# A Set of New Stable, Explicit, Second Order Schemes for the Non-Stationary Heat Conduction Equation 

Endre Kovács * © , Ádám Nagy and Mahmoud Saleh<br>Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary; fizadam@uni-miskolc.hu (Á.N.); mhmodsalh84@gmail.com (M.S.)<br>* Correspondence: fizendre@uni-miskolc.hu

Citation: 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. https://doi.org/10.3390/ math9182284

Academic Editor: José A. Tenreiro Machado

Received: 31 July 2021
Accepted: 11 September 2021
Published: 16 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ $4.0 /$ ).


#### Abstract

This paper introduces a set of new fully explicit numerical algorithms to solve the spatially discretized heat or diffusion equation. After discretizing the space and the time variables according to conventional finite difference methods, these new methods do not approximate the time derivatives by finite differences, but use a combined two-stage constant-neighbour approximation to decouple the ordinary differential equations and solve them analytically. In the final expression for the new values of the variable, the time step size appears not in polynomial or rational, but in exponential form with negative coefficients, which can guarantee stability. The two-stage scheme contains a free parameter $p$ and we analytically prove that the convergence is second order in the time step size for all values of $p$ and the algorithm is unconditionally stable if $p$ is at least 0.5 , not only for the linear heat equation, but for the nonlinear Fisher's equation as well. We compare the performance of the new methods with analytical and numerical solutions. The results suggest that the new algorithms can be significantly faster than the widely used explicit or implicit methods, particularly in the case of extremely large stiff systems.


Keywords: heat equation; explicit time-integration; stiff equations; parabolic partial differential equations; unconditional stability

## 1. Introduction

It is generally known that conductive heat transfer is described by a second-order linear parabolic partial differential equation (PDE), the so-called heat equation:

$$
\begin{equation*}
\frac{\partial u}{\partial t}=\alpha \nabla^{2} u+q \tag{1}
\end{equation*}
$$

or more generally

$$
\begin{equation*}
c \rho \frac{\partial u}{\partial t}=\nabla(k \nabla u)+c \rho q \tag{2}
\end{equation*}
$$

where $u=u(x, t)$ or, in the case of Equation (2), $u=u(\vec{r}, t)$ is the temperature, $\alpha=$ $k /(c \rho)>0$ is the thermal diffusivity, $q, k, c$, and $\rho$ are the intensity of heat sources (electromagnetic radiation, chemical reactions, radioactive decay, etc.), heat conductivity, specific heat and (mass) density, respectively. These equations and their generalizations such as the convection (or advection)-diffusion equation and reaction-diffusion equation are used to model the diffusion of various particles in chemical and biological systems [1] as well as in electronic devices [2]. Mathematically very similar equations are used to treat the flow of fluids through porous media [3], like moisture [4], ground water or crude oil in reservoirs. When diffusion or fluid flow is modelled, the $u$ variable denotes the concentration or the pressure. We will examine one of the nonlinear reaction-diffusion equations, namely Fisher's equation [5], which originally was introduced to model how advantageous gene-variants spread in space and time.

As applied science and measurement techniques are developing, higher and higher resolutions are required in numerical simulations. Because of this trend, the investigated spatial domain has to be divided into smaller and smaller cells during the discretization process, which means that the number of cells, i.e., the number of unknowns and that of equations are also rapidly growing. On the other hand, in real systems, the physical properties like the specific heat and the heat conductivity can be completely different at different points close to one another ([6], p. 15). Therefore the coefficients and thus the eigenvalues of the problem may have a range of several orders of magnitude, which implies that the problem can be highly stiff. It is well known that conventional explicit methods (either Runge-Kutta or multistep Adams-Bashforth types) are only conditionally stable; thus, the time steps must be reduced to very small values regardless of the actual accuracy requirements. This is why the implicit methods with much better stability properties are widely considered to be superior $[7,8]$ and almost exclusively used to solve these kinds of problems [9]. On the other hand, they require the solution of a system of algebraic equations at each time step, often by iterative methods. For huge matrices their solution is still rather slow, especially in more than one space dimension, where the matrix is not tridiagonal. To overcome the difficulties, enormous efforts have been made and numerous techniques have been developed. More and more sophisticated preconditioners are used [10,11], geometric, algebraic and hybrid (i.e., combined geometric and algebraic) multigrid systems are applied $[12,13]$ and even the implicit Euler method is supplemented by a linear time filter to remove its overdamping and increase its accuracy [14]. These improvements are impressive, but unfortunately, the parallelization of implicit algorithms is still not straightforward, although some moderate progress has been achieved [15,16]. However, as the clock frequency increase of CPUs has drastically slowed down in the last two decades, the trend towards increasing parallelism continues in high performance computing [17].

Nevertheless, it is not true that all explicit methods have such restrictive stability conditions. To the contrary, a couple of explicit methods exist which are more stable or even unconditionally stable. The most notable examples are the Dufort-Frankel scheme ([18], p. 88), the Runge-Kutta-Chebyshev methods [19], the Hopscotch method [20,21], Alternating Direction Explicit (ADE) and Alternating Group Explicit (AGE) methods [22-24], the D'Yakonov fully explicit variant of the iterative alternating decomposition method [25] and the positivity preserving schemes of Chen-Charpentier and others [26,27]. We have to note that, according to our knowledge, these methods haven't even been listed together in any single document apart from our work, let alone being systematically tested against one another and against popular methods. These methods also have weak points, for example, some of them are only conditionally convergent or consistent, they are usually less accurate, they can be complicated to code or hardly be applied for irregular grids [28-32]. These disadvantages explain, at least partially, that the investigation and application of these methods are rather rare (see for example [4,16,33,34]) and they remained relatively unknown. Even those scholars who work with semi-implicit schemes, also called mixed explicit-implicit schemes [35], do not really use these explicit methods. From the information collected above, one may draw the conclusion that known methods do not provide a truly convenient solution for this problem. In order to fill this growing gap, we started to elaborate a class of fundamentally new explicit methods. The simplest, first order member of this class has been already published [36]. At the beginning of the next section, we restate this simple method, because this is the basis of the second order methods; thus, it is necessary to understand this first. Then we introduce the new algorithms, first for the simplest, one dimensional grid, then for general, arbitrary grids. In Section 3 we analyse their properties, then we turn to the numerical tests in the next sections. We present here two analytical cases in Section 4 for verification (heat equation and Fisher's equation) and then two numerical cases in Section 5. We mention that the method we introduce here can be considered as a generalization of our previous CN2 method [37], which was only first order because it used only integer time steps.

## 2. The Description of the Methods

To solve Equation (1) numerically, we begin with the most typical starting step, as in the standard method of lines. We discretize the second space derivatives by the most widely used second order central difference formula ([38], p. 48)

$$
\begin{equation*}
\frac{\partial^{2}}{\partial x^{2}} f\left(x_{\mathrm{i}}, t_{\mathrm{j}}\right) \approx \frac{\frac{f\left(x_{\mathrm{i}+1}, \mathrm{t}_{\mathrm{j}}\right)-f\left(x_{\mathrm{i}}, t_{\mathrm{j}}\right)}{\Delta x}+\frac{f\left(x_{\mathrm{i}-1}, \mathrm{t}_{\mathrm{j}}\right)-f\left(x_{\mathrm{i}}, t_{\mathrm{j}}\right)}{\Delta x}}{\Delta x} \tag{3}
\end{equation*}
$$

First, we examine the one dimensional heat equation in a homogeneous medium. After defining an equidistant grid of $N$ grid nodes we get the following ordinary differential equation (ODE) system:

$$
\frac{d u_{\mathrm{i}}}{d t}=\alpha \frac{u_{\mathrm{i}-1}-2 u_{\mathrm{i}}+u_{\mathrm{i}+1}}{\Delta x^{2}}+Q_{\mathrm{i}}
$$

where $Q_{\mathrm{i}}$ is the value of the $q(x)$ source function at node $i$. This system of equations can be written in a briefer matrix-form:

$$
\begin{equation*}
\frac{d \vec{u}}{d t}=M \vec{u}+\vec{Q} \tag{4}
\end{equation*}
$$

The diagonal elements of the matrix $M$ are the sums of the non-diagonal elements of the appropriate rows with a negative sign:

$$
m_{\mathrm{ii}}=-\frac{2 \alpha}{\Delta x^{2}}(1<i<N), \quad m_{11}=m_{N N}=-\frac{\alpha}{\Delta x^{2}}
$$

We introduce the characteristic time or time-constant of the cell i:

$$
\tau_{\mathrm{i}}=\frac{-1}{m_{\mathrm{ii}}}=\frac{\Delta x^{2}}{2 \alpha}
$$

### 2.1. The One-Stage Constant-Neighbour (CN) Method

In this point we will recall the already published [36] method to solve the ODE system (4), which can be derived through the following steps:

