UNIVERSITY OF MISKOLC
FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

Enhancing solar efficiency: Numerical simulation and optimization of the temperature-dependent power output

TITLE

PHD THESES

Prepared by

Husniddin Khayrullaev
Electrical Engineering , Electrical Technology and Electrical mechanics (BSc)
Electrical Mechanic (by branches) -a specialization (MSc)

ISTVÁN SÁLYI DOCTORAL SCHOOL OF MECHANICAL ENGINEERING SCIENCES
TOPIC FIELD OF BASIC ENGINEERING SCIENCES
TOPIC GROUP OF TRASPORT PROCESSES AND MACHINES

Head of Doctoral School

Dr. Gabriella Bognár
DSc, Full Professor

Head of Topic Group

Prof. Dr. László Baranyi
CSc, Full Professor

Scientific Supervisor

Dr. Endre Kovács
PhD, Associate Professor

Miskolc
2026

CONTENTS

SUPERVISOR'S RECOMMENDATIONS

Date

Supervisor

LIST OF SYMBOLS AND ABBREVIATIONS

NOMENCLATURE

SYMBOLS			GREEK LETTERS	
c	*Specific heat* [J/(kgK)]		ρ	*Mass density* [kg/m^3]
C	*Heat capacity* [J/K]		Δ	*Difference*
u	*Temperature* [K]		α	*Thermal diffusivity* [m^2/s]
L	*Thickness* [m]		Δt	*Time step size* [s]
K	*Convection coefficient* [1/s]		σ^*	*Value of Stefan-Boltzmann constant modified for non-black bodies* [W/(m^2. K^4)]
k	*Thermal conductivity* [W/(m. K)]		σ	*Coefficient of the radiation term* [s^{-1}K^{-3}]
h	*Heat transfer coefficient* [W/(m^{2}K)]		ε	*Emisivity*
R	*Thermal resistance* [K/W]		δ	*Factor to indicate daily mean temperature*
Q	*Heat transfer rate* [W]		M	*The system matrix*
q^*	*heat generation* [W/m^2]		r	*Mesh ratio*
q	*Heat source rate* [K/s]		***Subscripts***	
U	*Overall heat transfer coefficient* [W/(m^{2}K)]		a	*Ambient air*
t	*Time* [s]		l	*Left side*
S	*Area* [m^2]		r	*Right side*
v	*The air velocity* [m/s]		b,p,g,ins	*Brick, Panel, Gypsum and Insulation*
			c	*Convection*

The Methods	
ode15s	*A first- to fifth-order (implicit) numerical differentiation formulas with variable-step and variable order (VSVO), developed for solving stiff problems*
LH	*Leapfrog–Hopscotch*
DF	*Dufort and Frankel*

1. INTRODUCTION

1.1. The Importance of Heat Equation

Fickian diffusion, or Fourier-type heat conduction, is one of the most fundamental processes for mass or energy transport. In the simplest case, it can be described by a single linear partial differential equation (PDE) in space and time:

$$\frac{\partial u(x,t)}{\partial t} = \alpha \frac{\partial^2 u(x,t)}{\partial x^2} + q\,, \tag{1.1}$$

where $x,\ t,\ \in \mathbb{R}, u = u(x,t)$ is the concentration (temperature in case of heat conduction) as a function of space and time, $\alpha = k/(c\rho)$ is the thermal diffusivity, where $c = c(\vec{r},t)$ $\rho = \rho(\vec{r},t)$, $k = k(\vec{r},t)$ are the specific heat, the density and the heat conductivity of the material, respectively. Here, q is the heat generation or source term. Generalisations of this equation, including the advection-diffusion equation and several types of advection-diffusion-reaction equations [1], can simulate particle movement and propagation in physical, chemical, and biological systems [2]. Similar PDEs are used to simulate fluid flow through porous media, including moisture, ground water [3,4].

If the conductivity depends on space, then, instead of dealing with the $\alpha \frac{\partial^2 u(x,t)}{\partial x^2}$ phrase in one space dimension, I have to deal with the term

$$\frac{1}{c(x)\rho(x)} \frac{\partial}{\partial x}\left(k(x) \frac{\partial u}{\partial x} \right). \tag{1.2}$$

One can use a more general form instead of (1.1), valid in arbitrary space dimensions:

$$\frac{\partial u}{\partial t} = \frac{\nabla(k \nabla u)}{c\rho} + q\,,$$

where ∇ is the Nabla differential vector operator, $\nabla = \vec{i}\left(\partial/\partial x\right) + \vec{j}\left(\partial/\partial y\right) + \vec{k}\left(\partial/\partial z\right)$.

Many numerical approaches have been developed to solve Equation (1.1) and its generalisations. Most of them belong to the broad family of finite difference schemes (FDM) [5,6], which frequently indicates a form of method of lines [1]. The spatial variables are discretised first to generate an ordinary differential equation (ODE) system, and then an ODE solver is applied. These techniques can be either explicit or implicit, with some having a mix of both [7]. If the physical parameters, or equation coefficients, are very non-uniform in space, the eigenvalues of the system matrix may vary by several orders of magnitude, indicating a stiff

problem. For conventional explicit time-integration schemes such as Runge–Kutta and Adams–Bashforth methods, the Courant–Friedrichs–Lewy (CFL) condition imposes a very restrictive stability limit, particularly for diffusion-dominated problems, where the time step size must remain below a small mesh Fourier number to avoid numerical instability, often leading to excessively slow simulations. Although implicit methods offer significantly improved stability properties, they necessitate the solution of large systems of algebraic equations, which can be computationally expensive for large-scale problems.

1.2. The Fisher's Equation

Nonlinear reaction–diffusion partial differential equations (PDEs) play an important role in modeling phenomena in many branches of science. Fisher's equation, commonly known as the Fisher-Kolmogorov-Petrovsky-Piskunov equation [8], adds an additional logistic reaction term to the usual diffusion term:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta u(1-u)\,. \tag{1.3}$$

Here, u is the unknown function and β is a non-negative parameter. In this case, the range of u is the unit interval $[0, 1]$. The equation is the conventional heat equation without the logistic element, where u is the temperature, which is frequently expressed in Kelvin units. The spread of gene variations in space, the growth and distribution of misfolded proteins in neuro-physiology [9], and the propagation of fronts in combustion processes [10] are all modelled by Eq. (1.3). Some chemotaxis models [11], can be considered as further generalization of Fisher's equation.

It is worth noting that the Cole-Hopf transformation was used by Ma and Fuchssteiner [12] to provide travelling wave analytical solutions to a type of Kolmogorov-Petrovsky-Piskunov equation (generalised Fisher's equation). Regarding this specific equation, exact N-soliton solutions in (1+1) dimensions can be supplied via binary Darboux transformations, while a broader approach to nonlinear integrable PDEs involves exploring multiple component extensions, such as four- and six-component integrable equations

Numerous research groups have numerically solved Fisher's equation. For instance, Tang and Weber [13] employed a Petrov-Galerkin finite element approach that included Gaussian elimination. A one-dimensional semi-implicit finite difference system with first-order accuracy in time and second-order accuracy in space was created by Chandraker et al. [14]. An exponential B-spline collocation method was proposed by Dag and Ersoy [15] which uses the exponential cubic B-spline in space and the Crank–Nicolson method in time. Tamsir and Huntul suggested [16] a new hybrid method based on cubic uniform algebraic trigonometric tension B-spline functions and differential quadrature method.

It is well known that the proper solution of the diffusion or heat equation always adheres to the minimum and maximum principles [17] (p. 87). The minimum principle implies the so-called positivity-preserving characteristic, which states that the equation does not generate negative

values if the initial and boundary values are non-negative. When a source or reaction term exists in the diffusion equation, these rules generally do not apply. However, in the case of Fisher's equation, a similar situation exists due to the logistic nature of the reaction term. In the case of Eq. (1.2), if the initial and boundary values of the u variable are in the unit interval, then u remains in this interval for any non-negative values of the parameter β. Therefore, the applied numerical schemes should also preserve this property of the solution, which may be called dynamical consistency. I note that the property 'dynamical consistency' of numerical algorithms can have several definitions [18], formulating that the numerical solutions reflect important properties of the true solutions of the original system. Standard finite difference or finite element approaches do not guarantee dynamical consistency. As a result, the solutions generated by these algorithms may occasionally contain negative and non-physical values. Subsequently, the solutions may oscillate, resulting in numerical instabilities.

Some researchers are using unconditionally positive schemes [19,20] to solve comparable equations. The positivity of solutions has also been included into numerical methods for solving ordinary differential equations [21]. Chertock and Kurganov developed a positivity-preserving approach for a set of advection-reaction-diffusion equations that define chemotaxis/haptotaxis models [22]. However, their solution maintains positivity only when the time step size is less than the CFL number, which is rather tiny. The same may be stated of nonstandard finite difference schemes (NSFD), which have been utilised to solve the Fisher and Nagumo equations by Agbavon et al. [23], [24], and to solve cross-diffusion equations by Chapwanya et al. [25].

1.3. The Huxley's Equation

Huxley's equation [26] is considered in the following form:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta u^2 (1-u) \tag{1.4}$$

where α and β are nonnegative real numbers. Solutions of the diffusion or heat equation satisfy the minimum and maximum principles. [27]. The minimum principle guarantees that the equation does not produce negative values if the initial and boundary values are non-negative, which is called the positivity-preserving property. When there is a reaction or other source term in the equation, these criteria are not always applicable. Implicit methods, such as the so-called shifted airfoil collocation method applied by Anjuman et al. [28] for nonlinear drift-diffusion-reaction equations, have better stability properties, albeit sometimes step size limitations can be required. However, they demand the solution of a nonlinear algebraic equation system, which can be time- and memory-consuming if the size of the equation system is large. In spite of this, diffusion-reaction equations are often solved by implicit methods. Moreover, the diffusion term is sometimes even treated exactly [29] in the case of a stiff problem.

A broad spectrum of approaches has been proposed, reflecting the problem's complexity. Spectral and high-order methods, such as the shifted Chebyshev spectral collocation technique [30], offer excellent accuracy for smooth solutions. Semi-analytical methods, including the

reduced differential transform method and variational iteration approaches, provide closed-form approximations under specific conditions [31, 32]. The advent of machine learning has introduced physics-informed neural networks (PINNs) and their improved variants as flexible tools for solving forward and inverse problems [33, 34]. Classical finite difference methodology has also evolved through weighted average formulations [35].

Despite this rich array of techniques, a critical examination reveals persistent limitations, particularly when transitioning from idealized benchmarks to realistic application scenarios. Many of the aforementioned methods are constructed and validated under restrictive assumptions: homogeneous media, low spatial dimensions (often one-dimensional), uniform spatial discretizations, and modest system sizes [29–35]. While these settings are valuable for verifying accuracy and convergence, they often shield algorithms from the practical difficulties encountered in more general problems. In real-world applications, models are frequently applied to heterogeneous media, leading to spatially variable coefficients [36]. Furthermore, inherent model simplifications and unavoidable measurement uncertainties in input data impose a practical limit on the meaningful precision of any numerical solution. Therefore, the primary objective in many applications shifts from pursuing extreme accuracy (e.g., driving errors below 10^{-8}) to developing methods that are computationally efficient in higher dimensions, robust across a wide range of parameters and stiffness regimes, and reliable in producing physically meaningful solutions [37].

Implicit methods, such as the Crank-Nicolson scheme or the shifted airfoil collocation method for nonlinear drift-diffusion-reaction equations, overcome these stability restrictions and are widely used for stiff systems [38, 39]. Their superior stability, however, comes at a high computational price: each time step requires the solution of a large, often nonlinear, system of algebraic equations. This process becomes memory- and time-intensive for problems in two or three spatial dimensions, limiting scalability. This creates a persistent and fundamental trade-off in the field: explicit methods are efficient per step but suffer from crippling stability constraints, while implicit methods are stable but computationally expensive per step.

This dichotomy is evident even in advanced methods designed for qualitative consistency. For instance, several positivity-preserving and dynamically consistent finite-difference schemes for the Burgers-Huxley, and cross-diffusion equations ultimately rely on an explicit treatment of the diffusion term [40, 41]. As a result, their favorable properties are guaranteed only for time steps respecting the explicit CFL condition, negating the potential for unconditional robustness.

1.4. Systems of the diffussion-reaction equations

Systems of diffusion-reaction equations are among the most widespread partial differential equations (PDEs) in science and technology. Therefore, analytical solutions are sought and found not only in the old times [42], but recently as well. For example, Simpson et al. [43] constructed analytical solutions of coupled linear reaction–diffusion equations where the space domain is growing in time. Escorcia and Suazo [44] very recently showed how to construct

solutions in explicit form via similarity transformations and a Ricatti-system for coupled reaction-diffusion equations where the coefficients depends on time. The group of Barna found very nontrivial solutions [45] a few years ago to regular and irregular diffusion equations containing Kummer or Whittaker functions using similarity transformations. Last year they extended their investigations to systems of diffusion-reaction equations [46] as well.

The landscape of numerical techniques developed to approximate solutions where analytics fall short is vast and continually evolving. Beyond the classical explicit-implicit dichotomy, significant research has focused on hybrid and specialized approaches that attempt to balance computational efficiency with stability and qualitative fidelity. A prominent strategy is the use of Implicit-Explicit (IMEX) methods, which treat stiff components (typically diffusion) implicitly while handling non-stiff or nonlinear parts (often reaction or advection) explicitly. Early frameworks for such methods [47] have been developed into high-order, linearly implicit Runge-Kutta schemes tailored for advection-reaction-diffusion equations [48]. These ideas have been applied to solve systems like the Lotka-Volterra reaction-diffusion model in two dimensions, employing implicit diffusion treatment combined with both traditional explicit and alternating direction implicit (ADI) schemes for the reaction terms [49]. More recently, novel IMEX-type Runge-Kutta methods have been designed specifically for stiff convection-diffusion-reaction systems, applying implicit treatment to both diffusion and reaction terms for enhanced stability [50].

Parallel developments include semi-implicit and semi-explicit predictor-corrector methods, often built from combinations of backward differentiation formulas (BDF) and Adams-Bashforth (AB) schemes, which aim to improve efficiency for the ODE systems resulting from method-of-lines discretizations [51]. The performance of such integrators can be further optimized through adaptive, variable step-size controllers [52]. Other advanced time-integration techniques leverage Gauss–Legendre quadrature and Padé approximations to achieve high accuracy and unconditional stability with controllable numerical dissipation [53].

In cases of extreme stiffness, a powerful technique involves treating the linear diffusion operator exactly within the time-stepping scheme. This approach is exemplified by Krylov implicit integration factor methods, which compute the effect of linear terms exactly and have been successfully applied to semilinear and fully nonlinear advection-diffusion-reaction equations [54]. Counterintuitively, some analyses suggest that for very large-scale systems, fully explicit schemes, even with severely restricted time steps, can outperform implicit solvers in terms of total computational time, highlighting a nuanced trade-off dependency on problem scale and architecture [55].

Another class of efficient explicit solvers includes odd-even hopscotch methods and Alternating Direction Explicit (ADE) methods, such as the Saulyev scheme. Comparative studies have shown these explicit ADE methods can achieve solutions faster than major implicit solvers for nonlinear PDEs [56, 57], though often with an accuracy trade-off.

1.5. Long-term transient simulation of insulated walls in cold cities

In recent years, concerns over climate change and rising heating energy demands have driven research toward more energy-efficient buildings, for example by optimizing insulation strategies [61]. Reducing energy consumption in buildings can be achieved through multiple approaches, including improvements in facade materials. For instance, studies suggest that stone cladding is more effective in minimizing cooling loads compared to aluminium composite panels, plaster, or polyurethane board systems [60]. Investments in energy efficiency become more financially viable as energy prices increase, yet studies show that consumer response to rising energy costs remains relatively inelastic in the short term. A 1% increase in household energy prices typically results in only a 0.1% to 0.4% reduction in energy demand [58]. Proper insulation significantly reduces energy consumption, fossil fuel dependency, and associated environmental pollution. While it lowers heating costs, it also increases the initial investment [62]. Several factors influence a building's heating load, including thermal mass, structural materials, and overall architectural design. The use of insulation materials has expanded significantly over the past few decades due to stricter building regulations that mandate greater insulation thickness [63]. Determining the optimal insulation thickness is a key aspect of energy-efficient building design. The insulation layer should strike a balance between installation costs and long-term energy savings. Research indicates that beyond a certain thickness, additional insulation provides diminishing returns in energy savings [64]. The ideal insulation thickness depends on various factors, including material properties, local energy costs, and climatic conditions [65].

Recent studies have applied combined economic and ecological assessments to determine optimal insulation thickness, particularly considering embodied emissions, structural constraints, or multi-criterion analyses. Gaarder et al. evaluated insulation thickness in cold climates by balancing embodied and operational greenhouse gas emissions [66]. Alrasheed optimized both insulation and structural layer thickness using computational life-cycle cost analysis [67]. Additionally, Uçar employed a comprehensive 4E framework energy, exergy, economic, and environmental to identify optimal insulation configurations, concludes that optimal thickness differs significantly depending on whether the criterion is economic or environmental [68].

While industry standards offer general guidelines, they may not always be optimal for every scenario, particularly when energy prices fluctuate [69]. The impact of insulation is not universal, as different heating systems operate through varying heat transfer mechanisms [70]. Studies conducted in various regions, such as Turkey and Tunisia, highlight how insulation performance varies based on local climate conditions and economic factors [71], [72]. The cost of insulation is influenced by production costs and its effectiveness in regulating temperature. As a result, multi-layer insulation systems are often employed to maximize economic and thermal performance [73], [74]. Expanded polystyrene (EPS) concrete panels, for example, are widely used due to their affordability, lightweight nature, and insulating properties, which contribute to faster and more cost-effective construction. Heat transfer analysis is crucial for understanding

energy efficiency in buildings. Most thermal studies focus on parameters such as heat flux, temperature distribution, and heat capacity [75], [76].

1.6. Transient Thermal Simulation of a Multi-Layer PV Panel

The global transition to sustainable energy has established solar photovoltaics (PV) as a cornerstone technology. The importance of renewables like solar power stems from their clean nature, declining costs, and long-term viability, evidenced by a rapid deployment that has led to a global installed capacity exceeding 2 Terawatts (TW) [77]. While the primary focus of PV research has often been on improving the electrical conversion efficiency of semiconductor materials, the operational temperature of the panel is a paramount factor that directly impacts this efficiency.

A well-documented phenomenon is the negative temperature coefficient of silicon solar cells, where efficiency decreases by approximately 0.3% to 0.5% for every degree Celsius increase in temperature above standard test conditions (25°C). Therefore, accurately predicting and managing the operating temperature is not merely an exercise in thermal engineering but a crucial aspect of performance optimization and lifetime forecasting.

A typical PV module is a complex composite structure. The standard lamination process bonds multiple layers with distinct thermophysical properties. From top to bottom, these typically include a tempered glass superstrate for protection and light transmission; a first layer of ethylene-vinyl acetate (EVA) for encapsulation; the silicon PV cells, where photon-to-electron conversion and associated heat generation occur; a second layer of EVA; and finally a back sheet, often made of a polymer or aluminum, for electrical insulation and environmental protection.

This multi-layer architecture necessitates robust thermal modeling. The application of the finite-difference method for this purpose is well-established, as demonstrated by Notton et al. [78] for a double-glass architecture. Furthermore, Shahverdian et al. [79] systematically quantified the trade-off between model complexity and accuracy, concluding that a five-layer representation is essential for capturing the detailed thermal behavior. Notably, Nagy et al. [80] developed a comprehensive 3D computational model in MATLAB to simulate coupled thermal and electrical behavior. Their work calculated the averaged silicon temperature to derive electrical parameters like short-circuit current and open-circuit voltage, validating their model against experimental data for both short-term and two-hour simulations.

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 Fishers, Huxley, Difussion equations, depending on a new way of thinking. Adapt the most successful methods (especially the Leapfrog-hopscotch 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 and Solar panel. The aim of this thesis is twofold: first, to develop and analyze explicit numerical methods for a class of

time-dependent PDEs under a novel theoretical approach; and second, to extend the most robust of these methods especially the Leapfrog-Hopscotch method to handle coupled convective and radiative boundary conditions. This extension enables the high-fidelity simulation of practical engineering problems, specifically the complex heat transfer processes occurring in building envelopes and photovoltaic solar panels.

1.7. Outline and the Aims of the Thesis

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 Fishers, Huxley, Difussion equations, depending on a new way of thinking. Adapt the most successful methods (especially the Leapfrog-hopscotch 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 and Solar panel. The aim of this thesis is twofold: first, to develop and analyze explicit numerical methods for a class of time-dependent PDEs under a novel theoretical approach; and second, to extend the most robust of these methods especially the Leapfrog-Hopscotch method to handle coupled convective and radiative boundary conditions. This extension enables the high-fidelity simulation of practical engineering problems, specifically the complex heat transfer processes occurring in building envelopes and photovoltaic solar panels.

My goal is to adapt and optimize numerical methods, and then, using these methods, to efficiently investigate heat transfer problems in building walls and solar panels. 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, Original Odd-Even Hopscotch (OOEH), Dufort and Frankel (DF) and Leapfrog-Hopscotch (LH), methods, other methods are also discussed in this chapter.

Chapter 3 studies explicit positivity-preserving numerical methods for the heat equation and diffusion equations with time and space-dependent diffusion coefficients. Numerical experiments are conducted to assess accuracy, stability, efficiency, and performance for stiff problems and different parameter values.

Chapter 4 introduces numerical methods for solving Fisher equations using a new way of thinking. I improved the efficiency of some methods by combining the hopscotch and leapfrog techniques, and I conducted numerical experiments to investigate their performance, select the best combinations, and test these algorithms on both small and large systems, stiff and non-stiff.

Chapter 5.1 studies dynamically consistent numerical methods for Huxley's equation. Operator-splitting techniques and different treatments of the nonlinear term are investigated. Numerical experiments in one and two space dimensions assess accuracy, stability, and

computational efficiency, including stiff problems. The performance of several efficient explicit methods is compared using running-time measurements.

Chapter 5.2 presents a range of stable explicit time-integration methods of first to fourth order, originally developed for the Huxley equation. New treatments of the nonlinear term ensure that the numerical solution remains within the physically meaningful unit interval, which implies unconditional stability. The most effective methods are identified from a large set of method combinations.

Chapter 6 combines two equations with time-dependent response terms. I collect eight efficient explicit numerical techniques that are unconditionally stable in the absence of reaction terms and apply them to the system of equations. I demonstrate that they significantly outperform typical explicit approaches when low or medium precision is desired.

Chapter 7, a numerical transient simulation is used to assess heat transmission through the southern and northern walls in five frigid cities over the six-month heating season. Temperature profiles and heat loss are investigated for both uninsulated and insulated walls, employing glass wool of varied thicknesses. The ideal insulation thickness is established by comparing thermal performance under various conditions. The findings will help decide how to improve both energy and cost efficiency in cold climates.

Chapter 8 presents a one-dimensional heat conduction model that incorporates internal heat generation within the silicon layer caused by light absorption as well as convective radiative boundary conditions at the external surfaces. The partial differential equation describing heat transmission was discretised using the finite difference technique. The explicit leapfrog-hopscotch approach is used to numerically solve this equation. It is widely known for its stability and second-order accuracy. After parameter estimation with the LH approach, the revised model was run for two hours. The results, presented as temperature-time graphs, clearly show thermal gradients inside the panel as well as the time lag required for heat to travel from the silicon layer to the outer surfaces.

2. EXPLICIT NUMERICAL DISCRETIZATION OF THE HEAT EQUATION

This section details explicit discretization of the spatial operator in the heat conduction equation. This process converts the governing partial differential equation (PDE) into a system of ordinary differential equations (ODEs). I establish this semi-discrete framework as the essential foundation for all subsequent numerical schemes presented in this dissertation. Following this derivation, I review explicit time-integration methods, both classic and novel, that I apply to the resulting ODE system. Finally, I conclude the chapter with an overview of the computational tools used to simulate and validate the techniques presented.

2.1. The Equations and Its Discretization

For the purposes of establishing the primary discretization framework in this chapter, I will consider the homogeneous case where this source term is absent, i.e., q=0 Eq (1.1).

In the case of Eq. (1.1) in one space dimension, I apply to the $\alpha\nabla^2 u$ term the most common central difference equation

$$\frac{\partial^2}{\partial x^2}u(x_i) \approx \frac{\frac{u(x_{i+1})-u(x_i)}{\Delta x}+\frac{u(x_{i-1})-u(x_i)}{\Delta x}}{\Delta x} = \frac{u_{i-1}-2u_i+u_{i+1}}{\Delta x^2}\ , \qquad (2.1)$$

which is second order in Δx, where $i=1,...,N-1$ and N is the overall number of nodes. Applying this process leads to the following spatially discretized formulation in one dimension:

$$\frac{du_i}{dt} = \alpha\frac{u_{i-1}-2u_i+u_{i+1}}{\Delta x^2}\ . \qquad (2.2)$$

Now let us give the discretisation of the heat transfer equation under the assumption that the parameters α, k, c, and ρ, which describe the properties of materials, are functions of space rather than fixed values Eq. (1.2).

In this example, the heat conduction equation can be discretised as follows:

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

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:

$$\frac{du_i}{dt} = \frac{1}{c_i\rho_i\Delta x}\left(k_{i,i+1}\frac{u_{i+1}-u_i}{\Delta x}+k_{i-1,i}\frac{u_{i-1}-u_i}{\Delta x}\right)\ . \qquad (2.4)$$

A cell's dimensions, measured along its length and across its (typical) cross-section, are denoted by Δx and S. Here u_i is the temperature of the cell i, $C_i = c_i\rho_i V$ is the heat capacity of that cell in $[\mathrm{J/K}]$ units, and $V = S\Delta x$ is the volume of the cell.

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 $R_{i,i+1} \approx d_{\mathrm{ij}} / \left(k_{i,i+1} S_{\mathrm{ij}}\right)$

Using these updated values, the time derivative of each cell variable can be expressed as follows:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i} C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i} C_i} \ . \tag{2.5}$$

As a generalization of Eq. (2.5) one may construct the ODE system for the time derivative of the cell variables for a generic grid by using the above approximations as follows:

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} \ . \tag{2.6}$$

This set of ordinary differential equations (ODEs) can be applied to a variety of grids, including unstructured grids with varying cell sizes and forms. It is vital to notice that unequal discretisation can impact spatial accuracy. For this purpose, I have used just rectangular cells.

Eq. (2.6) for can be written into a matrix form

$$\frac{d\vec{u}}{dt} = M\vec{u} + \vec{Q} \, . \tag{2.7}$$

In the one-dimensional case of Eq. (2.2), the matrix M is tridiagonal with the following elements:

$$m_{\mathrm{i,i}} = -\frac{2\alpha}{\Delta x^2} \ (1 < i < N), \ \ m_{\mathrm{i,i-1}} = \frac{\alpha}{\Delta x^2} \ (1 < i \leq N), \ \ m_{\mathrm{i,i+1}} = \frac{\alpha}{\Delta x^2} \ (1 \leq i < N) \, . \tag{2.8}$$

In the general case of Eq. (2.6), the nonzero elements of the matrix can be given as:

$$m_{\mathrm{i,j}} = \frac{1}{R_{\mathrm{i,j}} C_{\mathrm{i}}} \, , \quad m_{\mathrm{i,i}} = -\sum_{\mathrm{j} \neq \mathrm{i}} m_{\mathrm{i,j}} \, . \tag{2.9}$$

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 15 cells in Figure 0.1 to aid the reader's imagination. I emphasize that the arrangement and shape of the cells are not necessarily regular.

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 [81].

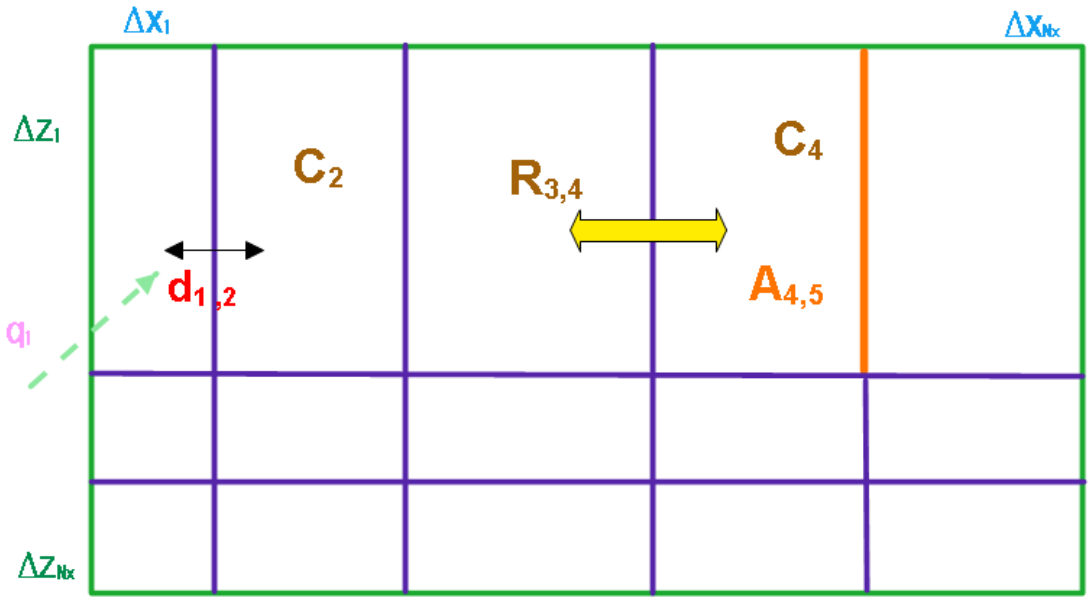


Figure 0.1. The generalized variables visualization for the case when the mesh is not necessarily regular. The orange arrow is for conduction between cells with neighboring through the resistance $R_{3,4}$.

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 for the heat or diffusion equation

In this area, I provide important information on the algorithms, some of which already exist and others that our group created. I begin by presenting their formula for the simplest case (a one-dimensional, equidistant mesh, Eq. (1.1)), followed by a generic, arbitrary mesh. The simplest form is relevant for comparison because numerical schemes are presented in this format in the majority of numerical techniques textbooks. The more general forms are necessary because I use only them in this work.

I introduce the usual mesh-ratio $r = \alpha\Delta t/\Delta x^2$ for the 1D equidistant mesh, e.g. for Eq. (1.1). On the other hand, for the case of the general mesh, I introduce the following notations.

$$r_i = \Delta t \sum_{j \neq i} \frac{1}{C_i R_{ij}} \text{ and } A_i = \Delta t \sum_{j \neq i} \frac{u_j^n}{C_i R_{ij}} \ . \tag{2.10}$$

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.

2.3. Forward-Time Central-Space (FTCS) method

First, the FTCS formula is applied to Equation (2.1), which is based on the explicit Euler time discretization

$$u_i^{n+1} = u_i^n + r\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right) \tag{2.11}$$

2.4. Adams–Bashforth (AB2) method

The Adams-Bashforth families is an explicit multistep numerical technique. I use its two-step second order variant, which predicts u_i^{n+1} using information from the previous two points, making it more accurate than single-step methods like Euler's method. In our case, the first step is performed using Euler's method in the same way as above. Now, I have *u* value for two time levels, and the value in the old time levels will be denoted by $u2_i^n$. Now, the formulas of the Adams-Bashforth method are the following:

$$fu_i^{n+1} = -2ru_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right) \text{ and } fuold_i^{n+1} = -2ru2_i^n + r\left(u2_{i-1}^n + u2_{i+1}^n\right). \tag{2.12}$$

Finally, the updated values are calculated as follows:

$$u_i^{n+1} = u_i^n + \frac{(3fu_i^{n+1} - fuold_i^{n+1})}{2} \ . \tag{2.13}$$

Although the AB2 method is theoretically stable for certain step sizes, in practice inhomogeneous forcing terms can amplify errors when the stability condition is violated.

