Available online at http://scik.org

J. Math. Comput. Sci. 2022, 12:162

https://doi.org/10.28919/jmcs/7407

ISSN: 1927-5307

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

ISSA OMLE[1], ALI HABEEB ASKAR[1,2], ENDRE KOVÁCS[3,*]

[1]Department of Fluid and Heat Engineering, University of Miskolc, 3515 Miskolc, Hungary

[2]Mechanical Engineering Department, University of Technology-Iraq, Baghdad 10066, Iraq

[3]Institute of Physics and Electric Engineering, University of Miskolc, 3515 Miskolc, Hungary

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

Keywords: heat equation; diffusion equation, positivity preserving methods; stiff systems; unconditional stability; hopscotch methods.

2010 AMS Subject Classification: 35K05.

*Corresponding author

E-mail address: fizendre@uni-misksolc.hu

Received April 02, 2022

1. INTRODUCTION

In recent years our group develops novel numerical methods for parabolic partial differential equations (PDEs), most importantly the heat conduction or diffusion equation and similar diffusion-reaction equation. In a homogeneous space, the diffusion equation has analytical solutions, old and new ones as well, [1], but when the material properties are functions of space, numerical integration is necessary. We are aware that many new and efficient methods are proposed and tested recently as well, [2]–[7], but here we are investigating the so-called positivity preserving methods. It is well known that the true solution of the heat or diffusion equation always follows the maximum and minimum principles [8] (p. 87). We know that these principles typically do not hold if there is a source term in the diffusion equation, either if it is constant, or if it is nonlinear, such as in the case of the Fisher and Huxley equations. But even in the latter cases, if the source term is necessarily positive, the dependent variable (e.g., the concentration) cannot be negative. It is important [9], [10] to have numerical schemes that preserve the positivity of the solution, because the quantities that are being modelled, concentrations of chemical species, populations sizes or number of neutrons, etc. are positive. Standard finite difference or finite element discretization does not explicitly incorporate the condition that the solutions be positive. So even though these algorithms usually yield positive values, due to truncation or other errors their solutions may become negative and non-physical. Then the solutions will usually start oscillating leading to numerical instabilities. That is why unconditionally positive schemes are investigated by some scholars [9]–[13]. We note that numerical methods that include the positivity of solutions have been used for ordinary differential equations, [14], [15] as well.

This work can be considered a continuation of our previous papers, [16]–[25], in which we developed novel numerical algorithms to solve the heat or diffusion equation. In our case, maximum and minimum principles hold, thus we say a method is positivity preserving if it never violates this principle. It means that we use this term in a stricter sense than most of the literature. In this paper we perform systematic tests for systems by gradually changing some parameters of the grid to examine how the performance of the individual methods changes and which of them is

the best choice under different circumstances.

2. THE EQUATIONS AND THEIR DISCRETIZATION

If we denote the unknown function (temperature, concentration, etc.) by *u*, then the simplest form of the diffusion equation is the following linear parabolic partial differential equation (PDE):

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \ . \tag{1}$$

Eq. (1) can be used if the medium where the diffusion takes place is homogeneous. Otherwise, one can use a more general form:

$$c\,\rho \frac{\partial u}{\partial t} = \nabla\left(k\,\nabla u\right), \tag{2}$$

where, in case of conductive heat transfer, $k = k(\vec{r},t)$, $c = c(\vec{r},t)$ and $\rho = \rho(\vec{r},t)$ are the heat conductivity, specific heat and mass density, respectively. These quantities cannot be negative and the $\alpha = k/(c\rho)$ relation connects the four quantities. We consider the initial function $u(t=0)$ as given, while the boundary conditions will be specified when we present the concrete examples.

