × 40, 80 × 80, and 120 × 120. I showed that the running time of my methods changes linearly with mesh size, unlike in the case of other methods. Three simple analytical solutions of the heat equation were used with an equidistant mesh for verification in the case of homogeneous material properties (one brick layer). All the methods used and the ANSYS solvers are confirmed to be convergent. These experiments suggested that my methods are better than all ANSYS solvers and MATLAB routines, whereas ANSYS was less accurate and slower, and I observed that the best performance was achieved by the leapfrog–hopscotch and the Dufort-Frankel algorithms with the pseudo-implicit treatment of the nonlinear radiation term [94].

T4. I developed and tested a simulation methodology based on fundamental physical principles and laws (ab initio approach) to study transient heat transfer in a two-dimensional wall without an insulator, with an insulator, and different types of thermal bridges. I made a long-term simulation (one full winter) of two-dimensional space systems to test the two best versions of the methods mentioned in T3. The real-life engineering problem I solved is the examination of thermal bridges with different shapes in buildings to increase energy efficiency. The temperature distribution and total heat losses of all cases were calculated. I found the straight thermal bridge to be energetically worse than others. I conclude that the numerical simulation methodology is established in my work [94].

T5. I made transient heat transfer simulations through multilayer walls with different materials and thicknesses that is subject to the typical external temperature and solar radiation specific to Hungary's environment using the leapfrog-hopscotch and modified Dufort-Frankel algorithms. I calculated the winter heating loss (across the walls) and temperature distribution of all cases according to wall orientations. I found the transient way to calculate the heat loss to be quick and accurate more than steady-state calculations based on methods like the Degree-days, which is frequently rather inaccurate [102].

T6. Using the leapfrog-hopscotch and modified Dufort-Frankel algorithms to make transient heat transfer simulations through multilayer walls, I determined the optimum thickness of insulation, energy savings, and payback time using an economic model that considers the orientation of the external walls, solar radiation, the cost of insulation materials, the present cost of energy consumption, and the cost over the 25-year lifetime of a building in Miskolc City, and a case is examined in the cold season. Additionally, it was looked at how well the walls conducted heat under optimal conditions. Comparing this study to others of a similar type, one of its unique characteristics was the use of less expensive local materials, to optimize investment on thermal insulation. I investigated three materials and a range of thicknesses: Expanded Polystyrene (EPS), glass-wool, and rock-wool. The results demonstrated a small, but noticeable impact of wall orientation on the thermal efficiency of the walls during the winter season of the environment under consideration. The small difference between northern and southern orientation can be explained by the fact that the first part of the winter is typically rather cloudy in Hungary. The optimal insulating properties can be achieved with 22 cm of thick glass wool, which only slightly depends on the wall orientation, according to the life cycle energy savings analysis [102].

ACKNOWLEDGEMENTS

First and foremost, I owe a tremendous debt of gratitude to Professor Dr. Endre Kovács, who has been instrumental in my success over the past four years and introduced me to this interesting research field. I feel incredibly fortunate to have had the guidance and mentorship of such a highly respected academic. I appreciate all of your wonderful counsel, support, and persistent and understanding assistance. Aside from studies, I'm happy to have met someone with such grace and friendliness. I highly valued the bidaily meetings we held, which not only served as crucial checkpoints to keep me on track academically, but also provided me with plenty of encouragement. I am profoundly grateful for the immeasurable contributions he made to my development.

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

I would like also to thank my friends Ali Habeeb Askar, and Mahmoud Saleh, you have been my biggest cheerleader throughout this journey, and I cannot thank you enough for your love and support.

Thanks for my team mates and colleagues, Ádám Nagy, Dr. Pszota Gabor, Tamás Jenyó who helped me in many different ways.

Lastly, I want to express my deepest gratitude to my family whose belief in my abilities and support. Your encouragement played an integral role in my accomplishments. To my mother Eman, and my brother Hasan and my sisters: Rasha and Maya, To the soul of my father, who has never left me since the beginning of my study: Thank you for everything. I dedicate this PhD thesis to you.

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 in my life, as it constituted a bright point in my professional life.