2.5. The alternating-direction explicit (ADE) method

The alternating-direction explicit (ADE) scheme [82, 83] is a nonconventional approach with a known consistency requirement. I have included it for comparison purposes. To calculate a one-dimensional equidistant mesh, sweep from left to right and vice versa independently. The equations below are solved left to right and right to left, accordingly:

$$uright_i^{n+1} = \frac{(1-r)u_i^n + r\left(u_{i+1}^n + uright_{i-1}^{n+1}\right)}{1+r} \text{ and } uleft_i^{n+1} = \frac{(1-r)u_i^n + r\left(uleft_{i+1}^{n+1} + u_{i-1}^n\right)}{1+r}. \tag{2.14}$$

The final values are the simple averages of the two half-sided terms:

$$u_i^{n+1} = (uright_i^{n+1} + uleft_i^{n+1})/2. \tag{2.15}$$

2.6. The constant neighbor (CNe) method

The constant neighbor (CNe) method [84,85] for is:

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

While for general grids it is:

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

2.7. The Two-Stage Constant neighbor (CpC) method

The CpC algorithm [86] 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:

$$u_i^{\mathrm{pred}} = u_i^n \cdot e^{-r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-r}\right) \text{ and } u_\mathrm{i}^{\mathrm{pred}} = u_i^n e^{-r_i/2} + \frac{A_i}{r_i}\left(1 - e^{-r_i/2}\right). \tag{2.17}$$

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

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^{\mathrm{pred}} + u_{i+1}^{\mathrm{pred}}}{2}\left(1 - e^{-2r}\right) \quad \text{and} \quad u_i^{n+1} = u_i^n \cdot e^{-r_\mathrm{i}} + \frac{A_i^{\mathrm{new}}}{r_\mathrm{i}}\left(1 - e^{-r_\mathrm{i}}\right). \tag{2.18}$$

2.8. The Linear-Neighbour(LNe) method

The next method is the 2-stage linear-neighbor (LNe or LNe2) method [87]. 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:

$$u_i^{n+1} = u_i^n e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1 - e^{-2r}\right) + s_i \frac{\Delta t^2}{2r}\left(1 - \frac{1 - e^{-2r}}{2r}\right). \tag{2.19}$$

For the general case,

$$A_i^{\mathrm{new}} = \Delta t \sum_{\mathrm{j}\neq\mathrm{i}} \frac{u_\mathrm{j}^{\mathrm{pred}}}{C_i R_{\mathrm{ij}}}, \tag{2.20}$$

with which I can make the corrector step as follows:

$$u_{\mathrm{i}}^{\mathrm{n+1}} = u_{\mathrm{i}}^{\mathrm{n}} e^{-r_{\mathrm{i}}} + \left(A_{\mathrm{i}} - \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}} \right) \frac{1 - e^{-r_{\mathrm{i}}}}{r_{\mathrm{i}}} + \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}. \qquad (2.21)$$

The corrector values given by Equation (2.21) are used to calculate A_i^{new} with (2.20) again, and thus one repeats (2.21) to obtain refined u values. This increases the total number of stages in one time step to three, and hence, the seventh algorithm is called LNe3 [87]. This iteration can be continued to obtain the LNe4 (LNe5) method, which has four (five) stages altogether. These are obtained by repeating (2.20) and (2.21). These iterations, although they yield slightly larger accuracy, unfortunately do not improve the order.

2.9. The Constant-Constan-Linear neighbor (CCL) method

The next method is called the CCL algorithm, which is an abbreviation of Constant‑Constant‑Linear neighbor. It is a one-step but three-stage method, recently published by our group [88], with third-order accuracy. The CNe formula is applied in the first and second stages and the so-called linear-neighbor (LNe) formula in the last stage. The first stage is a predictor with a Δt/3-sized time step

$$au0_i = \frac{u_{i-1}^n + u_{i+1}^n}{2} \text{ and } u_i^C = u_i^n \cdot e^{-2r/3} + au0_i \left(1 - e^{-2r/3}\right). \qquad (2.22)$$

Then, the first corrector stage is performed:

$$au1_i = \frac{u_{i-1}^C + u_{i+1}^C}{2} \text{ and } u_i^{CC} = u_i^n \cdot e^{-4r/3} + au1_i \left(1 - e^{-4r/3}\right). \qquad (2.23)$$

The last stage is as follows:

$$au2_i = \frac{u_{i-1}^{CC} + u_{i+1}^{CC}}{2} \text{ and } u_i^{n+1} = u_i^n \cdot e^{-2r} + au0_i \left(1 - e^{-2r}\right) + \frac{3\left(au2_i - au0_i\right)}{2} \left(1 - \frac{e^{-2r}}{2r}\right). \qquad (2.24)$$

2.10. The Constant–Linear–Quadratic (CLQ) neighbor method

The first two stages of the three-stage Constant-Linear-Quadratic-neighbor (CLQ) method [89] are the same as those of the LNe method, with the exception that the second LNe stage has to be made with not only a full but also a half time step size using

$$au_i^C = \frac{u_{i+1}^n + u_{i-1}^n}{2} \text{ and } u_i^C = u_i^n \cdot e^{-2r} + au_i^C \left(1 - e^{-2r}\right), \qquad (2.25)$$

Then, the first corrector stage is performed:

$$au_i^{CL}=\frac{u_{i+1}^C+u_{i-1}^C}{2}\ \text{and}\ u_i^{CL}=u_i^n\cdot e^{-2r}+au_i^{CL}\left(1-e^{-2r}\right)+\left(au_i^{CL}-au_i^C\right)\left(1-\frac{e^{-2r}}{2r}\right),$$
$$u_i^{CLmid}=u_i^n\cdot e^{-r}+au_i^{CL}\left(1-e^{-r}\right)+\frac{au_i^{CL}-au_i^C}{2}\left(1-\frac{e^{-r}}{r}\right) \quad (2.26)$$

The third stage contains the calculations below:

$$au_i^{CLQ}=\frac{u_{i+1}^{CL}+u_{i-1}^{CL}}{2}\ \text{and}\ au_{i,mid}^{CLQ}=\frac{u_{i+1}^{CLmid}+u_{i-1}^{CLmid}}{2}$$
$$su=4au_i^{CLQ}-au_i^{CLQ}-3\cdot au_i^C\ ,\ gu=2\cdot(au_i^{CLQ}-2au_{i,mid}^{CLQ}+au_i^C) \quad (2.27)$$
$$u_i^{CLQ}=u_i^n\cdot e^{-2\cdot r}+(1-e^{-2\cdot r})\cdot(gu/2\cdot r^2-su/2\cdot r+au_i^C)+gu-gu/r+su.$$

I can add one more stage by using the values in (2.25). In this case, besides (2.25) at the third stage, the midpoint values must also be calculated by a similar formula. Using these, one can calculate au_i^{CLQ} and $au_{i,mid}^{CLQ}$ values by (2.27) at the fourth stage, with which one repeats the calculations of the third stage, including those of *su* and *gu*. With these, one has four stages altogether, which constitute the CLQ2 algorithm. One can further repeat this iteration by re-calculating au_i^{CLQ} and $au_{i,mid}^{CLQ}$, then applying (2.27) again, etc. If one has three or four quadratic stages we call the obtained methods CLQ3 and CLQ4, respectively [89].

To proceed to the next group of methods, let us recall that the following general time discretization

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

which implies the so-called theta method:

$$u_i^{n+1}=u_i^n+r\left[\theta\left(u_{i-1}^n-2u_i^n+u_{i+1}^n\right)+(1-\theta)\left(u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}\right)\right], \quad (2.28)$$

where $\theta\in[0,1]$. For $\theta=0,\ ½,$ and 1 one have the (standard) implicit Euler, the Crank–Nicolson and the explicit Euler scheme, respectively [17]. If $\theta<1$, the theta method is clearly 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 (2.17) instead of $u_{i\pm1}^{n+1}$ 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 update, the final formula becomes totally explicit:

$$u_i^{n+1}=\frac{(1-2r\theta)u_i^n+r\left(u_{i-1}^n+u_{i+1}^n\right)}{1+2r(1-\theta)}\ . \quad (2.29)$$

2.11. The UPFD method

The UPFD (unconditionally positive finite difference) The UPFD method [19] is a simple one-stage algorithm with the formula obtained via the substitution of $\theta = 0$ to Eq. (2.29)

$$u_i^{\mathrm{n}+1} = \frac{u_i^{\mathrm{n}} + r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+2r} \ . \tag{2.30}$$

2.12. The original odd-even hopscotch (OOEH) method

The original odd-even hopscotch (OOEH) method [90] a classic example of the explicit methods which is stable for the linear diffusion equation. It requires a special spatial structure. In essence, the mesh must be divided into two parts, the so called odd and even nodes (or cells), where the closest neighbour of the even cells are odd and vice versa. First, the FTCS formula is applied to the odd cells, followed by the BTCS (Backward-time Central-space) formula, which is based on implicit Euler time discretisation. After every time step, the odd and even labels are switched, as it is illustrated in Figure 2.1. The equations being used are as follows:

$$\text{First stage: } u_i^{n+1} = u_i^n(1-r) + r\left(u_{i-1}^n + u_{i+1}^n\right) \ . \tag{2.31}$$

$$\text{Second stage: } u_i^{n+1} = \frac{u_i^n + r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+r} . \tag{2.32}$$

OOEH-CNe method:. It means that the odd values are refreshed first, and then the even values. However, after each time step, the odd–even labels are interchanged. The length of the time steps is always the full-, and the CNe formula (2.16) is employed in each stage.

2.13. The Leapfrog–Hopscotch (LH) method

The recently invented leapfrog hopscotch (LH) approach [91] also requires the odd-even space structure. Moreover, it has a structure made up of many full time steps and two half time steps. Using the initial values, the calculation begins by taking a half-sized time step for the odd nodes. Full-time steps are then taken strictly alternately for the even and odd nodes until the last timestep is reached, which should be halved for odd nodes to reach the same final time point as the even nodes, as shown in Figure 2.1. Since the first stage's time step is halved, the following general formula is applied:

$$u_i^{1/2} = \frac{u_i^0 + r\left(u_{i-1}^0 + u_{i+1}^0\right)/2}{1+r/2} \ . \tag{2.33}$$

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

$$u_i^1 = \frac{u_i^0(1-r/2) + r\left(u_{i-1}^{1/2} + u_{i+1}^{1/2}\right)}{1+r/2} \ . \tag{2.34}$$

Full time steps in the same manner are then taken for the odd and even nodes in turn. Lastly, the computations for the odd nodes must be closed using a half-length time step.

$$u_i^1 = \frac{u_i^0(1-r/2)+r\left(u_{i-1}^{1/2}+u_{i+1}^{1/2}\right)}{1+r/2} \quad . \tag{2.35}$$

The LH-CNe method has the same time-space structure as the LH method, but the CNe formula is used in each stage with the appropriate time step size instead of formulas (2.33)…(2.35).

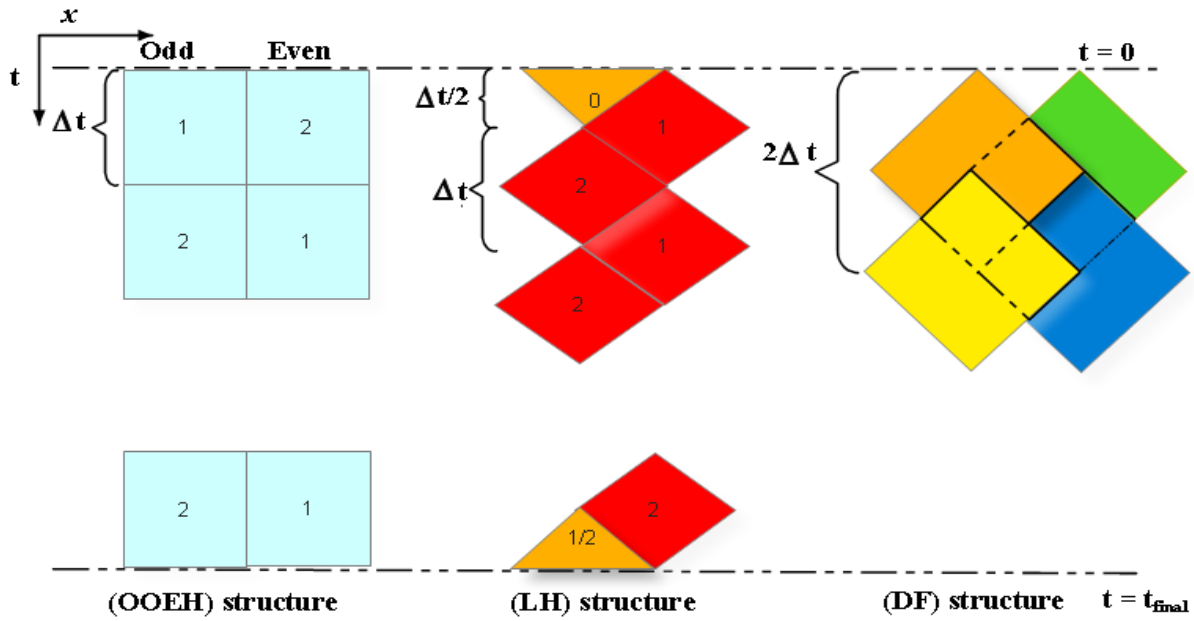


Figure 2.1. The structures of Dufort–Frankel (DF), leapfrog-hopscotch (LH) and original odd-even hopscotch (OOEH) methods.

2.14. The Dufort–Frankel (DF) algorithm

This method can be obtained from the so called leapfrog explicit scheme by a modification [92] (p. 313). It is a known explicit unconditionally stable scheme that has the formula in the special and general case:

$$u_i^{n+1} = \frac{(1-2r)u_i^{n-1}+2r\left(u_{i-1}^n+u_{i+1}^n\right)}{1+2r} \quad \text{and} \quad u_i^{n+1} = \frac{(1-r_i)u_i^{n-1}+2A_i}{1+r_i} \quad . \tag{2.36}$$

As one can see, it is a one-stage but two-step method (the formula contains u_i^{n-1}), which is not 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).

2.15. The pseudo-implicit (PI) method

The pseudo-implicit (PI) two-stage method was developed in [93] (Algorithm 5 there, applied to the pure diffusion Equation (1)). A half time step is taken to obtain the predictor values and then a full time step for the corrector values. The first-stage formulas are the following:

Stage 1: $u_i^{\text{pred}} = \dfrac{u_i^n + r/2\left(u_{i-1}^n + u_{i+1}^n\right)}{1+r}$ and Stage 2: $u_i^{n+1} = \dfrac{(1-r)u_i^n + r\left(u_{i-1}^{\text{pred}} + u_{i+1}^{\text{pred}}\right)}{1+r}$. (2.37)

2.16. Crank–Nicolson (CrN) method

I will also extensively test an implicit method, which is the widely used Crank–Nicolson (CrN) scheme. It is well known that, applying to Eq. (2.6), it leads to the matrix equation

$$\underbrace{\left(I - \frac{\Delta t}{2}M\right)}_{A} U^{n+1} = \underbrace{\left(I + \frac{\Delta t}{2}M\right)}_{B} U^n, \tag{2.39}$$

where I is the unit matrix of size $N \times N$. Since A and B are time-independent in the current work, one can spare running time by calculating $Y = A^{-1}B$ prior to the first time step. In this way, only a matrix-multiplication must be performed in each time step: $U^{n+1} = Y U^n$.

3. A THOROUGH INVESTIGATION OF EXPLICIT, POSITIVITY-PRESERVING METHODS FOR THE HEAT EQUATION.

In this section of the series, I investigate the performance of 12 explicit non-conventional algorithms. All of them have the convex combination property; therefore, they are unconditionally stable and preserve the positivity of the solution when applied to the heat equation. In this part of the series, I construct several two-dimensional systems to examine how the errors depend on the time-step size.

3.1. Case study 1: long-wave initial function

In recent years, our research group developed new numerical algorithms for the heat conduction or diffusion equation and similar diffusion-reaction equations (1.1). The maximum numerical errors are calculated by comparing the numerical solutions u_j^{num} produced using the examined method with the analytical reference solution u_j^{ref} at the final time t^{fin} using the formula

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

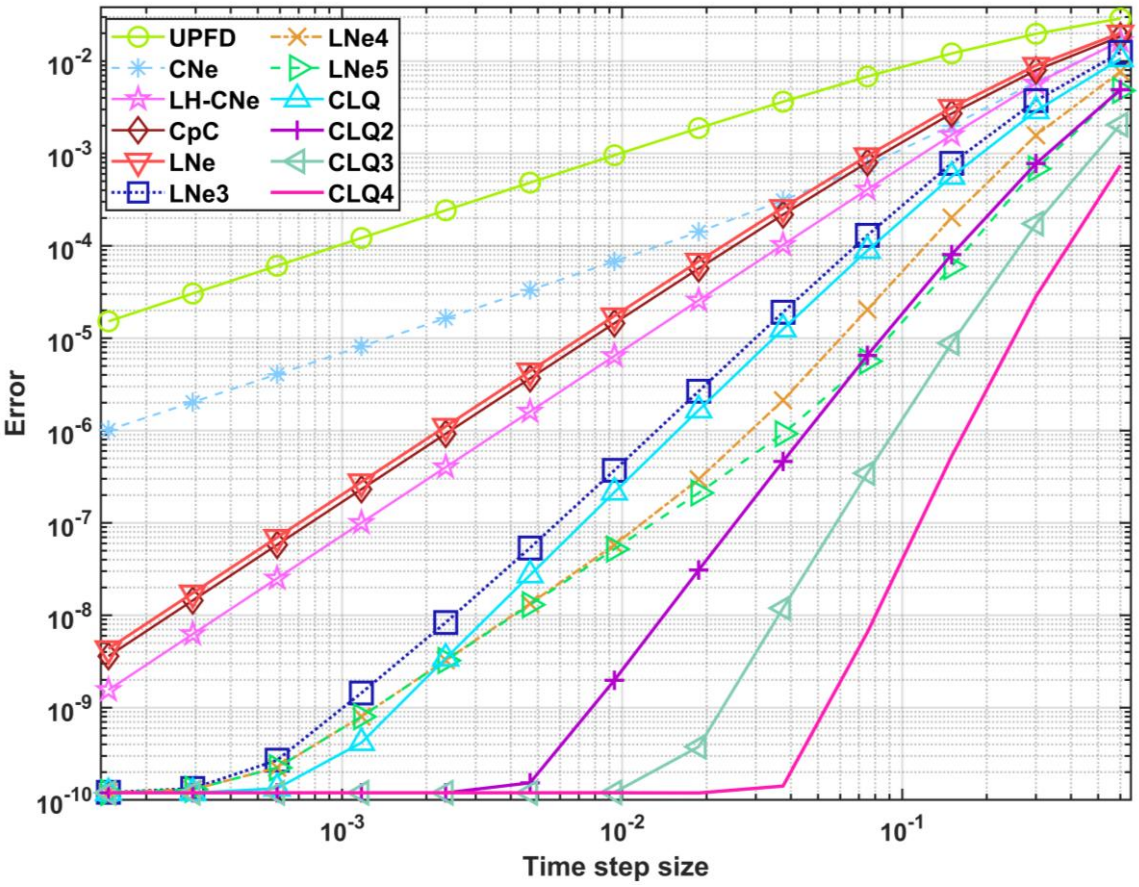


Figure 3.1. The error as a function of time step size Δt for Case study 1.

Random values will be generated for the heat capacities and the resistances with a log-uniform distribution using the formulas:

$$C_i = 10^{(a_C - b_C \times rand)},\ R_{x,i} = 10^{(a_{Rx} - b_{Rx} \times rand)},\ R_{y,i} = 10^{(a_{Ry} - b_{Ry} \times rand)}. \tag{3.2}$$

If one varies the a and b parameters, one can produce highly different systems. I solve the spatially discretized PDE (1.1) on a square-shaped system with $N_x = 41$, $N_y = 41$, and $t_{\text{fin}} = 2.4$. All a and b exponents in Eq. (3.2) are zero, which means $C = 1$, $R_x = R_z = 1$ for all cells, which yields $SR = 1.36 \cdot 10^3$ and $h_{MAX}^{FTCS} = 0.25$. Figure 3.1 shows the numerical results. The initial condition is the product of two cosine functions with wavelengths equivalent to the system size:

$$u(x, y, t = 0) = \big(\cos(2\pi x) - 1\big)\big(\cos(2\pi y) - 1\big) / 4. \tag{3.3}$$

3.2. Case study 2: a very stifff system

Now $N_x = 91$, $N_y = 100$, and $t_{\text{fin}} = 0.2$. The initial condition is $u_i^0 = rand$ to ensure that short and long-wavelengths components are present. The distribution of the capacities and the resistances has a width of six order of magnitude: $a_C = a_{Rx} = a_{Ry} = 3$, $b_C = b_{Rx} = b_{Ry} = 6$. This yields a very large stiffness and low CFL limit: thus $SR = 1.17 \cdot 10^{11}$ and $h_{MAX}^{FTCS} = 1.4 \cdot 10^{-6}$.

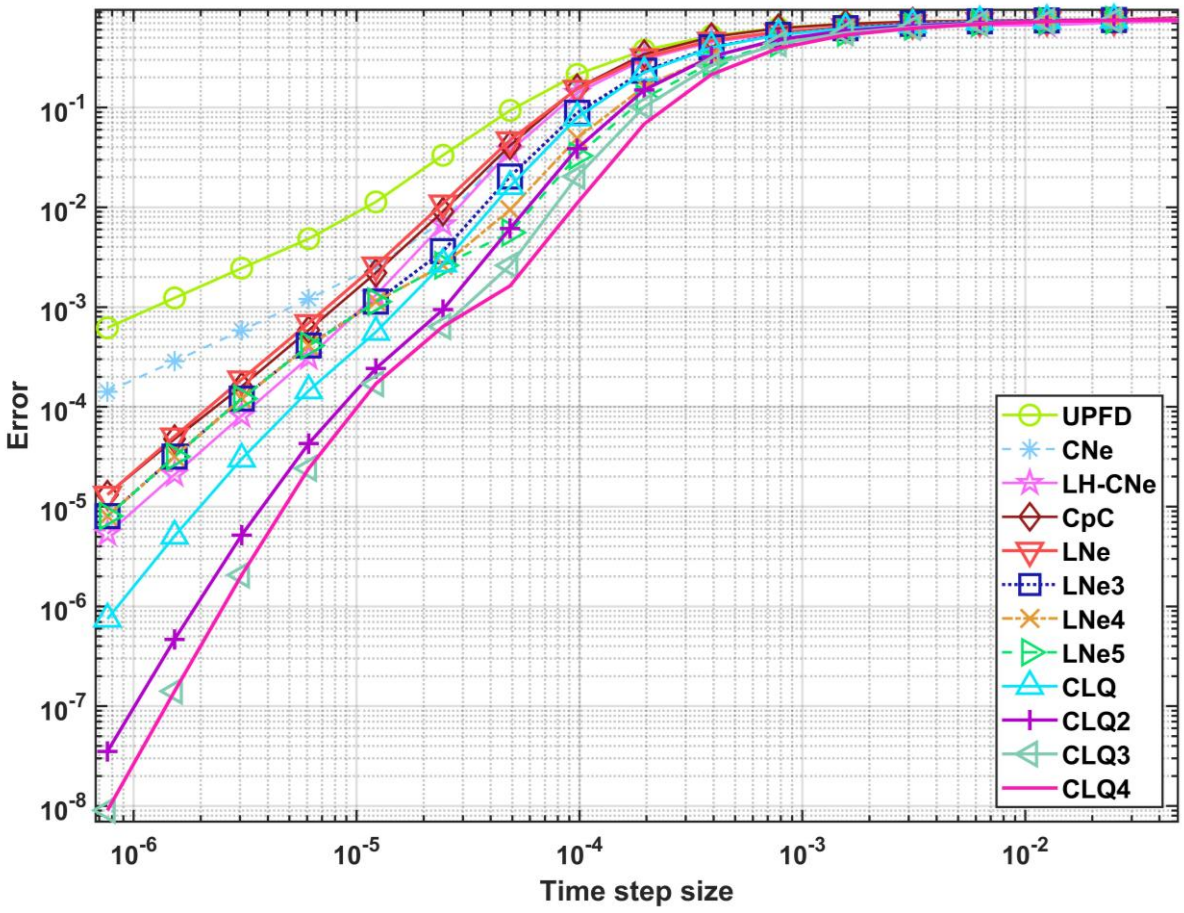


Figure 3.2. The error as a function of time step size Δt for Case study 2.

Since the system is quite large, the fluctuations in the running times are relatively low. The result of this numerical experiment are presented in Fig 3.2.

3.3. Case study 3 Random anisotropy with smooth initial function

Now $N_x = N_y = 51,$, and $t_{\text{fin}} = 0.2$. The parameters of the system have a quite wide range again: $a_C = 3,\ b_C = 6,\ a_{Rx} = 2,\ b_{Rx} = 2,\ a_{Ry} = 0,\ b_{Ry} = 2$. One can see that the resistances in the x direction are two orders of magnitude larger than in the y direction, but now this is a random variable. I obtained that $SR = 1.11 \cdot 10^{10}$ and $h_{MAX}^{FTCS} = 1.84 \cdot 10^{-5}$. The initial condition is the same product of cosine functions as in (3.3). The results of this numerical experiment are presented in Fig 3.3.

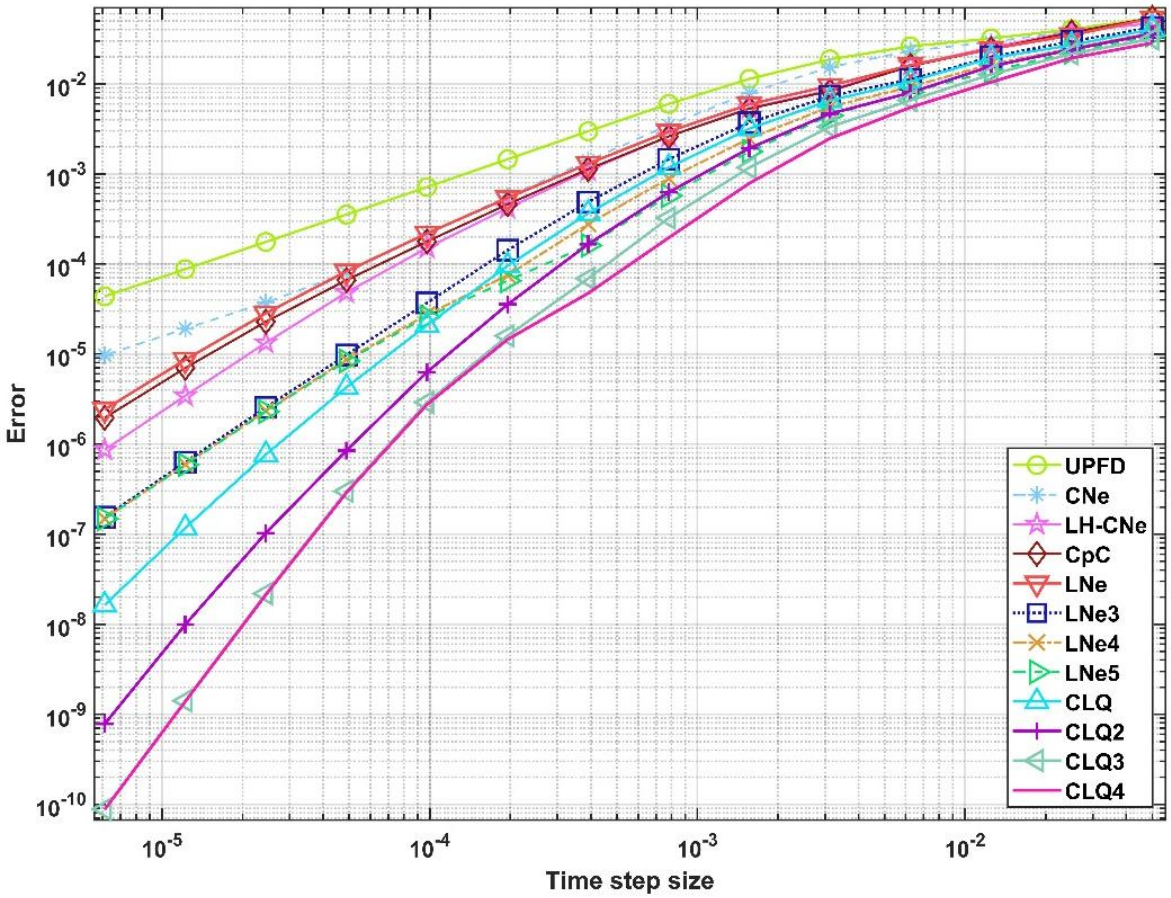


Figure 3.3. The error as a function of time step size Δt for Case study 3.

The numerical case studies presented here confirmed this statement, since the methods behaved well without the slightest sign of instability even for the stiff cases.

If a low or medium accuracy is required, generally the LH-CNe is the most efficient among the methods since it serves quite accurate results in very short time according to the running time measurements. If higher accuracy is required, the higher order CLQ family is the best choice, even if one time step is much slower than in the case of the low order algorithms. If the stiffness is very large, the differences between the methods in accuracy and efficiency is low due to order-reduction. If the initial temperature function is smooth, the multi-stage LNe methods can be as

accurate as the higher order CLQ and CLQ2 methods for large time step sizes. On the other hand, when the very short wavelength initial function was used, there was a large gap between the accuracy of the first, the second and the higher order algorithms. In these cases, the LH-CNe scheme can lose its advantage in the low and medium accuracy region.

3.4. Positivity-preserving numerical methods for diffusion equations with time-space dependent diffusion

The diffusivity has spatial and temporal dependence, which is justified by the fact that in several engineering problems, the properties of the materials widely change [4] because of natural or artificial inhomogeneities. Some new types of analytic solutions were found [45],[94] and plenty of numerical algorithms [4] were applied for different types of diffusion equations. In this work, I assume a specific type of diffusivity function, namely

$$\alpha(x,t) = D\left(\frac{x}{\sqrt{t}}\right)^m,$$

where D is a positive constant. I take analytical solutions for this case in our group's previous paper [95], which contain the Whittaker functions, and therefore highly nontrivial. I set $D=1$ and with this, one has the following form of the solution:

$$u(x,t) = c\sqrt{\frac{t^{1/2-2\alpha}}{x}}\cdot \mathrm{e}^{\frac{\left(x/\sqrt{t}\right)^{2-m}}{4(m-2)}}\cdot W_{\frac{4\alpha-1}{2m-4},\frac{m-1}{2m-4}}\left(\frac{\left(x/\sqrt{t}\right)^{2-m}}{2(m-2)}\right). \tag{3.4}$$

My current work can be considered as the continuation of that our group previous work [95], where we applied several numerical schemes to solve Eq. (1.1) and the function (3.3) was used as the reference solution. Now I focus on only the positivity preserving methods and include the very recently published CLQ type family [89] which will be described later.