Eq. (1) is usually spatially discretized in one dimension by applying the well-known central difference formula for obtain a system of ordinary differential equations (ODEs) for nodes $i = 1,\ldots,N-2$:

$$\frac{du_{\mathrm{i}}}{dt} = \alpha \frac{u_{\mathrm{i-1}} - 2u_{\mathrm{i}} + u_{\mathrm{i+1}}}{\Delta x^{\,2}} \tag{3}$$

In this paper, we don't exactly follow this formalism as explained in the following. We examine 2 space-dimensional, topologically rectangular meshes where the total number of cells is $N = N_x N_z$. The space step size (dimensions of the cells) in the horizontal and vertical directions is denoted by Δx_i, $i = 1,\ldots,N$ and Δz_k, $k = 1,\ldots,N$, respectively. The mesh is general in the sense that these quantities are not necessarily equal, i.e., $\Delta x_i \neq \Delta x_{i+1} \neq \Delta z_k$ can happen. Moreover, the *k*, *c*, and ρ quantities, which represent the material properties, can be functions of the space variables. Now

we give briefly the essential elements of the generalization procedure which is explained in more details in our previous papers, Kovács et al., [18], [22], [26] (which is based on the appropriate literature, e.g. Chapter 5 of the book, M. Munka and J. Pápay [27]). The (2-dimensional) volume of the cell can be given as $V_i = \Delta x_i \, \Delta z_i$, and with this the heat capacity of the cells can be calculated as $C_i = c_i \rho_i V_i$, while the thermal resistance between the cells in the x and z direction is $Rx_{i,i+1} \approx \frac{\Delta x_i}{k_{i,i+1} \Delta z_i}$ and $Rz_{i,Nx+i} \approx \frac{\Delta z_i}{k_{i,Nx+i} \Delta x_i}$, provided that the cell has a right and lower neighbor, respectively. Figure 1 is presented to help the reader to visualize this system.

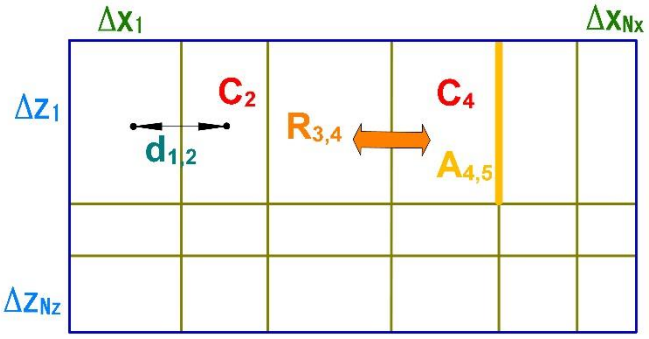

Figure 1. Arrangement of the generalized variables for the case when the mesh is not necessarily regular. The red double arrow is for conduction between cells with neighboring capacities indexed with number 3 and 4 through the resistance R_{34}.

Now the spatially discretized form of Eq. (2) can be written as follows:

$$\frac{du_\mathrm{i}}{dt} = \sum_\mathrm{j} \frac{u_\mathrm{j} - u_\mathrm{i}}{R_\mathrm{ij} C_\mathrm{i}}, \tag{4}$$

where the summation is going over all neighbors of the cell i, i.e., $j \in \{i-1, i+1, i-N_x, i+N_x\}$ for an inner cell. Eq. (4) is the generalization of Eq. (3) in the sense that if the definition of the volume, capacity and resistance is substituted back to (4) in the case of an equidistant 1D mesh, we obtain (3).

Equation systems (3) and (4) can be written into a matrix-form:

$$\frac{d\vec{u}}{dt} = M\vec{u} \tag{5}$$

In the one-dimensional case of Eq. (3), the matrix M is tridiagonal with the following elements:

$$m_{\mathrm{ii}} = -\frac{2\alpha}{\Delta x^2}\ (1 < i < N),\quad m_{\mathrm{i,i+1}} = \frac{\alpha}{\Delta x^2}\ (1 \le i < N),\quad m_{\mathrm{i,i-1}} = \frac{\alpha}{\Delta x^2}\ (1 < i \le N) \tag{6-a}$$

In the general case of Eq. (4), the nonzero elements of the matrix can be given as:

$$m_{\mathrm{ij}} = \frac{1}{R_{\mathrm{i,j}} C_{\mathrm{i}}},\quad m_{\mathrm{ii}} = -\sum_{\mathrm{j}\neq\mathrm{i}} m_{\mathrm{ij}} \tag{6-b}$$