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 remarkable how life leads us to unexpected places and introduces us to remarkable people when we least expect it.

REFERENCES

[1] "Yunus Cengel, Afshin Ghajar-Heat and Mass Transfer_ Fundamentals and Applications-McGraw-Hill Science_Engineering_Math (2014)".

[2] F. P. Incropera, D. P. DeWitt, T. L. Bergman, A. S. Lavine, and et al, *Fundamentals of heat and mass transfer*, vol. 6. Wiley New York, 1996.

[3] A. Giri and P. E. Hopkins, "A Review of Experimental and Computational Advances in Thermal Boundary Conductance and Nanoscale Thermal Transport across Solid Interfaces," *Adv Funct Mater*, vol. 30, no. 8, p. 1903857, Feb. 2020, doi: 10.1002/ADFM.201903857.

[4] L. M. Jiji, "Heat conduction: Third edition," *Heat Conduction (Third Edition)*, pp. 1–418, 2009, doi: 10.1007/978-3-642-01267-9/COVER.

[5] P. Nithiarasu, R. W. Lewis, and K. N. Seetharamu, "Fundamentals of the Finite Element Method for Heat and Mass Transfer Second Edition."

[6] A. Standard, "Thermal environmental conditions for human occupancy," *ANSI/ASHRAE, 55*, vol. 5, 1992.

[7] A. Guide and et al, "Environmental design," *Chartered Institute of Building Services Engineers (CIBSE)*, 2006.

[8] B. Berggren and M. Wall, "State of knowledge of thermal bridges—A follow up in Sweden and a review of recent research," *Buildings*, vol. 8, no. 11, p. 154, 2018.

[9] A. Hilke and L. Ryan, "Mobilising investment in energy efficiency," 2012.

[10] C. Liu, S. Sharples, and H. Mohammadpourkarbasi, "A Review of Building Energy Retrofit Measures, Passive Design Strategies and Building Regulation for the Low Carbon Development of Existing Dwellings in the Hot Summer--Cold Winter Region of China," *Energies (Basel)*, vol. 16, no. 10, p. 4115, 2023.

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

[12] B. Dai *et al.*, "Environmental and economical analyses of transcritical CO2 heat pump combined with direct dedicated mechanical subcooling (DMS) for space heating in China," *Energy Convers Manag*, vol. 198, p. 111317, 2019.

[13] S. Amiri, M. Mazaheri, and N. Bavandpouri Gilan, "Introducing a new method for calculating the spatial and temporal distribution of pollutants in rivers," *International Journal of Environmental Science and Technology*, pp. 1–18, 2021.

[14] Swedish Standards Institute, "Thermal Performance of Buildings—Transmission and Ventilation Heat Transfer Coefficients—Calculation Method. EN ISO 13789: 2017." Accessed: Apr. 20, 2023. [Online]. Available: https://www.iso.org/standard/65713.html

[15] Swedish Standards Institute, "Thermal Bridges in Building Construction—Heat Flows and Surface Temperatures—Detailed Calculations. EN ISO 10211: 2017." Accessed: Apr. 20, 2023. [Online]. Available: https://www.iso.org/%0Astandard/65710.html

[16] Swedish Standards Institute, "Thermal Bridges in Building Construction—Linear Thermal Transmittance— Simplified Methods and Default Values. EN ISO 14683: 2017." Accessed: Apr. 20, 2023. [Online]. Available: https://www.iso.org/standard/65706.html

[17] A. Alhawari and P. Mukhopadhyaya, "Thermal bridges in building envelopes--An overview of impacts and solutions," *International Review of Applied Sciences and Engineering*, vol. 9, no. 1, pp. 31–40, 2018.

[18] D. Borelli, P. Cavalletti, A. Marchitto, and C. Schenone, "A comprehensive study devoted to determine linear thermal bridges transmittance in existing buildings," *Energy Build*, vol. 224, p. 110136, 2020.

[19] K. Karabulut, E. Buyruk, and A. Fertelli, "Numerical investigation of the effect of insulation on heat transfer of thermal bridges with different types," *Thermal Science*, vol. 20, no. 1, pp. 185–195, 2016.