3.5. Numerical results for different parameter values

We introduce the so-called aggregated errors (AgE). In the case of the maximum errors, it is given by the following formula

$$\mathrm{AgE}(L_\infty) = \frac{1}{S}\sum_{s=1}^{S}\log\left(\mathrm{Error}(L_\infty)\right). \tag{3.5}$$

A similar aggregated error can be calculated using the average absolute error:

$$\mathrm{Error}(L_1) = \frac{1}{N}\sum_{0\le \mathrm{j}\le N}\left|u_{\mathrm{j}}^{\mathrm{ref}}\left(t^{\mathrm{fin}}\right) - u_{\mathrm{j}}^{\mathrm{num}}\left(t^{\mathrm{fin}}\right)\right|. \tag{3.6}$$

The third possibility provides the energy error in the case of the heat equation, thus I traditionally refer to it as the energy error:

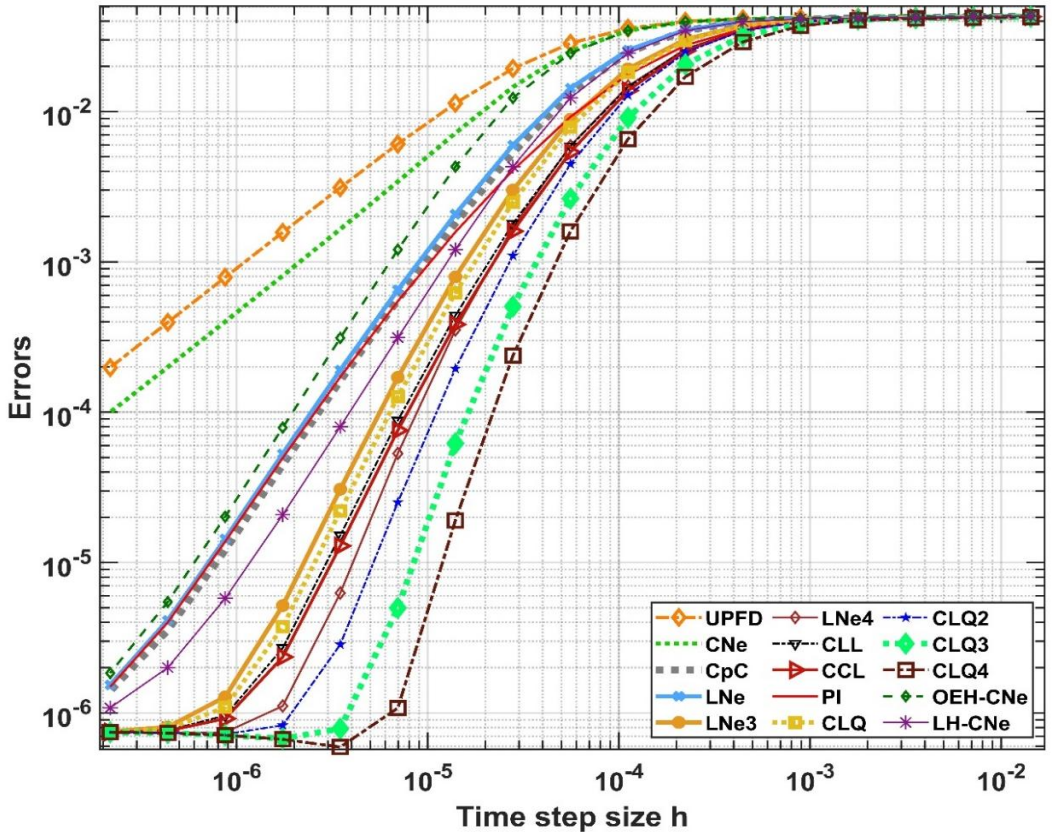


Figure 3.4 The maximum error as a function of the time step size for m=2.1.

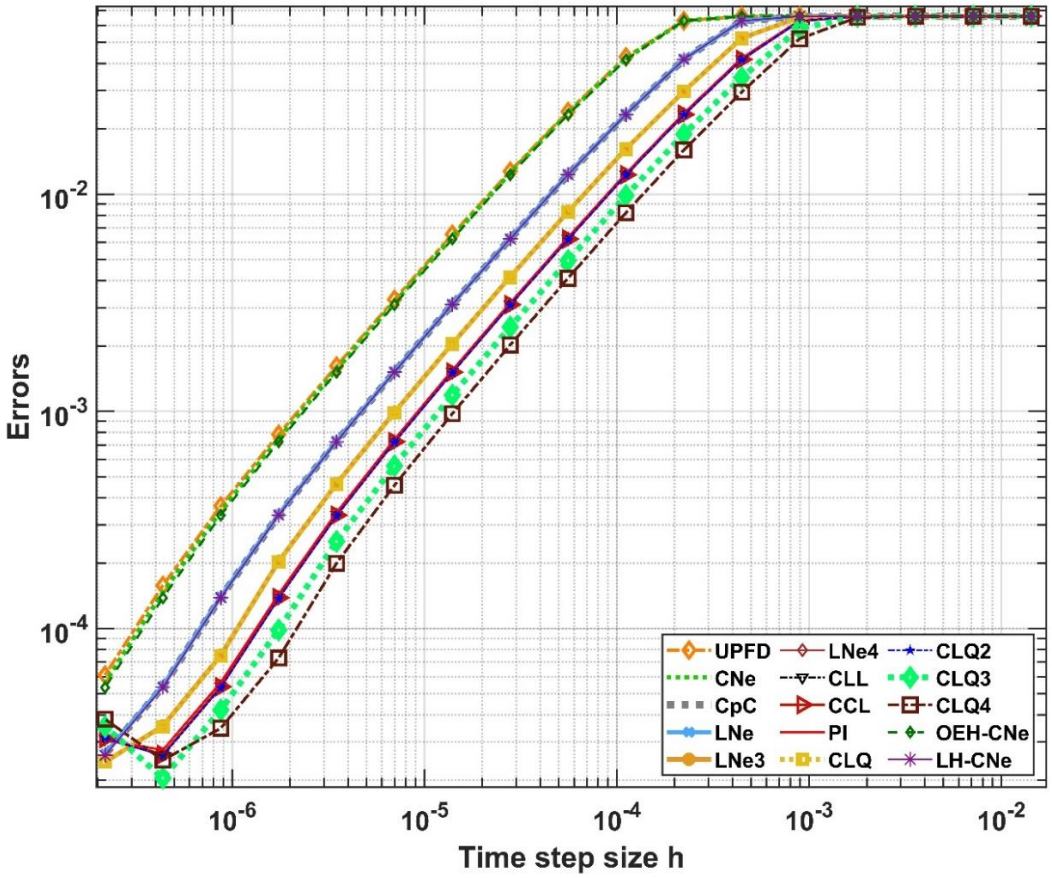


Figure 3.5 The maximum error as a function of the time step size for m=20.

$$\mathrm{Error}(Energy) = \sum_{1\le j\le N} C_j \left| u_j^{\mathrm{ref}}\left(t^{\mathrm{fin}}\right) - u_j^{\mathrm{num}}\left(t^{\mathrm{fin}}\right)\right|. \tag{3.7}$$

Lastly, it is also possible to compute the three types of errors' simple average:

$$\mathrm{AgE} = \frac{1}{3}\left(\mathrm{AgE}(L_\infty) + \mathrm{AgE}(L_1) + \mathrm{AgE}(Energy)\right) \tag{3.8}$$

I used S =17 different time step sizes for all the examined methods. Note that some data about the running times can be found in our previous papers. In the first experiment, the used parameter values are the following: $\alpha = 0.1,\ N = 100,\ x_0 = 0.25,\ \Delta x = 0.01,\ t^0 = 0.1,\ t^{\mathrm{fin}} = 0.2$ and $m = 2.1,$ I obtained that $SR = 1.5\cdot 10^5$ and $h_{MAX}^{FTCS} = 6.4\cdot 10^{-6}$**.**The value of c is always set to be a normalization constant, i.e. the largest absolute values of the functions are always unity. After this first experiment, I changed m to m=20, CFL limit: thus $SR = 2.5\cdot 10^{14}$ and $h_{MAX}^{FTCS} = 1.1\cdot 10^{-13}$. The errors as a function of the time step size are displayed in Fig. 3.4 and 3.5 for the first and second experiment, respectively. One sees that the accuracy difference between the algorithms became much smaller for this large value of m.

3.6. Summary of this chapter

The transient diffusion equation with space- and time-dependent diffusivity was investigated, where increasing the parameter m leads to higher stiffness ratios and a decreasing CFL stability limit. For large m, the CFL limit becomes extremely small and varies in time due to the temporal dependence of the diffusivity, making the use of traditional explicit methods highly risky. All fifteen examined explicit algorithms reproduced the reference solution accurately; however, differences in accuracy diminished as m increased, indicating order reduction. This confirms that higher-order methods provide clear benefits mainly for moderately stiff problems, while lower-order schemes are sufficient for highly stiff cases. Among the methods, the PI scheme showed a notable performance improvement with increasing m, surpassing LH-CNe, LNe3, and even the third-order CLQ method, whereas the second-order OEH-CNe method deteriorated significantly and converged to first-order performance.

4. EXPLICIT, DYNAMICAL CONSISTENT METHODS FOR FISHER'S EQUATION

In this chapter, I present new numerical approaches for solving Fisher's equation, which comprises both a linear diffusion term and a nonlinear logistic term. The conventional explicit finite difference techniques are only conditionally stable for this equation, and even when stable, they can provide concentrations that are less than zero or more than one. I compiled a list of algorithms here which are not only stable and explicit, but possess the convex combination property for the diffusion equation. Based on these, I construct All of them are unconditionally dynamically consistent with Fisher's equation, so the concentration remains in the unit interval for any parameter. I thoroughly examine the performance of numerical schemes by performing tests in the case of 1D and 2D systems to explore how the errors depend on the coefficient of the nonlinear term, the stiffness ratio, and the anisotropy of the system. I also measure running times and recommend which algorithms should be used in specific circumstances.

4.1. The numerical treatments of the nonlinear term

The nonlinear logistic term is implemented via operator splitting, whereby I address the effect of the diffusion term and the nonlinear term independently. The outcome of those algorithms can be denoted by u_i^{diff}, which represents the concentration-value when the diffusion term is fully considered. The effect of the reaction term, which has the form

$$\beta u\left(1-u\right)=\beta u-\beta uu\,, \tag{4.1}$$

is calculated in two ways. The first way is called "pseudo-implicit" treatment [89] and involves a specialapproxiamtion of increment by the selective replacement of u in the right hand side of (4.1) by the u_i^{diff} values and the new, unknown value u_i^{n+1}. After this we obtain

$$\frac{u_i^{n+1}-u_i^{\text{diff}}}{\Delta t}=\beta u_i^{\text{diff}}-\beta u_i^{\text{diff}}u_i^{n+1}\,. \tag{4.2}$$

This can be easily rearranged to become fully explicit:

$$u_i^{n+1}=\frac{1+\beta\Delta t}{1+\beta\Delta t u_i^{\text{diff}}}u_i^{\text{diff}}\,. \tag{4.3}$$

Observe that this formula's right-hand side has no negative terms. I can obtain the other method by solving the ODE

$$\frac{du}{dt}=\beta u\left(1-u\right), \tag{4.4}$$

analytically. To be more exact, the appropriate initial value problem is solved, in which the initial time is the first point of the actual time step, and the u_i^{diff} values are the initial values of u for each cell. This yields the formula:

$$u_i^{n+1} = \frac{1}{u_i^{\text{diff}} + \left(1 - u_i^{\text{diff}}\right) e^{-\beta \Delta t}} u_i^{\text{diff}} . \tag{4.5}$$

A part from the operator splitting itself, this so-called quasi exact method has the advantage of avoiding approximation and linearisation. However, there is no analytical solution for some more complex nonlinear elements, such as the Nagumo equation, but the pseudo-implicit treatment can be applied without any problems [89].

4.2. One space dimension using an exact solution

The analytical solution [23], [96], valid for $\alpha = 1$:

$$u^{\text{exact}}(x,t) = \left(1 + e^{\sqrt{\frac{\beta}{6}} x - \frac{5}{6}\beta t} \right)^{-2} , \tag{4.6}$$

is used as reference solution of PDE (1.3). The initial condition is obtained simply by substituting the initial time into Eq. (4.6).

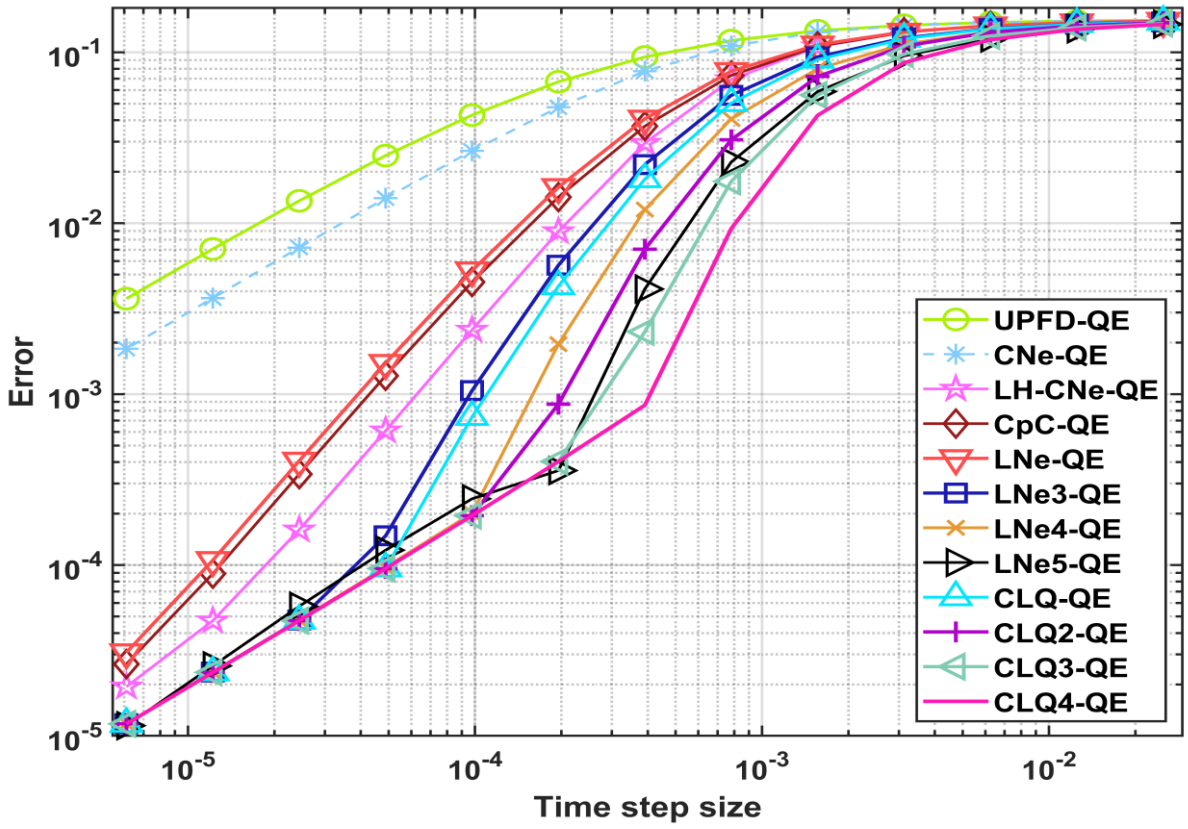


Figure 4.1. Maximum errors as a function of the time step size for QE treatment.

The appropriate Dirichlet boundary conditions are prescribed by substitution of the left and right-side coordinate of the examined interval, which is $x \in [0, 2]$. The initial and the final times are

$t^0 = 0$ and $t^{\text{fin}} = 0.1$. This is discretized $x_j = x_0 + j\Delta x$, $j = 1,...,100$, $\Delta x = 0.02$ by dividing it into 100 equal parts . The nonlinear coefficient is quite large, $\beta = 29$.

In my experiments, I calculated the error for *S*=13 different time step sizes for all the examined methods. The results for three different kinds of treatment are presented in Fig 4.1. Very similar plots have been obtained for the three other treatments and for other values of parameters β, t^{fin}, etc. The algorithms exhibit smooth convergence and show no signs of instability. However, as noted above, the inconsistent terms in the truncation error cause this convergence to be slow for larger time step sizes. The algorithms exhibit smooth convergence and show no signs of instability. In order to compare the overall accuracy of the 72 different combinations.

It is obvious that algorithm-combinations with larger absolute value AgE are more accurate. The AgE values are tabulated in Table 4.1 and visualized in Figure 4.2. One can observe that the quasi-exact treatment is almost always more accurate than the pseudo-implicit one. Strang-splitting yields significantly larger accuracy usually for the higher-order, more accurate methods, e.g., for the CLQ3 and CLQ4 schemes. The mixed treatments are rarely competitive, and since they are more complicated to code, I omit them from future investigations. Only the QE and QE-Strang treatments for all the 12 methods, 24 combinations altogether, will be carried forward to further examinations.

Table 4.1. Aggregated error (AgE) values for each methods combined with each different treatment.

Numerical method	Treatment of the nonlinear term					
	PI	QE	PI-St	QE-St	PI- QE-St	QE- PI-St
UPFD	-9.93	-10.74	-10.25	-10.74	-10.48	-10.48
CNe	-11.46	-12.32	-11.80	-12.31	-12.043	-12.04
LH-CNe	-18.92	-21.685	-21.27	-20.11	-22.34	-22.41
CpC	-18.03	-20.15	-18.72	-19.96	-19.29	-19.32
LNe	-17.67	-19.64	-18.37	-19.50	-18.90	-18.89
LNe3	-20.57	-24.47	-21.84	-24.74	-22.99	-23.03
LNe4	-21.82	-26.33	-23.38	-27.30	-24.90	-24.98
LNe5	-23.28	-27.56	-25.28	-28.99	-27.71	-27.81
CLQ	-20.90	-25.07	-22.18	-25.47	-23.42	-23.42
CLQ2	-22.33	-27.19	-23.92	-29.07	-25.61	-25.61
CLQ3	-23.16	-28.47	-24.94	-31.34	-26.91	-26.91
CLQ4	-23.71	-29.34	-25.63	-32.94	-27.81	-27.79

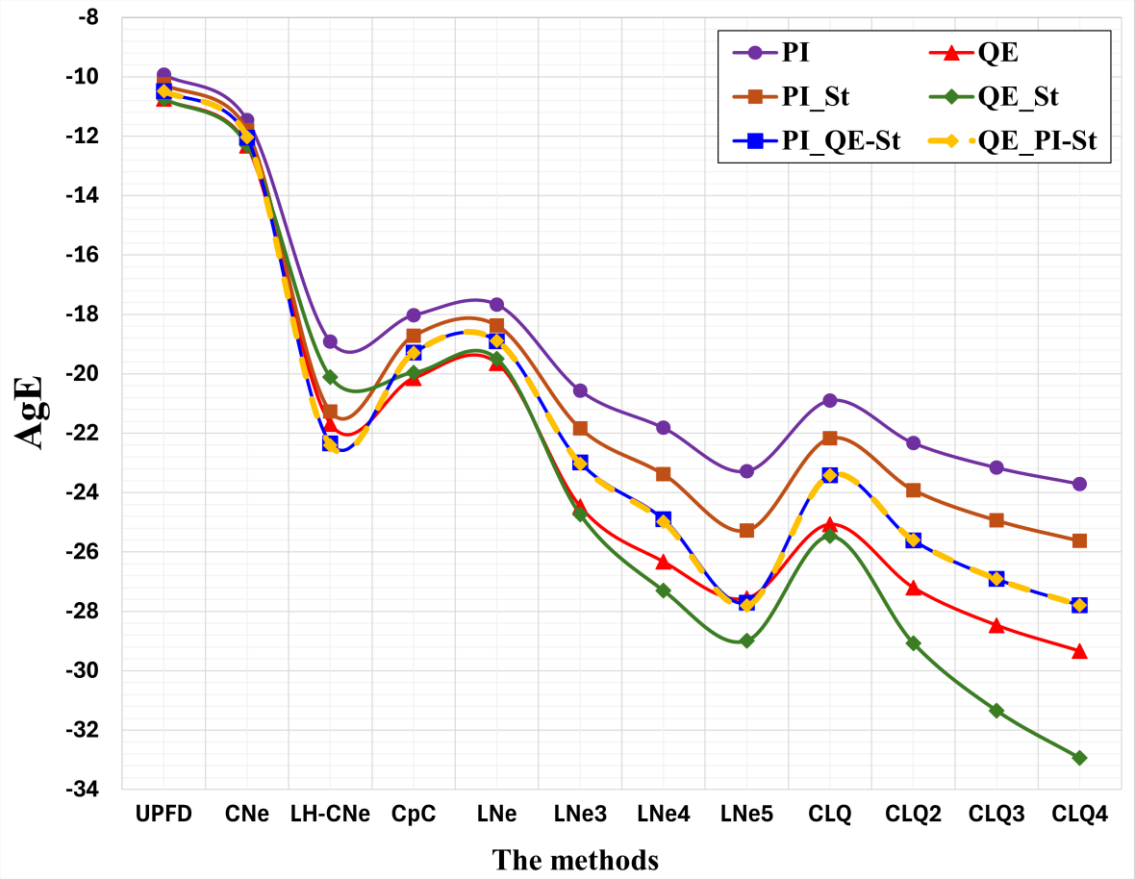


Figure 4.2. AgE errors for the methods and for all treatments of the nonlinear term.

4.3. Testing of performance with running time measurements

From this point I will have two space dimensions and I will use the capacity-resistivity model. I generate random values for the capacitance and resistance using a log-uniform distribution, as defined in Eq (4.2). I able to produce highly different test problems. Here rand are random numbers generated by MATLAB in the unit interval. In order to minimize the effect of the random fluctuations, the calculations are performed 100 times subsequently, and then the average running times are calculated. From this point, the reference solution was provided by the MATLAB ode15 solver with a very stringent tolerance. Now PDE (1.3) with $\alpha = 1$ is going to be solved using 24 algorithm-combinations. These parameters yield $SR = 1.92 \cdot 10^{9}$ and $h_{MAX}^{FTCS} = 1.83 \cdot 10^{-5}$, which means that the system is rather stiff. The used system size and final time are $N_x = 25, N_y = 50, t^{\text{fin}} = 0.4$, while $a_C = 3, b_C = 6, a_{Rx} = 1, b_{Rx} = 2, a_{Ry} = 2, b_{Ry} = 4$. and $\beta = 6$, The initial concentrations are random numbers: $u_i^0 = rand$. I present the obtained errors as a function of the running times. Since the curves are very close to one another, I separately display the low accuracy and the high accuracy part in Figs. 4.3 and 4.4, respectively. One can see that the methods behave well for this case as well.

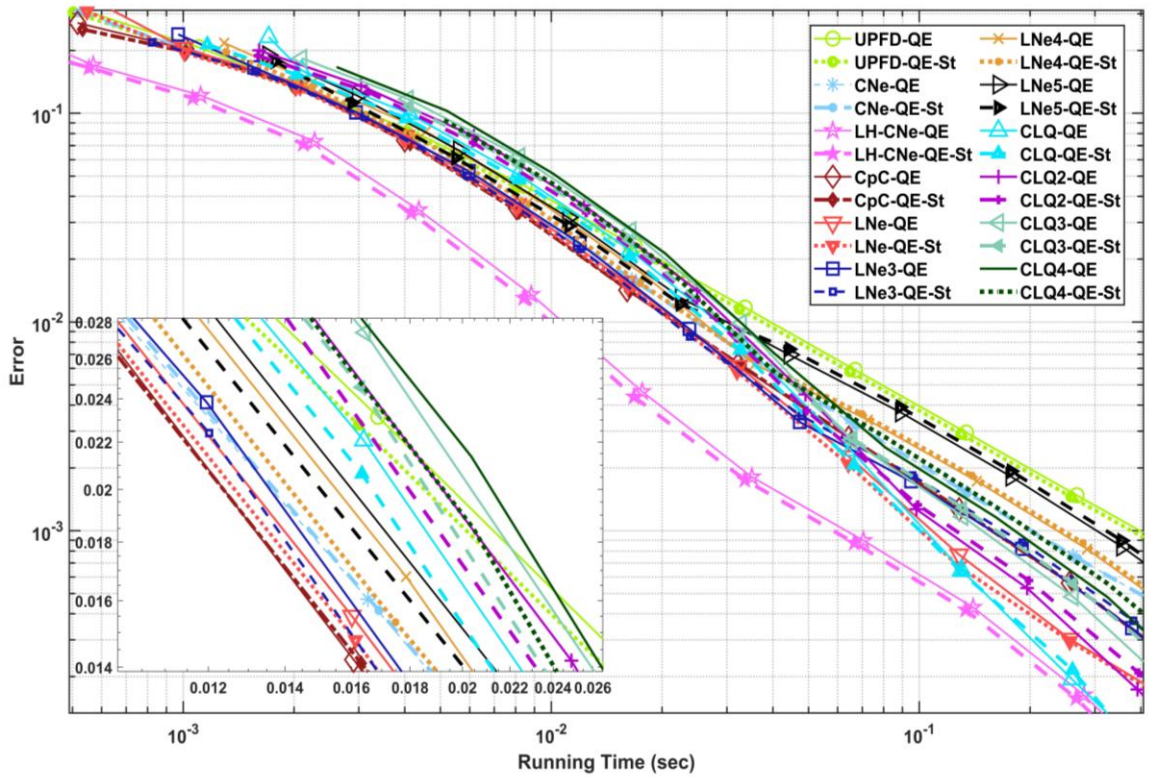


Figure 4.3. Maximum errors as a function of the running time, low accuracy part.

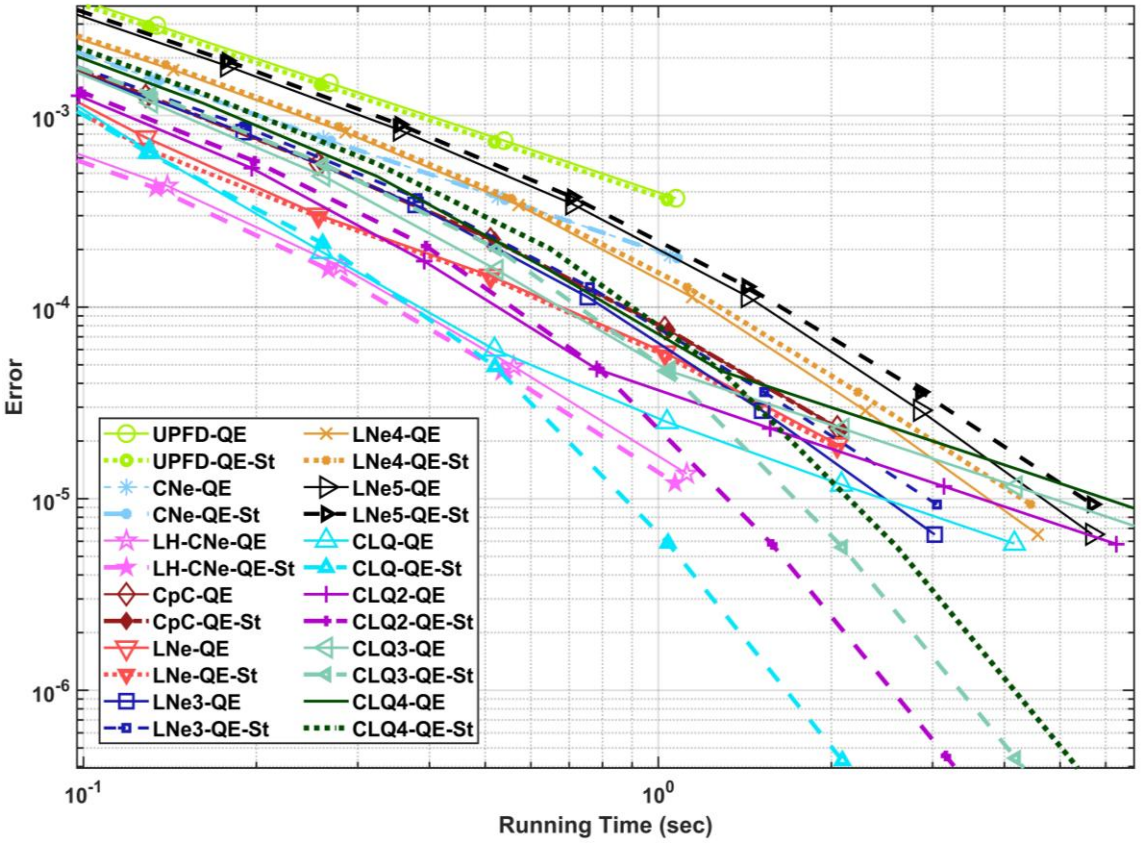


Figure 4.4. Maximum errors in relation to high accuracy and running time.

In this anysotropic system, the used system size and final time are $N_x = 15, N_y = 80, t^{\text{fin}} = 0.4,$ the coefficient of the Fisher term is $\beta = 9$, while the exponents are $a_C = 1, b_C = 2$ and $a_{Rx} = 2, b_{Rx} = 2, a_{Ry} = 0, b_{Ry} = 2$. These parameters give $SR = 5.66 \cdot 10^5$ and $h_{MAX}^{FTCS} = 10^{-4}$, which means that the system is only moderately stiff. However, the resistances in the x direction are two orders of magnitude larger than in the y direction, therefore the system is quite anisotropic. The initial concentrations are random numbers in the left half of the unit interval: $u_i^0 = rand / 2$. I present the maximum errors as a function of the running times in Fig. 4.5.

One can see that all of the examined 24 combinations behave well for this case as well. In Fig. 4.3 ,4.4 and 4.5, the LH method is the most efficient with and without Strang-splitting when only low accuracy is required. The LNe and CpC methods also have acceptable performance. For high accuracy, the CLQ family with Strang-splitting clearly outperforms the other methods. Based on the results of above I choose LH-CNE-Qe, LH-CNe-QE-St, CpC-QE, LNe3-QE-St, LNe4-QE-St, LNe5-QE-St, CLQ-QE-St, CLQ2-QE-St, CLQ3-QE-St, and CLQ4-QE-St as top 10 methods for further investigation.

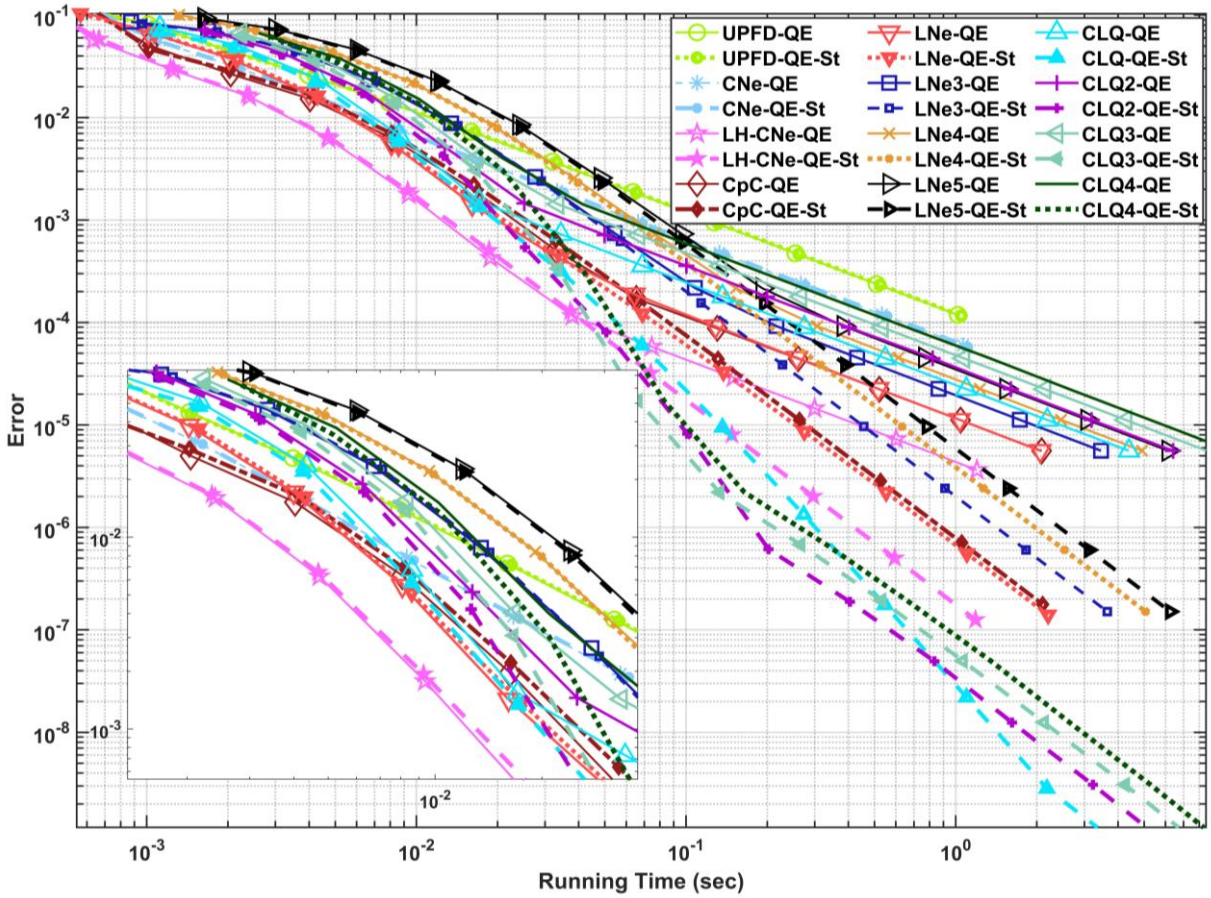


Figure 4.5. Maximum errors as a function of the running time.