The characteristic time or time constant τ_{i} of cell i can be introduced, which is always a non-negative quantity:

$$\tau_{\mathrm{i}} \doteq -\left(m_{\mathrm{ii}}\right)^{-1} \tag{7}$$

The time variable is discretized by setting time levels according to the standard rules: $t_n = nh, n = 0,...,T-1$. We introduce the following notations for practical reasons

$$r_i = \frac{h}{\tau_i} \text{ and } A_i = h\sum_{\mathrm{j}\neq i} m_{\mathrm{ij}} u_{\mathrm{j}}^n = h\sum_{\mathrm{j}\neq\mathrm{i}} \frac{u_{\mathrm{j}}^n}{C_i R_{\mathrm{ij}}} \tag{8}$$

The quantity r_{i} is similar to the standard mesh ratio $r = \frac{\alpha h}{\Delta x^2}$, while the A_{i} collects the effect coming from the neighbors of the cell i.

3. DESCRIPTION OF THE APPLIED METHODS

First, we need to recall the original UPFD, the CNe, the two- and three-stage linear-neighbor (LNe and LNe3) methods and the CpC algorithms.

The unconditionally positive finite-difference (UPFD) method was introduced by Chen-Charpentier and Kojouharov [9], [11] in 2013 for advection–diffusion reaction equations. In the case or Eq. (1) the algorithm reads as follows:

$$u_i^{\mathrm{n+1}} = \frac{u_i^{\mathrm{n}} + r\left(u_{i-1}^n + u_{i+1}^n\right)}{1+2r}\ , \tag{9}$$

and the general form for Eq. (2) is:

$$u_i^{\mathrm{n}+1} = \frac{u_i^{\mathrm{n}} + A_i}{1+2r_i} \tag{10}$$

We emphasize that this method was not invented by us but other scholars, while all the upcoming 8 methods are our constructions.

The second formula we use is the constant neighbor (CNe) method, which is introduced and tested in our papers, Kovács et al., [17], [18]. In the simplest case of Eq. (1),

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^n + u_{i+1}^n}{2}\left(1-e^{-2r}\right) \tag{11}$$

While for general grids it is:

$$u_i^{n+1} = u_i^n \cdot e^{-r_{\mathrm{i}}} + \frac{A_i}{r_{\mathrm{i}}}\left(1-e^{-r_{\mathrm{i}}}\right) \tag{12}$$

The third and fourth methods we use are the 2 and 3 stage linear-neighbor (LNe and LNe3) methods, which are introduced in our paper, Kovács [18]. They are based on the CNe method, which is used as a predictor to calculate new $u_{\mathrm{i}}^{\mathrm{CNe}}$ values valid at the end of the actual time step. Using them we can recalculate $A_i^{\mathrm{new}} = h\sum_{\mathrm{j}\neq\mathrm{i}} \frac{u_{\mathrm{j}}^{\mathrm{CNe}}}{C_i R_{\mathrm{ij}}}$, with which we can make the corrector step for the two-stage LNe method as follows:

$$u_{\mathrm{i}}^{\mathrm{n}+1} = u_{\mathrm{i}}^{\mathrm{n}} e^{-r_{\mathrm{i}}} + \left(A_{\mathrm{i}} - \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}}\right)\frac{1-e^{-r_{\mathrm{i}}}}{r_{\mathrm{i}}} + \frac{A_{\mathrm{i}}^{\mathrm{new}} - A_{\mathrm{i}}}{r_{\mathrm{i}}} \tag{13}$$

The values given in (13) can be used to recalculate again A_i^{new} which makes sense to repeat (13) to obtain even more accurate results, which is called three-stage LNe3 method. We mention that this iteration can be repeated even more times to obtain LNe4, etc. methods, but as we showed in our paper, Kovács [18], usually these don't significantly improve the accuracy.

The fifth method we examine is the two-stage Constant-neighbor (briefly: CpC) algorithm which is introduced in our paper, Kovács et al., [25]. This generally starts with a fractional time step with length ph, but here we take $p = 1/2$, because it is the simplest and usually the most accurate choice.