1. We make an approximation: When we calculate the new value of a variable $u_{\mathrm{i}}^{\mathrm{n}+1}$, we neglect that other variables are also changing during the time step $h=\Delta t$. This means that we consider $u_{\mathrm{j}}$ a constant if $j \neq i$, so we have to solve uncoupled, linear ODEs:

$$
\begin{equation*}
\frac{d u_{\mathrm{i}}(t)}{d t}=a_{\mathrm{i}}-\frac{u_{\mathrm{i}}(t)}{\tau_{\mathrm{i}}} \tag{5}
\end{equation*}
$$

where the $u_{\mathrm{i}}$ unknowns are considered as a function of time with initial values $u_{\mathrm{i}}^{n}$, which are the temperatures at the beginning of the $n$-th time step. We introduced

$$
a_{\mathrm{i}}=\sum_{\mathrm{j} \neq \mathrm{i}} m_{\mathrm{ij}} u_{\mathrm{j}}^{\mathrm{n}}+Q_{\mathrm{i}}=\alpha \frac{u_{\mathrm{i}-1}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}}{\Delta x^{2}}+Q_{\mathrm{i}} .
$$

2. The ODE system (5) is a highly simplified approximation of the original PDE, and it is straightforward to solve it analytically:

$$
u_{\mathrm{i}}(t)=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{t}{\tau_{\mathrm{i}}}}+a_{\mathrm{i}} \tau_{\mathrm{i}} \cdot\left(1-e^{-\frac{t}{\tau_{\mathrm{i}}}}\right) \quad 0<t<h
$$

We use this solution at the end of the time step to obtain the new values of the $u$ variable:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{h}{\tau_{\mathrm{i}}}}+a_{\mathrm{i}} \tau_{\mathrm{i}} \cdot\left(1-e^{-\frac{h}{\tau_{\mathrm{i}}}}\right) . \tag{6}
\end{equation*}
$$

For a standard one dimensional homogeneous system with an equidistant grid, we have the following one-stage first order method, which was called "constantneighbour method" or CN scheme.
The CN method algorithm

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{2 \alpha h}{\Delta x^{2}}}+\left[\frac{u_{\mathrm{i}-1}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}}{2}+\frac{\Delta x^{2} Q_{\mathrm{i}}}{2 \alpha}\right]\left(1-e^{-\frac{2 \alpha h}{\Delta x^{2}}}\right) \tag{7}
\end{equation*}
$$

while the actual values can be denoted as $u_{\mathrm{i}}^{C N}$. Formula (7) can be written as:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}}+\left(\frac{u_{\mathrm{i}-1}^{\mathrm{n}}-2 u_{\mathrm{i}}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}}{2}+\frac{\Delta x^{2} Q_{\mathrm{i}}}{2 \alpha}\right)\left(1-e^{-\frac{2 \alpha h}{\Delta x^{2}}}\right) \tag{8}
\end{equation*}
$$

where the simple identity $e^{\varphi}=1+\left(e^{\varphi}-1\right)$ has been applied. Using the power series of the exponential function, we get:

$$
1-e^{-\frac{2 \alpha h}{\Delta x^{2}}}=\frac{2 \alpha h}{\Delta x^{2}}-2\left(\frac{\alpha h}{\Delta x^{2}}\right)^{2}+\ldots
$$

Inserting this back to the previous (8) formula and keeping only the first term yields:

$$
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}}+\alpha \frac{u_{\mathrm{i}-1}^{\mathrm{n}}-2 u_{\mathrm{i}}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}}{\Delta x^{2}} h+Q_{\mathrm{i}} h
$$

which is exactly the explicit Euler method, also called Forward Time Central Space (FTCS) scheme in the case of PDEs. One can see that regarding one time step, the one-stage CN method is identical to the explicit Euler method, but only up to first order. However, in spite of this apparent similarity, there is a fundamental difference between the methods: During step (1) we do not fix the actual variable $u_{\mathrm{i}}$ to its initial value (we fix only the neighbours), but keep it as a variable. This means that at the end of this step we still have differential equations and not difference equations to be solved analytically, i.e., here time derivatives are not approximated by finite difference formulas as in Finite Difference Methods (FDM). In our CN method, the time step size $h$ in the Euler/FTCS scheme is replaced by $\left[1-\exp \left(-\frac{2 \alpha h}{\Delta x^{2}}\right)\right] \frac{\Delta x^{2}}{2 \alpha}$ and because of these exponential terms, Formulas (6)-(8) contain $h$ up to infinite order, which is crucial for the unconditional stability.

### 2.2. The Two-Stage Constant-Neighbour Method

Now we show how to create a more accurate, second order two-stage "combined CN" algorithm. In Table 1, one can see the Butcher tableau of the well-known explicit generic second order Runge-Kutta (RK) method. In three cases these methods obtained distinct names:

- for $p=1 / 2$, it is called the midpoint method.
- for $p=1$, one has the explicit trapezoidal rule, which is also called Heun's method or improved Euler's method.
- for for $p=2 / 3$, it is called the Ralston-method, which provides a minimum bound on the truncation error for the second-order RK algorithms [39].

Table 1. Butcher tableau of the explicit generic second order Runge-Kutta (RK) method, $p \in \mathbb{R}^{+}$.

| 0 | 0 | 0 |
| :---: | :---: | :---: |
| $p$ | $p$ | $p$ |
|  | $1-\frac{1}{2 p}$ | $\frac{1}{2 p}$ |

The first attempt could be to replace the time step $h$ in these second order RK schemes by $\left[1-\exp \left(-\frac{2 \alpha h}{\Delta x^{2}}\right)\right] \frac{\Delta x^{2}}{2 \alpha}$ to stabilize it. We obtained that although this replacement indeed
yields stable algorithms, they are still not second order. That is why we must return to the original logic by which we derived the first order CN method and combine it with the logic of the two-stage RK method above. At the first stage, we can calculate new "predictor" values of the variables, using (6) but with $h_{1}=p h$ time step:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{C}_{\mathrm{p}}}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{p h}{\tau_{\mathrm{i}}}}+a_{\mathrm{i}} \tau_{\mathrm{i}} \cdot\left(1-e^{-\frac{p h}{\tau_{\mathrm{i}}}}\right) . \tag{9}
\end{equation*}
$$

At the second stage, we can calculate the linear combination:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{comb}}=\left(1-\frac{1}{2 p}\right) u_{\mathrm{i}}^{\mathrm{n}}+\frac{1}{2 p} u_{\mathrm{i}}^{\mathrm{C}_{\mathrm{p}}} \tag{10}
\end{equation*}
$$

and take a full time step using these combinations as the more accurate constant values $u_{\mathrm{j}}^{\mathrm{comb}}$ of the neighbours towards which the $u_{\mathrm{i}}$ unknown functions are tending with increasing time step size $h$ because $e^{-\frac{p h}{\tau_{\mathrm{i}}}} \rightarrow 0$ if $h \rightarrow 0$. This means we have to replace $a_{\mathrm{i}}$ in (6) to:

$$
\begin{gather*}
a_{\mathrm{i}}^{\mathrm{comb}}=\sum_{\mathrm{j} \neq \mathrm{i}} m_{\mathrm{ij}} u_{\mathrm{j}}^{\mathrm{comb}}+Q_{\mathrm{i}},  \tag{11}\\
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{h}{\tau_{\mathrm{i}}}}+a_{\mathrm{i}}^{\mathrm{comb}} \tau_{\mathrm{i}}\left(1-e^{-\frac{h}{\tau_{\mathrm{i}}}}\right) . \tag{12}
\end{gather*}
$$

For a standard one dimensional homogeneous equidistant grid, we suggest the following Algorithm 1:

Algorithm 1. (Two-stage Constant-neighbour with ph time step and Constant-neighbour
with full time step (briefly: CpC ) method):
Stage 1. Take a $h_{1}=p h$ time step with the CN method with $p \in \mathbb{R}^{+}$:
$u_{\mathrm{i}}^{\mathrm{c}_{\mathrm{p}}}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{2 \alpha p h}{\Delta x^{2}}}+\left[\frac{u_{\mathrm{i}-1}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}}{2}+\frac{\Delta x^{2} \mathrm{Q}_{\mathrm{i}}}{2 \alpha}\right]\left(1-e^{-\frac{2 \alpha p h}{\Delta x^{2}}}\right)$.
Stage 2. Take a full $h$ time step with the CN method:
$u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}} \cdot e^{-\frac{2 \alpha h}{\Delta x^{2}}+\left[\frac{u_{\mathrm{i}-1}^{\mathrm{comb}}+u_{\mathrm{i}+1}^{\mathrm{comb}}}{2}+\frac{\Delta x^{2} \mathrm{Q}_{\mathrm{i}}}{2 \alpha}\right]\left(1-e^{-\frac{2 \alpha h}{\Delta x^{2}}}\right) .{ }^{\text {con }}}$
where
$u_{\mathrm{j}}^{\mathrm{comb}}=\left(1-\frac{1}{2 p}\right) u_{\mathrm{j}}^{\mathrm{n}}+\frac{1}{2 p} u_{\mathrm{j}}^{\mathrm{c}_{\mathrm{p}}}$.