I note that I performed these running time measurements in the case of the PI treatments just to check whether it is not significantly faster than the QE treatment. I found that choosing QE or PI does not noticeably influence the running times, which is logical since the exponential

expression $e^{-\beta\Delta t}$ does not depend on time or space, thus it has to be calculated only once. However, in cases where the coefficient β depends on both space and time, calling the exponential function could significantly slow down the calculations when they use the QE treatment, thus the PI could become competitive.

4.4. Testing of performance with parameter-sweep for the Top 10 methods

Test problems with different stiffness ratios have been constructed using the exponents of the mesh-cells data distribution. To be more specific, we made use of the factors listed in Table 4.2. The nonlinear coefficient was $\beta = 5$, the size of the grid is fixed to $N_x = 21$ and $N_y = 20$, the initial values are $u_i^0 = rand$, while the final time is $t^{\text{fin}} = 0.4$. I applied *S*=15 different time step sizes and then applied the formula (3.8) of the aggregated error.

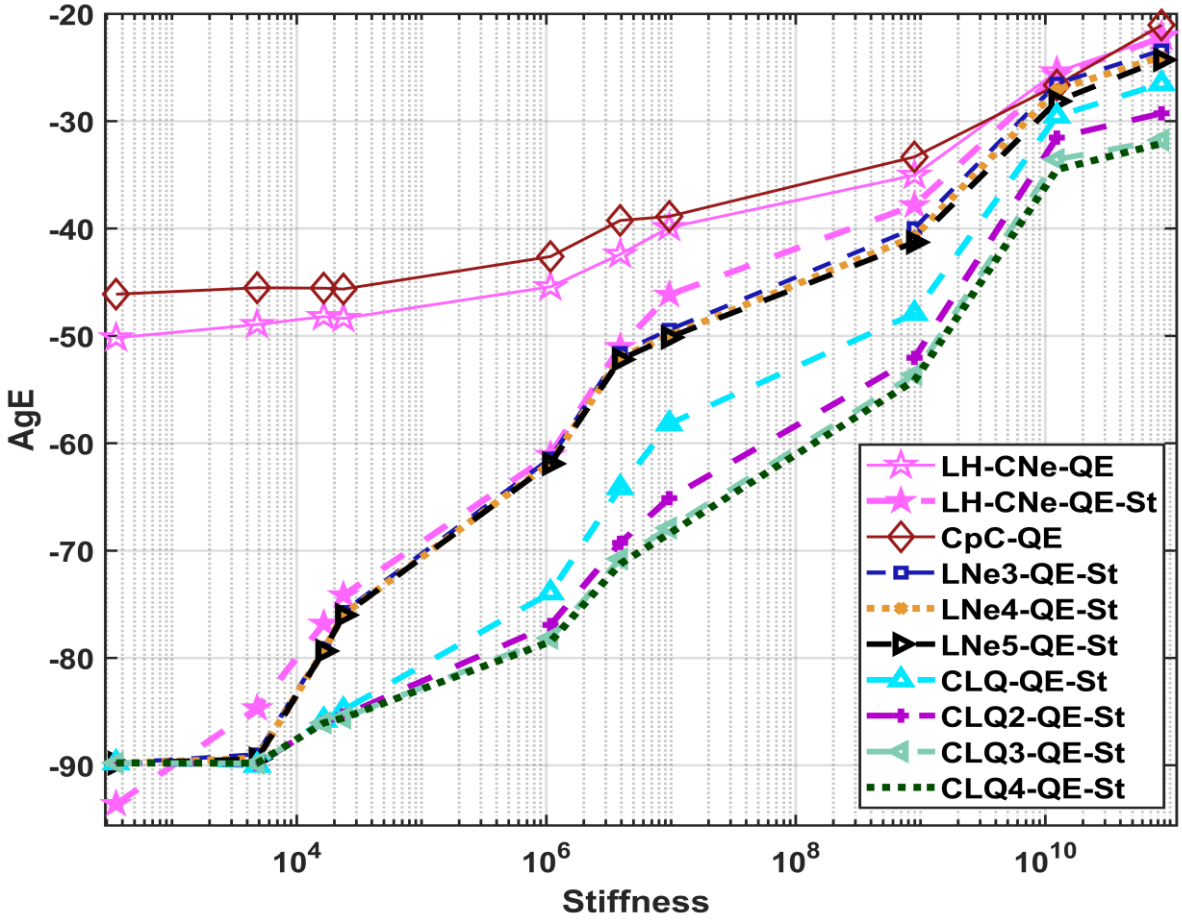


Figure 4.6. The AgE errors as a function of stiffness in the case of the positivity preserving methods.

The AgE errors as a function of stiffness ratio is displayed in Figure 4.6 and Table 4.3. I see that the methods' accuracy decreases with rising stiffness ratios and decreasing CFL limit, as it is expected. For low stiffness, the Strang-splitting treatment has a substantial advantage, which is diminishing with increasing stiffness. The relative advantage of the high-order CLQ-family is large for medium stiffness, but starts to vanish for very high values of the stiffness ratio due to the so-called order-reduction.

Table 4.2. The exponents of the capacities and resistances.

Number	Type	a_c	b_c	a_{RX}	b_{Rx}	a_{Rz}	b_{Rz}	Stiffness ratios	h_{Max}
1	Non Stiff	0	0	0	0	0	0	356	0.251
2		-1	1	0	0	0	0	4806.8	0.048
3	Medium Stiff	-1	1	-1	1	0	0	1.64×10^{4}	0.012
4		-1	1	-1	1	-1	1	2.36×10^{4}	0.007
5	Moderately Stiff	-2	2	-1	1	-1	1	1.11×10^{6}	0.001
6		-2	2	-2	2	-1	1	3.92×10^{6}	2.98×10^{-4}
7		-2	2	-2	2	-2	2	9.74×10^{6}	1.45×10^{-4}
8	Very Stiff	-3	3	-2	2	-2	2	8.97×10^{8}	2.04×10^{-5}
9		-3	3	-3	3	-2	2	1.34×10^{10}	1.48×10^{-6}
10		-3	3	-3	3	-3	3	8.56×10^{10}	1.48×10^{-6}

Table 4.3. Aggregated error (AgE) values for positivity preserving methods of different treatment.

Algorithms	The stiffness values									
	356	4806.8	1.6×10^{4}	2.4×10^{4}	1.1×10^{6}	3.9×10^{6}	9.7×10^{6}	8.9×10^{8}	1.3×10^{10}	8.5×10^{10}
	AgE Errors									
LH-CNe-QE	-50.18	-48.94	-48.20	-48.36	-45.42	-42.39	-39.90	-35.03	-25.41	-22.08
LH-CNe-QE-St	-93.60	-84.68	-76.80	-74.15	-61.08	-51.02	-46.17	-37.83	-25.72	-22.21
CpC-QE	-46.11	-45.52	-45.55	-45.63	-42.61	-39.25	-38.87	-33.33	-26.64	-21.05
LNe3-QE -St	-89.78	-88.97	-79.07	-75.70	-61.40	-51.58	-49.43	-39.96	-26.50	-23.40
LNe4-QE -St	-89.80	-89.34	-79.31	-75.96	-61.82	-51.99	-49.97	-40.76	-27.05	-24.01
LNe5-QE -St	-89.80	-89.41	-79.35	-76.00	-61.90	-52.19	-50.14	-41.28	-28.14	-24.28
CLQ-QE -St	-89.76	-89.99	-85.79	-84.86	-73.93	-64.14	-58.19	-47.9	-29.52	-26.51
CLQ2-QE -St	-89.77	-89.86	-86.03	-85.36	-76.92	-69.28	-65.11	-52.04	-31.52	-29.27
CLQ3-QE -St	-89.77	-89.81	-86.03	-85.57	-78.15	-70.77	-67.88	-53.59	-33.56	-31.76
CLQ4-QE-St	-89.77	-89.80	-86.04	-85.59	-78.48	-71.23	-68.40	-54.16	-34.50	-32.08

Here, I used the $\beta \in \{0, 1, 2, 4, 6, 10, 15, 20, 25, 30\}$ values of the nonlinear coefficient to calculate the AgE values. All *a* and *b* exponents are zero, which means $C = 1$, $R_x = R_y = 1$ for all cells. The size of the grid is fixed to $N_x = 101$ and $N_y = 2$, while the final time is $t^{\text{fin}} = 0.1$. The initial values are $u_i^0 = 1/i$.The AgE errors as a function of the β parameter are displayed in Fig.3.7

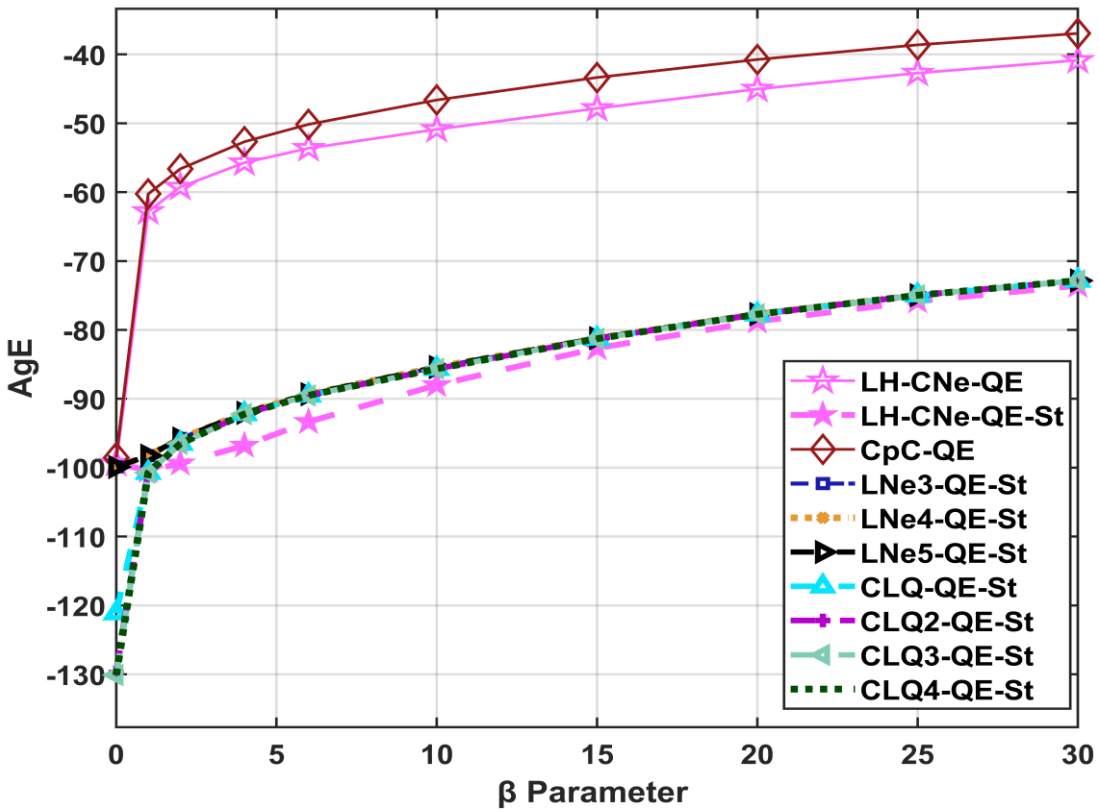


Figure 4.7. AgE errors as a function of the β parameter in the case of the positivity preserving methods.

4.5. Summary of this chapter

I studied numerical methods about which it is analytically proven that they keep the values of the concentration function in the unit interval for arbitrary time step size and arbitrary values of the nonlinear parameter β when they are applied to Fisher's equation. The numerical case studies confirmed these results, since all methods behaved well without the slightest sign of instability.

According to the running time measurements, generally the LH-CNe is the most efficient among the methods since it serves quite accurate results in very short time. However, if stiffness or anisotropy increases, its advantage vanishes. In those cases, the CLQ or the CLQ2 methods are the optimal choice.

When the nonlinearity is strong, i.e., the β coefficient is large, the quasi-exact treatment of the nonlinear term combined with Strang-splitting is recommended, especially when the method for the diffusion-part is relatively accurate, usually due to higher order of convergence. However, if the accuracy is limited by the anisotropy, or more importantly, by the stiffness of the problem, Strang-splitting can be a waste of time and the time step size should be decreased instead.

5.1. DYNAMICALLY CONSISTENT METHODS FOR HUXLEY'S EQUATION

The efficiency of various numerical methods for solving Huxley's equation which includes a diffusion term and a nonlinear reaction term is investigated. Conventional explicit finite difference algorithms often suffer from severe stability limitations and can yield unphysical concentration values. In this section, I collect a range of stable, explicit time integration methods of first to fourth order, originally developed for the diffusion equation, and design treatments of the nonlinear term that ensure that the solution remains within the physically meaningful unit interval. This property, called dynamical consistency, is analytically proven and implies unconditional stability. In addition to this, the most effective ones are identified from the large number of constructed method combinations. I conduct systematic tests in one and two spatial dimensions, also evaluating computational efficiency in terms of CPU time. Our results show that higher-order schemes are not always the most efficient: in certain parameter regimes, second-order methods can outperform their higher-order counterparts.

5.1.1. The operator-splitting treatments of the nonlinear term

During a time step, I independently take into account the effect of the diffusion term and the Huxley term. The diffusion term is treated by 13 methods listed in Section 2. After fully considering the diffusion term by those schemes, I obtain a concentration value u_i^{diff}, which I temporarily denote by p for brevity. The reaction term is local and can be written into the form $\beta u^2 - \beta u^3$. A very efficient way to take into account this nonlinear term is to solve the initial value problem

$$\begin{cases} \dfrac{du}{dt} = \beta u^2 - \beta u^3, \\ u(t=0) = u_i^{\text{diff}} = p, \; t \in [0, \Delta t] \end{cases}, \tag{5.1}$$

analytically, as in the previous chapter in the case of Fisher's Equation. However, I was unable to solve it, even with the assistance of Maple 18 program. Hence I choose an approach which starts with a selective replacement of u by the p values in all possible 12 ways:

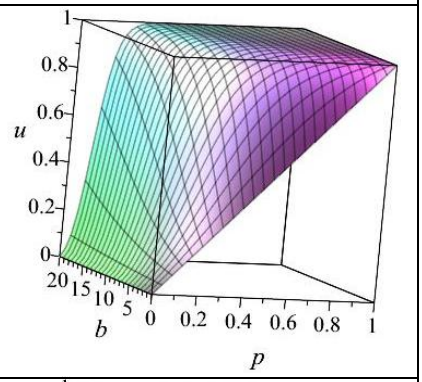
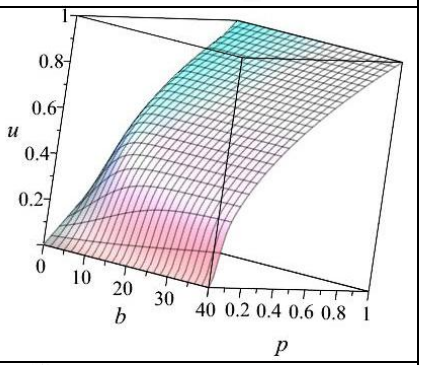
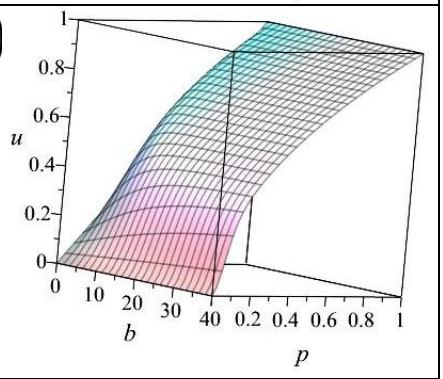
$$\begin{array}{lll} \beta u^2 - \beta u^3 & \beta up - \beta u^3 & \beta p^2 - \beta u^3 \\ \beta u^2 - \beta u^2 p & \beta up - \beta u^2 p & \beta p^2 - \beta u^2 p \\ \beta u^2 - \beta up^2 & \beta up - \beta up^2 & \beta p^2 - \beta up^2 \\ \beta u^2 - \beta p^3 & \beta up - \beta p^3 & \beta p^2 - \beta p^3 \end{array}. \tag{5.2}$$

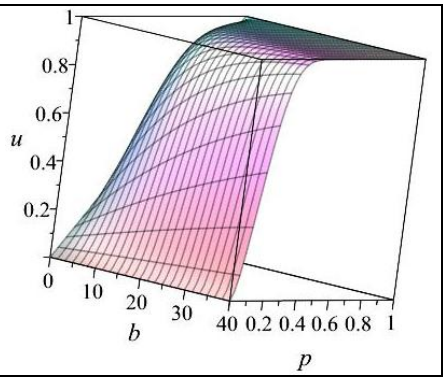
Three of these ways, namely $\beta up - \beta up^2$, $\beta up - \beta p^3$ and $\beta p^2 - \beta up^2$ can be considered as a kind of linearization, while the last one without u is the equivalent to the explicit Euler treatment. After placing each of the expressions (5.1) into the right-hand side of (5.2), I attempt to solve the acquired initial value problems, for example

$$\begin{cases} \dfrac{du}{dt} = \beta pu - \beta pu^2, \\ u(t=0) = u_i^{\text{diff}} = p,\ t \in [0, \Delta t] \end{cases}, \tag{5.3}$$

again analytically. If the solution is found, I first check whether the obtained analytical solutions guarantee dynamical consistency by plotting it as a function of p and $b := \beta\Delta t$. I found that there are two circumstances in which there is no analytical solution, 6 cases (including the linearization $\beta up - \beta up^2$) in which the solution exists but can be bigger than one, and 4 cases in which the solution is within the unit interval. These latter four cases will be used to treat the nonlinear term, and they are listed in the following Table 5.1.

Table 5.1. Four treatments of the nonlinear term.

Treatment	substitution	Solution	3D plot of the solution
1	$\beta up - \beta u^2 p$	$u(\Delta t) = \dfrac{p}{p + (1-p)e^{-pb}}$ or $u_i^{n+1} = \dfrac{u_i^{\text{diff}}}{u_i^{\text{diff}} + \left(1 - u_i^{\text{diff}}\right) e^{-u_i^{\text{diff}} \beta t}}$ (5.4)	
2	$\beta up - \beta u^3$	$u(\Delta t) = \dfrac{p}{\sqrt{p + (1-p)e^{-2pb}}}$ or $u_i^{n+1} = \dfrac{u_i^{\text{diff}}}{\sqrt{u_i^{\text{diff}} + \left(1 - u_i^{\text{diff}}\right) e^{-2u_i^{\text{diff}} \beta t}}}$ (5.5)	
3	$\beta p^2 - \beta u^2 p$	$u(\Delta t) = \sqrt{p}\ \tanh\left(b p^{3/2} + \operatorname{arctanh}\left(\sqrt{p}\right)\right)$ or $u_i^{n+1} = \sqrt{u_i^{\text{diff}}}\ \tanh \left(\beta t \left(u_i^{\text{diff}}\right)^{3/2} + \operatorname{arctanh}\left(\sqrt{u_i^{\text{diff}}}\right)\right)$ (5.6)	

4	$\beta p^2 - \beta p^2 u$	$u(\Delta t) = 1-(1-p)e^{-p^2 b}$ or $u_i^{n+1} = 1-\left(1-u_i^{\text{diff}}\right)e^{-\left(u_i^{\text{diff}}\right)^2 \beta t}$ (5.7)	

5.1.2. Verification 1D

Now PDE (1.4) with $\alpha = 1$ is going to be solved in the domain $(x,t) \in [x_0, x_N] \times \left[t^{\text{ini}}, t^{\text{fin}}\right]$. The analytical solution [26](p. 34) is used as the reference.

$$u^{\text{exact}}(x,t) = \frac{e^{\sqrt{\frac{\beta}{2}}x + \frac{\beta}{2}t}}{e^{\sqrt{\frac{\beta}{2}}x + \frac{\beta}{2}t} + c}. \tag{5.8}$$

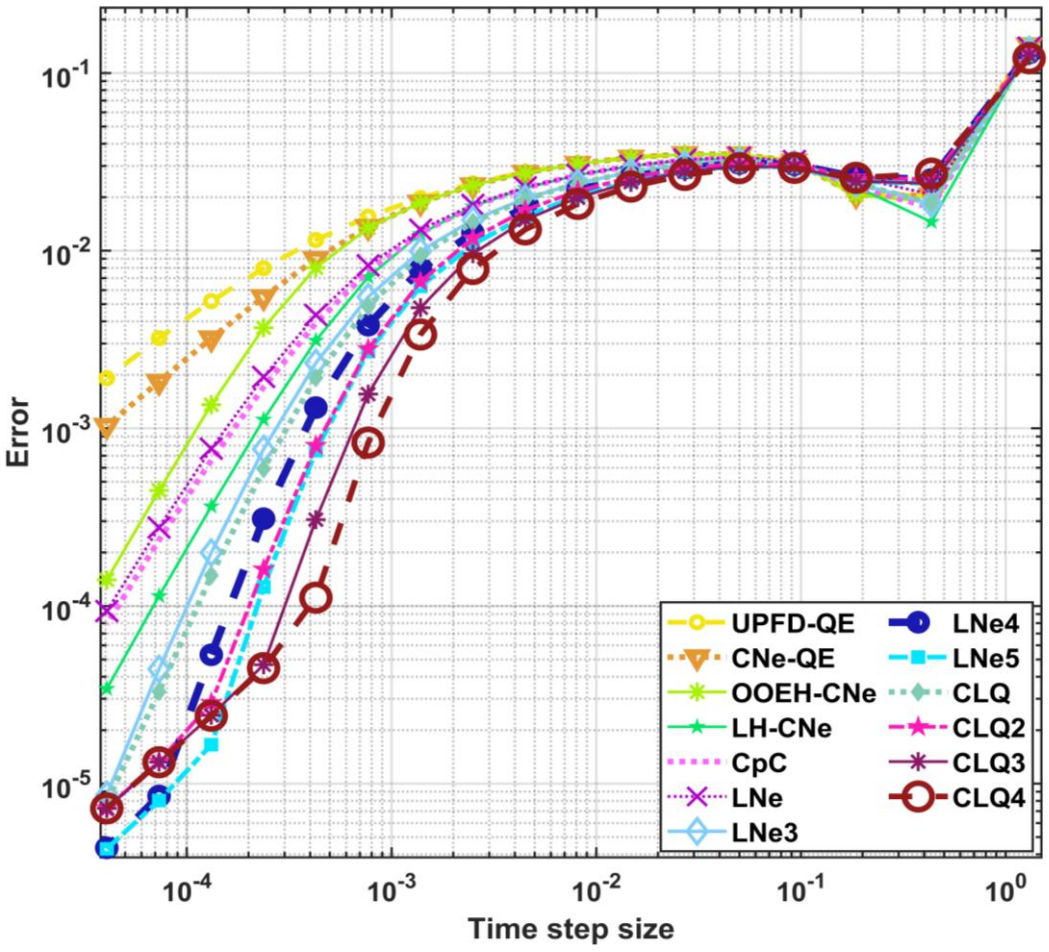


Figure 4.1. Maximum errors as a function of the time step size for the PI treatment.

The initial and the Dirichlet boundary conditions are obtained by evaluating this function at the time and space points belonging to the initial time $u(x,0)$, and the boundaries $u(x_0,t), u(x_N,t)$, respectively. PDE (1.4) is discretized on the $x_0=0, x_{N=200}=4, t^{\text{ini}}=0.00, t^{\text{fin}}=1.3$ domain using $\Delta x=0.02$ space step size. The nonlinear coefficient is $\beta=4.3$, while c=1 in Eq. (5.8). The remaining four reaction-treatments are organized with halved time steps by Strang-splitting as well. In this way, 12 treatments are constructed and tested with the 13 diffusion-solvers, i.e. $12\times13=156$ algorithm-combinations are tested. Considering that I have two test problems and S=13 different time step sizes, I performed 4056 runs with error calculations.

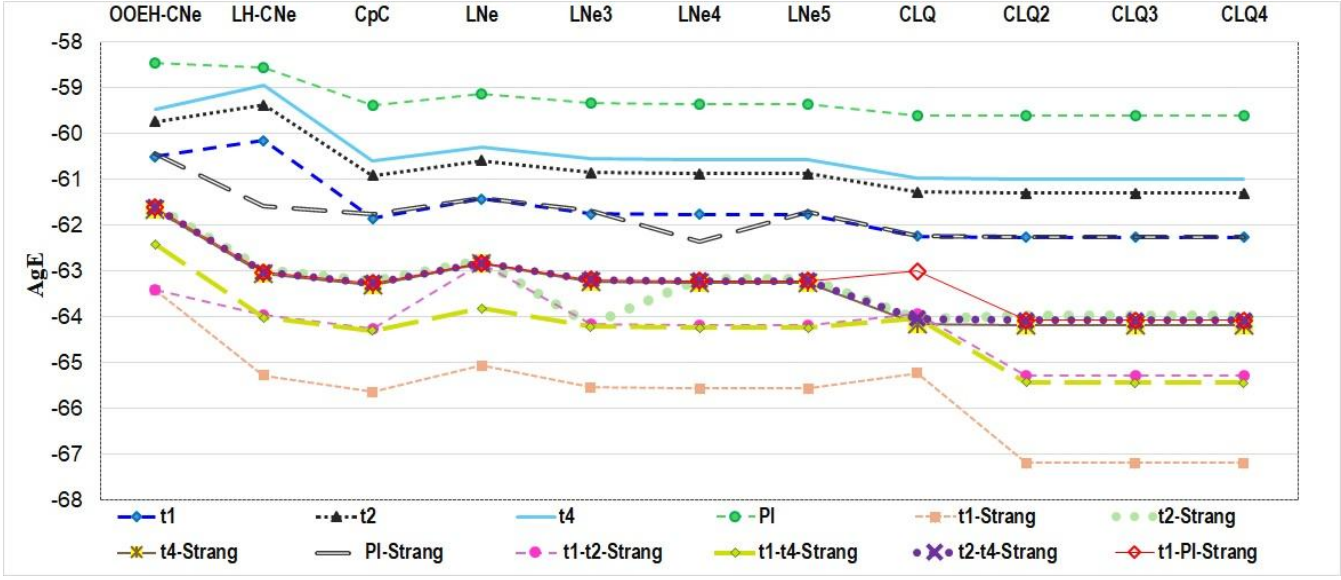


Figure 5.2. The AgE errors for different treatments of the diffusion and the nonlinear.

Table 5.1. The AgE errors for different treatments of the diffusion and the nonlinear term.

	Treatment of the nonlinear term											
method	t1	t2	t4	PI	t1-Strang	t2-Strang	t4-Strang	PI-Strang	t1-t2-Strang	t1-t4-Strang	t2-t4-Strang	t1-PI-Strang
UPFD	-41.98	-41.99	-41.98	-41.99	-42.00	-42.00	-42.00	-42.00	-42.00	-41.99	-41.99	-41.99
CNe	-45.99	-46.00	-45.97	-45.98	-45.99	-46.00	-45.99	-45.99	-46.00	-45.99	-45.99	-45.99
OOEH-CNe	-60.50	-59.73	-59.47	-58.46	-63.41	-61.58	-61.64	-60.42	-63.41	-62.41	-61.61	-61.62
LH-CNe	-60.15	-59.37	-58.93	-58.55	-65.26	-62.96	-63.06	-61.58	-63.95	-64.01	-63.03	-63.02
CpC	-61.85	-60.91	-60.60	-59.38	-65.63	-63.21	-63.31	-61.75	-64.24	-64.29	-63.25	-63.25
LNe	-61.42	-60.59	-60.30	-59.13	-65.05	-62.76	-62.82	-61.39	-62.82	-63.81	-62.82	-62.82
LNe3	-61.74	-60.85	-60.55	-59.33	-65.53	-64.16	-63.22	-61.68	-64.16	-64.21	-63.18	-63.19
LNe4	-61.76	-60.87	-60.57	-59.34	-65.56	-63.16	-63.24	-62.36	-64.18	-64.23	-63.20	-63.21
LNe5	-61.77	-60.88	-60.57	-59.35	-65.57	-63.17	-63.25	-61.70	-64.19	-64.24	-63.21	-63.22
CLQ	-62.23	-61.28	-60.98	-59.60	-65.22	-64.03	-64.15	-62.24	-63.92	-64.03	-64.03	-62.99
CLQ2	-62.26	-61.29	-60.99	-59.61	-67.18	-63.95	-64.18	-62.26	-65.27	-65.42	-64.07	-64.07
CLQ3	-62.26	-61.29	-60.99	-59.61	-67.18	-63.96	-64.18	-62.26	-65.27	-65.42	-64.07	-64.07
CLQ4	-62.26	-61.29	-60.99	-59.61	-67.18	-63.96	-64.18	-62.26	-65.27	-65.42	-64.07	-64.07

The AgE values are tabulated in Tables 5.1 and visualized in Figures 5.2. I note that the errors of the different treatments are sometimes very close to each other, the difference between them can be only in the third or fourth decimals.

5.1.3. Numerical experiments:selection by the performance with running time measurements in 2D

The stiff system with weaker nonlinearity figure 4.3-4.4 size is $N_x = 51, N_y = 60,$ the final time is $t^{\text{fin}} = 0.01$ the coefficient of the reaction-term $\beta = 2.9,$ while the exponents are $a_C = 3, b_C = 6, a_{Rx} = 1, b_{Rx} = 2, a_{Ry} = 2, b_{Ry} = 4$. The problem we obtained using these parameters is characterized by the $SR = 2.68 \cdot 10^9$ and $h_{MAX}^{FTCS} = 2.1 \cdot 10^{-5}$ numbers, hence the system is rather stiff. The initial concentration-function is smooth: $u_i^0 = \frac{1}{\sqrt{1+i}} + 0.001$.

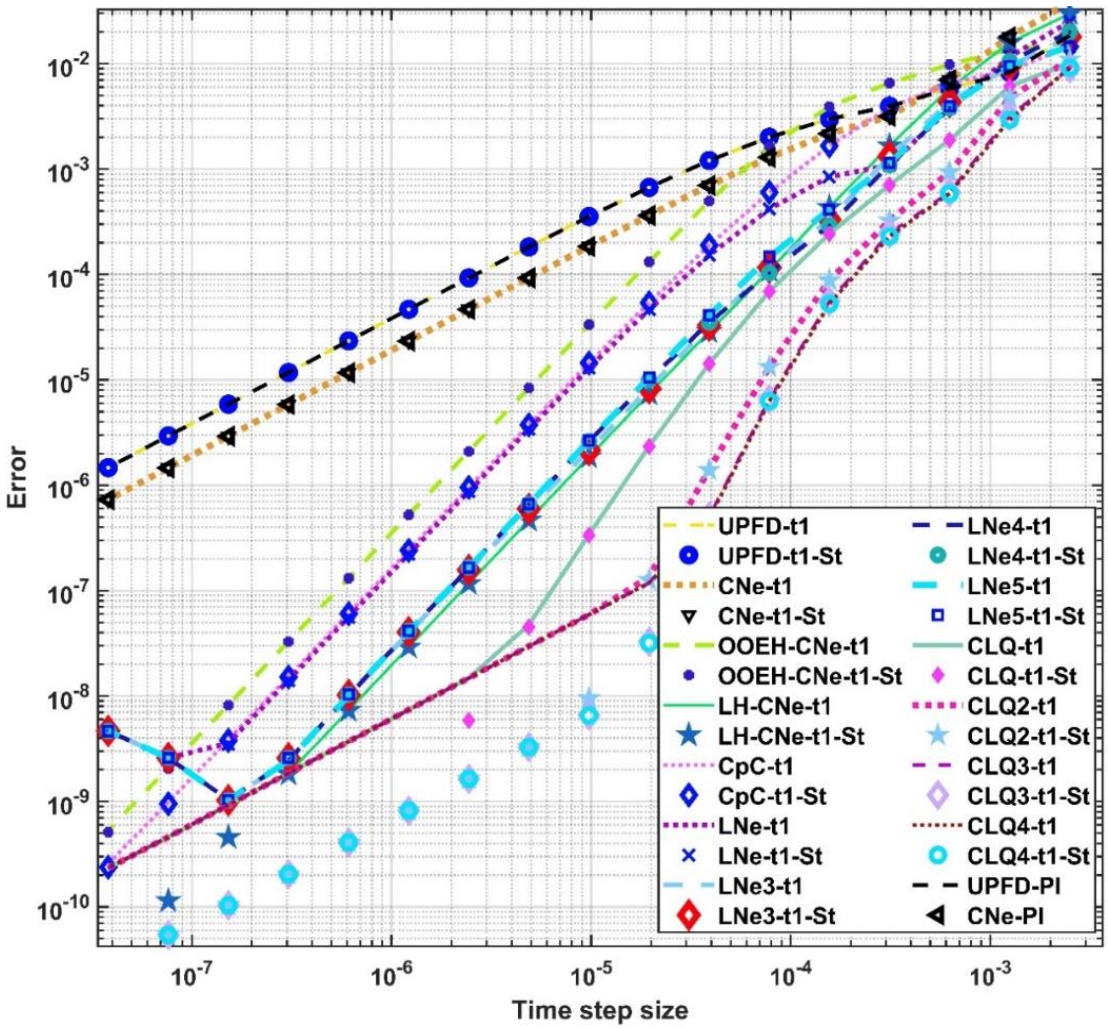


Figure 5.4. Maximum errors as a function of the time step size in stiff system.