So, at the first stage, we can calculate new predictor values of the variables, but with a $h_1 = h/2$ time step:

$$u_{\mathrm{i}}^{\mathrm{Cp}} = u_{\mathrm{i}}^{\mathrm{n}} e^{-r_i/2} + \frac{A_{\mathrm{i}}}{r_{\mathrm{i}}}\left(1 - e^{-r_i/2}\right). \tag{14}$$

At the second stage, we can recalculate the values

$$A_i^{\mathrm{new}} = h\sum_{\mathrm{j}\neq\mathrm{i}} \frac{u_{\mathrm{j}}^{\mathrm{Cp}}}{C_i R_{\mathrm{ij}}}, \tag{15}$$

and take a full-time step size corrector step. Thus, the final values at the end of the time step are

$$u_i^{n+1} = u_i^n \cdot e^{-r_{\mathrm{i}}} + \frac{A_i^{\mathrm{new}}}{r_{\mathrm{i}}}\left(1 - e^{-r_{\mathrm{i}}}\right) \tag{17}$$

Now let us turn the attention to the hopscotch-type methods. First, we recall the original OEH method, where we need to construct a bipartite grid, in which the set of nodes (or cells) is divided into two similar subsets, the odd and even nodes, such that all nearest neighbors of even nodes are odd nodes and vice versa, like on a checkerboard. We treat the odd nodes in the first stage in odd time steps, and only the *u* values belonging to the beginning of the time step are used. Then, in the second stage, the new values of odd neighbors, which have just been calculated, are used, when the values of their even neighbors are calculated. The roles of the nodes are interchanged when even time steps start, as shown in Figure (2-a).

The original OEH method applies the standard explicit Euler formula in the first stage and the implicit Euler formula in the second stage. This is a powerful explicit method which is unconditionally stable for the heat equation, but it is far from being positivity preserving. In fact, as we have shown in, Kovács et al., [21], the deviation from the true values can be very large in case of stiff systems. Therefore, we substitute all the formulas by the CNe scheme (12) to obtain our sixth method denoted by OEH-CNe.

For the seventh method, we modify the time and space structures of the original OEH method to construct the so called shifted-hopscotch method, Kovács et al., [22], as follows. The calculation starts with taking a half-sized time step for the odd nodes which is symbolized by a light green box

containing the number 0 in Figure (2-b). Then, a full-time step is taken for the even nodes, then for the odd nodes and for the even nodes again. Finally, a half-size time step closes the calculation of the values. We use the latest available u values of the neighbors in each stage, so that the method is fully explicit and the previous values do not need to be stored at all. We have a structure consisting of five stages, which correspond to five partial time steps, which altogether spanned two-time steps for odd and even cells, too. In Figure (2-b), the number "0" means that this stage is implemented only once and is not repeated. Also, T must be an even integer because the stages 1–4 are repeated $T/2$ times (including the last, halved stage 4). So, finally, we have five different stages from 0 to 4.

Whereas in the case of the new leapfrog-hopscotch method, the calculation starts with taking a half-sized time step for the odd nodes using the initial values, Kovács et al., [23]. Then, for the even and odd nodes, full-time steps are taken strictly alternately until the end of the last timestep (dark green box in Figure 2-c), which should also be halved for odd nodes to reach exactly the same time point as the even nodes.

The last of our method is called asymmetric hopscotch (ASH), Fig. (2-d), similar to what was published in a previous paper, Kovács et al., [24]. This method uses only three stages instead of four. It means that the calculation starts with taking a half-sized time step for the odd nodes and full-time step for the even nodes, finally, a half-size time step closed the calculation of the values.