The $C \frac{1}{2} C, C \frac{2}{3} C, C 1 C$ versions are analogous to the (explicit) midpoint, Ralston and trapezoidal methods, respectively. In Section 3 we will analyse the general CpC algorithm and prove that it is second order in the time step size $h$. One might think that this whole procedure of constructing new methods using the already known RK schemes is trivial. To refute this opinion it is enough to note that following the logic explained above and using the Butcher-tableau of third order RK methods, the obtained methods are not third order, sometimes not even second order.

### 2.3. Extension to a General Grid

For a realistic system, the discretization must reflect the geometrical and material properties of the system; thus, we start from the more general Equation (2), where $\alpha, k, c$ and $\rho$ are not considered uniform in space. Using Formula (3) for a one dimensional, equidistant grid, we have

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

Now we change to cell variables, where the indices refer to whole cells:

$$
\frac{d u_{\mathrm{i}}}{d t}=\frac{A}{c_{\mathrm{i}} \rho_{\mathrm{i}} A \Delta x}\left(k_{\mathrm{i}, \mathrm{i}+1} \frac{u_{\mathrm{i}+1}-u_{\mathrm{i}}}{\Delta x}+k_{\mathrm{i}-1, \mathrm{i}} \frac{u_{\mathrm{i}-1}-u_{\mathrm{i}}}{\Delta x}\right)+Q_{\mathrm{i}} .
$$

Here $u_{\mathrm{i}}$ is the temperature of the cell $i$, is the heat capacity of the cell in units $[\mathrm{J} / \mathrm{K}], \Delta x$ is the length, $V=A \Delta$ is the volume while $m$ is the mass of the cell. We introduce two other quantities, the heat source term $Q$ of the cells:

$$
Q_{\mathrm{i}}=\frac{1}{V_{\mathrm{i}}} \int_{V_{\mathrm{i}}} q d V, \text { in }\left[\frac{K}{s}\right] \text { units, }
$$

and the thermal resistance $R_{\mathrm{ij}}=\frac{\Delta x}{k_{\mathrm{ij}} A}$ in $[K / W]$ units. In the case of an irregular grid, the distances between the cell-centres are $d_{\mathrm{ij}}=\left(\Delta x_{\mathrm{i}}+\Delta x_{\mathrm{j}}\right) / 2$ and the resistances can be approximated as $R_{\mathrm{ij}} \approx \frac{d_{\mathrm{ij}}}{k_{\mathrm{ij}} A_{\mathrm{ij}}}$. Using these notations we have:

$$
\frac{d u_{\mathrm{i}}}{d t}=\frac{u_{\mathrm{i}-1}-u_{\mathrm{i}}}{R_{\mathrm{i}-1, \mathrm{i}} C_{\mathrm{i}}}+\frac{u_{\mathrm{i}+1}-u_{\mathrm{i}}}{R_{\mathrm{i}+1, \mathrm{i}} C_{\mathrm{i}}}+Q_{\mathrm{i}}
$$

This can be generalised easily to obtain the ODE system for a general (possibly unstructured) grid, which gives the time derivative of each temperature independently of any coordinate-system:

$$
\frac{d u_{\mathrm{i}}}{d t}=\sum_{\mathrm{j} \neq \mathrm{i}} \frac{u_{\mathrm{j}}-u_{\mathrm{i}}}{R_{\mathrm{i}, \mathrm{j}} C_{\mathrm{i}}}+Q_{\mathrm{i}}
$$

Formally we can write this equation system into the same matrix-form as in (4). Now an off-diagonal $m_{\mathrm{ij}}=1 /\left(R_{\mathrm{ij}} C_{\mathrm{i}}\right)$ element of the $M$ matrix can be nonzero only if the cells i and $j$ are neighbours, i.e., there is direct heat conduction between them. Using this fact we can write the diagonal elements of the matrix as:

$$
m_{\mathrm{ii}}=\sum_{\mathrm{j} \in \text { neighbours }(\mathrm{i})} \frac{-1}{R_{\mathrm{i}, \mathrm{j}} C_{\mathrm{i}}} \doteq \frac{-1}{\tau_{\mathrm{i}}}
$$

After this generalization of the system matrix and time constant $\tau$ we can immediately use not only the CN algorithm (6) but the Formulas (9)-(12) as well for the CpC method.

## 3. The Properties of the Methods for the Heat Equation

The new CN and CpC methods can be used to solve general ODE systems as well as the special ODE systems obtained by the semidiscretization of PDEs. In the following pages, we will consider both aspects, but we certainly do not state that they can be recommended for any kind of ODE systems and PDEs. First, we collect those properties of the methods which are obvious and do not require to be proved:

- They are explicit; we use matrices only to store data in a structured way, but do not perform operations with them and in the loop for the time steps, we use only vectors, not matrices. It also implies that the methods are easily parallelizable.
- They are one-step methods. When the $u_{\mathrm{i}}^{n+1}$ values are calculated, only the values $u_{j}^{n}$ at the beginning of the current timestep are used. This implies that the methods are self-starting, the step size can be changed without any difficulty and the memory requirements are quite low.
- They can be easily applied for any space dimensions, unstructured grid or inhomogeneous heat conduction medium provided that the heat capacity of the cells and the thermal resistance between the cells can be calculated.
At this point, we have to underline again that these new methods may not be considered as Finite Difference Methods. Here time derivatives are not approximated by finite
difference formulas as in FDM. Because of the exponential terms, our methods might also seem to be similar to the so-called generalized Runge-Kutta Formula [40] or exponential integrators [41]. However, those methods use matrix exponentials, while we do not even need to use matrices during the calculation; thus, our methods are fundamentally different from those.


### 3.1. Convergence

In this section, we analyse the convergence properties of the methods as solvers for the spatially discretized systems. In Section 3.3 these will be examined regarding the methods as PDE solvers.

Theorem 1. The new CpC method, defined by Equations (9)-(12), is a second order numerical method for arbitrary nonzero $p \in \mathbb{R}$ and for the general:

$$
\begin{equation*}
\frac{d \vec{u}}{d t}=M \vec{u}+\vec{Q}, \quad \vec{u}(\mathrm{t}=0)=\vec{u}(0) \tag{13}
\end{equation*}
$$

linear ODE initial value problem, where $M$ is an arbitrary nonsingular constant matrix, while $\vec{Q}$ and $\vec{u}(0)$ are arbitrary vectors.

Proof of Theorem 1. To help the reader follow the argument, we present the matrix form of the ODE system:

$$
\frac{d}{d t}\left(\begin{array}{c}
u_{1} \\
\vdots \\
u_{\mathrm{N}}
\end{array}\right)=\left(\begin{array}{ccc}
m_{11} & \ldots & m_{1 \mathrm{~N}} \\
\vdots & \vdots & \vdots \\
m_{\mathrm{N} 1} & \ldots & m_{\mathrm{NN}}
\end{array}\right)\left(\begin{array}{c}
u_{1} \\
\vdots \\
u_{\mathrm{N}}
\end{array}\right)+\left(\begin{array}{c}
Q_{1} \\
\vdots \\
Q_{\mathrm{N}}
\end{array}\right)
$$

The exact solution of the initial value problem (13) can be written as follows:

$$
\begin{aligned}
\vec{u}(t) & =\mathrm{e}^{M t} \vec{u}(0)+\left(\mathrm{e}^{M t}-1\right) M^{-1} \vec{Q} \\
& =\left(1+M t+M^{2} \frac{t^{2}}{2}+M^{3} \frac{t^{3}}{3!}+\ldots\right) \vec{u}(0)+\left(t+M \frac{t^{2}}{2}+M^{2} \frac{t^{3}}{3!}+\ldots\right) \vec{Q} .
\end{aligned}
$$

Without the loss of generality we examine only $u_{1}$ at the first time step. It implies that we use the initial values $u_{\mathrm{j}}(0)$ at the beginning of the time step. The $0^{\text {th }}$ and first order terms in the exact solution at $t=h$ are the following

$$
\begin{equation*}
u_{1}(h)=u_{1}(0)\left(1+m_{11} h\right)+h \sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}(0)+Q_{1} h . \tag{14}
\end{equation*}
$$

The second order terms:

$$
\begin{equation*}
u_{1}(h)=\frac{h^{2}}{2} \sum_{\mathrm{j}=1}^{\mathrm{N}} m_{1 \mathrm{j}} \sum_{\mathrm{k}=1}^{\mathrm{N}} m_{\mathrm{jk}} u_{\mathrm{k}}(0)+\frac{h^{2}}{2} \sum_{\mathrm{j}=1}^{\mathrm{N}} m_{1 \mathrm{j}} Q_{\mathrm{j}} \tag{15}
\end{equation*}
$$

When we apply the CpC method for the general initial value problem (13), at the first stage we obtain:

$$
u_{1}^{\mathrm{Cp}}(p h)=u_{1}(0) \mathrm{e}^{m_{11} p h}+\frac{a_{1}}{m_{11}}\left(\mathrm{e}^{m_{11} p h}-1\right)
$$

where

$$
a_{1}=\sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}(0)+Q_{1} .
$$

The result of the second stage in the CpC method is the following:

$$
u_{1}^{\mathrm{CpC}}(h)=u_{1}(0) e^{m_{11} h}-\frac{a_{1}^{\mathrm{comb}}}{m_{11}}\left(1-e^{m_{11} h}\right)
$$