[20] X. Cao, X. Dai, and J. Liu, "Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade," *Energy Build*, vol. 128, pp. 198–213, 2016.

[21] T. Li, J. Xia, C. S. Chin, and P. Song, “Investigation of the Thermal Performance of Lightweight Assembled Exterior Wall Panel (LAEWP) with Stud Connections,” *Buildings*, vol. 12, no. 4, p. 473, 2022.

[22] K. A. Barber and M. Krarti, “A review of optimization based tools for design and control of building energy systems,” *Renewable and Sustainable Energy Reviews*, vol. 160, no. January, p. 112359, 2022, doi: 10.1016/j.rser.2022.112359.

[23] D. I. Kolaitis, E. Malliotakis, D. A. Kontogeorgos, I. Mandilaras, D. I. Katsourinis, and M. A. Founti, “Comparative assessment of internal and external thermal insulation systems for energy efficient retrofitting of residential buildings,” *Energy Build*, vol. 64, pp. 123–131, 2013.

[24] N. Daouas, “A study on optimum insulation thickness in walls and energy savings in Tunisian buildings based on analytical calculation of cooling and heating transmission loads,” *Appl Energy*, vol. 88, no. 1, pp. 156–164, 2011, doi: 10.1016/j.apenergy.2010.07.030.

[25] N. Ghrab-Morcos, “CHEOPS: a simplified tool for thermal assessment of Mediterranean residential buildings in hot and cold seasons,” *Energy Build*, vol. 37, no. 6, pp. 651–662, 2005.

[26] P. A. A Aïssani, A Chateauneuf, JP Fontaine, “Cost model for optimum thicknesses of insulated walls considering indirect impacts and uncertainties,” *Energy Build*, vol. 84, pp. 21–32, 2014.

[27] M. K. Nematchoua *et al.*, “Study of the economical and optimum thermal insulation thickness for buildings in a wet and hot tropical climate: Case of Cameroon,” *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1192–1202, 2015.

[28] A. Ucar and F. Balo, “Determination of environmental impact and optimum thickness of insulation for building walls,” *Environ Prog Sustain Energy*, vol. 30, no. 1, pp. 113–122, 2011.

[29] N. Daouas, “A study on optimum insulation thickness in walls and energy savings in Tunisian buildings based on analytical calculation of cooling and heating transmission loads,” *Appl Energy*, vol. 88, no. 1, pp. 156–164, 2011.

[30] M. Subhash, J. A. Usmani, and S. Bhagman, “Study on energy saving in residential building using life cycle cost analysis,” *Int J Eng Math Sci*, vol. 4, pp. 8–17, 2013.

[31] D. P. P. Meddage, A. Chadee, M. T. R. Jayasinghe, and U. Rathnayake, “Exploring the applicability of expanded polystyrene (EPS) based concrete panels as roof slab insulation in the tropics,” *Case Studies in Construction Materials*, vol. 17, no. May, p. e01361, 2022, doi: 10.1016/j.cscm.2022.e01361.

[32] A. Hasan, “Optimizing insulation thickness for buildings using life cycle cost,” *Appl Energy*, vol. 63, no. 2, pp. 115–124, 1999, doi: 10.1016/S0306-2619(99)00023-9.

[33] B. Y. K Çomaklı, “Optimum insulation thickness of external walls for energy saving,” *Appl Therm Eng*, vol. 23, no. 4, pp. 473–479, 2003.

[34] M. Ozel, “Cost analysis for optimum thicknesses and environmental impacts of different insulation materials,” *Energy Build*, vol. 49, pp. 552–559, 2012.

[35] D. Acharjee, D. Jyoti Basu, and D. Bandyopadhyay, “A numerical study on the through thickness shear behavior of EPS sandwich panels,” *Mater Today Proc*, vol. 62, pp. 4379–4385, 2022, doi: 10.1016/j.matpr.2022.04.876.