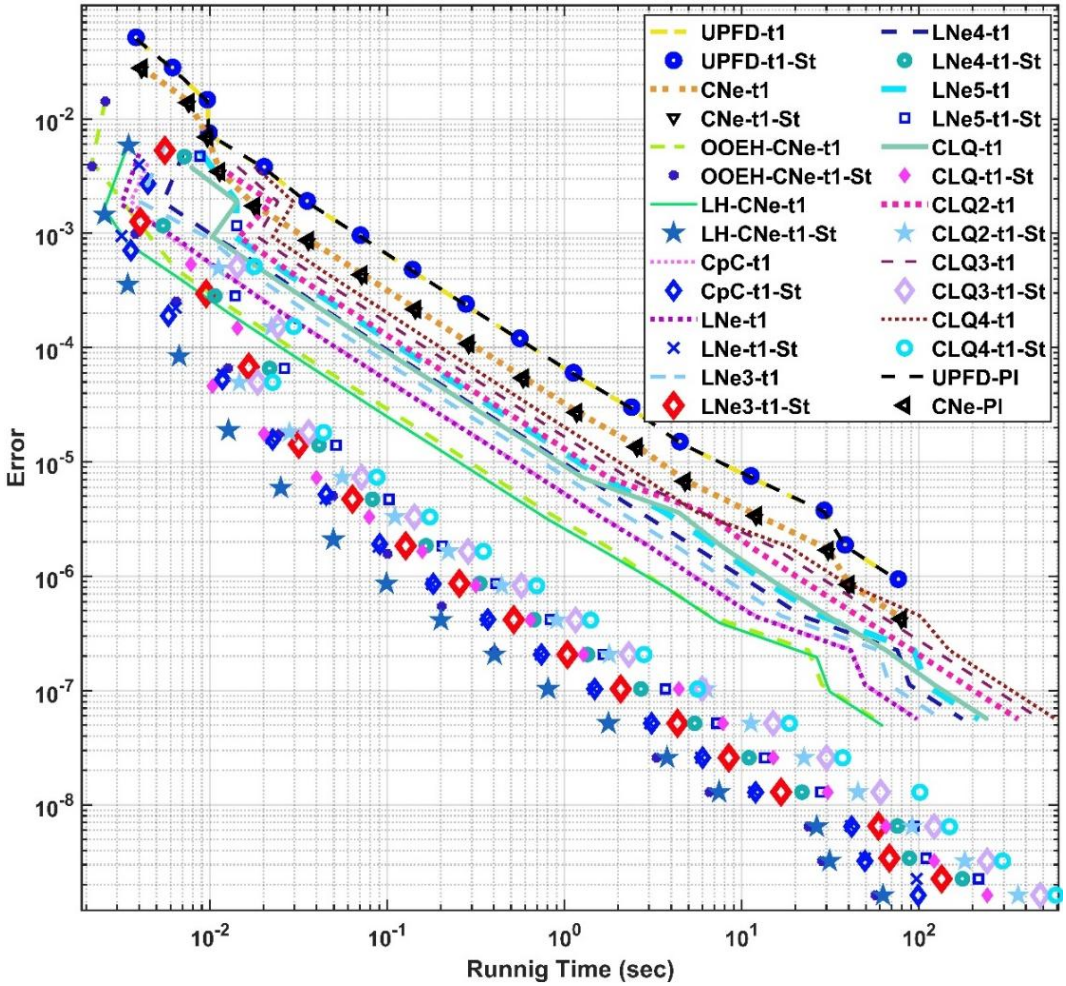


Figure 5.4. Maximum errors as a function of the running time in stiff system.

5.2. Solution of Huxley's equation by efficient numerical methods

In the previous subsection, I focused on only dynamically consistent methods. However, as I already mentioned in Chapter 5.1, these qualitative properties are usually contradict with the requirement of fast convergence. Thus, in this subsection, I start from those four explicit and stable numerical schemes which are proven to be among the most efficient solvers for the diffusion equation in the earlier work of our group. Using these diffusion solvers, several new methods are constructed for the nonlinear Huxley equation. Then, based on extensive numerical case studies in one- and two-dimensional spatial domains, the least-performing methods are gradually eliminated, leaving only the best-performing ones. During the tests, as above, not only a single time-step size but all relevant time-step sizes are considered, and corresponding run-time measurements are performed.

5.2.1. Further treatments of the nonlinear term

In this subsection, some operator splitting is applied using Equations (5.4), (5.5), and (5.7) from the previous subsection to compute the quantities marked respectively as t1, t2, and t8 in the figure, but I include a new treatment, labelled as t3, which is not necessarily preserves

dynamical consistency, but might be more efficient. It is derived from the expression $\beta u^2 - \beta u^2 p$ and has the explicit formula follows:

$$u_i^{n+1} = \frac{u_i^{\text{diff}}}{\left(\left(u_i^{\text{diff}}\right)^2 \beta\Delta t\right) - \left(u_i^{\text{diff}} \beta\Delta t\right) + 1}. \tag{5.9}$$

Besides these, I include the following further treatments.

Inside treartment. The Huxley term is evaluated at the beginning of the actual time step to obtain $\beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2$ and inserted during the space- and time discretization of the equation. It is multiplied by the time step size (Δt) to assess the increment due to the Huxley term. In this approach, the nonlinear term turns up in the numerator of the formulas, as listed below. LH method: First stage: $u_i^{n+1} = \dfrac{u_i^n + \beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2 / 2 + A_i / 2}{1 + r_i / 2}$. (5.10)

Second stage: $u_i^{n+1} = \dfrac{\left(1 - r_i / 2\right) u_i^n + 0.5\beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2 + A_i}{1 + r_i / 2}$. (5.11)

DF scheme: First, UPFD step: $u_i^{n+1} = \dfrac{u_i^n + A_i + \beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2}{1 + 2r_i}$. (5.12)

All other steps: $u_i^{n+1} = \dfrac{\left(1 - r_i\right) u_i^{n-1} + 2A_i + \beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2}{1 + r_i}$. (5.13)

OOEH: First, UPFD step: $u_i^{n+1} = \left(1 - r_i\right) u_i^n + A_i + \beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2$. (5.14)

Second steps: $u_i^{n+1} = \dfrac{u_i^n + A_i + \beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2}{1 + r_i}$. (5.15)

Pseudo implicit treatment (PI). This case is similar than the previous one, but one of the u_i-s are evaluated in the end of the time step, thus $\beta\Delta t\left(1-u_i^{n+1}\right)\left(u_i^n\right)^2$ is inserted instead of $\beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2$. When we rearrange the formula to express u_i^{n+1}, the nonlinear term appears in the denominator as well. The goal of this trick is to enhance stability.

LH method: $u_i^{n+1} = \dfrac{u_i^n\left(1 + \beta\Delta t u_i^n / 2\right) + A_i / 2}{1 + r_i / 2 + \beta\Delta t\left(u_i^n\right)^2 / 2}$ and $u_i^{n+1} = \dfrac{u_i^n\left(1 - r_i / 2 + \beta\Delta t u_i^n\right) + A_i}{1 + r_i / 2 + \beta\Delta t\left(u_i^n\right)^2}$. (5.16)

DF: First $u_i^{n+1}=\dfrac{u_i^n\left(1+\beta\Delta t u_i^n\right)+A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2}$ Second stage $u_i^{n+1}=\dfrac{\left(1-r_i\right)u_i^{n-1}+\beta\Delta t\left(u_i^n\right)^2+2A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2}$. (5.17)

OOEH: First: $u_i^{n+1}=\dfrac{u_i^n\left(1-r_i+\beta\Delta t u_i^n\right)+A_i}{1+\beta\Delta t\left(u_i^n\right)^2}$ Second stage $u_i^{n+1}=\dfrac{u_i^n\left(1+\beta\Delta t u_i^n\right)+A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2}$. (5.18)

Mixed treatment. Mixed treatment means that simply the average of the previous two expressions $\beta\Delta t\left(1-u_i^n\right)\left(u_i^n\right)^2$ and $\beta\Delta t\left(1-u_i^{n+1}\right)\left(u_i^n\right)^2$ is inserted during the discretization. This treatment was proved to be successful in our previous work [98] for the convection term. In this case, we have the following formulae. LH method :

$$\text{First stage: } u_i^{n+1}=\frac{u_i^n\left(1+0.5\beta\Delta t u_i^n\right)\left(1-u_i^n/2\right)+A_i/2}{1+r_i+\beta\Delta t\left(u_i^n\right)^2/4}. \tag{5.19}$$

$$\text{Second stage: } u_i^{n+1}=\frac{u_i^n\left(1-r_i/2+\beta\Delta t u_i^n\right)\left(1-u_i^n/2\right)+A_i}{1+r_i/2+\beta\Delta t\left(u_i^n\right)^2/2}. \tag{5.20}$$

$$\text{DF scheme: First, UPFD step: } u_i^{n+1}=\frac{u_i^n\left(1+\beta\Delta t u_i^n\right)\left(1-u_i^n/2\right)+A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2/2}. \tag{5.21}$$

$$\text{All other steps: } u_i^{n+1}=\frac{\left(1-r_i\right)u_i^{n-1}+2\beta\Delta t\left(1-u_i^n/2\right)\left(u_i^n\right)^2+2A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2}. \tag{5.22}$$

$$\text{OOEH:First, UPFD step: } u_i^{n+1}=\frac{\left(1-r_i\right)u_i^n+\beta\Delta t u_i^n\left(1-u_i^n/2\right)+A_i}{1+\beta\Delta t\left(u_i^n\right)^2/2}. \tag{5.23}$$

$$\text{Second steps: } u_i^{n+1}=\frac{u_i^n\left(1+\beta\Delta t u_i^n\right)\left(1-u_i^n/2\right)+A_i}{1+r_i+\beta\Delta t\left(u_i^n\right)^2/2}. \tag{5.24}$$

5.2.2. One space dimension using an exact solution

Next the figure 5.5 is on the $x_0=0, x_{N=500}=2, t^{\text{ini}}=0.01, t^{\text{fin}}=0.1$ domain using $\Delta x=0.004$ space step size. The nonlinear coefficient is $\beta=23$, while c=19 in Eq. (4.8).

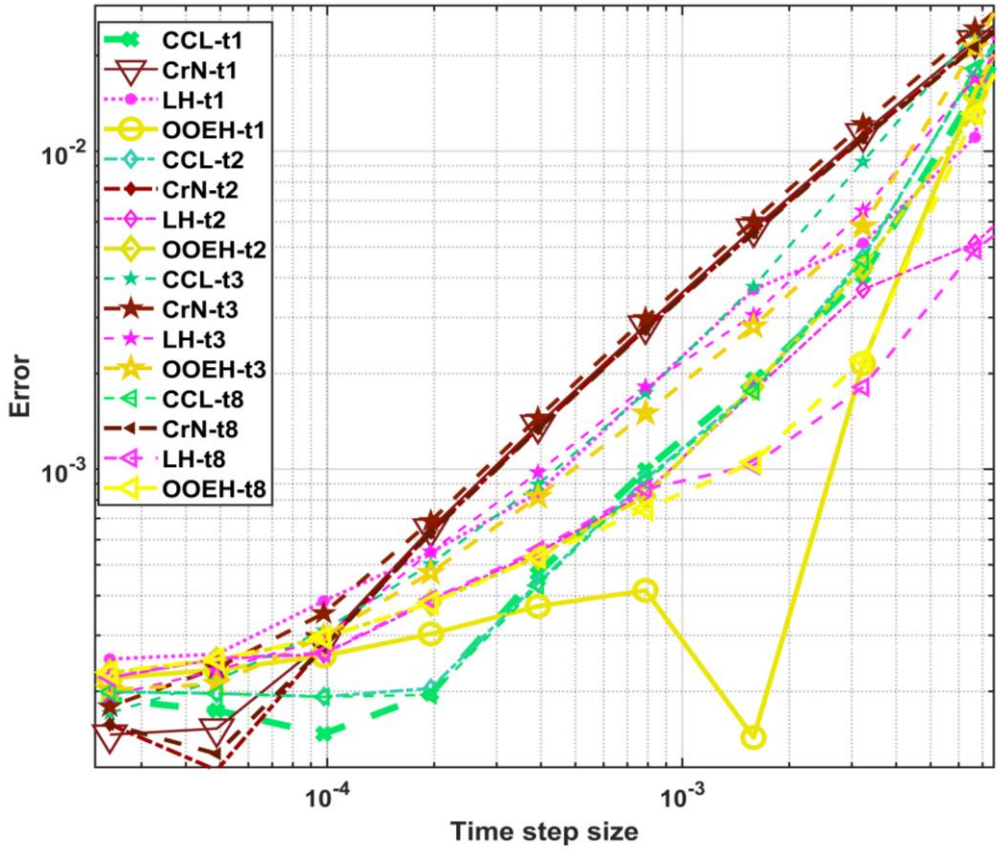


Figure 5.5. Maximum errors as a function of time step size for different operator-splitting treatments.

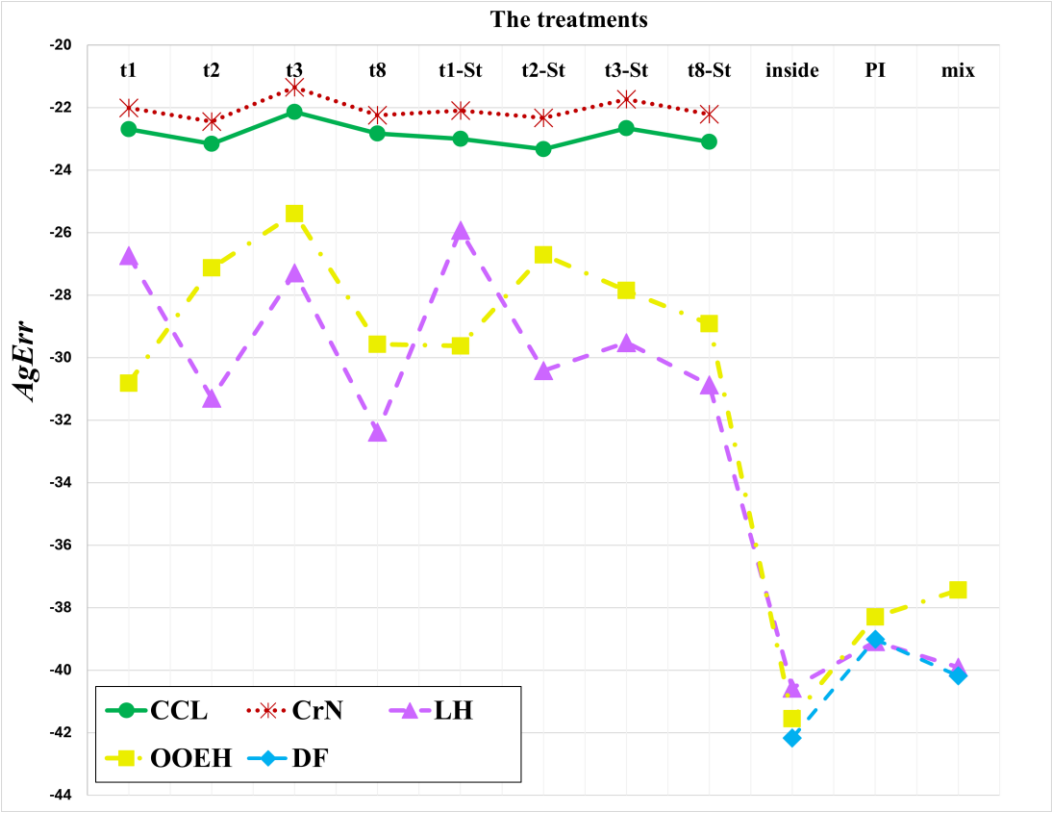


Figure 5.6. The aggregated errors (AgErr) for different diffusion-solvers and treatments.

The aggregated errors for different diffusion-solvers and treatments of the Huxley are shown in Figure 5.6 and Table 5.2.

Table 5.2. The aggregated error (AgErr) values for different treatment.

Methods	Treatment of the nonlinear term										
	t_1	t_2	t_3	t_8	t_1-St	t_2-St	t_3-St	t_8-St	inside	PI	mix
	AgErr										
CCL	-22.70	-23.16	-22.13	-22.83	-23.00	-23.33	-22.66	-23.10	-	-	-
CrN	-22.00	-22.44	-21.34	-22.24	-22.10	-22.33	-21.73	-22.21	-	-	-
LH	-26.73	-31.29	-27.29	-32.38	-25.92	-30.42	-29.51	-30.87	-40.58	-39.08	-39.90
OOEH	-30.81	-27.13	-25.40	-29.60	-29.62	-26.71	-27.85	-28.91	-41.55	-38.29	-37.44
DF	-	-	-	-	-	-	-	-	-42.17	-39.01	-40.18

5.2.3. 2D Stiff system

I tested Stiff 2D system problems by changing the *a* and *b* values. The variable rand are MATLAB-generated random numbers within the unit interval. The Stiff system fig(5.7-5.8), the system size and the final time used are $N_x = 31, N_y = 13, t^{\text{fin}} = 0.3$, and $\beta = 8.3$, while the exponents are $a_C = 3, b_C = 6, a_{Rx} = 1, b_{Rx} = 2, a_{Ry} = 1, b_{Ry} = 2$. These variables give $SR = 9.1 \cdot 10^8$ and $h_{MAX}^{FTCS} = 1.7 \cdot 10^{-5}$. This indicates that the system is moderately stiff only. The initial concentration function is the sum of a smooth function and some random numbers: $u_i^0 = 1/\sqrt{i+4} + 0.2 rand$

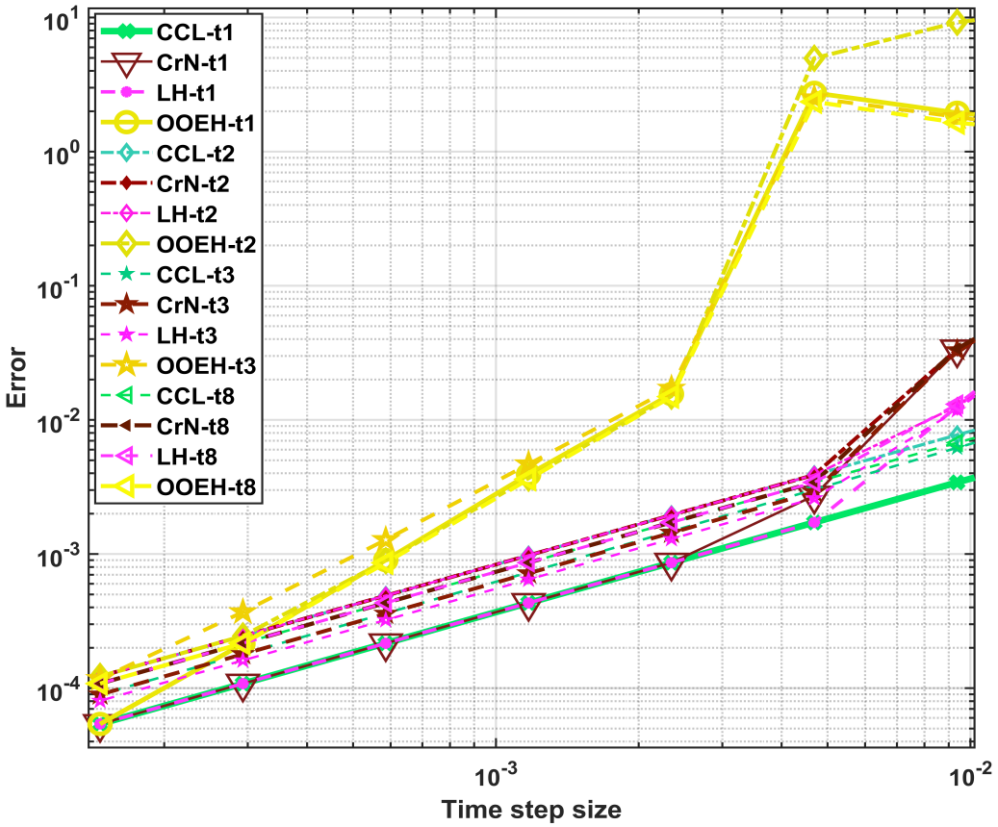


Figure 5.7. Maximum errors as a function of time step size for different operator-splitting.

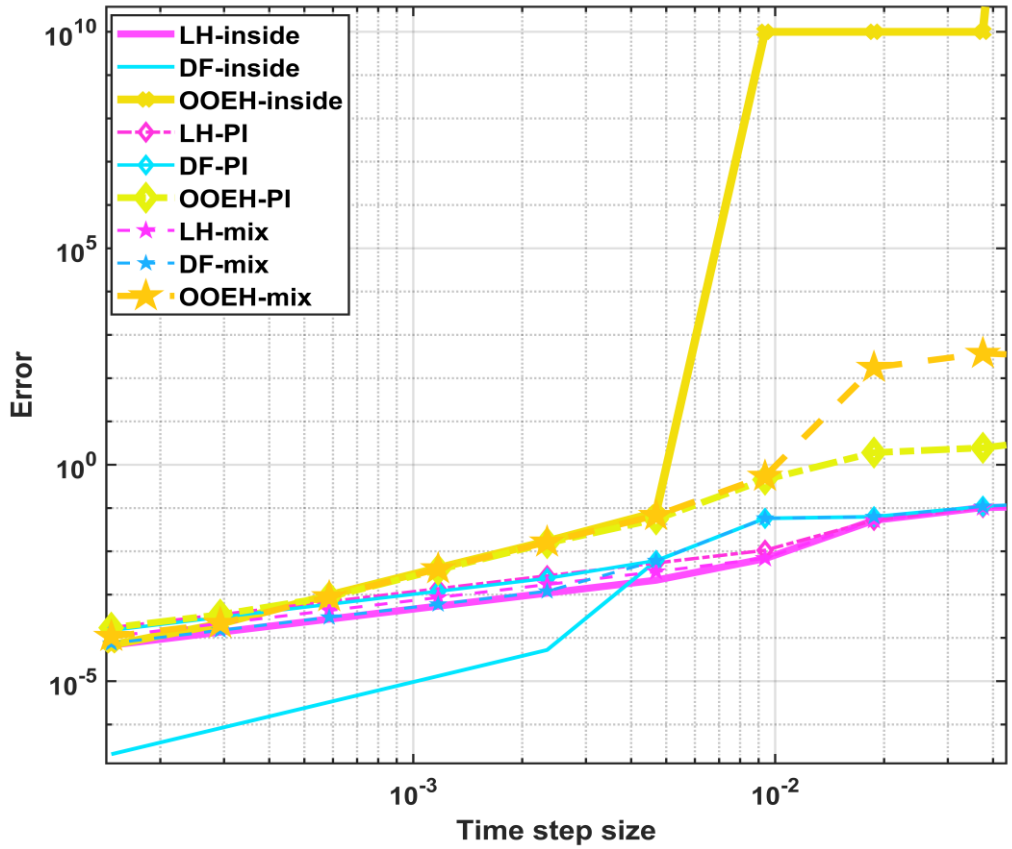


Figure 5.8. Maximum errors as a function of time step size for the remaining methods.

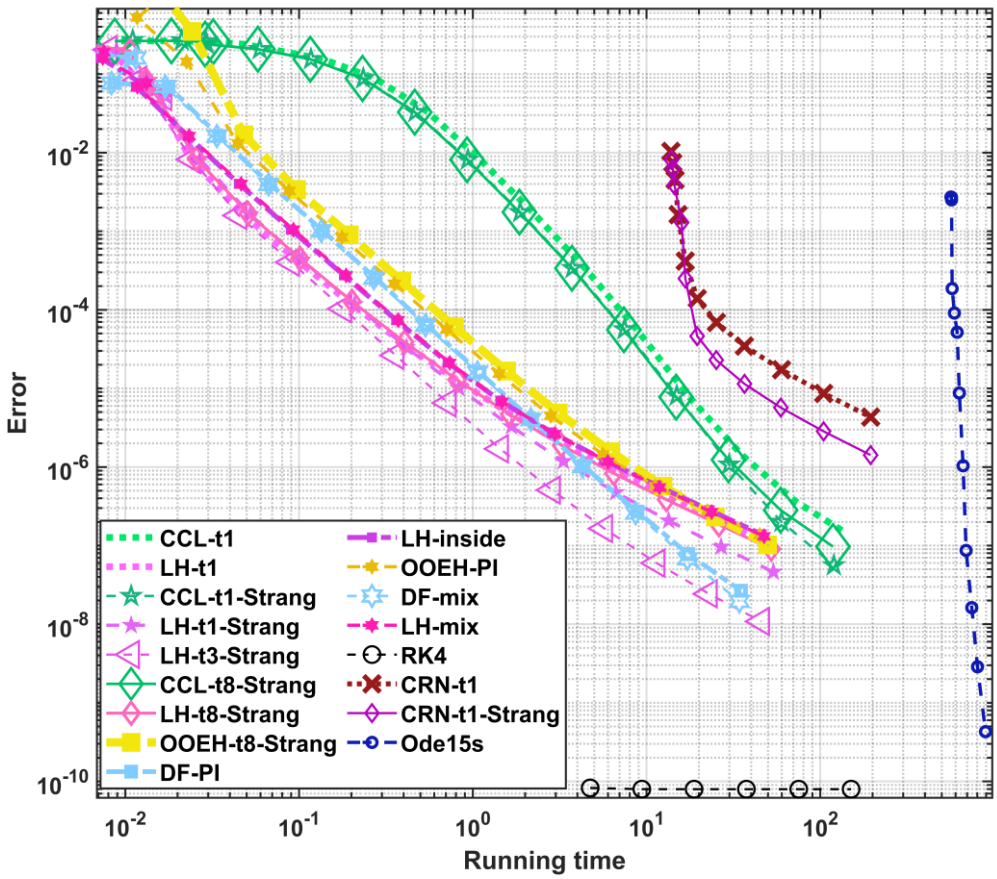


Figure 5.9. Maximum errors as a function of the running time moderately stiff system.

Moderately stiff system fig-5.9, while the exponents are $a_C = a_{Rx} = a_{Ry} = 0, b_C = b_{Rx} = b_{Ry} = 2$ the system size and the final time used are $N_x = 101, N_y = 120, t^{\text{fin}} = 0.1$, and $\beta = 21$, These variables give $SR = 8 \cdot 10^5$ and $h_{MAX}^{FTCS} = 7.73 \cdot 10^{-5}$. This indicates that the system is moderately stiff only. The initial concentrations contain random numbers in the unit interval as follows:

$$u_i^0 = 0.3 cos\left(\frac{x}{N_x}\frac{\pi}{2}\right) cos\left(\frac{y}{N_y}\frac{\pi}{2}\right) + 0.6 rand \cdot$$

5.2.4. Summary of this chapter

I performed extensive numerical tests to solve Huxley's equation. The goal was to explore the performance of some algorithms about which unconditional stability is analytically proven for the linear diffusion equation, thus excellent stability properties for the nonlinear case were expected. For some, but not all of the methods, these expectations are fulfilled well. According to the numerical tests, the LH with the t1 treatment and Strang-splitting is generally the most efficient and reliable among the examined methods. It provides relatively accurate results quite quickly for all the examined sets of parameters, and no signs of instability have been detected even for extremely stiff systems of strong nonlinearity. The LH-t3-Strang combination can also give very accurate results, but in some cases, for example, if the simulated time interval is long, it is not so reliable. However, if the geometry is complicated and the odd-even division of the cells faces difficulties, the CCL-t1-Strang or the DF-mixed methods can be used instead of the LH methods, since they are also reliable.

The original odd-even hopscotch method can be proposed only for an equidistant mesh with a constant diffusion parameter. In these cases, this method with the PI treatment is much simpler to code than the LH scheme and its accuracy is roughly the same.

The Crank-Nicolson method with operator-splitting has no advantage against the LH method. Its execution time strongly increases with the system size, so we can advise using it only when the number of nodes is small. However, in that case, the CrN method without operator-splitting, i.e. with standard Newton iterations, would be more accurate.

6. SOLUTION OF COUPLED SYSTEM OF TWO REACTION-DIFFUSION EQUATIONS

In this chapter I deal with an equation system consist of two diffusion-reaction equations, coupled by explicitly time-dependent reaction terms. I collect eight efficient explicit numerical schemes which are unconditionally stable if the reaction terms are missing, and apply them to the system of equations. I will use nontrivial analytical solutions, containing the Kummer functions as reference solutions. I present that they severely outperform the standard explicit methods when low or medium accuracy is required. I thoroughly investigate how the performance of the methods depend on the coefficients and parameters such as the examined time interval by parameter-sweeps. I obtained that, similarly to the single-equation case, the leapfrog-hopscotch method is usually the most efficient to solve these problems.

6.1. Coupled Diffusion Equations and their solution

In Eq. (19) in the work [46] of Barna and Mátyás, the following system of PDEs were studied:

$$\begin{aligned}\frac{\partial u(x,t)}{\partial t} &= D\frac{\partial^2 u(x,t)}{\partial x^2} + F(u(x,t),w(x,t)),\\ \frac{\partial w(x,t)}{\partial t} &= D\frac{\partial^2 w(x,t)}{\partial x^2} + G(u(x,t),w(x,t)),\end{aligned} \tag{6.1}$$

where F and G are general functions. The special case I study will be the following:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\frac{\partial^2 u}{\partial x^2} + a\cdot w\cdot t^{A},\\ \frac{\partial w}{\partial t} &= D\frac{\partial^2 w}{\partial x^2} + b\cdot u\cdot t^{-A-2}.\end{aligned} \tag{6.2}$$

where A is an arbitrary real parameter. According to [46], the analytical solution of Equation System (6.2) is the following:

$$u(x,t) = \frac{x}{t^{\alpha+1/2}}\cdot e^{-\frac{x^2}{4Dt}}\left(\begin{array}{l} c_1 M\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_2 M\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right]\\ +c_3 U\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_4 U\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right]\end{array}\right) \tag{6.3}$$

and

$$w(x,t) = \frac{x}{2a\cdot t^{A+\alpha+3/2}}\cdot e^{-\frac{x^2}{4Dt}}\left\{\begin{array}{l}\left(c_1 M\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_3 U\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right]\right)(\gamma-\alpha+2q)\\ +\left(c_2 M\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_4 U\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right]\right)(\gamma-\alpha-2q)\end{array}\right\}. \tag{6.4}$$

where α and γ are independent real parameters and the abbreviations $p = 1-\frac{\gamma}{2}-\frac{\alpha}{2}$ and $q = \sqrt{4ab-2\alpha\gamma+\gamma^2+\alpha^2}/2$ were introduced. It is clear that if $a\cdot b \geq 0$ then q is real regardless of the values of α and γ.