The first term, up to second order is simple:

$$
\begin{equation*}
u_{1}(0)\left(1+m_{11} h+\frac{m_{11}^{2} h^{2}}{2}+\ldots\right) \tag{16}
\end{equation*}
$$

To examine the second term we have to do the substitutions using (11) and (10):

$$
\begin{aligned}
& -\frac{1}{m_{11}}\left(\sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}^{\mathrm{comb}}+Q_{1}\right)\left(1-e^{m_{11} h}\right) \\
& =-\frac{1}{m_{11}}\left(\sum_{\mathrm{j}>1} m_{1 \mathrm{j}}\left[\left(1-\frac{1}{2 p}\right) u_{\mathrm{j}}(0)+\frac{1}{2 p} u_{\mathrm{j}}^{\mathrm{pred}}\right]+Q_{1}\right)\left(1-e^{m_{11} h}\right) \\
& =-\frac{1}{m_{11}}\left(\sum_{\mathrm{j}>1} m_{1 \mathrm{j}}\left[\left(1-\frac{1}{2 p}\right) u_{\mathrm{j}}(0)+\frac{1}{2 p}\left\{u_{\mathrm{j}}(0) \mathrm{e}^{m_{j j} p h}+\frac{1}{m_{j j}}\left(\sum_{\mathrm{k} \neq j} m_{\mathrm{jk}} u_{k}(0)+Q_{\mathrm{j}}\right)\left(\mathrm{e}^{m_{j j} p h}-1\right)\right\}\right]+Q_{1}\right) \times\left(1-e^{m_{11} h}\right)
\end{aligned}
$$

which can be written as:

$$
\left(\sum_{\mathrm{j}>1} m_{1 \mathrm{j}}\left[\left(1-\frac{1}{2 p}\right) u_{\mathrm{j}}(0)+\frac{1}{2 p}\left\{\begin{array}{l}
u_{\mathrm{j}}(0)\left(1+m_{j j} p h+\frac{\left(m_{j j} p h\right)^{2}}{2}+\ldots\right)+\frac{1}{m_{j j}} \times \\
\times\left(\sum_{\mathrm{k} \neq j} m_{\mathrm{jk}} u_{k}(0)+Q_{\mathrm{j}}\right)\left(m_{j j} p h+\frac{\left(m_{j j} p h\right)^{2}}{2}+\ldots\right)
\end{array}\right\}\right]+Q_{1}\right) \times \frac{\left(m_{11} h+\frac{\left(m_{11} h\right)^{2}}{2}+\ldots\right)}{-m_{11}} .
$$

As the last term contains terms at least first order, we can omit the second order terms in the large bracket and obtain:

$$
\begin{align*}
& \left(\sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}(0)\left(1-\frac{1}{2 p}+\frac{1+m_{\mathrm{j} j} p h}{2 p}+\ldots\right)+\frac{1}{2 p} \sum_{\mathrm{j}>1} m_{1 \mathrm{j}}\left(\sum_{\mathrm{k} \neq \mathrm{j}} m_{\mathrm{jk}} u_{\mathrm{k}}(0)+Q_{\mathrm{j}}\right) p h+Q_{1}\right)\left(h+m_{11} \frac{h^{2}}{2}+\ldots\right) \\
& +O\left(h^{3}\right)=h \sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}(0)+Q_{1} h+\frac{h^{2}}{2} \sum_{\mathrm{j}>1} m_{1 \mathrm{j}} m_{\mathrm{jj}} u_{\mathrm{j}}(0)+\frac{h^{2}}{2} m_{11} \sum_{\mathrm{j}>1} m_{1 \mathrm{j}} u_{\mathrm{j}}(0)  \tag{17}\\
& +\frac{h^{2}}{2} \sum_{\mathrm{j}>1} m_{1 \mathrm{j}} \sum_{\mathrm{k} \neq \mathrm{j}} m_{\mathrm{jk}} u_{\mathrm{k}}(0)+\frac{h^{2}}{2} \sum_{\mathrm{j}>1}^{\mathrm{N}} m_{1 \mathrm{j}} Q_{\mathrm{j}}+\frac{h^{2}}{2} m_{11} Q_{1}+O\left(h^{3}\right) .
\end{align*}
$$

Notice that the parameter $p$ has been completely cancelled out. Now we can easily see that the zeroth and first order terms in (16) and (17) together are the same as in (14). Let us collect the second order terms:
which is equivalent to (15), i.e., the CpC method is (at least) 2 nd order.

### 3.2. Stability

We recall that when the stability of the ODE solvers are examined, typically the following scalar equation is used

$$
\frac{d u}{d t}=-\lambda u, \quad \lambda \geq 0
$$

Absolute stability means that arbitrarily large time steps yield a bounded solution for this equation [42] and [43] (p. 13). It is well-known that the stability function of any explicit

Runge-Kutta method is a polynomial, so they can never be A-stable, which limits their use for solving PDEs ([43], p. 24). Now, for this scalar equation, i.e., in the case of one cell, our methods can be defined only in one meaningful way: To provide the analytical solution. This means that in this sense, they are trivially A-stable. We therefore examine the stability when $u$ is a vector, as it is the case for spatially discretized PDEs. Unlike Runge-Kutta methods, Formulas (6), (9) and (12) use an exponential expression of the time step size $h$ to calculate the new values of the variable $u$. At the first stage, Equation (9) or Stage 1 in Algorithm 1 the temperature $u_{\mathrm{i}}$ of each cell tends to $a_{\mathrm{i}} \tau_{\mathrm{i}}$ with increasing $h$, which is the weighted average of the temperatures of the neighbouring cells at the beginning of the time step. At the second stage, $u_{\mathrm{i}}$ tends to the combination of the beginning temperatures of the neighbours and the neighbours of the neighbours with increasing $h$. We will formulate this statement in Theorem 2, but first we evoke the following simple lemma, the associativity of convex combinations ([44], p. 28) on which we build during the proof of the theorem.

Lemma 1. A convex combination Lemma: A convex combination $x=\sum a_{i} x_{i}$ of convex combinations $x_{\mathrm{i}}=\sum b_{\mathrm{ij}} y_{\mathrm{ij}}$ is still a convex combination:

$$
x=\sum \sum\left(a_{\mathrm{i}} b_{\mathrm{ij}}\right) y_{\mathrm{ij}}
$$

for any $y_{\mathrm{ij}} \in \mathbb{R}^{\mathrm{n}}$.
Theorem 2. In the case of the heat equation, if the $Q_{i}$ source terms are zero, the new values $u_{i}^{C p C}$ are the convex combination of the initial values $u_{j}(0), j=1, \ldots, N$ for any $p \geq \frac{1}{2}$.

Proof of Theorem 2. We will use the facts that $\tau_{\mathrm{i}}=-1 / m_{\mathrm{ii}}$ and $m_{\mathrm{ij}, \mathrm{j} \neq \mathrm{i}}$ are non-negative quantities; thus, $0<\mathrm{e}^{-h / \tau_{\mathrm{i}}}=\mathrm{e}^{m_{\mathrm{ij}} h} \leq 1$ holds because of physical reasons, namely the second law of thermodynamics. Again, without the loss of generality, we examine only $u_{1}$ at the first time step.
(A) Let us examine the first stage:

$$
\begin{equation*}
u_{1}^{C p}=u_{1}(0) \cdot \mathrm{e}^{-\frac{p h}{\tau_{1}}}+a_{1} \tau_{1} \cdot\left(1-\mathrm{e}^{-\frac{p h}{\tau_{1}}}\right) \tag{18}
\end{equation*}
$$

The term $a_{1} \tau_{1}$ is the following:

$$
a_{1} \tau_{1}=\frac{-1}{m_{11}} \sum_{\mathrm{j}>1} m_{\mathrm{ij}} u_{\mathrm{j}}(0) .
$$

Let us calculate the coefficients of the $u_{\mathrm{j}}(0)$ initial values:

$$
\frac{-m_{1 \mathrm{j}}}{m_{11}}=\frac{1}{C_{1} R_{1 \mathrm{j}}} \frac{1}{\sum_{\mathrm{k}>1} \frac{1}{C_{1} R_{1 \mathrm{k}}}}=\frac{\frac{1}{R_{1 \mathrm{j}}}}{\sum_{\mathrm{k}>1} \frac{1}{R_{1 \mathrm{k}}}}
$$

One can see that they are nonnegative and their sum is one, which implies that $a_{1} \tau_{1}$ is a convex combination $u_{\mathrm{j}}(0)$. Now it is clear that the coefficients $\mathrm{e}^{-\frac{p h}{\tau_{1}}}$ and $1-\mathrm{e}^{-\frac{p h}{\tau_{1}}}$ in (18) are also nonnegative and their sum is one, and therefore $u_{1}^{\mathrm{Cp}}$ is also a convex combination of the initial values $u_{j}(0), j=1, \ldots, N$.
(B) The final solution at the end of the first time step is:

$$
u_{1}^{\mathrm{CpC}}=u_{1}(0) \cdot \mathrm{e}^{-\frac{h}{\tau_{1}}}+a_{1}^{\mathrm{comb}} \tau_{1} \cdot\left(1-\mathrm{e}^{-\frac{h}{\tau_{1}}}\right),
$$

where

$$
a_{1}^{\mathrm{comb}} \tau_{1}=\frac{-1}{m_{11}} \sum_{\mathrm{j}>1} m_{1 \mathrm{j}}\left[\left(1-\frac{1}{2 p}\right) u_{\mathrm{j}}(0)+\frac{1}{2 p} u_{\mathrm{j}}^{\mathrm{Cp}}\right]
$$

The square bracket is a convex combination if $0 \leq \frac{1}{2 p} \leq 1$ and $0 \leq 1-\frac{1}{2 p} \leq 1$, which holds if $\frac{1}{2} \leq p$. As we have seen at point A) the factors $\frac{-m_{1 j}}{m_{11}}$ are nonnegative and their sum is one; thus, $u_{1}^{\mathrm{CpC}}$ is a convex combination of the initial values.