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

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

[38] I. Pisarenko and E. Ryndin, “Numerical Drift-Diffusion Simulation of GaAs p-i-n and Schottky-Barrier Photodiodes for High-Speed AIIIBV On-Chip Optical Interconnections,” *Electronics (Basel)*, vol. 5, no. 4, p. 52, Sep. 2016, doi: 10.3390/electronics5030052.

[39] David Fisher, “Defects and Diffusion in Carbon Nanotubes Abstracts [1] Defects: 5-1DB.”

[40] C. Fradin, “On the importance of protein diffusion in biological systems: The example of the Bicoid morphogen gradient,” *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, vol. 1865, no. 11, pp. 1676–1686, Nov. 2017, doi: 10.1016/j.bbapap.2017.09.002.

[41] 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.

[42] 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.

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

[44] S. Haq and I. Ali, "Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials," *Eng Comput*, vol. 38, pp. 2059–2068, Aug. 2022, doi: 10.1007/s00366-021-01327-5.

[45] F. Suárez-Carreño and L. Rosales-Romero, "Convergency and stability of explicit and implicit schemes in the simulation of the heat equation," *Applied Sciences (Switzerland)*, vol. 11, no. 10, May 2021, doi: 10.3390/app11104468.

[46] S. A. Lima, M. Kamrujjaman, and M. S. Islam, "Numerical solution of convection-diffusion-reaction equations by a finite element method with error correlation," *AIP Adv*, vol. 11, no. 8, p. 85225, Aug. 2021, doi: 10.1063/5.0050792/968138.

[47] M. Ivanovic, M. Svicevic, and S. Savovic, "Numerical solution of Stefan problem with variable space grid method based on mixed finite element/finite difference approach," *Int J Numer Methods Heat Fluid Flow*, vol. 27, no. 12, pp. 2682–2695, 2017, doi: 10.1108/hff-11-2016-0443.

[48] D. A. Anderson, J. C. Tannehill, R. H. Pletcher, R. Munipalli, and V. Shankar, "Computational Fluid Mechanics and Heat Transfer; Fourth Edition." [Online]. Available: https://www.routledge.com/

[49] 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, 2021.

[50] J. Zhang, C. Zhao, and T. Tang, "Sharp error estimate of BDF2 scheme with variable time steps for molecular beam expitaxial models without slop selection," 2021, doi: 10.13140/RG.2.2.24714.59842.

[51] 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, 2022, doi: 10.1080/00207160.2021.1954621.

[52] I. Ali, S. Haq, K. S. Nisar, and S. U. Arifeen, "Numerical study of 1D and 2D advection-diffusion-reaction equations using Lucas and Fibonacci polynomials," *Arabian Journal of Mathematics*, vol. 10, no. 3, pp. 513–526, Dec. 2021, doi: 10.1007/s40065-021-00330-4.

[53] S. Haq, M. Hussain, and A. Ghafoor, "A computational study of variable coefficients fractional advection–diffusion–reaction equations via implicit meshless spectral algorithm," *Eng Comput*, vol. 36, no. 4, pp. 1243–1263, Oct. 2020, doi: 10.1007/s00366-019-00760-x.

[54] F. Gagliardi, M. Moreto, M. Olivieri, and M. Valero, "The international race towards Exascale in Europe," *CCF Transactions on High Performance Computing*, vol. 1, no. 1. Springer, pp. 3–13, May 06, 2019. doi: 10.1007/s42514-019-00002-y.

[55] I. Z. Reguly and G. R. Mudalige, "Productivity, performance, and portability for computational fluid dynamics applications," *Comput Fluids*, vol. 199, p. 104425, Mar. 2020, doi: 10.1016/J.COMPFLUID.2020.104425.

[56] F. Sanjaya and S. Mungkasi, "A simple but accurate explicit finite difference method for the advection-diffusion equation," in *Journal of Physics: Conference Series*, 2017, p. 12038.

[57] A. Y. Al-Bayati, S. A. Manaa, and A. M. 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.

[58] C. Nwaigwe, "An Unconditionally Stable Scheme for Two-Dimensional Convection-Diffusion-Reaction Equations," *no. January*, 2022, [Online]. Available: https://www.researchgate.net/publication/357606287_An_Unconditionally_Stable_Scheme_for_Two-Dimensional_Convection-Diffusion-Reaction_Equations

[59] 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.