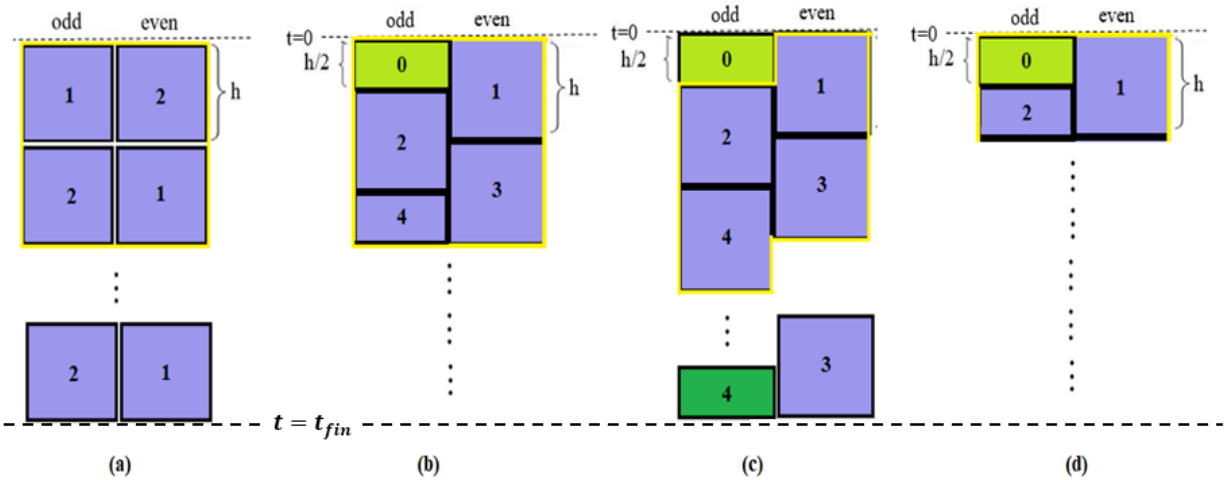

Figure 2. The stencil of the odd-even hopscotch structures for two nodes. (a) The OEH. (b) The shifted-hopscotch. (c) The leapfrog-hopscotch. (d) The asymmetric hopscotch.

Here it should be noted that we used only CNe formula with all stages in these structures because in previous papers, Kovács et al., [22], [23] we showed that among the positivity preserving combinations the full CNe was the best. In the case of halved time steps, the quantities r_i and A_i must be divided by 2.

As one can see in Figure 2, in case of the SH method, there were five-time steps (five stages) altogether, instead of four, while in the case of the ASH scheme, there were 3-time steps instead of two. Therefore, for the sake of correctness, we used the effective time step size as $h_{\mathrm{EFF}} = 4/5\, h$ and $h_{\mathrm{EFF}} = 2/3\, h$ for the SH and the ASH method, respectively, when the errors were plotted as a function of the time step sizes.

In our previous publications it has been proved that for all the methods presented here, the new value of u_i^{n+1} is the convex combination of the old values u_j^n, thus all of these methods are automatically unconditionally positive when applied to the linear heat equation.

4. Results of the Systematic Testing and Discussion

4.1. General definitions and investigation circumstances

We consider the 2D system described in Section 2 as thermally isolated (zero Neumann boundary conditions). 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, $\lambda_{MAX}/\lambda_{MIN}$ gives the stiffness ratio of the system, and the maximum possible time step size for the FTCS (explicit Euler) scheme is exactly given by $h_{MAX}^{FTCS} = \left|2 / \lambda_{MAX}\right|$, above which the solutions are expected to diverge due to instability. This $h_{\mathrm{MAX}}^{\mathrm{FTCS}}$ threshold time step size is frequently called the CFL limit and valid for the second order explicit Runge-Kutta (RK) methods as well. We stress again that this restriction is absolutely not valid for the methods presented here, but these two numbers give information about the difficulty level of the problem.

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

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

have been given to the cells. The parameters α_C , β_C , α_{Rx} , β_{Rx} , α_{Rz} , β_{Rz} of the distribution of the mesh-cells data have been chosen to construct test problems with various stiffness ratios. More concretely, we used the parameters shown in Table 1.

Table 1 The parameters used in algorithms

Type	No.	α_c	β_c	α_{Rx}	β_{Rx}	α_{Rz}	β_{Rz}
Non Stiff	1	0	0	0	0	0	0
	2	-1	1	0	0	0	0
Mildly Stiff	3	-1	1	-1	1	0	0
	4	-1	1	-1	1	-1	1
Medium Stiff	5	-2	2	-1	1	-1	1
	6	-2	2	-2	2	-1	1
	7	-2	2	-2	2	-2	2
Very Stiff	8	-3	3	-2	2	-2	2
	9	-3	3	-3	3	-2	2
	10	-3	3	-3	3	-3	3

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

The numerical error is calculated by comparing our numerical solutions $u_{\mathrm{j}}^{\mathrm{num}}$ produced by the examined method with the reference solution $u_{\mathrm{j}}^{\mathrm{ref}}$ at final time t_{fin} .The reference solution is given by the “ode15s” routine of MATLAB with large prescribed accuracy. We use the following three types of (global) error. The first one is the maximum of the absolute differences:

$$\mathrm{Error}(L_\infty) = \max_{0 \le \mathrm{j} \le N} \left| u_{\mathrm{j}}^{\mathrm{ref}}(t_{\mathrm{fin}}) - u_{\mathrm{j}}^{\mathrm{num}}(t_{\mathrm{fin}}) \right| \tag{18}$$