Theorem 2 implies that any error produced during the calculation will be distributed quickly and cannot be amplified; thus, the methods are stable for the heat equation (and as we will show later, for Fisher's equation). Moreover, in the homogeneous $(Q=0)$ case, Theorem 2 guarantees that the solution follows the maximum and minimum principles ([43], p. 87), i.e., the extreme values of $u$ occur among the initial or the prescribed boundary values. This means that starting from positive initial values and assuming no heat sinks are present, the variables always remain positive. This property is especially advantageous when Equations (1) and (2) are used to model physical processes, where the temperature measured in Kelvin (or analogous variables such as concentrations of chemical species) cannot be negative.

### 3.3. Analysis of the Errors

We analytically calculated the first two terms of the error of the CpC scheme for a system of 2 identical cells. At the end of the first time step, the error for the first cell is the following:
$u_{1}^{\text {exact }}(h)-u_{1}^{\text {num }}(h)=\frac{1+p}{4 \tau^{3}}\left(u_{2}(0)-u_{1}(0)\right) h^{3}+\frac{1+3 p}{12 \tau^{2}} Q_{2} h^{3}+\frac{1}{6 \tau^{2}}\left(Q_{1}-Q_{2}\right) h^{3}+O\left(h^{4}\right)$.
We note that we obtained similar expressions for larger systems and for the higher order error terms as well, but of course with more terms and with larger exponents. We can draw two conclusions.

First, for not too large time steps, the error is smaller for smaller parameter values $p$, so $p$ should be chosen as small as possible. Unfortunately, this advice contradicts the requirement of stability, which is not guaranteed for $p<0.5$. We will return to this question during the discussion of the numerical experiments.

For the second consequence we recall that $\tau \sim \Delta x^{2}$, which means that for small $\Delta x$, the main error terms contain powers of $\frac{h}{\Delta x^{2}}$. We have to conclude that the methods are only conditionally consistent in the sense that when the size of the spatial cells $\Delta x$ tends to zero, the time step size $h$ also has to go to zero faster than the second power of $\Delta x$. If $\Delta x \rightarrow 0$ while $h /(\Delta x)^{2}$ is kept constant, then the global errors tend to some constant values, and if one decreases $h$ faster, the errors tend to zero. However, if $h$ is decreased slower, the errors are growing, but not without limits: They approach their maximum possible values which are restricted by the above mentioned maximum and minimum principles. It implies that considering as solvers for the original PDE, the methods are conditionally convergent. However, as we have proved above, they are unconditionally convergent for any fixed spatial grid, i.e., the solution converges to the exact solution of the ODE system obtained by the discretization of space variables when the time step size $h$ tends to zero. It is worth mentioning that this second consequence holds for the CN method as well.

## 4. Comparison with Exact Results

We note that all simulations are performed using the MATLAB R2019b software on a desktop computer with an Intel Core i3-8100 as CPU.

### 4.1. The Heat Equation

We solve PDE (1) on the interval $x \in[0,1]$ with the following heat source term:

$$
q(x, t)=q_{0} \sin (\pi x)
$$

where $q_{0}=2$ and $\alpha=1$. The initial condition is also a sine function but with smaller wavelength:

$$
u(x, t=0)=\sin (3 \pi x)
$$

and we use zero Dirichlet boundary conditions:

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

It is easy to check that the analytical solution of this problem is:

$$
u(x, t)=\frac{q_{0}}{\pi^{2}} \sin (\pi x)\left(1-\mathrm{e}^{-\pi^{2} t}\right)+\sin (3 \pi x) \mathrm{e}^{-9 \pi^{2} t}
$$

We discretize $x$ by setting:

$$
x_{\mathrm{j}}=j \Delta x, j=0, \ldots, 100, \quad \Delta x=0.01
$$

The solutions are examined and compared at final time $t_{\text {fin }}=0.2$. We define the (global) error as the maximum of the absolute value of the difference between the exact temperature $u_{\mathrm{i}}^{\text {exact }}$ and the temperature $u_{\mathrm{i}}^{\text {num }}$ obtained by our numerical methods at the end of the examined time interval:

$$
\begin{equation*}
\text { Error }=\max _{\mathrm{i}}\left|u_{\mathrm{i}}^{\text {exact }}\left(t_{\text {fin }}\right)-u_{\mathrm{i}}^{\text {num }}\left(t_{\text {fin }}\right)\right| . \tag{20}
\end{equation*}
$$

In Figure 1, one can see these global errors as a function of time step size $h$. One can see that the error tends to a small nonzero value in all cases. This non-vanishing error is caused by the discretization of the space variable. More precisely, it is the truncation errors ([43], p. 7)

$$
\begin{equation*}
\varepsilon_{\mathrm{i}}=-\frac{\Delta x^{2}}{12} f^{\prime \prime \prime \prime}\left(\eta_{\mathrm{i}}\right) \tag{21}
\end{equation*}
$$

of the central difference Formula (3) accumulated during the subsequent time steps. In Figure 2, we illustrate the behaviour of the maximum errors for different space steps and for $p=0.5$.

The minimum of these functions appears in the form of a cusp in each case. The reason for this is that the largest error terms (19) and (21), coming from the time-integration method and the spatial discretization, respectively, are of opposite signs and cancel each other at these special, optimal time step size and grid spacing combinations. Unfortunately, we observed this behaviour only when the source term $Q$ strongly dominates the effect of the (non-uniform) initial conditions. Moreover, the location of the cusp, i.e., the optimal $h$ depends on so many factors that this favourable property can hardly be exploited in practical problems.


Figure 1. Maximum errors defined by formula (20) as a function of the time step size for our constant-neighbour ( CN ) and two-stage CpC methods in the case of the heat equation.


Figure 2. Maximum errors defined by Formula (20) as a function of the time step size in the case of our different grid spacings $\Delta x$ for the two-stage combined $\mathrm{C}^{1 / 2} \mathrm{C}$ methods applied for the heat equation.

### 4.2. The Nonlinear Fisher's Equation

We consider the following equation

$$
\begin{equation*}
\frac{\partial u}{\partial t}=\alpha \nabla^{2} u+\beta u(1-u) \tag{22}
\end{equation*}
$$

with $\alpha=1$ and $\beta>0$, subject to the following initial condition:

$$
u(x, t=0)=\left(1+e^{\sqrt{\frac{\beta}{6}} x}\right)^{-2}
$$

The analytical solution of this problem is the following travelling wave function [45,46]

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

The appropriate Dirichlet boundary conditions are prescribed at the two ends of the interval:

$$
u\left(x=x_{0}, t\right)=\left(1+e^{\sqrt{\frac{\beta}{6}} x_{0}-\frac{5}{6} \beta t}\right)^{-2}, \quad u\left(x=x_{\mathrm{fin}}, t\right)=\left(1+e^{\sqrt{\frac{\beta}{6}} x_{\mathrm{fin}}-\frac{5}{6} \beta t}\right)^{-2}
$$

The values of the exact solution obviously lie in the unit interval, $u(x, t) \in[0,1], x, t \in \mathbb{R}$. We take the nonlinear term into account in a "semi-implicit" way:

$$
u_{\mathrm{i}}^{\mathrm{n}+1}=u_{\mathrm{i}}^{\mathrm{n}+1, \text { pred }}+\beta u_{\mathrm{i}}^{\mathrm{n}+1, \operatorname{pred}}\left(1-u_{\mathrm{i}}^{\mathrm{n}+1}\right) h .
$$

where $u_{\mathrm{i}}^{\mathrm{n}+1, \mathrm{pred}}$ is the result of the first and the second stage using the same formulas as in the previous section, i.e., with taking into account the effect of the diffusion term only. Note that now the new value $u_{\mathrm{i}}^{\mathrm{n}+1}$ appeared on the right hand side of the equation as well. However, this equation can be rearranged into a fully explicit form:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{n}+1}=\frac{1+\beta h}{1+\beta h u_{\mathrm{i}}^{\mathrm{n}+1, \text { pred }}} u_{\mathrm{i}}^{\mathrm{n}+1, \text { pred }} \tag{23}
\end{equation*}
$$

This operation is performed in a separate third stage after Stage 2 (a separate second stage after the first one in case of the CN method), where a loop is going through all the nodes. We have done several tests with this semi-implicit treatment and obtained that the methods behave similarly well as in the linear case. The errors as a function of the time step size $h$ are presented in Figure 3 for $\beta=2.5, x_{0}=0, x_{\mathrm{fin}}=4, t_{\mathrm{fin}}=2$, and $\Delta x=0.01$.