[60] S. Mahmoud, N. Ádám, and K. Endre, "Construction and investigation of new numerical algorithms for the heat equation: Part III," *Multidiszciplináris Tudományok*, vol. 10, no. 4, pp. 349–360, 2020.

[61] Á. Nagy, M. Saleh, I. Omle, H. Kareem, and E. Kovács, "New stable, explicit, shifted-hopscotch algorithms for the heat equation," *Mathematical and Computational Applications*, vol. 26, no. 3, p. 61, 2021.

[62] 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.

[63] I. Omle, A. H. Askar, and E. Kovács, “Systematic testing of explicit positivity preserving algorithms for the heat-equation,” *J. Math. Comput. Sci.*, vol. 12, no. 0, p. Article ID 162, 2022, doi: 10.28919/10.28919/JMCS/7407.

[64] P. Gordon, “Nonsymmetric difference equations,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 13, no. 3, pp. 667–673, 1965.

[65] A. R. GOURLAY, “Hopscotch: a Fast Second-order Partial Differential Equation Solver,” *IMA J Appl Math*, vol. 6, no. 4, pp. 375–390, 1970, doi: 10.1093/imamat/6.4.375.

[66] 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.

[67] 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.

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

[69] O. Issa, “New explicit algorithm based on the asymmetric hopscotch structure to solve the heat conduction equation,” *Multidiszciplináris Tudományok*, vol. 11, no. 5, pp. 233–244, 2021.

[70] 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.

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

[72] 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.

[73] M. Munka and J. Pápay, *4D Numerical Modeling of Petroleum Reservoir Recovery*. Budapest: Akadémiai Kiadó, 2001.

[74] 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.

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

[76] 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.

[77] 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.

[78] M. Saleh, Á. Nagy, and E. Kovács, “Part 3: Construction and investigation of new numerical algorithms for the heat equation,” *Multidiszciplináris tudományok*, vol. 10, no. 4, pp. 349–360, 2020, doi: 10.35925/j.multi.2020.4.38.

[79] 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.

[80] 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.

[81] “Heun’s method - Wikipedia.” Accessed: Jul. 30, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Heun%27s_method

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

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

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

[85] 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.

[86] “Leapfrog integration.” Accessed: Jan. 12, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Leapfrog_integration

[87] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. Boca Raton: Taylor & Francis, 1989.

[88] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, vol. 1. Elsevier, 2001.

[89] A. Iserles, “Generalized Leapfrog Methods,” *IMA Journal of Numerical Analysis*, vol. 6, no. 4, pp. 381–392, Oct. 1986, doi: 10.1093/IMANUM/6.4.381.

[90] J. G. Verwer and B. P. Sommeijer, “Stability Analysis of an Odd-Even-Line Hopscotch Method for Three-Dimensional Advection-Diffusion Problems,” *Source: SIAM Journal on Numerical Analysis*, vol. 34, no. 1, pp. 376–388, 1997.

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

[92] H. Kareem Jalghaf, I. Omle, and E. Kovács, “A Comparative Study of Explicit and Stable Time Integration Schemes for Heat Conduction in an Insulated Wall,” *Buildings*, vol. 12, no. 6, p. 824, 2022.

[93] M. Bastani and D. K. Salkuyeh, “A highly accurate method to solve Fisher’s equation,” *Pramana - Journal of Physics*, vol. 78, pp. 335–346, 2012, doi: 10.1007/s12043-011-0243-8.

[94] 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, Jun. 2023, doi: 10.3390/en16124604.

[95] E. M. Alawadhi, *Finite element simulations using ANSYS*. CRC Press, 2009.

[96] R. S. Amando, “Numerical computation of internal and external flows, volume 1: Fundamentals of numerical discretization: C. Hirsch John Wiley \& Sons, Ltd., 1988.” Elsevier, 1989.

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

[98] “Weather-Online Website.” Accessed: May 17, 2023. [Online]. Available: https://www.worldweatheronline.com/miskolc-weatherhistory/miskolc/hu.aspx

[99] 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, pp. 125–134, 2023.