6.2. Simple coupling Diffusion for Large nonlinear coefficient

The L2 error is determined by taking the square root of the sum of the squared differences between the numerical solution and the analytical (reference) solution at each point:

$$L2_u = \left(\sum_{i=1}^{N} (u_i^n - u^{ref}(x_1, t_n))^2 \right)^{1/2}, \qquad (6.5)$$

$$L2_w = \left(\sum_{i=1}^{N} (w_i^n - w^{ref}(x_1, t_n))^2 \right)^{1/2}.$$

In the case of simple coupling, the analytical solution given by Equations (6.3)–(6.4) is evaluated using a completely different set of parameter values. I always use the same coefficients $c_1 = 1,\ c_2 = 0,\ c_3 = 0,\ c_4 = 0$ in each case study. I used the analytical solution (6.3) and (5.4) with the following parameters: $x_0 = -2,\ L_x = 4,\ N_x = 1000,\ t^{\text{in}} = 0.04,\ \text{TIME} = 0.1,\ a = 4,\ b = 9$ and $A = -0.6,\ \alpha = 0.87$. First I present the graphs of the initial function $u(x, t = 0.04)$, the reference analytical solution $u^{exact}(x, t_{fin} = 0.14)$, and two corresponding numerical solutions for $h = 10^{-4}$ in Figure 6.1.

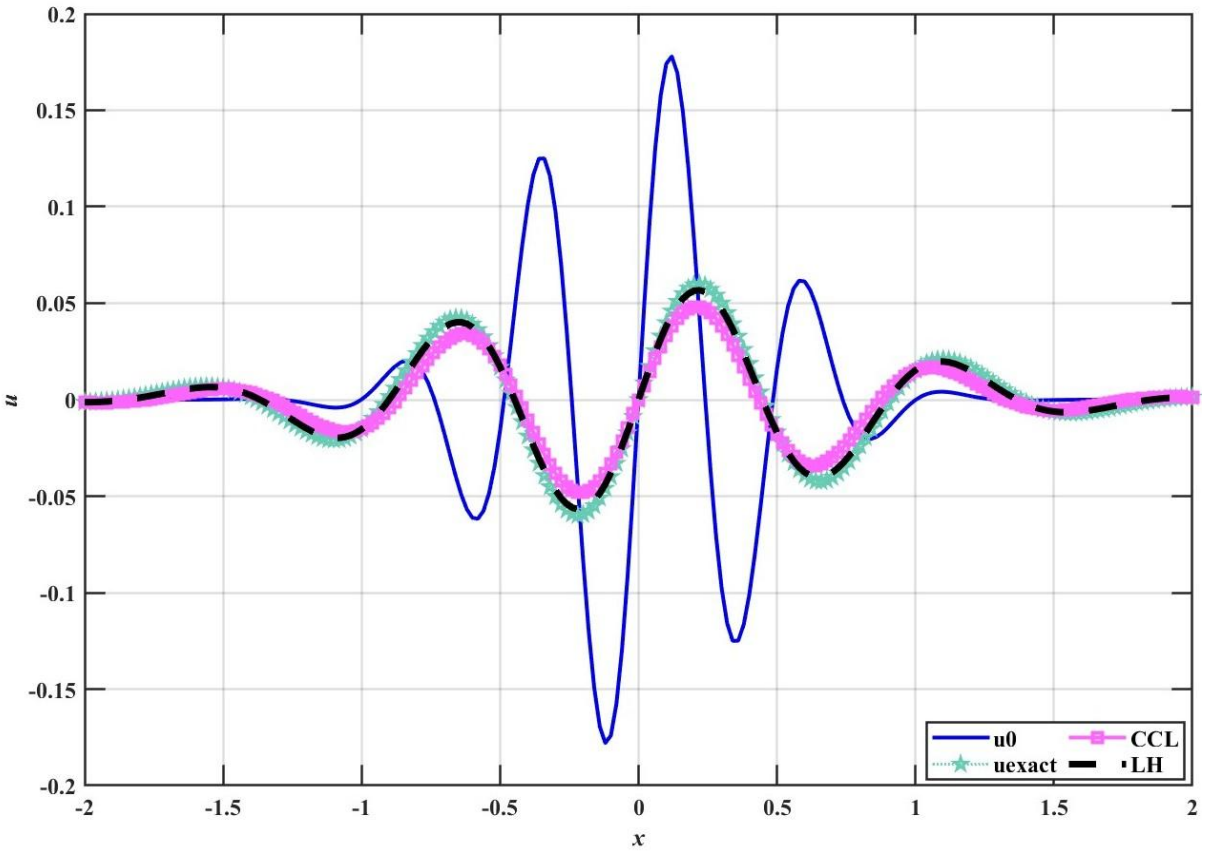


Figure 6.1. The concentration of u in term of x in the case of the initial function u^0, for u^{exact} analytical solution, CLL and LH are the numerical methods.

Then, in Fig. 6.2, I display the errors calculated by Formula (6.5) as a function of the time step size for both u and w for each method. I consider the verification successful, since for decreasing Δt, the errors are decreasing as expected based on our similar experiments conducted for single diffusion-type equations. I note that the FTCS and the AB2 methods produce meaningful solutions only below $\Delta t = 10^{-5}$ (bottom left corner of the figure).

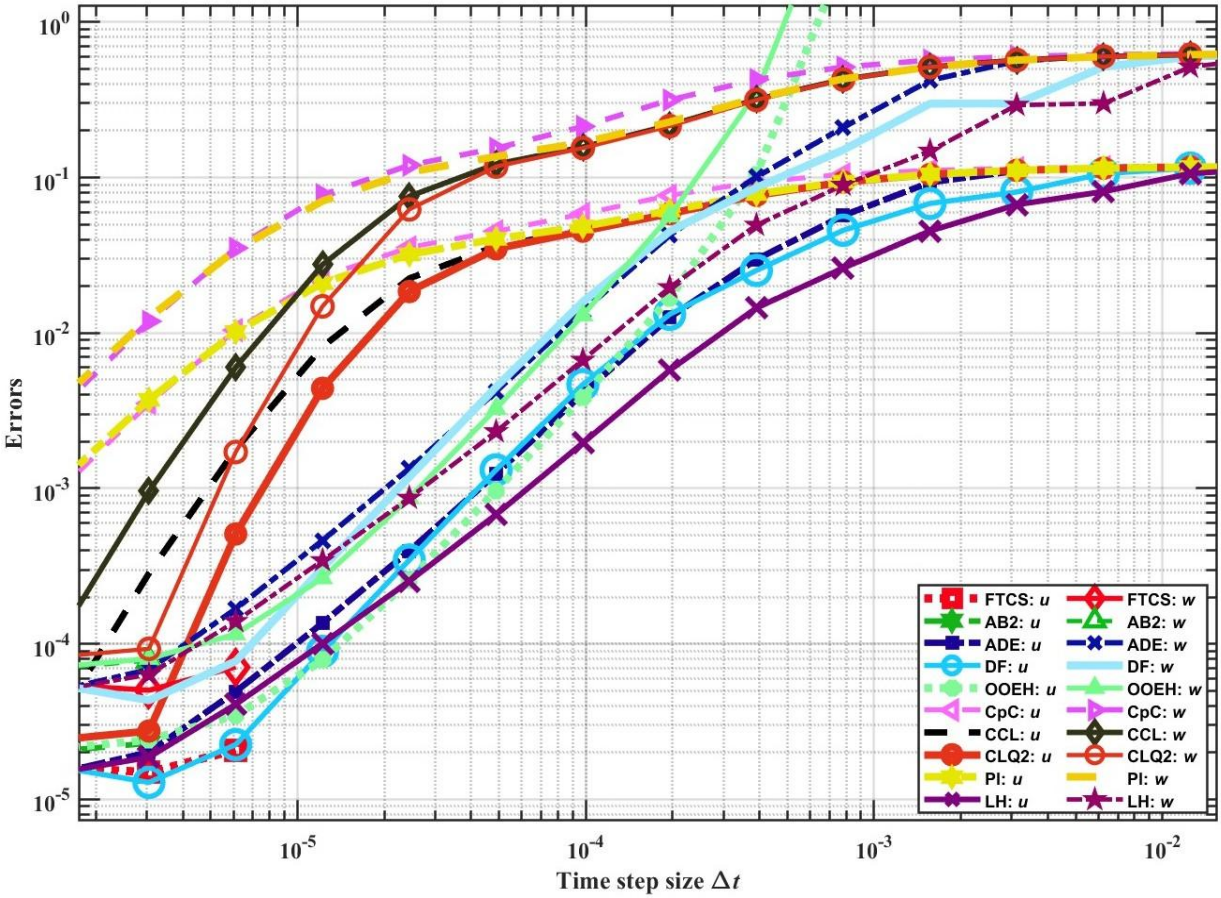


Figure 5.2. L_2 errors as a function of the time step size Δt the simple coupling case.

6.3. Parameter-sweeps for further testing of the methods

In this section I conduct parameter-sweep experiments to explore how the performance of the methods depend on the parameters. In the case of simple coupling the analytical solutions (6.3)-(6.4) will be used for reference purposes with the $c_1 = 1,\ c_2 = 0,\ c_3 = 0,\ c_4 = 0$ choice in each simulation. Since the FTCS and AB2 methods produce "*NaN*" (not-a-number) values above their CFL limit, the aggregated errors cannot be calculated for them, therefore I proceed only with the remaining eight methods with good stability properties. Errors as a function of parameter a for the simple coupling case. The following series is used for the values of the coefficient: $a \in \{0.1, 0.3, 1, 3, 6, 10, 20, 20, 100, 300\}$. The coefficients have the following values: $b = 2a,\ A = -0.9,\ \alpha = 0.6$, which means that the two coupling coefficients grow together. The PDE (6.1) is discretized using $x_0 = 1,\ L_x = 0.8,\ N_x = 100,\ t^{\text{in}} = 0.1,\ \text{TIME} = 0.1,\ t^{fin} = t^{\text{in}} + \text{TIME}.$

The error is calculated for S=9 different time step sizes for all the examined methods. The aggregated errors are displayed in Table 6.1 and Fig.6.3.

Table 6.1. The aggregated errors *(AgErr)* for simple coupling of the methods.

Numerical methods	0.1	0.3	1	3	6	10	20	30	100	300
ADE*u*	-20.45	-18.70	-12.14	-8.56	-2.21	4.33	24.69	38.61	166.61	420.26
ADE*w*	-16.91	-16.05	-10.02	-6.51	-0.16	6.36	26.69	40.61	168.60	422.25
DF*u*	-20.29	-16.63	-13.09	-11.74	-5.26	-0.79	13.52	23.05	87.91	215.24
DF*w*	-16.92	-14.68	-10.84	-9.73	-3.22	1.22	15.56	25.08	89.93	217.23
OOEH*u*	-18.71	-17.06	-10.68	-4.49	6.10	18.11	50.65	75.71	267.83	676.56
OOEH*w*	-15.08	-14.04	-8.44	-2.19	8.58	20.86	53.98	79.51	273.61	684.90
CpC*u*	-9.40	-6.12	-4.96	-2.95	-1.26	-1.36	0.29	3.41	33.03	123.56
CpC*w*	-6.54	-3.79	-2.69	-0.84	0.80	0.69	2.31	5.44	35.03	125.56
CCL*u*	-10.27	-7.05	-6.38	-4.30	-1.89	-1.14	3.81	9.31	60.57	208.27
CCL*w*	-7.42	-4.71	-4.17	-2.22	0.16	0.90	5.82	11.33	62.57	210.27
CLQ2*u*	-10.46	-7.23	-6.61	-4.51	-1.96	-1.05	4.33	10.23	65.33	228.88
CLQ2*w*	-7.61	-4.90	-4.41	-2.43	0.09	0.98	6.33	12.25	67.33	230.88
PI*u*	-9.80	-6.58	-5.78	-3.67	-1.68	-1.38	1.97	6.32	47.37	165.17
PI*w*	-6.95	-4.24	-3.57	-1.57	0.37	0.66	3.98	8.34	49.37	167.16
LH*u*	-23.69	-19.97	-17.01	-15.08	-6.31	-0.86	17.25	29.56	113.61	278.26
LH*w*	-20.43	-18.05	-14.78	-12.98	-4.29	1.14	19.28	31.58	115.61	280.25

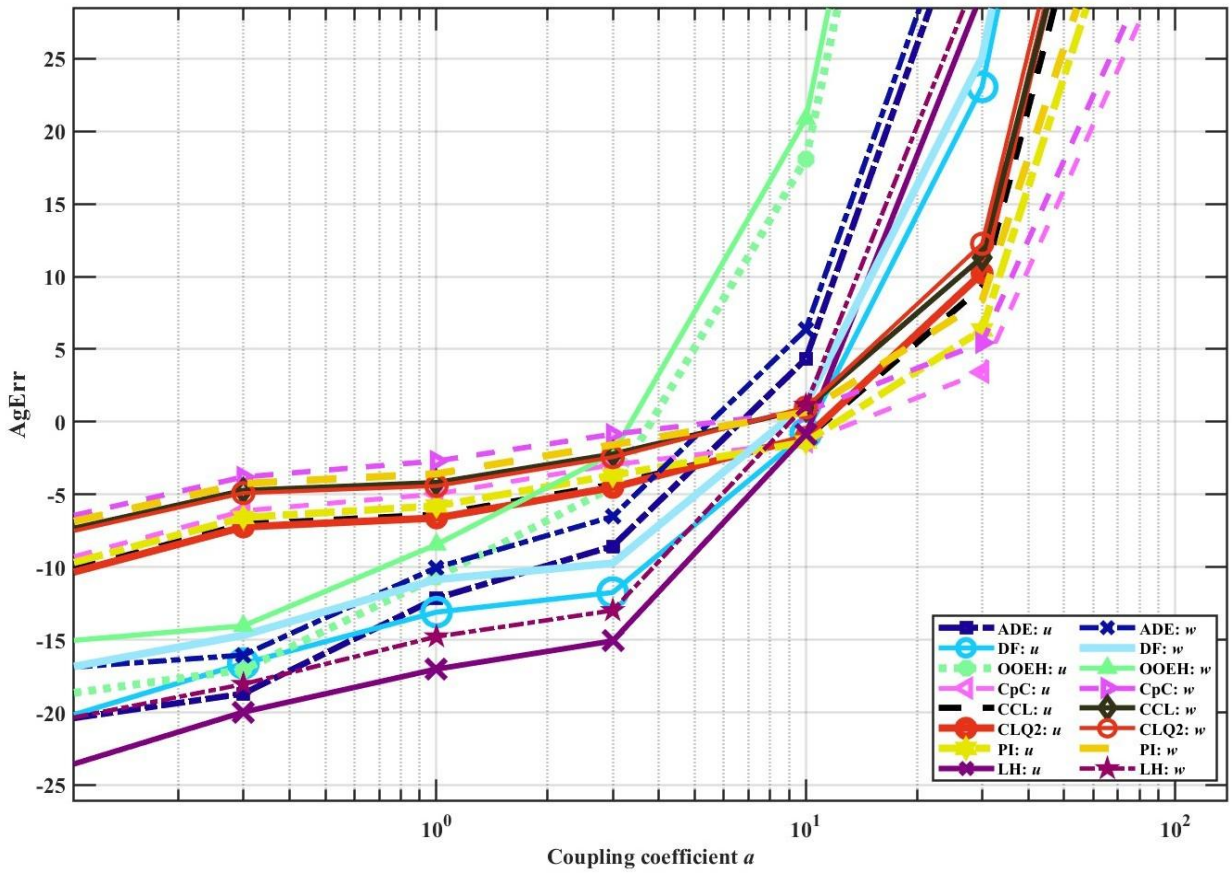


Figure 6.3. Maximum errors as a function of the number of a for the remaining methods .

6.4. Errors as a function of the b parameter, simple coupling case

Table 6.2. The aggregated errors *(AgErr)* for simple coupling case.

Numerical methods	0.1	0.3	1	3	10	30	100	300
ADE*u*	-16.03	-12.90	-9.03	-8.85	-5.46	-3.17	4.56	12.58
ADE*w*	-21.48	-16.69	-10.41	-8.08	-2.41	2.00	12.07	22.22
DF*u*	-12.74	-9.49	-12.40	-8.31	-8.02	-6.01	-1.46	5.60
DF*w*	-18.70	-13.19	-13.79	-7.64	-4.96	-0.83	6.06	15.30
OOEH*u*	-14.57	-11.11	-7.42	-6.09	-1.32	5.14	21.09	42.46
OOEH*w*	-19.50	-14.86	-8.60	-4.92	2.02	10.82	29.48	53.50
CpC*u*	-2.93	-0.99	-2.88	-1.18	-3.12	-4.28	-5.87	-7.54
CpC*w*	-8.62	-4.53	-4.21	-0.44	-0.02	0.92	1.67	2.09
CCL*u*	-3.89	-2.15	-3.82	-1.92	-3.74	-4.51	-4.72	-4.03
CCL*w*	-9.59	-5.68	-5.20	-1.19	-0.65	0.68	2.80	5.59
CLQ2*u*	-4.08	-2.35	-4.01	-2.08	-3.87	-4.53	-4.52	-3.52
CLQ2*w*	-9.77	-5.88	-5.40	-1.35	-0.78	0.66	3.01	6.10
PI*u*	-3.42	-1.64	-3.36	-1.53	-3.51	-4.56	-5.53	-5.84
PI*w*	-9.11	-5.17	-4.73	-0.80	-0.42	0.64	2.00	3.78
LH*u*	-16.09	-12.95	-16.06	-10.53	-9.53	-6.72	-0.28	9.53
LH*w*	-22.09	-16.70	-17.45	-9.87	-6.48	-1.55	7.23	19.21

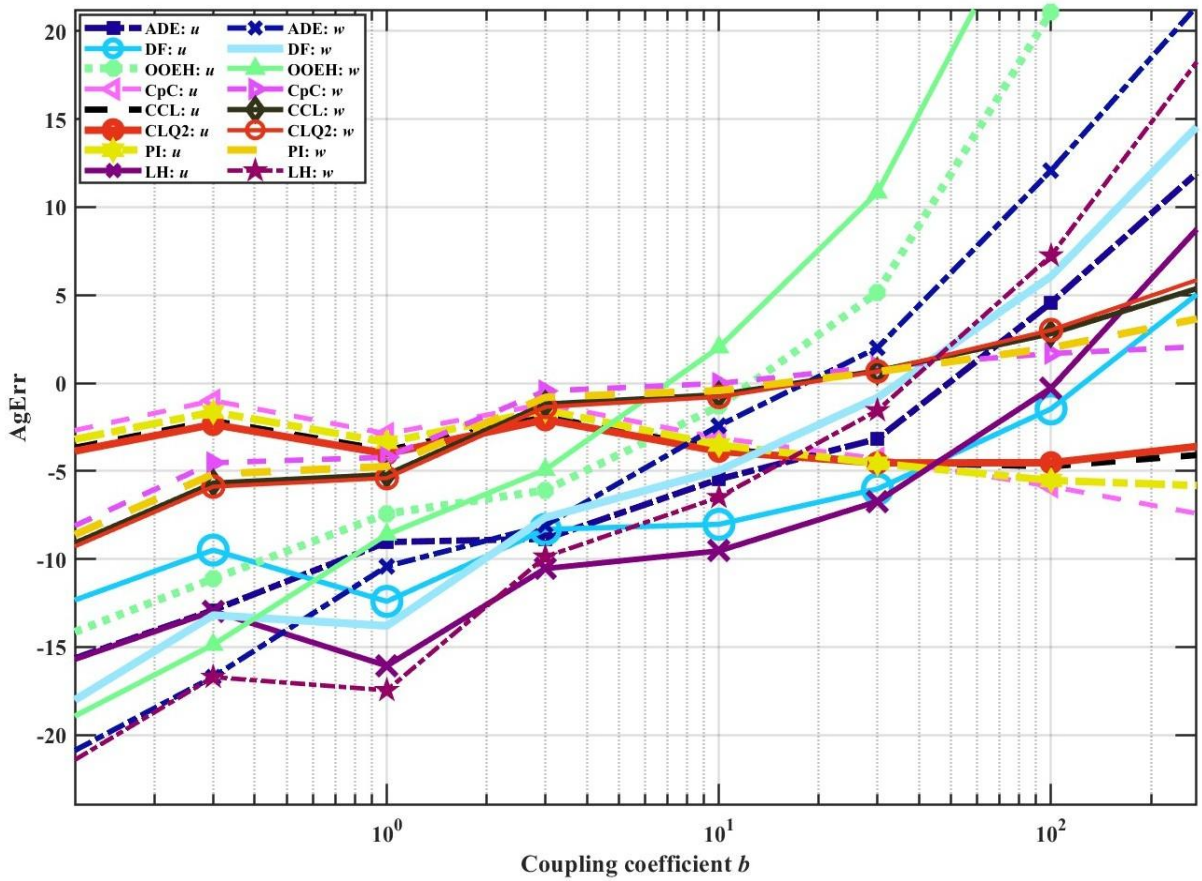


Figure 6.4. Maximum errors as a function of the number of *b* for the remaining methods in the case.

Errors as a function of the *b* parameter Table 6.1 and fig 6.4. The following series is used for the values of the coefficient: $b \in \{0.1, 0.3, 1, 3, 10, 30, 100, 300, \}$. The used coefficients are $a = 3,\ A = -0.9$ and $\alpha = 0.6$, so the first coupling coefficient is kept fixed and only the second is increased gradually. I calculated the error for *S*=9 different time step sizes for all the examined methods.

6.5. Summary of this chapter

I investigated coupled sets of reaction-diffusion equations in one space dimension with two distinct, time-dependent reaction terms. I then reproduced the analytical solution found in the literature using ten explicit numerical methods, including the widely used FTCS and Adams–Bashforth schemes. The remaining 8 schemes are unconditionally stable for the diffusion equation without the reaction term. First, I validated the methods by demonstrating that the errors properly decrease with the time step size for both types of reaction terms. Then I ran sweeps on two critical parameters to see how the approaches behaved as a function of these variables. The first result is that the methods are not unconditionally stable for the diffusion-reaction equation system under study. If the reaction term's coefficients are big, the error increase over time. However, approaches that are stable for the diffusion equation remain more stable when a response term is present than standard FTCS and AB2 schemes. The OOEH approach is the least stable of these methods, while the CpC method is the most stable and has superior qualitative qualities. Nonetheless, the LH approach typically has the minimal error, although the DF, OOEH, and ADE schemes work well as long as they are not near their stability limits.

7. IMPACT OF CLIMATE VARIABILITY ON OPTIMAL THICKNESS OF WALL INSULATION: A NUMERICAL STUDY ACROSS FIVE COLD CITIES USING LONG-TERM NUMERICAL SIMULATIONS

The section is structured as follows: The first subsection outlines the equations and numerical method used in the simulation. The second one details the simulation setup, including wall materials, geometry, initial and boundary conditions. The thrid one presents the results, including heat transfer analysis for different cities. Finally, I summarize the key results and conclusions.

7.1. The equations and leapfrog-hopscotch method

The terms for radiation, convection, and the heat source are added to the heat conduction equation (2.1) as follows:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + q - K \cdot u - \sigma \cdot u^4 \ , \tag{7.1}$$

where the terms K and σ are present only at the boundary of the solid material. The convective transfer of heat between a moving fluid and a surface is referred to as $K(u_a - u)$ in accordance with Newton's law of cooling. The fluid, which may be either a liquid or a gas and is propelled by variations in temperature and density, moves during this process. The fluid velocity, the temperature differential between the fluid and the surface, and the fluid's physical characteristics, including its viscosity and thermal conductivity, all affect the rate of heat transfer [99].

In this case, the ambient temperature (in Kelvin), u_a , is unaffected by the unknown variable u. Because of this, the term Ku_a should be added to the expression q, which represents the heat source. According to the Stefan-Boltzmann equation [98], where the proportionality constant σ is the product of the positive Stefan-Boltzmann constant and the surface area, the radiative heat loss from a surface may be expressed by the term $-\sigma u^4$. Similar to the previously stated Ku_a term, the incoming radiation, which includes direct sunlight, will be included into the source term q. The following is the one-dimensional discretised version of the heat transfer equation (6.1) that I derive using this approximation method:

$$\frac{du_i}{dt} = \alpha \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + q - Ku_i - \sigma {u_i}^4 \tag{7.2}$$

The following formula is obtained for the time derivative of each cell variable using the new quantities:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i-1,i} C_i} + \frac{u_{i+1} - u_i}{R_{i+1,i} C_i} + q - Ku_i - \sigma {u_i}^4 \tag{7.3}$$

where K and σ are nonzero only for the computational cells on the surface of the wall.

The leapfrog-hopscotch (LH) formulas for Eq. (7.3) are the following: The first stage has the length of a halved time step, thus we have the following formulas

$$u_i^{1/2} = \frac{u_i^0 + r_i / 2(u_{i-1}^0 + u_{i+1}^0) - K\Delta t u_i^0 / 4}{1 + r_i + K\Delta t / 4 + \sigma\Delta t (u_i^0)^3 / 2} \tag{7.4}$$

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

$$u_i^1 = \frac{u_i^0 + r_i (u_{i-1}^{1/2} + u_{i+1}^{1/2}) - K\Delta t u_i^0 / 2}{1 + r_i + K\Delta t / 2 + \sigma\Delta t (u_i^0)^3}, \tag{7.5}$$

7.2. Numerical simulation of the wall

All transient simulations were performed using a constant time step of Δt =50 s. The reason is that in our group previous work [100], where the used LH method was verified, it was demonstrated that this time step size yield errors below 0.01°C, and this accuracy is more than enough for our purpose. It is also shown there that the explicit Euler or Runge-Kutta CFL limit is significantly smaller than this time step size, and this is the main advantage of the LH method. The simulated time interval is from 15 of October to 15 of April, with duration T=184×24×3600=15894000 s, and it represents the whole heating season in most cities. It follows that there are always NT=T/Δt=317880 time steps in total.

The weather data was obtained hourly from the website over the whole 2023–2024 winter season [101]. After then, linear interpolation was used to compute the data every 50 s. The studied cities Warsaw, Hokkaido, Edmonton, Ulaanbaatar, and Copenhagen are characterized by continental climates (Köppen D-type). Warsaw, Hokkaido, Edmonton, and Copenhagen have a humid continental climate with warm summers (Dfb), whereas Ulaanbaatar exhibits a subarctic climate with dry winters (Dwc). Key meteorological conditions in mid-January, including average outdoor temperature, solar radiation, and wind speed, strongly influence building heat loss and insulation performance. Table 7.1 summarizes these parameters, which were used as boundary conditions in the transient simulations. Understanding these climatic factors helps optimize insulation thickness and energy savings in cold climates.

Table 7.1. Mid-January Climatic Conditions and Köppen Classification of the Studied Cities.

City	Köppen Climate	Avg. Temp (°C)	Solar Radiation (W/m²)	Wind Speed (m/s)
Copenhagen	Dfb	0.0	90	4.0
Edmonton	Dfb	-10.0	120	5.0
Hokkaido	Dfb	-6.5	130	4.2
Ulaanbaatar	Dwc	-20.1	150	5.5
Warsaw	Dfb	-3.2	110	3.5

The studied wall models are seen in Fig. 7.2. First, two layers of panel and gypsum plaster were used, as shown in Fig. 7.2A. The dimension 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 glass wool with several thicknesses with a range of [0.02 m to 0.36 m], as illustrated in Fig. 7.2B.

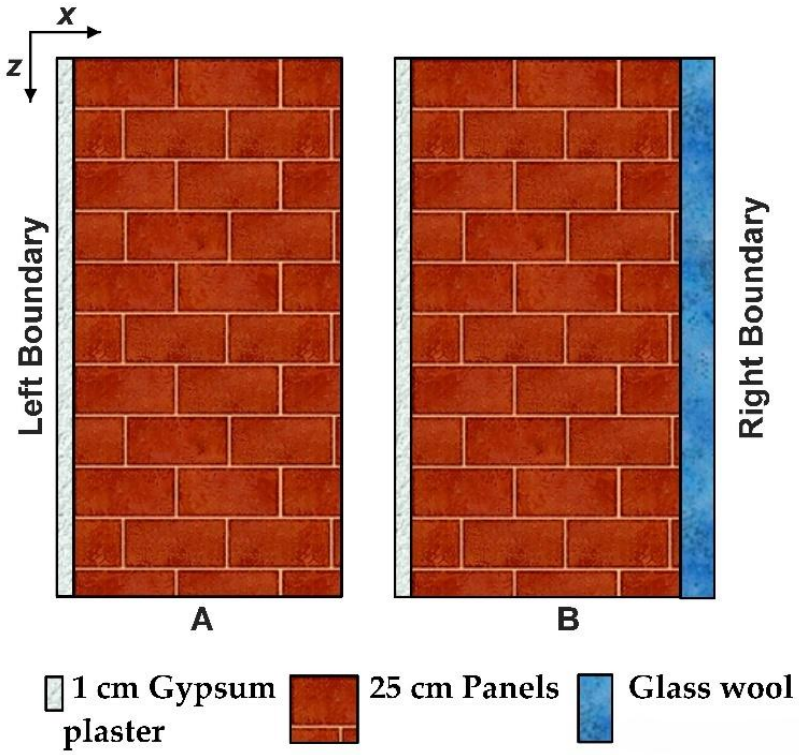


Figure 7.2. Shows the wall A, and B, with different thicknesses and insulation materials.

In this investigation, the true material properties listed in Table 7.2 are taken into account. Although these coefficients have a discontinuity at the material's boundary, they are constants inside the material and do not vary with temperature, time, or space.

Table 7.2. Properties of the materials used in the simulation [102].

Material	$c\ \left[\mathrm{Jkg^{-1}K^{-1}}\right]$	$\rho\left[\mathrm{kgm^{-3}}\right]$	$k\ \left[\mathrm{Wm^{-1}K^{-1}}\right]$
Gypsum plaster	977	805	0.29
Panels	800	1600	0.743
Glass wool	700	120	0.039

The simulations were carried out for the exterior wall of buildings in five cities, as depicted in Fig. 7.2, using an insulator made of glass wool with different thicknesses [103], with the overall heat transfer coefficient determined by the HAP as indicated in Table 7.3. These simulations were then compared to the wall with two layers without insulation. The convection coefficient, solar radiation, and outside air temperature values for these cities were utilised in the simulations.

Table 7.3. Components and heat transfer coefficient of enclosure structure of the building.

Name	Components of the wall	The insulation	The thickness	Total Resistance *R* [m²K/W]	Heat transfer coefficient, *U* $\left[\mathrm{Wm^{-2}K^{-1}}\right]$
External wall	1cm Gypsum plaster + 25cm Panel	Without insulation	-	0.54096	1.85
		Glass wool	10 cm	3.103	0.322
			15 cm	4.378	0.228
			20 cm	5.653	0.176

7.3. Initial and Boundary Conditions

For all wall situations and all boundaries in the simulations, zero Neumann boundary conditions are applied to prevent the passage of conductive heat. To do this, we set the matrix components describing heat conduction over the border to zero. Nonetheless, it has been assumed that the left and right sides of the wall experience convective and radiative heat transmission in the *x* direction, respectively, as shown in Table 7.4. The inner components cannot lose or absorb heat by convection or radiation and they lack a heat source, hence the coefficients *K*, σ and *q* are zero in Eq. (7.1) for the interior cells. Regarding radiation, the Stefan–Boltzmann constant is a universal number $5.67\cdot10^{-8}\ \mathrm{W/m^2\cdot K^4}$. The surface is not an ideal black body; thus, we multiplied the Stefan–Boltzmann constant by the appropriate emissivity constant to obtain realistic values for σ^*.