Remark 1. Suppose that $0 \leq u_{\mathrm{i}}^{\mathrm{n}+1, \text { pred }} \leq 1$. Now from Equation (23), it is obvious that the new value $u_{\mathrm{i}}^{\mathrm{n}+1}$ cannot be negative, and also cannot exceed 1 ; thus, $0 \leq u_{\mathrm{i}}^{\mathrm{n}+1} \leq 1$.

Corollary 1. In the case of Fisher's equation (22) if the initial values are in the interval $u_{\mathrm{i}}^{0} \in[0,1]$, then the values of $u$ remain in this interval for arbitrary $\beta$ and time step size $h$, for the whole calculation. This is a special case of the maximum and minimum principles, which implies that the CN and the CpC methods supplemented with (23) is unconditionally stable for Fisher's equation.

Proof. Theorem 2 and Remark 1 immediately imply the statement.
However, we do not state that we have found the optimal algorithm to solve Fisher's equation. The purpose of this subsection is to demonstrate that these methods can be successfully applied for nonlinear equations as well. The systematic investigation of the application possibilities of the methods for nonlinear equations is clearly out of the scope of this paper and planned to be published in a series of papers in the future.


Figure 3. Maximum errors defined by Formula (23) as a function of the time step size in the case of our constant-neighbour ( CN ), the two-stage CpC methods for the nonlinear Fisher's equation.

## 5. Comparison with Numerical Results

In this section rectangle-shaped lattices are examined with $N=N_{\mathrm{x}} \times N_{\mathrm{z}}$ cells, where the edge of the system is thermally isolated regarding conductive type heat transfer, which means closed (zero Neumann) boundary conditions. To help the reader to imagine the system, we present the arrangement of the variables in Figure 4. We emphasize that the size and the shape of the cells are not necessarily identical.


Figure 4. Arrangement of the examined system. The outer double line represents thermal isolation and the inner thin lines represent cell borders. The red double arrows symbolize conductive heat transport through the resistances $R_{\mathrm{ij}}$ while $Q_{N}$ is an external heat source.

To obtain a reference solution, we used the implicit ode15s solver of MATLAB, which is a variable-step, variable-order solver based on the numerical differentiation formulas (NDFs) of orders 1 to 5 , where the letter s indicates that the codes were suggested to use in the case of stiff systems. During this calculation, we applied stringent error tolerance ('RelTol' and 'AbsTol' were both $10^{-16}$ ). We compare the performance of the new CN and CpC methods with professionally coded and widely used MATLAB routines. In addition to ode15s, the following solvers are employed, with loose error tolerance to increase the speed:

- ode45, the explicit Runge-Kutta-Dormand-Prince formula of order 4(5).
- ode23, the explicit Runge-Kutta-Bogacki-Shampine method of order 2(3).
- ode23t, an implementation of the trapezoidal rule using a "free" interpolant .
- ode23s, a modified Rosenbrock formula of order 2.

Besides the above listed MATLAB solvers, we have coded the previously mentioned unconditionally positive finite-difference (UPFD) method of Chen-Charpentier et al. [27] for comparison purposes because of its simplicity. Applying that scheme to (1) we obtain:

$$
\frac{u_{\mathrm{i}}^{\mathrm{n}+1}-u_{\mathrm{i}}^{\mathrm{n}}}{h}=\frac{\alpha}{\Delta x^{2}}\left(u_{\mathrm{i}-1}^{\mathrm{n}}-2 u_{\mathrm{i}}^{\mathrm{n}+1}+u_{\mathrm{i}+1}^{\mathrm{n}}\right)+Q_{\mathrm{i}}
$$

This formula can be rewritten in the following explicit form:

$$
u_{\mathrm{i}}^{\mathrm{n}+1}=\left(\frac{u_{\mathrm{i}}^{\mathrm{n}}}{h}+\frac{\alpha}{\Delta x^{2}}\left(u_{\mathrm{i}-1}^{\mathrm{n}}+u_{\mathrm{i}+1}^{\mathrm{n}}\right)+Q_{\mathrm{i}}\right)\left(\frac{1}{h}+\frac{2 \alpha}{\Delta x^{2}}\right)^{-1}
$$

which can be generalized in the manner we explained in Section 2.3 , so the UPFD formula we use:

$$
\begin{equation*}
u_{\mathrm{i}}^{\mathrm{n}+1}=\frac{\frac{u_{\mathrm{i}}^{\mathrm{n}}}{h}+\sum_{\mathrm{j} \neq i} \frac{u_{\mathrm{i}}^{\mathrm{n}}}{C_{\mathrm{i}} R_{\mathrm{ij}}}+Q_{\mathrm{i}}}{\frac{1}{h}+\frac{1}{\tau_{\mathrm{i}}}} \tag{24}
\end{equation*}
$$

Here we define the (global) error in the same way as in (20) but now we have a reference solution instead of the exact analytical solution:

$$
\begin{equation*}
\text { Error }=\max _{\mathrm{i}}\left|u_{\mathrm{i}}^{\mathrm{ref}}\left(t_{\mathrm{fin}}\right)-u_{\mathrm{i}}^{\mathrm{num}}\left(t_{\mathrm{fin}}\right)\right| . \tag{25}
\end{equation*}
$$

### 5.1. First Case: A Very Stiff System

We set $N_{x}=100, N_{z}=40$; thus, the number of cells is $N=4000$. Different random values are given to the heat capacities and to the thermal resistances:

$$
C_{\mathrm{i}}=10^{(2-4 \times r a n d)}, \quad R_{x, \mathrm{i}}=10^{(3-6 \times r a n d)}, \quad R_{z, \mathrm{i}}=10^{(3-6 \times r a n d)}
$$

where rand is a random number generated by the MATLAB uniformly in the $(0,1)$ interval for each quantity. This means that the capacities (the resistances) follow a log-uniform distribution between 0.01 and 100 (between 0.001 and 1000). The initial temperatures have a uniform distribution between 0 and 1, i.e., $u_{\mathrm{i}}(0)=$ rand, while the source-terms have a uniform distribution between -0.5 and 0.5 . Starting from $t_{0}=0 \mathrm{~s}$ we solve this system for the temperatures at final time $t_{\text {fin }}=1 \mathrm{~s}$.

As the system is thermally isolated, the system matrix has a zero eigenvalue, all other eigenvalues are negative. If we denote the eigenvalues with the largest (smallest) absolute value with $\lambda_{\text {MAX }}\left(\lambda_{\text {MIN }}\right)$, then the stiffness ratio can be calculated as

$$
\frac{\lambda_{\mathrm{MAX}}}{\lambda_{\mathrm{MIN}}}=1.8 \times 10^{9}
$$

For the explicit Euler method, the maximum possible time step size is:

$$
h_{\mathrm{MAX}}^{\mathrm{EE}}=\left|\frac{2}{\lambda_{\mathrm{MAX}}}\right|=1.3 \times 10^{-5} \mathrm{~s}
$$

above this threshold instability necessarily appears. In Figure 5, we present the errors for the one-stage CN and the two-stage CpC methods in the case of different $p$ parameters as a function of the step size $h$. From the figure one can see that for small $h$, the global error decreases with the first power of the step size in the case of the Constant-neighbour methods and with the second power for the CpC methods, which confirms that the methods are first and second order, respectively. It can be seen that for $p=1 / 3$ the method is unstable in the region of large step sizes, but in the region of small $h$-s it is slightly more accurate than for
larger $p$-s. For $p=1 / 2$ the error-function has a relative elevation around $h=10^{-2} \ldots 10^{-3}$, because this parameter value is at the border of the stability region.

We also examine the global errors as a function of the total running times. The results are presented in Figure 6. For smaller step sizes the running times of several runs have been averaged to avoid random fluctuations due to very short running times.


Figure 5. Maximum error defined by Formula (25) as a function of the time step size $h$ in the case of our constant-neighbour ( CN ), the two-stage CpC methods and the UPFD method defined by (22) applied for the system consisting of $N=4000$ cells described in Section 5.1.

We note that the explicit solvers ode 23 and ode 45 can hardly provide any meaningful results, because for large error tolerance they diverge, while tightening the tolerance yields an error message "array exceeds maximum array size".