[100] 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.

[101] “Accurate Weather Forecasts for Any Location.” [Online]. Available: https://open-meteo.com/

[102] 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*, p. 101476, 2023.

[103] “properties of insulating materials,” hoszigetelesplaza.hu. Accessed: Apr. 01, 2023. [Online]. Available: https://www.hoszigetelesplaza.hu/eps-80-homlokzati-hoszigetelo-lemez

[104] B. Bolló, “Authentic Energy Certificate by Decree No. 7/2006.,” 2021.

[105] L. Area, V. Velocity, D. Force, M. Pressure, and D. V. Kinematic, *Heat Transfer Tenth Edition*.

[106] “Heating degree days in Hungary, 2000-2020.” Accessed: May 17, 2023. [Online]. Available: https://www.iea.org/data-and-statistics/charts/heating-degree-days-in-hungary-2000-2020

[107] “Degree Day Calculator.” Accessed: May 17, 2023. [Online]. Available: https://energy-data.io/members/degree-day-calculator-free/

[108] “World Bank, T., World Bank open data,” 2018. [Online]. Available: https://data.worldbank.org/country/hungary

LIST OF PUBLICATIONS RELATED TO THE TOPIC OF THE RESEARCH FIELD

(1) Omle, I.; Kovács, E.; Bolló, B. Applying Recent Efficient Numerical Methods for Long-Term Simulations of Heat Transfer in Walls to Optimize Thermal Insulation. *Elsevier, Results Eng.* 2023, 101476, doi: 10.1016/j.rineng.2023.101476.

(2) 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.

(3) Kareem, H.; Omle, I.; Kovács, E. A Comparative Study of Explicit and Stable Time Integration Schemes for Heat Conduction in an Insulated Wall. Buildings 2022, 12. https://doi.org/10.3390/buildings12060824.

(4) 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.

(5) 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, p. 92, 2021.https://doi.org/10.3390/computation9080092

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

(7) Khayrullaev, H.; Omle, I.; Kovács, E. "Systematic Investigation of the Explicit, Dynamically Consistent Methods for Fisher's Equation". Computation 2024, 12, 49. https://doi.org/10.3390/computation12030049.

(8) 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.

(9) Omle, I. "New explicit algorithm based on the asymmetric hopscotch structure to solve the heat conduction equation," Multidiszcip. Tudományok, vol. 11, no. 5, pp. 233–244, 2021.

(10) Omle, I.; Kovács, E.; Bolló, B. "Multi-objective optimization and simulation for multiple models of walls to estimate heat gain using Artificial Neural Network model." Multidiszciplináris Tudományok 13.3 (2023): 159-174. https://doi.org/10.35925/j.multi.2023.3.17.

(11) Omle, I.; Kovács, E.; "Evaluate Recent Numerical Methods for Long-Term Simulation to Study the Effect of Different Shapes of Thermal Bridges in Walls". Advances in Mechanical Engineering (2024).

(12) Omle, I.; Kovács, E.; Bolló, B. "The Effect of Surface Triangular Roughness Inspired by Nature with Different Angles on the Interaction between Free Convection and Surface Radiation in a Square Cavity". Heat Transfer Research 2023, DOI: 10.1615/HeatTransRes.2023048854.

(13) 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

(14) Omle, I.; Kovács, E.; Bolló, B. "Testing the effect of radiation emissivity on heat convection inside a square enclosure". 9th International Zeugma Conference on Scientific Research: The book of full texts, Istanbul, Turkey : (2023) pp. 430-437. , 8 p. ISBN: 9786256404762.

(15) Abed, H. H., Omle, I., Askar, A. H., & Kovács, E. (2023). Experimental Study of the Stability and Thermophysical Properties for Different Particle Size of Al_2o_3-H_2o Nanofluid. Journal of Engineering Science and Technology, 18(6), 2793-2808.

(16) Omle, I.; Askar, A.H.; Kovács, E. "Optimizing the Design of Container Houses Using Argon and Recycled Plastic Materials". 2024 (**Under Review**, Q1)