Table 7.4. For both types of walls, the convection, radiation, and heat source parameters on both sides of the wall.

	h_c $\left[\frac{\mathrm{W}}{\mathrm{m^2K}}\right]$	σ^* $\left[\frac{\mathrm{W}}{\mathrm{m^2K^4}}\times10^{-8}\right]$	ε
Left Elements (inside)	8	3.97	0.7
Right Elements (outside)	0.6 – 25	4.82	0.85

I use the following formulae to determine the coefficient values for our equations:

$$K=\frac{h_c}{c\rho\Delta x},\ \sigma=\frac{\sigma^*}{c\rho\Delta x},\ q=\frac{q^*}{c\rho\Delta x}+\frac{h_c}{c\rho\cdot\Delta x}\cdot u_{\mathrm{a}}, \tag{7.6}$$

Also, I assumed that the right and left components have the following heat sources:

For the inside components: $q_l=\frac{1}{c\rho}\times q_l^*+\frac{h_{cl}}{c\rho\cdot\Delta x}\times 295\ \mathrm{K}$

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

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

Where $q_l^* = \varepsilon_l \sigma_l (295)^4$ and $q_r^*(t) = \alpha_{\text{sun}} G_{cr}(t) + \alpha_{\text{Low}} \varepsilon_r \sigma_r \left[u_r(t)\right]^4$ [104].

The following formula is used to determine the convection heat transfer coefficient for external components as a function of air velocity :

$$h_{cr}(t) = 0.6 + 6.64\sqrt{v}(t)\ , \qquad (7.7)$$

$v(t)$: According to the website data [100], every 50 seconds, we took measurements of the air velocity [m/s] in October, November, December, January, February, March, and April, ignoring the relative orientation of the wind and the 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].

For every 50 seconds during the months of October, November, December, January, February, March, and April, the outdoor air temperature [°C] varies according on the weather [101].

u_l : The inside ambient temperature 22 °C ≈ 295 K on the left side.

$G_{cr}(t)$: According to website data [101], the sun radiation is measured every 50 seconds in October, November, December, January, February, March, and April [W/m^2].

α_{sun} : The absorption of solar radiation by the surface, calculated to be 0.95.

α_{Low} : The painted surface's absorptivity to low-temperature thermal radiation is 0.93.

We approximated the initial temperature within the wall using the assumption that a stationary heat flow with constant flux had formed between the designated boundary values of the internal and external air temperatures before the simulation period at midnight on the 15 of October. For the starting condition, it produces two linear functions of the *x* variable in the case of two layers:

For the part of gypsum plaster: $u(x, t=0) = (u_{\text{mid}} - u_l) \cdot x / L_g + u_l \qquad (7.8)$

For the part of panel:

$$u(x, t=0) = \left[(u_{\text{r,initial}} - u_{\text{mid}}) \cdot x / L_p\right] - \left[(u_{\text{r,initial}} - u_{\text{mid}}) \cdot L_g / L_p\right] + u_{\text{mid}} \qquad (7.9)$$

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

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

where $u_{\text{r,initial}} = 282$ K and the subscripts g and p stand for the gypsum and panel layers, respectively, L_g, for example, is the gypsum plaster's thickness. For the remaining cases, the assumption of stationary heat conduction with initial values at the boundaries refers that we have to use the two linear functions of the x variable for the initial condition:

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

For the part of panel:

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

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

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

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

For the part of insulation:

$$u\left(x, t=0\right) = \left[\left(u_{\text{r,initial}} - u_{\text{mid},2}\right) \cdot x / L_{ins}\right] - \left[\left(u_{\text{r,initial}} - u_{\text{mid},2}\right) \cdot \left(L_g + L_p\right) / L_{ins}\right] + u_{\text{mid},2} \qquad (7.10)$$

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

7.4. Result for the simulationof the outside the wall

The results are calculated for the south and north outside walls in the five cold cities. The east- and west-facing walls typically exhibit thermal behavior that falls between that of the north- and south-facing walls due to their moderate and time-dependent solar exposure. For this reason, simulations were focused on the two extreme cases of the north and south walls while east and west orientations were excluded from the analysis.

7.5. Calculation of Heat Loss through walls

The total energy requirement Q in kWh units is calculated from Eq. (7.11)

$$Q = \sum_{t=1}^{Time} \left(\Delta t Q_{\text{cell}}(t)\right), \qquad (7.11)$$

where $Q_{\text{cell}} = q^*_{\text{cell}} \cdot S$ and q^* is the current density of heat loss, and, using Fourier's law, we can calculate it as $q^*_{\text{cell}} = \frac{\left((u_2 - u_1)K_{\text{cond}}\right)}{\Delta x}$, i. e. the heat flux between the two computational cells on the inner surface of the wall.

7.6. Life-Cycle Economic Analysis

The life cycle technique, a thorough and practical approach to assessing economic performance, is what we have employed, where the inflation and interest rates are 2% and 4%, respectively.

By multiplying the current power price (per kWh) by the quantity of heat losses, we can now determine the cost of the energy consumed to compensate for the energy loss.

The following approximate equations are used in our calculations:

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

Where the cost of installation (e.g. wage of construction workers) 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 material (7.136 Euro) + the estimated cost of installation (18 Euro) = 25.136 Euro.

Total annual energy cost = price of kWh × total heat loss in the examined heating season.

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.

I note that I ignored heating or cooling costs outside of the heating season, as well as the cost of the maintenance in my calculations.

7.7. Comparison of the effect of changing the city for two layers south wall without insulation

Figure 7.3 illustrates the total heat loss (in W/m²) over time for a two-layer south-facing wall without insulation in five different cold cities. Ulaanbaatar (dashed black line) exhibits the highest average heat loss, peaking above 120 W/m², indicating a harsh winter climate with extreme temperature variations.

Overall, the figure highlights the impact of climate conditions on heat loss through an uninsulated south-facing wall. Cities with more extreme winter temperatures, such as

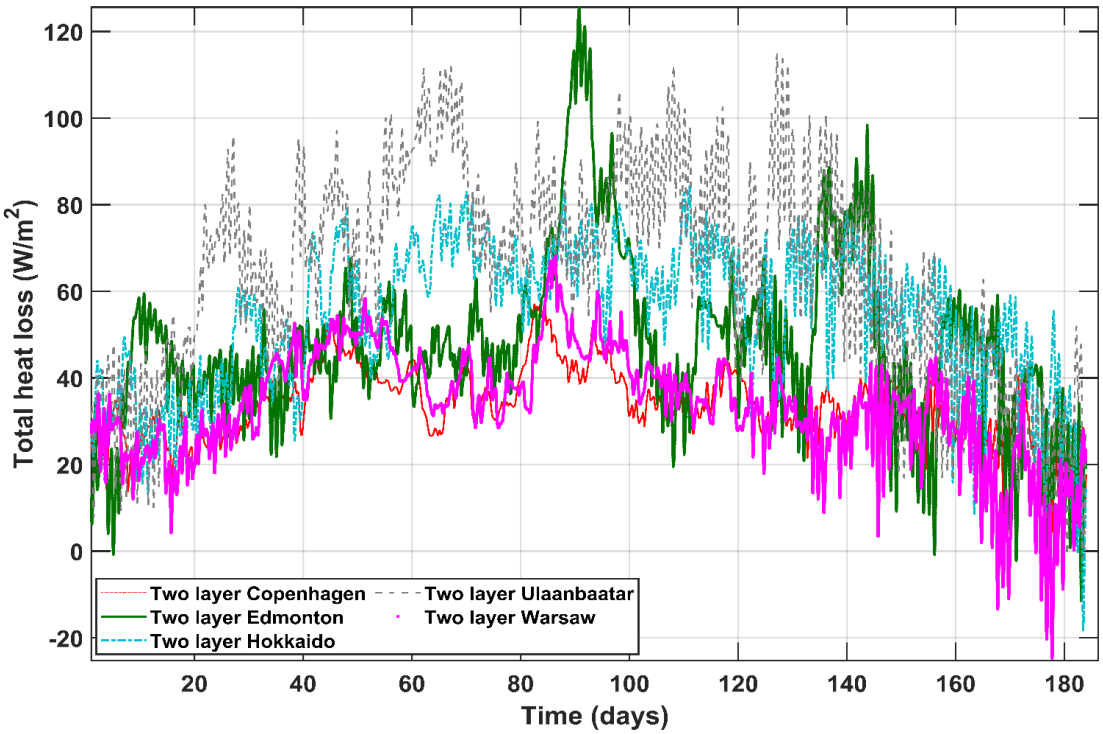


Figure 7.3. The heat loss in W units as a function of time in days for the simulation of the five cities for the south wall without insulation.

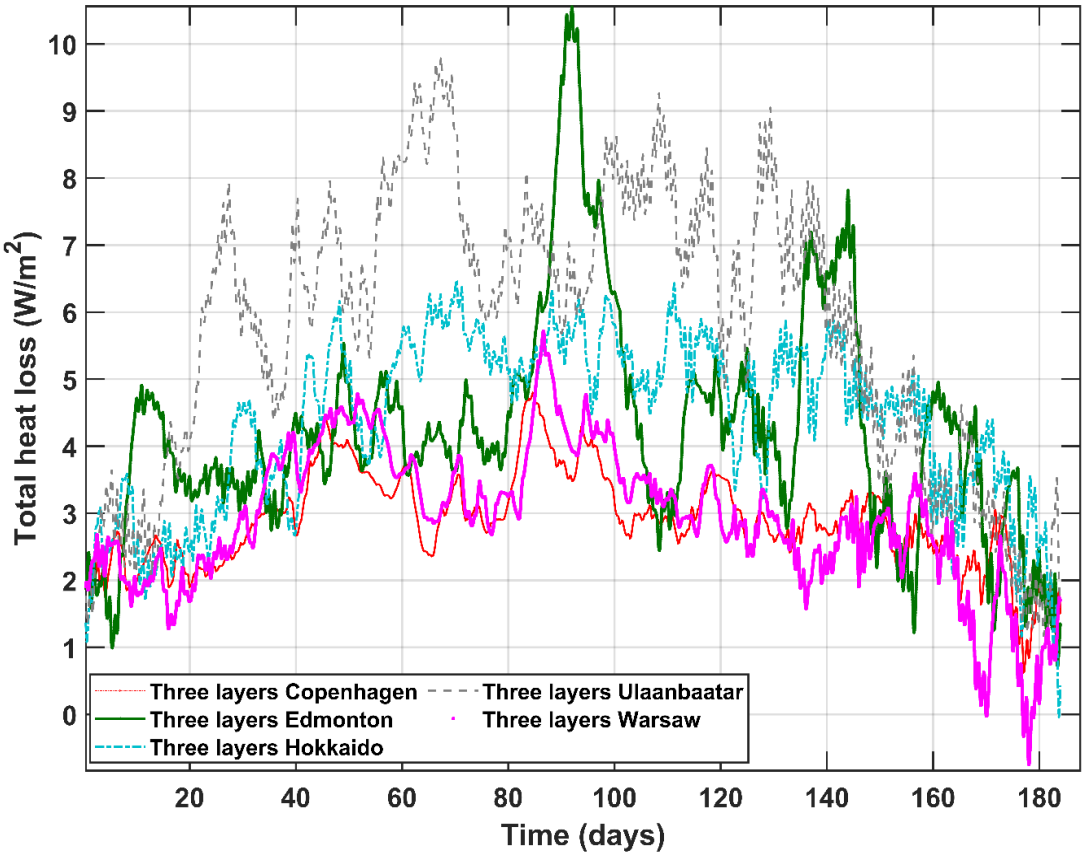


Figure 7.4. The heat loss in W/m2 units as a function of time in days at 0.2 m thickness of the insulator for the simulation of the five cities in the case of three layers wall.

Ulaanbaatar and Edmonton, experience higher and more fluctuating heat loss, while cities with milder winters, such as Warsaw and Copenhagen, exhibit lower values.

Figure 7.4 presents the total heat loss (in W/m²) through a three-layer wall with 0.2 m insulation thickness on the south wall, simulated for five cold cities. Heat loss varies over time, generally starting low, increasing mid-winter (more heating demand), and decreasing toward the end as temperatures rise again. One can see that climate greatly influences heat loss even in the same Köppen climate-category, and hence, insulation optimization should be climate- or location-specific. For cities like Ulaanbaatar and Hokkaido, a thicker insulation might be more effective in reducing the high heat losses observed.

7.8. Calcualtion of the optimial thicknesses

To determine the optimal insulation thickness, the economic calculations were based directly on the simulated heat loss values presented in Table 6.4. The heat loss (in kWh/m² per heating season) was converted to monetary cost using the current electricity price applied uniformly to all cases. Since the calculation uses purchased electricity to compensate for the heat loss, the effective system efficiency is $\eta = 1.0$. The energy price used in the study is the same value employed to compute the 'Total Annual Energy Cost' in Table 7.5.

The total initial cost consists of two components:

(1) the cost of insulation material, proportional to its thickness; and
(2) a fixed installation cost of 18 € per m², as described previously in the Life-Cycle Economic Analysis section.

These values are explicitly used in Table 7.5.

Annual energy savings were computed as the differ ence between the annual energy cost of the uninsulated wall (Wall A) and the insulated wall (Wall B). Payback time was then calculated using:

$$\text{Payback} = \frac{\text{Total Initial Cost}}{\text{Annual Savings}}.$$

As shown in Table 7.5, the combination of low installation cost and a relatively large reduction in heat loss leads to short payback periods for all cities, including Warsaw. For Warsaw, the optimized insulation thickness of 26 cm yields a U-value of 0.1387 W/m²K which satisfies the Polish thermal standard WT2021 requiring $U \leq 0.20$ W/m²K.

Figure 7.5 illustrates the heat loss as a function of insulation thickness of the south wall for the five cities. We can see all cities show a large decrease in heat loss as insulation thickness increases. Design strategies for energy efficiency must consider climate-specific requirements, as colder regions benefit more from thicker insulation. Adding insulation to the south wall is highly effective up to a point, especially in colder cities such as Ulaanbaatar.

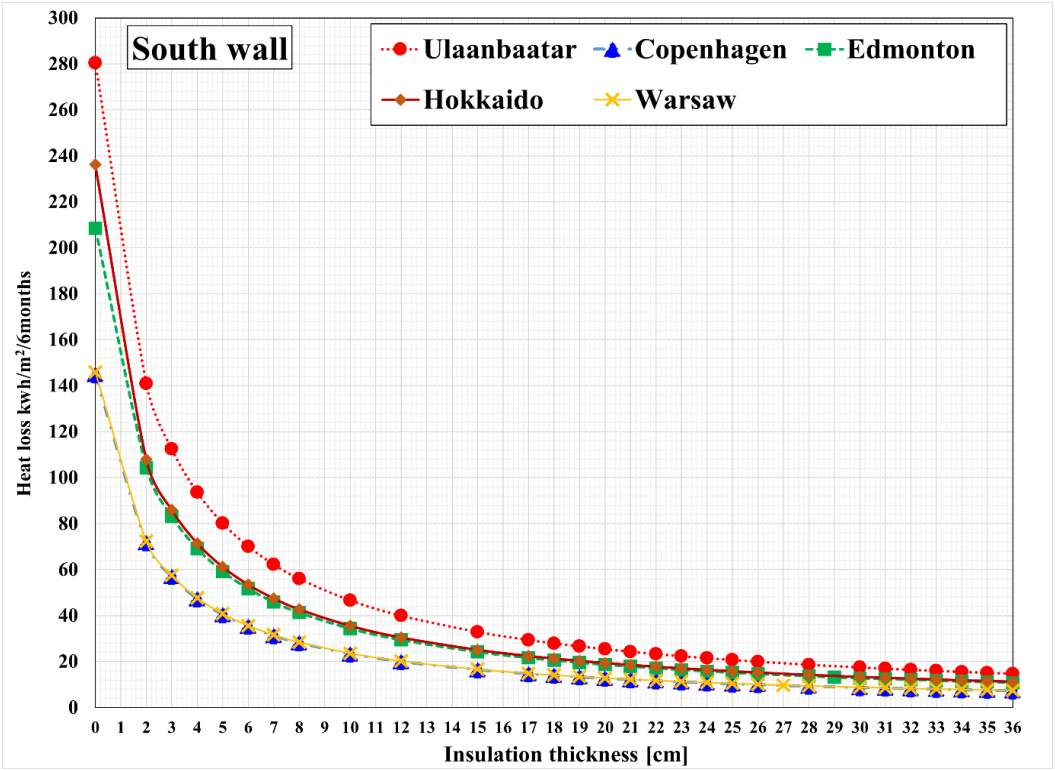


Figure 7.5. The heat loss in kWh/m^2/6 months as a function of insulation thickness of the south wall for the five cities.

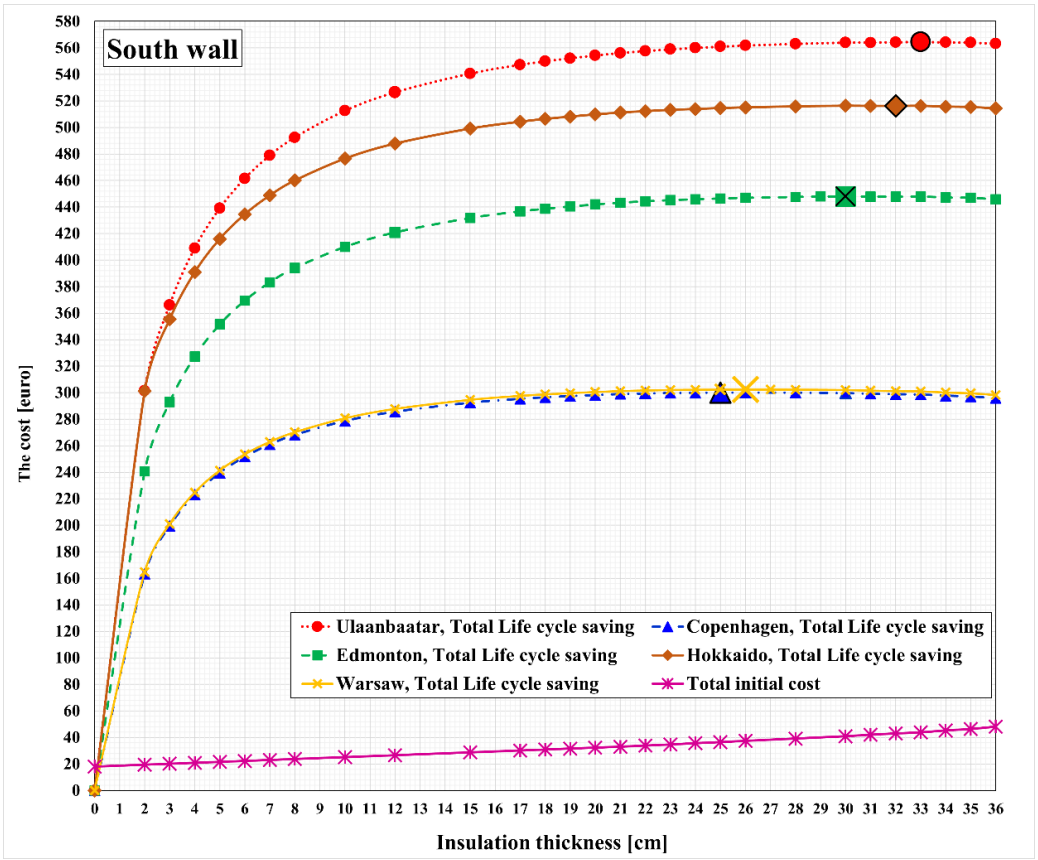


Figure 7.6. The total life cycle saving in 25 years and the total initial cost as a function of insulation thickness to calculate the optimum thickness of the south wall for the five cities.

Figure 7.6 illustrates the relationship between insulation thickness (cm) and cost savings (Euros) over 25 years, as well as the initial cost of insulation for the south-facing wall in different cities. This figure highlights that while adding insulation leads to significant long-term energy savings, there is an optimal thickness beyond which additional insulation is not cost-effective. The optimal thickness varies by city, depending on climate conditions, with colder cities like Ulaanbaatar and Hokkaido benefiting the most from thicker insulation.

Table 7.5 below shows the costs associated with energy use and heat loss when the insulation thickensses are around their optimum. Figure 7.7 then shows the payback time as a function of the thickness for the south wall of the five cities.

Table 7.5. Heat loss through the 1 m^2 of the two types of the south walls in the case of transient simulation and the energy cost in Euro in addition to payback time in years.

Wall orientation	Wall type	Insulator thickness (cm)	Cost of insulation (Euro)	Total Initial cost (Euro)	Heat loss kWh/m^2/6months	Total Annual energy cost (Euro)	Payback time (Years)	Annual saving energy cost (Euro)	Total saving energy cost in 25 years (Euro)	Total Life cycle saving (Euro)
Ulaanbaatar	Wall (A)	-	-	18	280.45	28.05	-	-	-	-
	Wall (B), glass	32	24.96	42.96	16.43	1.51	1.77	24.29	607.24	564.30
		33	25.76	43.76	15.96	1.47	1.79	24.33	608.32	564.56
	wool	34	27.2	45.2	15.51	1.43	1.85	24.37	609.34	564.14
Copen-hagen	Wall (A)	-	-	18	144.95	14.50	-		-	-
	Wall (B), glass	24	17.6	35.6	10.76	1.07	2.65	13.42	335.48	299.87
		25	18.448	36.448	10.36	1.03	2.71	13.46	336.47	300.12
	wool	26	19.36	37.36	9.98	0.99	2.76	13.49	337.40	300.04
Edmonton	Wall (A)	-		18	208.41	20.84	-	-	-	-
	Wall (B), glass	29	21.92	39.92	13.26	1.32	2.04	19.51	487.87	447.95
		30	22.88	40.88	12.85	1.28	2.09	19.55	488.89	448.01
	wool	31	24	42	12.46	1.24	2.14	19.59	489.86	447.86
Hokkaido	Wall (A)	-	-	18	236.16	23.61	-		-	-
	Wall (B), glass	31	24	42	12.88	1.28	1.88	22.33	558.21	516.21
		32	24.96	42.96	12.49	1.25	1.92	22.36	559.17	516.41
	wool	33	25.76	43.76	12.14	1.21	1.95	22.40	560.06	516.02
Warsaw	Wall (A)	-	-	18	145.97	14.60	-		-	-
	Wall (B), glass	25	18.448	36.448	10.45	1.04	2.69	13.55	338.81	302.36
		26	19.36	37.36	10.07	1.01	2.75	13.59	339.75	302.38
	wool	27	20.24	38.24	9.72	0.97	2.80	13.62	340.61	302.37

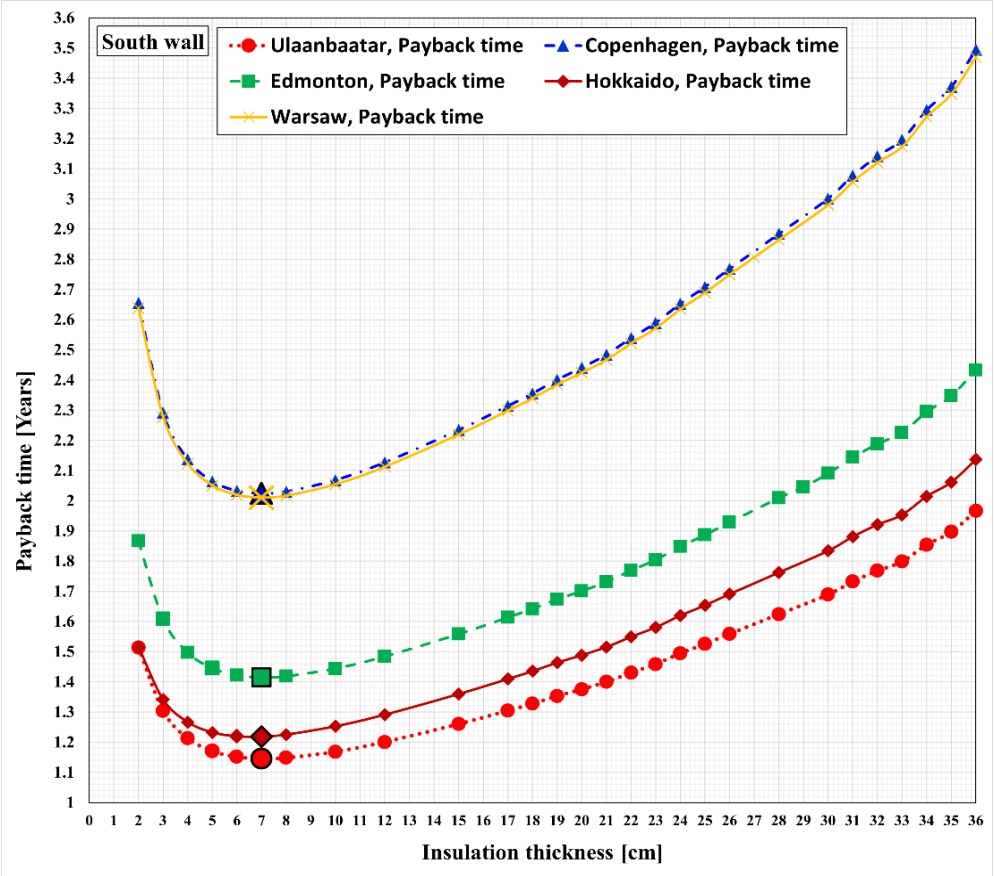


Figure 7.7. The payback time for the south wall of the five cities.

7.9. Summary of chapter

In this chapter presented a detailed numerical study on the optimal insulation thickness for north- and south-facing walls in five cold-climate cities using a transient simulation approach over a six-month winter period. The analysis incorporated real weather data, including outdoor temperature, wind speed, and solar radiation, to evaluate the thermal behavior of walls with and without insulation. The results demonstrate that insulation significantly reduces heat loss, and the effectiveness varies depending on local climate conditions. Colder cities such as Ulaanbaatar and Hokkaido benefit from thicker insulation, both in terms of thermal performance and long-term economic savings. Also showed that optimal insulation thickness depends not only on climatic severity but also on solar gains, especially on south-facing walls. In the later months of the heating season, notably March and April, increased solar radiation leads to greater differences in heat loss between south- and north-facing walls, highlighting the influence of solar exposure. The life cycle cost analysis further confirmed that selecting the optimal insulation thickness results in substantial long-term savings and short payback periods across all locations. These findings can support energy-efficient building design. The findings showed that, during the winter months in the location, wall orientation had a little but noticeable effect on the walls' thermal efficiency. The fact that these cold cities usually experience a typically rather cloudy initial portion of the winter can be used to explain the little difference between northern and southern orientation. The differences rise in March and early April, when the days are longer and the weather is more sunny.

8.TRANSIENT THERMAL SIMULATION OF A MULTI-LAYER PHOTOVOLTAIC PANEL USING THE LEAPFROG-HOPSCOTCH METHOD

In this section, I present a detailed transient thermal analysis of a representative five-layer solar panel structure consisting of a glass front sheet, two ethylene–vinyl acetate (EVA) encapsulant layers, a silicon cell layer, and an aluminum back sheet.

8.1. The physical model and the corresponding equations

The thermal behaviour of the photovoltaic (PV) module is described using a one-dimensional heat transfer model that includes heat conduction, light absorption, and convective and radiative heat exchange with the surrounding environment.

I assumed that the PV-module is placed into standard test conditions (STC), which practically means that I used standard data for the ambient temperature $u_a = 25°C$. An incident solar irradiance of 800 W/m² is applied to the top surface of the panel (see Fig 8.1), taken into account in the heat source term. Heat exchange inside the material of the PV-panel happens clearly by conductive heat transfer. As a result of the incoming light, difference between the ambient temperature and that of the *x*-*y* surfaces of the PV-panel increases during the turning on period. The extracted electrical power, which would mean an additional heat removal from the physical system, is not taken into account. The light absorption in the glass layer is 3%, while the EVA layer absorbs 7% of the incident radiation. The silicon layer receives and absorbs approximately 80% of the incoming light. These values indicate that the majority of the incident solar radiation passes through the encapsulation layers and is primarily absorbed by the silicon layer

The radiation, convection, and heat source terms are incorporated into the heat conduction equation (8.1) as follows:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + q - K \cdot u - \sigma \cdot u^4 \ , \qquad (8.1)$$

where $u(x,t)$ notes the temperature distribution within the PV module as a function of space and time, and α is the thermal diffusivity of the material. The first term on the right-hand side represents heat conduction inside the PV layer, which governs the spatial redistribution of thermal energy. where the terms K and σ are present only at the boundary of the solid material. The convective transfer of heat between a moving fluid and a surface is referred to as $K(u_a - u)$ in accordance with Newton's law of cooling. This term represents the rate of heat transfer per unit area between the solar panel surface (temperature u) and the ambient air (temperature u_a). According to the Stefan-Boltzmann equation [80], where the proportionality constant σ is the product of the positive Stefan-Boltzmann constant and the surface area, the radiative heat loss from the surface may be expressed by the term $-\sigma u^4$.

The partial differential equation (7.2) is discretized using a finite difference approach and solved numerically with the explicit leapfrog–hopscotch method Eq.(7.4-7.5), since, as I demonstrated in the previous chapters, it proved to be reliable and accurate.

8.2. Geometry and material properties of the PV

A one-dimensional heat conduction model is formulated Eq.(8.1). The schematic in Figure 8.1 depicts the one-dimensional heat transfer model used in the simulation. The composite structure, comprising glass, EVA, silicon, and aluminium layers, facilitates heat conduction through its thickness. The primary cooling mechanisms are convective heat loss to the ambient air and radiative heat loss to the surroundings, both occurring from the external glass surface.

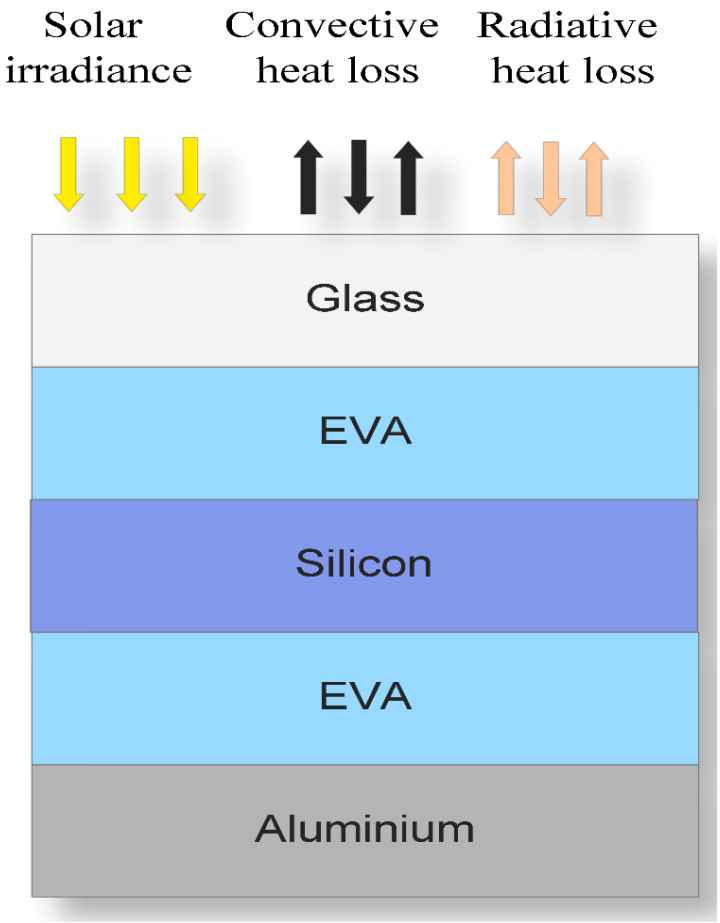


Figure 8.1. One-dimensional heat transfer model of the photovoltaic panel.

In this investigation, the real material properties listed in Table 8.1 are taken into account. Similarly to the previous chapter, these coefficients have a discontinuity at the material's boundary, but they are constants inside the material and do not vary with temperature, time, or space.

Table 8.1. Properties of the materials used in the simulation.