In Table 2 we summarize some results obtained by MATLAB routines ode15s, ode23s, ode23, ode23t, the UPFD scheme, and our methods $C_{1 / 2} C$ and $C_{2 / 3} C$. One can see that, if we set the same moderate precision requirements, our methods are much faster than the conventional explicit or implicit methods, even without adaptive stepsize control, parallelization, or vectorization. The UPFD method is also very fast, but its accuracy is far from being optimal.


Figure 6. Maximum errors defined by Formula (25) as a function of the total running times in the case of our constant-neighbour (CN), the two-stage CpC methods with different values of parameter $p$, three different MATLAB solvers and the UPFD method applied for the system consisting of $N=4000$ cells described in Section 5.1.

Table 2. Performance of different solvers for the system consisting of $N=4000$ cells described in Section 5.1. CpC means the new two-stage method. The absolute and the relative tolerance 'RelTol' and 'AbsTol' have been set to the same values denoted by 'tol'. The error is defined by (25).

| Method | Running Time (s) | Error |
| :---: | :---: | :---: |
| ode15s, tol $=200$ | 43.6 | 0.017 |
| ode15s, tol $=0.03$ | 51.1 | 0.0017 |
| ode23s, tol $=200$ | 608 | $5.2 \times 10^{-4}$ |
| ode23, tol $=0.5$ | 2039 | 0.496 |
| ode23t, tol $=200$ | 45.7 | 0.067 |
| ode23t, tol $=1$ | 50.5 | 0.0014 |
| $C_{2 / 3} C, h=5 \times 10^{-4}$ | 0.3 | 0.06 |
| $C_{1 / 2} C, h=2 \times 10^{-5}$ | 7.55 | $3.3 \times 10^{-4}$ |
| $C_{1 / 2} C, h=5 \times 10^{-6}$ | 30.2 | $2.7 \times 10^{-5}$ |
| UPFD, $h=10^{-5}$ | 6.51 | 0.0045 |

### 5.2. Second Case: A Highly Anisotropic System

Now $N_{x}=100, N_{z}=100$; thus, $N=10,000$. The capacities and the thermal resistances followed a log-uniform distribution again:

$$
C_{\mathrm{i}}=10^{(1-2 \text { rand })}, \quad R_{\mathrm{x}, \mathrm{i}}=10^{(3-2 \text { rand })}, \quad R_{\mathrm{y}, \mathrm{i}}=10^{(-1-2 \text { rand })} .
$$

Note that, on average, there is four orders of magnitude difference between the resistances in the $x$ and in the $z$ direction. The initial temperatures are $u_{\mathrm{i}}(0)=2$ rand -1 while the source-terms are $Q_{i}=$ rand -0.5 . The task is to solve this initial value problem for the temperatures at $t_{\text {fin }}=1 \mathrm{~s}$ starting from $t_{0}=0 \mathrm{~s}$.

The stiffness ratio is rather high, $2.15 \times 10^{9}$. The maximum possible time step size for the explicit Euler method is $h_{\mathrm{MAX}}^{\mathrm{EE}}=9.6 \times 10^{-5} \mathrm{~s}$, a few times larger than in the first case.

In Figure 7, we present the error for the CN and the CpC methods as a function of the step size $h$.

We also present the global errors as a function of the total running times in Figure 8. In this case, due to the larger $h_{\mathrm{MAX}}^{\mathrm{EE}}$, the explicit solvers ode 23 and ode 45 are able to produce some results, albeit still in a relatively narrow region of tolerance values 'RelTol' and 'AbsTol'.


Figure 7. Maximum errors defined by Formula (25) as a function of the time step size in the case of our constant-neighbour ( CN ), the two-stage CpC methods, and the UPFD method applied for the system consisting of $N=10,000$ cells described in Section 5.2.

One can see that our methods are much faster than the conventional explicit or implicit methods. One might think that the two-stage $\mathrm{C}^{1 ⁄ 2} \mathrm{C}$ method produces too large errors for medium step sizes, e.g., for $h=5 \times 10^{-4}$. However, in this case the average absolute error is only 0.002 and from Figure 9 one can see that the graph of the produced solutionfunction is almost indistinguishable from the reference curve. This means that our methods can produce acceptable results three orders of magnitude faster than the well-established MATLAB routines.


Figure 8. Maximum errors defined by Formula (25) as a function of the total running time in the case of our constant-neighbour ( CN ), the two-stage CpC methods, three different MATLAB solvers and the UPFD method applied for the system consisting of $N=10,000$ cells described in Section 5.2.

In Table 3, we summarize some results obtained by MATLAB solvers ode23, ode45, ode15s, ode23s, ode23t, the UPFD scheme, and our two-stage CpC methods.

Table 3. Performance and errors of different solvers for the system consisting of $N=10,000$ cells. For other notations, see caption of Table 2.

| Method | Running Time (s) | Error |
| :---: | :---: | :---: |
| ode15s, tol $=200$ | 716 | 0.014 |
| ode15s, tol $=0.03$ | 898 | $1.4 \times 10^{-4}$ |
| ode23s, tol $=1$ | 11,893 | $5 \times 10^{-4}$ |
| ode23, tol $=0.9$ | 1476 | 0.056 |
| ode23t, tol $=2$ | 804 | 0.004 |
| ode45, tol $=2.5$ | 2491 | 0.25 |
| $\mathrm{C}_{1 / 2} \mathrm{C}, h=5 \times 10^{-4}$ | 0.76 | 0.028 |
| $\mathrm{C}_{1 / 2} \mathrm{C}, h=2 \times 10^{-5}$ | 18.9 | $9.9 \times 10^{-5}$ |
| $\mathrm{C}_{1 / 2} \mathrm{C}, h=10^{-6}$ | 379 | $2.6 \times 10^{-7}$ |
| UPFD, $h=2 \times 10^{-5}$ | 8.04 | 0.0042 |



Figure 9. The variable $u$ as a function of the cell index i for the first row (first 100 cells, see Figure 4) in the case of the system consisting of $N=10,000$ cells described in Section 5.2. The red continuous line represents the high-precision solution while the green dotted line with diamond markers are the values produced by our $\mathrm{C}^{1 / 2} \mathrm{C}$ algorithm for $h=5 \times 10^{-4}$ in 0.76 s , orders of magnitude faster than the conventional solvers.

We emphasize that the $N=10,000$ number of cells is still much smaller than in many real-world applications. Since for larger systems, implicit methods have a more serious disadvantage, we think that our methods are even more competitive in those cases. We note that the reason we have not used a larger number of cells for testing purposes is that our computer cannot solve them using the MATLAB routines because of the too large memory requirements.

## 6. Conclusions

The purpose of this paper is to introduce and examine a set of new explicit one-step numerical schemes. We propose these methods to solve the ODE system which is obtained after spatially discretizing diffusion term. We prove that one-stage "constant-neighbour" (CN) method is first order, while the two-stage " CpC " methods are second order for the heat equation in the time step size $h$ for arbitrary parameter values $p$. For $p \geq 0.5$ the variables at the end of the time-step are the convex combinations of the values at the beginning of the time step for the heat equation in the absence of external sources; thus, the methods are stable for arbitrarily large values of the time step size. This holds for the nonlinear Fisher's equation as well, if one treats the nonlinear term as we suggest. These statements have been proved analytically and confirmed by several numerical experiments.

The schemes have been verified by comparisons to analytical solutions of the inhomogeneous heat equation and Fisher's equation. The performance of the methods has been tested in the case of two discretized two-dimensional systems with highly inhomogeneous random parameters and initial conditions. The obtained data show that if quick results are required, the proposed schemes have a significant advantage in speed, even without adaptive stepsize control, parallelization or vectorization. We must add that they are easy to implement and parallelize and they can be applied regardless of space dimensions and
grid irregularity. On the one hand, increasing number of cells makes our methods more competitive against the widely used implicit schemes. On the other hand, if the range of the eigenvalues of the system matrix is wide (which is in connection with stiffness), the new methods have a significant advantage over the traditional explicit schemes, which are only conditionally stable. However, when highly accurate results are required, first and second order methods are not the best choice; thus, we have already started to search for higher-order versions of the new methods.

Author Contributions: Conceptualization and methodology, E.K.; software, Á.N.; validation, E.K., and M.S.; formal analysis, E.K.; investigation, Á.N. and M.S.; resources, E.K.; data curation, Á.N.; writing-original draft preparation, E.K.; writing-review and editing, M.S.; visualization, M.S.; supervision, E.K.; project administration, Á.N. All authors have read and agreed to the published version of the manuscript.
Funding: The research was supported by the ÚNKP-21-3 new national excellence program of the ministry for innovation and technology from the source of the national research, development and innovation fund.

Data Availability Statement: Data is available at the following link: https://github.com/Drendre/ New-CpC-method-dataset/tree/main (accessed on: 15 September 2021).
Conflicts of Interest: The authors declare no conflict of interest.