Material	Specific Heat (J/kg·K)	Density (kg/m³)	Thermal Conductivity (W/m·K)	Absorption	Thickness (m)
Glass	840	2500	0.78	0.1	0.006
EVA	2090	950	0.35	0.15	0.002

Silicon	705	2330	148	0.75	0.0006
Aluminum	237	2700	237	0.0	0.001

8.3. Results for the Simulation of the PV

The transient thermal response of the PV panel, simulated over a two-hour period with a computational time step of $\Delta t = 10$ seconds, is shown in Figure 8.2.

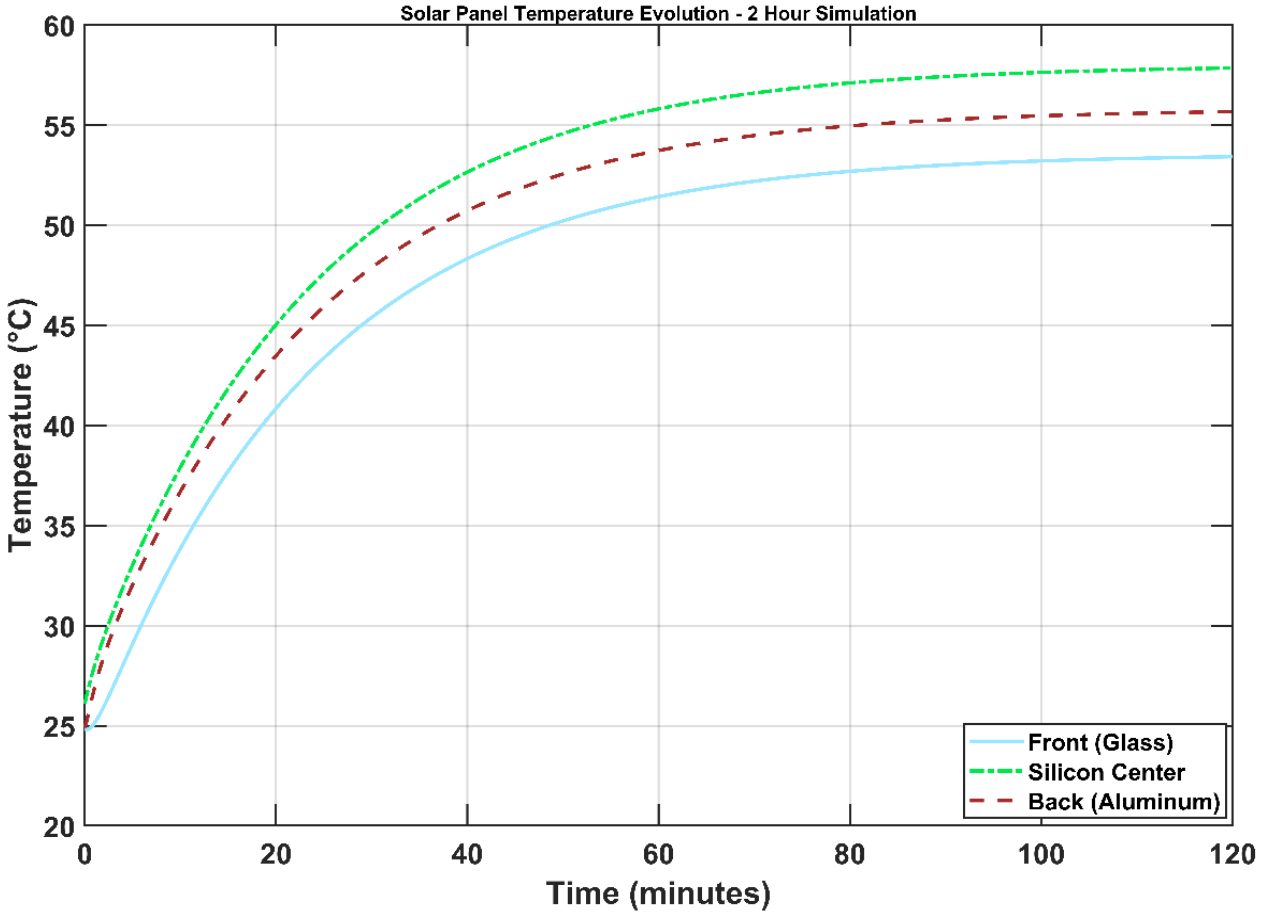


Figure 8.2. The temperature of the PV panel as a function of time in minutes.

The temperature profiles at the three critical locations the front glass, the silicon cell, and the aluminum back sheet exhibit a rapid initial heating phase followed by convergence toward a near steady-state condition.

The silicon cell consistently acts as the thermal hotspot, reaching a peak temperature of approximately 58.5 °C, which is higher than both the aluminum back sheet (~55.5 °C) and the front glass (~52.5 °C). The resulting maximum temperature differential of about 6 °C across the module's thickness demonstrates a substantial and persistent thermal gradient. These quantitative results, obtained using a stable explicit numerical scheme, directly characterize the thermal state of the PV module. The elevated silicon-cell temperature is a key contributor to electrical efficiency losses, while the significant temperature gradient through the layered structure highlights the level of thermal stress that can affect long-termmechanical reliability.

8.4. Summary of chapter

This chapter developed a transient thermal model for a five-layer photovoltaic (PV) module and solved it using the Leapfrog–Hopscotch explicit scheme. The approach demonstrated strong numerical stability and computational efficiency for multilayer heat-conduction problems.

By integrating realistic material properties and boundary conditions, the model accurately captured the dynamic thermal behaviour of the PV structure. The analysis provides essential insight into how heat propagates through the layered assembly and establishes a framework for assessing thermal stresses and their impact on module reliability and performance.

Overall, the modelling strategy presented here offers a practical and adaptable tool for future optimisation of PV thermal management, enabling improved design, enhanced cooling strategies, and better prediction of operating conditions.

9. THESES – NEW SCIENTIFIC RESULTS

T1. I examined several numerical algorithms possessing the convex combination property for the heat conduction equation and accurately reproduced the analytical reference solution in 1 space dimension and the numerical reference solution in 2 space dimension. I found that as the stiffness parameter is increased, the differences in accuracy between the methods decreased, indicating the presence of order reduction, which confirms that higher-order methods provide clear advantages mainly for non-stiff problems, while lower-order schemes are sufficient for very stiff cases. All methods satisfying the maximum–minimum principle remained stable for arbitrary time step sizes, even under extremely stiff conditions; among them, LH-CNe proved most efficient for low to medium accuracy requirements, whereas higher-order CLQ methods are preferable when high accuracy is needed, and multi-stage LNe methods can achieve comparable accuracy for smooth initial conditions.

T2. I systematically investigated explicit numerical algorithms for Fisher's equation that are unconditionally dynamically consistent and preserve positivity. I combined twelve diffusion schemes with pseudo-implicit and quasi-exact treatments of the nonlinear logistic term, including Strang-splitting, resulting in 72 algorithm combinations. Through comprehensive numerical testing in one- and two-dimensional systems with exact reference solutions and stiff, anisotropic random media, I evaluated the accuracy, stability, and computational efficiency of the method. Based on aggregated error measures and running time analysis, I identified the ten most effective methods and demonstrated that quasi-exact treatments combined with higher-order CLQ schemes and Strang-splitting provide superior accuracy for strong nonlinearity, while simpler LH-CNe-based methods are optimal for low-accuracy or weakly stiff problems.

T3. I solved the nonlinear Huxley equation, whose true solution remains within the unit interval, using numerous algorithm combinations with operator-splitting. I employed methods which preserve this unit-interval property for arbitrary spatial meshes, time steps, and nonlinear coefficient β. Through numerical experiments, I identified the most effective schemes under different conditions. I recommend the LH-CNe solver with t1 treatments and Strang-splitting when speed is prioritized, stiffness is moderate, or β is large, while the CLQ family is preferable for high accuracy in stiff problems. I also found that Strang-splitting is unnecessary for very small nonlinear coefficient. For complicated geometries, the CpC-t1-Strang solver offers high speed with acceptable accuracy.

T4. I performed extensive numerical tests to solve Huxley's equation, focusing on the most efficient methods with proven unconditional stability for the linear diffusion equation. I found that the LH solver with t1 treatment and Strang-splitting is generally the most efficient and reliable, providing accurate results quickly even for extremely stiff systems with strong nonlinearity. For complicated geometries where odd-even cell division is difficult, the CCL-t1-Strang or DF-mixed methods proved to be reliable alternatives, while the original odd-even

hopscotch method is suitable only for equidistant meshes with constant diffusion parameters. The Crank-Nicolson method is computationally expensive for large systems and offers no clear advantage over LH, though standard Newton iterations may improve accuracy for small systems. For extreme accuracy requirements, higher-order methods such as RK4 perform better than low-order schemes.

T5. I analyzed a coupled system of two reaction–diffusion equations with time-dependent reaction terms and used analytical solutions involving Kummer functions as reference solutions. I applied eight explicit numerical methods with excellent stability properties and verified their performance using the analytical solutions. By extensive parameter-sweep experiments, I demonstrated that these methods substantially outperform standard explicit schemes, and that the leapfrog–hopscotch method is generally the most efficient for solving such coupled systems.

T6. I developed a transient numerical framework to investigate the impact of climate variability on the optimal insulation thickness of building walls using long-term simulations driven by real meteorological data. By extending the heat conduction equation to include convection, radiation, and solar heat gains and solving it with the explicit leapfrog–hopscotch method, I simulated heat transfer through multilayer walls for five cold-climate cities with different Köppen classifications. I demonstrated that the leapfrog–hopscotch method enables stable and accurate simulations over an entire heating season using time steps far beyond the CFL limits of standard explicit schemes. Using life-cycle economic analysis, I identified climate-specific optimal insulation thicknesses and showed that colder cities benefit from significantly thicker insulation, while solar gains on south-facing walls reduce the economically optimal thickness in milder climates. The results confirm that optimal insulation design must account for both climatic severity and wall orientation, and that long-term transient simulations provide more realistic guidance than steady-state approaches.

T7. I developed a transient one-dimensional thermal model for a multilayer photovoltaic panel incorporating incoing sunshine and electric energy production in the form of positive and negative internal heat generation, respectively, radiative and convective boundary conditions, and realistic material properties. The governing heat conduction equation was discretized using a finite difference approach and solved with the explicit leapfrog–hopscotch method, demonstrating computational efficiency. Numerical simulations revealed persistent thermal gradients across the panel thickness, with the silicon cell acting as the dominant thermal hotspot. These results provide a reliable numerical framework for evaluating photovoltaic thermal behaviour and optimizing thermal management and module design.

ACKNOWLEDGEMENTS

…

REFERENCES

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

[2] Acton, Q.A. *Issues in Biophysics and Geophysics Research and Application: 2011 Edition*; ScholarlyEditions: Atlanta, GA, USA, 2012; ISBN 9781464964299.

[3] Zhokh, A.; Strizhak, P. Advection–diffusion in a porous medium with fractal geometry: Fractional transport and crossovers on time scales. *Meccanica* 2022, 57, 833–843.

[4] Yu, H.; Yao, L.; Ma, Y.; Hou, Z.; Tang, J.; Wang, Y.; Ni, Y. The moisture diffusion equation for moisture absorption of multiphase symmetrical sandwich structures. *Mathematics* 2022, 10, 2669.

[5] Savović, S.M.; Djordjevich, A. Numerical solution of diffusion equation describing the flow of radon through concrete. *Applied Radiation and Isotopes* 2008, 66, 552–555.

[6] Suárez-Carreño, F.; Rosales-Romero, L. Convergency and stability of explicit and implicit schemes in the simulation of the heat equation. *Applied Sciences* 2021, 11, 4468.

[7] Lima, S.A.; Kamrujjaman, M.; Islam, M.S. Numerical solution of convection–diffusion–reaction equations by a finite element method with error correlation. *AIP Advances* 2021, 11, 1–12.

[8] Li, Y.; van Heijster, P.; Marangell, R.; Simpson, M.J. Travelling wave solutions in a negative nonlinear diffusion–reaction model. *Journal of Mathematical Biology* 2020, 81, 1495–1522.

[9] Weickenmeier, J.; Jucker, M.; Goriely, A.; Kuhl, E. A physics-based model explains the prion-like features of neurodegeneration in Alzheimer's disease, Parkinson's disease, and amyotrophic lateral sclerosis. *Journal of the Mechanics and Physics of Solids* 2019, 124, 264–281.

[10] Campos, D.; Llebot, J.E.; Fort, J. Reaction–diffusion pulses: A combustion model. *Journal of Physics A: Mathematical and General* 2004, 37, 6609–6621.

[11] Columbu, A.; Frassu, S.; Viglialoro, G. Refined criteria toward boundedness in an attraction–repulsion chemotaxis system with nonlinear productions. *Applicable Analysis* 2024.

[12] Ma, W.X.; Fuchssteiner, B. Explicit and exact solutions to a Kolmogorov–Petrovskii–Piskunov equation. *International Journal of Non-Linear Mechanics* 1996, 31, 329–338.

[13] Tang, S.; Weber, R.O. Numerical study of Fisher's equation by a Petrov–Galerkin finite element method. *Journal of the Australian Mathematical Society, Series B* 1991, 33, 27–38.

[14] Chandraker, V.; Awasthi, A.; Jayaraj, S. A numerical treatment of Fisher equation. *Procedia Engineering* 2015, 127, 1256–1262.

[15] Dag, I.; Ersoy, O. The exponential cubic B-spline algorithm for Fisher equation. *Chaos, Solitons & Fractals* 2016, 86, 101–106.

[16] Tamsir, M.; Huntul, M.J. A numerical approach for solving Fisher's reaction–diffusion equation via a new kind of spline functions. *Ain Shams Engineering Journal* 2021, 12, 3157–3165.

[17] Holmes, M.H. *Introduction to Numerical Methods in Differential Equations*; Springer: New York, NY, USA, 2007.

[18] Anguelov, R.; Lubuma, J.M.-S.; Shillor, M. Dynamically consistent nonstandard finite difference schemes for continuous dynamical systems. *Contemporary Mathematics* 2009, 34–43.

[19] Chen-Charpentier, B.M.; Kojouharov, H.V. An unconditionally positivity preserving scheme for advection–diffusion–reaction equations. *Mathematical and Computer Modelling* 2013, 57, 2177–2185.

[20] Appadu, A.R. Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *International Journal of Numerical Methods for Heat & Fluid Flow* 2017, 27, 1412–1429.

[21] Dimitrov, D.T.; Kojouharov, H.V. Positive and elementary stable nonstandard numerical methods with applications to predator–prey models. *Journal of Computational and Applied Mathematics* 2006, 189, 98–108.

[22] Chertock, A.; Kurganov, A. A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models. *Numerische Mathematik* 2008, 111, 169–205.

[23] Agbavon, K.M.; Appadu, A.R.; Khumalo, M. On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term. *Advances in Difference Equations* 2019, 146.

[24] Agbavon, K.M.; Appadu, A.R. Construction and analysis of some nonstandard finite difference methods for the FitzHugh–Nagumo equation. *Numerical Methods for Partial Differential Equations* 2020, 36, 1145–1169.

[25] Chapwanya, M.; Lubuma, J.M.S.; Mickens, R.E. Positivity-preserving nonstandard finite difference schemes for cross-diffusion equations in biosciences. *Computers & Mathematics with Applications* 2014, 68, 1071–1082.

[26] Bradshaw-Hajek, B. *Reaction–Diffusion Equations for Population Genetics*. Ph.D. Thesis, University of Wollongong, Wollongong, Australia, 2004.

[27] Qin, X.; Jiang, Z.; Yan, C. Strong stability preserving two-derivative two-step Runge–Kutta methods. *Mathematics* 2024, 12, 2465.

[28] Anjuman, A.; Leung, A.Y.T.; Das, S. Two-dimensional time-fractional nonlinear drift reaction–diffusion equation arising in electrical field. *Fractal and Fractional* 2024, 8, 456.

[29] Chou, C.S.; Zhang, Y.T.; Zhao, R.; Nie, Q. Numerical methods for stiff reaction–diffusion systems. *Discrete and Continuous Dynamical Systems – Series B* 2007, 7, 515–525.

[30] Sharma, S.; Prabhakar, N. Numerical simulation of generalized FitzHugh–Nagumo equation by shifted Chebyshev spectral collocation method. *IAENG International Journal of Applied Mathematics* 2024, 54.

[31] Lukonde, J.; Kasumo, C. On numerical and analytical solutions of the generalized Burgers–Fisher equation. *Journal of Innovative Applied Mathematics and Computational Science* 2023, 3.

[32] El-Danaf, T.E.; Zaki, M.A.; Moenaaem, W. New numerical technique for solving the fractional Huxley equation. *International Journal of Numerical Methods for Heat & Fluid Flow* 2014, 24, 1736–1754.

[33] Shi, J.; Yang, X.; Liu, X. A novel fractional physics-informed neural networks method for solving the time-fractional Huxley equation. *Neural Computing and Applications* 2024, 36, 19097–19119.

[34] Bai, Y.; Chaolu, T.; Bilige, S. Solving Huxley equation using an improved PINN method. *Nonlinear Dynamics* 2021, 105, 3439–3450.

[35] Loyinmi, A.C.; Sanyaolu, M.D.; Gbodogbe, S. Exploring the efficacy of the weighted average method for solving nonlinear partial differential equations: A study on the Burger–Fisher equation. *Educational Journal of Science, Mathematics and Technology* 2025, 12.

[36] March, N.G.; Carr, E.J.; Turner, I.W. Numerical investigation into coarse-scale models of diffusion in complex heterogeneous media. *Transport in Porous Media* 2021, 139, 467–489.

[37] Kondratenko, P.S.; Matveev, A.L.; Vasiliev, A.D. Numerical implementation of the asymptotic theory for classical diffusion in heterogeneous media. *European Physical Journal B* 2021, 94, 50.

[38] Manaa, S.; Sabawi, M. Numerical solution and stability analysis of Huxley equation. *AL-Rafidain Journal of Computer Sciences and Mathematics* 2005, 2, 85–97.

[39] Rieth, Á.; Kovács, R.; Fülöp, T. Implicit numerical schemes for generalized heat conduction equations. *International Journal of Heat and Mass Transfer* 2018, 126, 1177–1182.

[40] Macías-Díaz, J.E. On an exact numerical simulation of solitary-wave solutions of the Burgers–Huxley equation through Cardano's method. *BIT Numerical Mathematics* 2014, 54, 763–776.

[41] Songolo, M.E. A positivity-preserving nonstandard finite difference scheme for parabolic systems with cross-diffusion equations and nonlocal initial conditions. *American Scientific Research Journal for Engineering, Technology, and Sciences* 2016, 18, 252–258.

[42] Lienhard, J.H., IV; Lienhard, J.H., V. *A Heat Transfer Textbook*, 5th ed.; Phlogiston Press: Cambridge, MA, USA, 2017.

[43] Simpson, M.J.; et al. Exact solutions of coupled multispecies linear reaction–diffusion equations on a uniformly growing domain. *PLOS ONE* 2015, 10, e0138894.

[44] Escorcia, J.M.; Suazo, E. On explicit solutions for coupled reaction–diffusion and Burgers-type equations with variable coefficients through a Riccati system. *Discrete and Continuous Dynamical Systems – Series S* 2025.

[45] Mátyás, L.; Barna, I.F. General self-similar solutions of diffusion equation and related constructions. *Romanian Journal of Physics* 2022, 67, 101.

[46] Barna, I.F.; Mátyás, L. Diffusion cascades and mutually coupled diffusion processes. *Mathematics* 2024, 12.

[47] Ascher, U.M.; Ruuth, S.J.; Wetton, B.T.R. Implicit–explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis* 1995, 32, 797–823.

[48] Calvo, M.P.; de Frutos, J.; Novo, J. Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations. *Applied Numerical Mathematics* 2001, 37, 535–549.

[49] Settanni, G.; Sgura, I. Devising efficient numerical methods for oscillating patterns in reaction–diffusion systems. *Journal of Computational and Applied Mathematics* 2016, 292, 674–693.

[50] Yadav, V.S.; et al. New RK-type time-integration methods for stiff convection–diffusion–reaction systems. *Computers & Fluids* 2023, 257, 105865.

[51] Beuken, L.; et al. Numerical stability and performance of semi-explicit and semi-implicit predictor–corrector methods. *Mathematics* 2022, 10.

[52] Fedoseev, P.; et al. New step size control algorithm for semi-implicit composition ODE solvers. *Algorithms* 2022, 15, 275.

[53] Ji, Y.; Xing, Y. Highly accurate and efficient time integration methods with unconditional stability and flexible numerical dissipation. *Mathematics* 2023, 11, 593.

[54] Jiang, T.; Zhang, Y.T. Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection–diffusion–reaction equations. *Journal of Computational Physics* 2013, 253, 368–388.

[55] Essongue, S.; Ledoux, Y.; Ballu, A. Speeding up mesoscale thermal simulations of powder bed additive manufacturing thanks to the forward Euler time-integration scheme: A critical assessment. *Finite Elements in Analysis and Design* 2022, 211, 103825.

[56] Al-Bayati, A.; Manaa, S.; Al-Rozbayani, A. Comparison of finite difference solution methods for reaction–diffusion systems in two dimensions. *AL-Rafidain Journal of Computer Sciences and Mathematics* 2011, 8, 21–36.

[57] Pourghanbar, S.; et al. An efficient alternating direction explicit method for solving a nonlinear partial differential equation. *Mathematical Problems in Engineering* 2020, 2020, 1–12.

[58] ANSI/ASHRAE. *Thermal Environmental Conditions for Human Occupancy*; ANSI/ASHRAE Standard 55; Atlanta, GA, USA, 1992.

[59] Hilke, A.; Ryan, L. *Mobilising Investment in Energy Efficiency*; International Energy Agency: Paris, France, 2012.

[60] Abu, S.; Ibrahim, T.; Shareef, S.; Mushtaha, E.; Alsyouf, I. Sustainable façade cladding selection for buildings in hot climates based on thermal performance and energy consumption. *Results in Engineering* 2022, 16.

[61] Ghrab-Morcos, N. CHEOPS: A simplified tool for thermal assessment of Mediterranean residential buildings in hot and cold seasons. *Energy and Buildings* 2005, 37, 651–662.

[62] Barber, K.A.; Krarti, M. A review of optimization-based tools for design and control of building energy systems. *Renewable and Sustainable Energy Reviews* 2022, 160, 112359.

[63] Kolaitis, D.I.; et al. Comparative assessment of internal and external thermal insulation systems for energy-efficient retrofitting of residential buildings. *Energy and Buildings* 2013, 64, 123–131.

[64] Aïssani, P.A.A.; Chateauneuf, A.; Fontaine, J.P. Cost model for optimum thicknesses of insulated walls considering indirect impacts and uncertainties. *Energy and Buildings* 2014, 84, 21–32.

[65] Nematchoua, M.K.; 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* 2015, 50, 1192–1202.

[66] Gaarder, J.E.; et al. Optimization of thermal insulation thickness pertaining to embodied and operational GHG emissions in cold climates. *Building and Environment* 2023, 234, 110187.

[67] Alrasheed, M.R.A. Estimating optimal cost, insulation layer thickness, and structural layer thickness of different composite insulation external walls using computational methods. *Buildings* 2023, 13, 2774.

[68] Uçar, A. Optimization of insulation thickness of walls and roofs using energy, exergy, economic and environmental (4E) analyses. *Journal of Mechanical Engineering and Sciences* 2024, 18, 9959–9975.

[69] Kaynakli, O. A study on residential heating energy requirement and optimum insulation thickness. *Renewable Energy* 2008, 33, 1164–1172.

[70] Cvetković, D.; Bojić, M. Optimization of thermal insulation of a house heated by using radiant panels. *Energy and Buildings* 2014, 85, 329–336.

[71] Daouas, N. A study on optimum insulation thickness in walls and energy savings in Tunisian buildings. *Applied Energy* 2011, 88, 156–164.

[72] Ucar, A.; Balo, F. Determination of environmental impact and optimum thickness of insulation for building walls. *Environmental Progress & Sustainable Energy* 2011, 30, 113–122.

[73] Çomaklı, K.; Yüksel, B. Optimum insulation thickness of external walls for energy saving. *Applied Thermal Engineering* 2003, 23, 473–479.

[74] Ozel, M. Cost analysis for optimum thicknesses and environmental impacts of different insulation materials. *Energy and Buildings* 2012, 49, 552–559.

[75] Omle, I.; Askar, A.H.; Kovács, E. Optimizing the design of container house walls using argon and recycled plastic materials. *Buildings* 2024, 14, 3944.

[76] Askar, A.H.; Omle, I.; Kovács, E. The role of roof angle and geographic location on the thermal performance of buildings. *International Journal of Thermofluids* 2025, 27, 101192.

[77] SolarPower Europe. *Global Market Outlook for Solar Power 2025–2029*; SolarPower Europe: Brussels, Belgium.

[78] Notton, G.; Cristofari, C.; Mattei, M.; Poggi, P. Modelling of a double-glass photovoltaic module using finite differences. *Applied Thermal Engineering* 2005, 25, 2854–2877.

[79] Shahverdian, M.H.; et al. Multi-layer modeling of bifacial photovoltaic panels: Evaluating the accuracy of one-, three-, and five-layer models. *Journal of Solar Energy Engineering* 2023, 145, 021001.

[80] Nagy, Á.; Bodnár, I.; Kovács, E. Simulation of the thermal behavior of a photovoltaic solar panel using recent explicit numerical methods. *Advanced Theory and Simulations* 2024.

[81] Muñoz-Matute, J.; Calo, V.M.; Pardo, D.; Alberdi, E.; van der Zee, K.G. Explicit-in-time goal-oriented adaptivity. *Computer Methods in Applied Mechanics and Engineering* 2019, 347, 176–200, doi:10.1016/j.cma.2018.12.028.

[82] Barakat, H.Z.; Clark, J.A. On the solution of the diffusion equations by numerical methods. *Journal of Heat Transfer* 1966, 88, 421–427.

[83] Liu, H.; Leung, S. An alternating direction explicit method for time evolution equations with applications to fractional differential equations. *Methods and Applications of Analysis* 2020, 26, 249–268.

[84] Kovács, E. A class of new stable, explicit methods to solve the non-stationary heat equation. *Numerical Methods for Partial Differential Equations* 2021, 37, 2469–2489.

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

[86] Kovács, E.; Nagy, Á.; Saleh, M. A set of new stable, explicit, second order schemes for the non-stationary heat conduction equation. *Mathematics* 2021, 9, 2284, doi:10.3390/math9182284.

[87] Kovács, E. A class of new stable, explicit methods to solve the non-stationary heat equation. *Numerical Methods for Partial Differential Equations* 2020, 37, 2469–2489, doi:10.1002/num.22730.

[88] Kovács, E.; Nagy, Á. A new stable, explicit, and generic third-order method for simulating conductive heat transfer. *Numerical Methods for Partial Differential Equations* 2023, 39, 1504–1528.

[89] Kovács, E.; Majár, J.; Saleh, M. Unconditionally positive, explicit, fourth order method for the diffusion- and Nagumo-type diffusion–reaction equations. *Journal of Scientific Computing* 2024, 98, 39.

[90] Gourlay, A.R.; McGuire, G.R. General hopscotch algorithm for the numerical solution of partial differential equations. *IMA Journal of Applied Mathematics* 1971, 7, 216–227.

[91] Nagy, Á.; Omle, I.; Kareem, H.; Kovács, E.; Barna, I.F.; Bognár, G. Stable, explicit, leapfrog–hopscotch algorithms for the diffusion equation. *Computation* 2021, 9, 92.

[92] Hirsch, C. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*; Wiley: New York, NY, USA, 1988.

[93] Jalghaf, H.K.; Kovács, E.; Majár, J.; Nagy, Á.; Askar, A.H. Explicit stable finite difference methods for diffusion–reaction type equations. *Mathematics* 2021, 9, 3308, doi:10.3390/math9243308.

[94] Barna, I.F.; Mátyás, L. Advanced analytic self-similar solutions of regular and irregular diffusion equations. *Mathematics* 2022, 10, 3281, doi:10.3390/math10183281.

[95] Saleh, M.; Kovács, E.; Barna, I.F. Analytical and numerical results for the transient diffusion equation with diffusion coefficient depending on both space and time. *Algorithms* 2023, 16, 184, doi:10.3390/a16040184.

[96] Bastani, M.; Salkuyeh, D.K. A highly accurate method to solve Fisher's equation. *Pramana – Journal of Physics* 2012, 78, 335–346, doi:10.1007/s12043-011-0243-8.

[97] Khayrullaev, H.; Omle, I.; Kovács, E. Systematic investigation of the explicit, dynamically consistent methods for Fisher's equation. *Computation* 2024, 12, 3, doi:10.3390/computation12030049.

[98] Askar, A.H.; Omle, I.; Kovács, E.; Majár, J. Testing some different implementations of heat convection and radiation in the leapfrog–hopscotch algorithm. *Algorithms* 2022, 15, 400, doi:10.3390/a15110400.

[99] Holman, J.P. *Heat Transfer*, 10th ed.; McGraw-Hill Education: New York, NY, USA, 2010; ISBN 0073529362.

[100] Jejeniwa, O.A.; Gidey, H.H.; Appadu, A.R. Numerical modeling of pollutant transport: Results and optimal parameters. *Symmetry* 2022, 14, 2616.

[101] Accurate weather forecasts for any location. 2023.

[102] Properties of insulating materials. *Hoszigetelesplaza.hu*, accessed April 1, 2023.

[103] Properties of insulating materials. *Hoszigetelesplaza.hu*, accessed April 1, 2023.

[104] Omle, I.; Askar, A.H.; Kovács, E. Comparison of the performance of new and traditional numerical methods for long-term simulations of heat transfer in walls with thermal bridges. 2023.

LIST OF PUBLICATIONS RELATED TO THE TOPIC OF THE RESEARCH FIELD

(1) Khayrullaev, H.; Zain, A.; Kovács, E. Solution of coupled systems of reaction–diffusion equations using explicit numerical methods with outstanding stability properties. Computation 2025, 13, 129, doi:10.3390/computation13060129.

(2) Khayrullaev, H.; Kovács, E. Unconditionally dynamically consistent numerical methods with operator-splitting for a reaction–diffusion equation of Huxley's type. Mathematics 2025, 13, 2848, doi:10.3390/math13172848.

(3) Khayrullaev, H.; Omle, I.; Kovács, E. Exploring the performance of some efficient explicit numerical methods with good stability properties for Huxley's equation. Mathematics 2025, 13, 207, doi:10.3390/math13020207.

(4) Khayrullaev, H.; Omle, I.; Kovács, E. Systematic investigation of the explicit, dynamically consistent methods for Fisher's equation. Computation 2024, 12, 49, doi:10.3390/computation12030049.

(5) Khayrullaev, H.; Kovács, E. Comprehensive investigation of the explicit, positivity preserving methods for the heat equation: Part 1. Multidiszcip. Tudományok 2024, 14, 46–59, doi:10.35925/j.multi.2024.1.5.

(6) Khayrullaev, H.; Kovács, E. Comprehensive investigation of the explicit, positivity preserving methods for the heat equation: Part 2. Multidiszcip. Tudományok 2024, 14, 60–70, doi:10.35925/j.multi.2024.1.6.

(7) Khayrullaev, H.; Kovács, E. Numerical performance of some positivity preserving methods for the diffusion equation where the diffusion coefficient depends on both time and space. International Journal of Advanced Natural Sciences and Engineering Researches 2024,

(8) Khayrullaev, H. Transient thermal simulation of a multi-layer photovoltaic panel using the leapfrog-hopscotch method. Conference Paper Dec. 2025

(9) Khayrullaev, H.; Omle, I.; Zain, A.; Kovács, E. Impact of climate variability on optimal thickness of wall insulation: A numerical study across five cold cities using long-term numerical simulations. Article Jan. 2026 (ResearchGate).

(10) Khayrullaev, H.; Omle, I.; Kovács, E. Supplementary material to "Exploring the performance of some efficient explicit numerical method with good stability properties for Huxley's equation" (data and supplementary files). Data Jan. 2025 (ResearchGate).

APPENDICES