## References

1. Murray, J.D. Mathematical biology: I. An introduction: Interdisciplinary applied mathematics. In Mathematical Biology; Springer: Berlin/Heidelberg, Germany, 2002.
2. Blaj, G.; Kenney, C.; Segal, J.; Haller, G. Analytical solutions of transient drift-diffusion in pn junction pixel sensors. arXiv 2017, arXiv:1706.01429.
3. Kovács, E.; Majár, J. On some analogous transient phenomena. In Proceedings of the XXXII microCAD International Multidisciplinary Scientific Conference C, Miskolc, Hungary, 5-6 September 2018; Volume 1.
4. Gasparin, S.; Berger, J.; Dutykh, D.; Mendes, N. Stable explicit schemes for simulation of nonlinear moisture transfer in porous materials. J. Build. Perform. Simul. 2018, 11, 129-144. [CrossRef]
5. Fisher, R.A. THE wave of advance of advantageous genes. Ann. Eugen 1937, 7, 355-369. [CrossRef]
6. John, H.; Lienhard, I.; Lienhard, V. A Heat Transfer Textbook; Phlogiston Press: Cambridge, MA, USA, 2008.
7. Appau, P.O.; Dankwa, O.K.; Brantson, E.T. A comparative study between finite difference explicit and implicit method for predicting pressure distribution in a petroleum reservoir. Int. J. Eng. Sci. Technol. 2019, 11, 23-40. [CrossRef]
8. Moncorgé, A.; Tchelepi, H.A.; Jenny, P. Modified sequential fully implicit scheme for compositional flow simulation. J. Comput. Phys. 2017, 337, 98-115. [CrossRef]
9. Chou, C.S.; Zhang, Y.T.; Zhao, R.; Nie, Q. Numerical methods for stiff reaction-diffusion systems. Discret. Contin. Dyn.-Syst.-B 2007, 7, 515. [CrossRef]
10. Liu, H.; Wang, K.; Chen, Z. A family of constrained pressure residual preconditioners for parallel reservoir simulations. Numer. Linear Algebra Appl. 2016, 23, 120-146. [CrossRef]
11. White, J.A.; Castelletto, N.; Klevtsov, S.; Bui, Q.M.; Osei-Kuffuor, D.; Tchelepi, H.A. A two-stage preconditioner for multiphase poromechanics in reservoir simulation. Comput. Methods Appl. Mech. Eng. 2019, 357, 112575. [CrossRef]
12. Lu, C.; Jiao, X.; Missirlis, N. A hybrid geometric+ algebraic multigrid method with semi-iterative smoothers. Numer. Linear Algebra Appl. 2014, 21, 221-238. [CrossRef]
13. Wang, Y.; Killough, J.E. Solver preconditioning using the combinatorial multilevel method on reservoir simulation. Comput. Geosci. 2015, 19, 695-708. [CrossRef]
14. DeCaria, V.; Layton, W.; Zhao, H. A time-accurate, adaptive discretization for fluid flow problems. arXiv 2020, arXiv:1810.06705.
15. Gumel, A.B.; Ang, W.T.; Twizell, E.H. Efficient parallel algorithm for the two-dimensional diffusion equation subject to specification of mass. Int. J. Comput. Math. 1997, 64, 153-163. [CrossRef]
16. Xue, G.; Feng, H. A new parallel algorithm for solving parabolic equations. Adv. Differ. Equ. 2018, 2018, 174. [CrossRef]
17. Gagliardi, F.; Moreto, M.; Olivieri, M.; Valero, M. The international race towards exascale in Europe. CCF Trans. High Perform. Comput. 2019, 1, 3-13. [CrossRef]
18. Yang, X.S. An Introduction to Computational Engineering with Matlab; Cambridge Int Science Publishing: Cambridge, MA, USA, 2006.
19. Van Der Houwen, P.J.; Sommeijer, B.P. On the Internal Stability of Explicit, m-Stage Runge-Kutta Methods for Large m-Values. Zeitschrift Angew. Math. Mech. 1980, 60, 479. [CrossRef]
20. Gourlay, A. Hopscotch: A fast second-order partial differential equation solver. IMA J. Appl. Math. 1970, 6, 375-390. [CrossRef]
21. Gourlay, A.; McGuire, G. General hopscotch algorithm for the numerical solution of partial differential equations. IMA J. Appl. Math. 1971, 7, 216-227. [CrossRef]
22. Schmidt, J.W. VK Saul'yev, Integration of Equations of Parabolic Type by the Method of Nets. (International Series of Monographs in Pure and Applied Mathematics, Volume 54) XVII+ 346 S. m. 26 Fig. Oxford/London/Edinburgh/New York/Paris/Frankfurt 1964. Pergamon Press. Preis geb. 80 s. net. Z. Angew. Math. Und Mech. 1965, 45, 149.
23. Barakat, H.; Clark, J. On the solution of the diffusion equations by numerical methods. J. Heat Transfer. 1966, 88, 421-427. [CrossRef]
24. Evans, D.; Abdullah, A. Group explicit methods for parabolic equations. Int. J. Comput. Math. 1983, 14, 73-105. [CrossRef]
25. Sahimi, M.; Sundararajan, E.; Subramaniam, M.; Hamid, N. The D'Yakonov fully explicit variant of the iterative decomposition method. Comput. Math. Appl. 2001, 42, 1485-1496. [CrossRef]
26. Sun, G.; Liu, G.; Li, M. An efficient explicit finite-difference scheme for simulating coupled biomass growth on nutritive substrates. Math. Probl. Eng. 2015, 2015, 708497. [CrossRef]
27. Chen-Charpentier, B.M.; Kojouharov, H.V. An unconditionally positivity preserving scheme for advection-diffusion reaction equations. Math. Comput. Model. 2013, 57, 2177-2185. [CrossRef]
28. Campbell, L.; Yin, B. On the stability of alternating-direction explicit methods for advection-diffusion equations. Numer. Methods Partial. Differ. Equ. Int. J. 2007, 23, 1429-1444. [CrossRef]
29. Özişik, M.N.; Orlande, H.R.; Colaço, M.J.; Cotta, R.M. Finite Difference Methods in Heat Transfer; CRC Press: Boca Raton, FL, USA, 2017; p. 224.
30. Gourlay, A.; McKee, S. The construction of hopscotch methods for parabolic and elliptic equations in two space dimensions with a mixed derivative. J. Comput. Appl. Math. 1977, 3, 201-206. [CrossRef]
31. Skvortsov, L.M. Explicit stabilized Runge-Kutta methods. Comput. Math. Math. Phys. 2011, 51, 1153-1166. [CrossRef]
32. Verwer, J.G.; Hundsdorfer, W.H.; Sommeijer, B.P. Convergence properties of the Runge-Kutta-Chebyshev method. Numer. Math. 1990, 57, 157-178. [CrossRef]
33. Tavakoli, R.; Davami, P. New stable group explicit finite difference method for solution of diffusion equation. Appl. Math. Comput. 2006, 181, 1379-1386. [CrossRef]
34. Vabishchevich, P.N.; Zakharov, P.E. Explicit-Implicit Splitting Schemes for Parabolic Equations and Systems; International Conference on Numerical Methods and Applications; Springer: Berlin/Heidelberg, Germany, 2014; pp. 157-166.
35. Rousseau, G.; Kapral, R. Asynchronous algorithm for integration of reaction-diffusion equations for inhomogeneous excitable media. Chaos Interdiscip. J. Nonlinear Sci. 2000, 10, 812-825. [CrossRef] [PubMed]
36. Kovács, E. New stable, explicit, first order method to solve the heat conduction equation. J. Comput. Appl. Mech. 2020, 15, 3-13. [CrossRef]
37. Kovács, E. A class of new stable, explicit methods to solve the non-stationary heat equation. Numer. Methods Partial. Differ. Equ. 2021, 37, 2469-2489. [CrossRef]
38. Ferziger, J.H.; Perić, M.; Street, R.L. Computational Methods for Fluid Dynamics; Springer: Berlin/Heidelberg, Germany, 2002; Volume 3.
39. Ralston, A. Runge-Kutta methods with minimum error bounds. Math. Comput. 1962, 16, 431-437. [CrossRef]
40. Hut'a, A.; Strehmel, K. Construction of explicit and generalized Runge-Kutta formulas of arbitrary order with rational parameters. Apl. Mat. 1982, 27, 259-276.
41. Hochbruck, M.; Ostermann, A. Exponential integrators. Acta Numer. 2010, 19, 209-286. [CrossRef]
42. Hairer, E.; Wanner, G. Singular Perturbation Problems and Differential Algebraic Equations. In Solving Ordinary Differential Equations II; Springer: Berlin/Heidelberg, Germany, 1991; pp. 399-545.
43. Holmes, M.H. Introduction to Perturbation Methods; Springer: Berlin/Heidelberg, Germany, 2012; Volume 20.
44. Hiriart-Urruty, J.-E.; Lemaréchal, C. Fundamentals of Convex Analysis; Springer: Berlin/Heidelberg, Germany, 2001.
45. Bastani, M.; Salkuyeh, D.K. A highly accurate method to solve Fisher's equation. Pramana 2012, 78, 335-346. [CrossRef]
46. Agbavon, K.M.; Appadu, A.R.; Khumalo, M. On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term. Adv. Differ. Equ. 2019, 2019, 1-33. [CrossRef]