The second one is the average absolute error:

$$\mathrm{Error}(L_1) = \frac{1}{N} \sum_{0 \le \mathrm{j} \le N} \left| u_{\mathrm{j}}^{\mathrm{ref}}(t_{\mathrm{fin}}) - u_{\mathrm{j}}^{\mathrm{num}}(t_{\mathrm{fin}}) \right|. \tag{19}$$

We call the third one energy error, because it gives the error in terms of energy in case of the heat equation:

$$\mathrm{Error}(Energy) = \sum_{1\le j\le N} C_j \left| u_j^{\mathrm{ref}}(t_{\mathrm{fin}}) - u_j^{\mathrm{num}}(t_{\mathrm{fin}}) \right| \tag{20}$$

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

$$\mathrm{ARE}(L_\infty) = \frac{1}{S}\sum_{i=1}^{S} \log\left(\mathrm{Error}(L_\infty)\right) \tag{21}$$

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

$$\mathrm{ARE} = \frac{1}{3}\left(\mathrm{ARE}(L_\infty) + \mathrm{ARE}(L_1) + \mathrm{ARE}(Energy)\right) \tag{22}$$

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

First, we examined the following concrete parameter-combination,

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

which configuration has number 3 in Table 1 and categorized as mildly stiff.

We have plotted the L_∞ errors as a function of the effective time step size h_{EFF} and as a function of the running times for all methods. In Figure 3 we present the L_∞ error as a function of the effective time step size h_{EFF} for the nine positivity preserving methods defined in Section 3, while in Figure 4 one can see the L_∞ errors vs. the total running times.

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

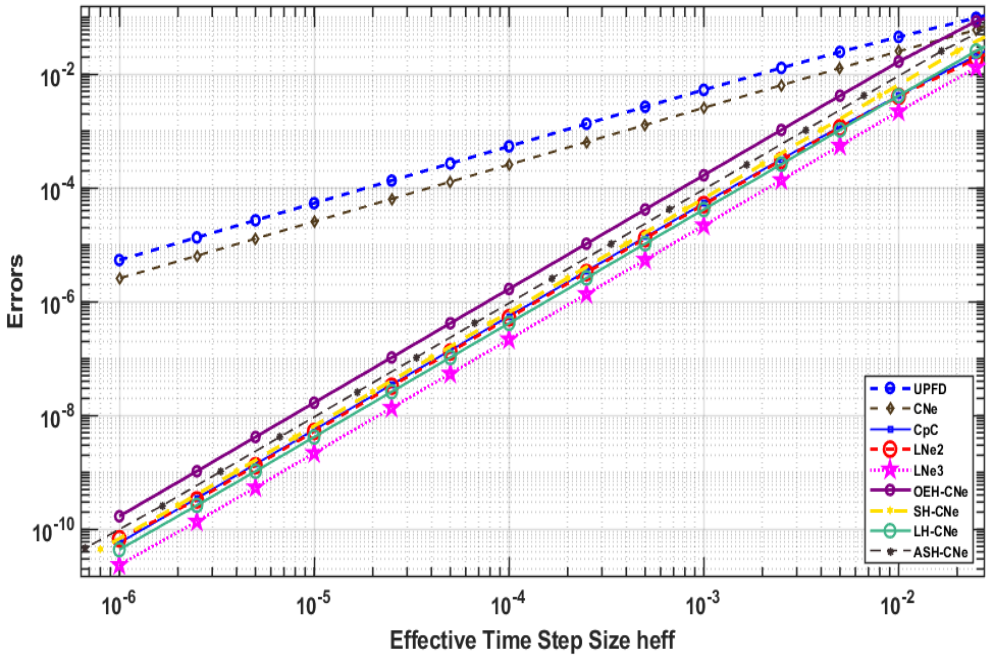

Figure 3. The L_∞ errors as a function of the effective time step size (h_{EFF}) for the (Mildly Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

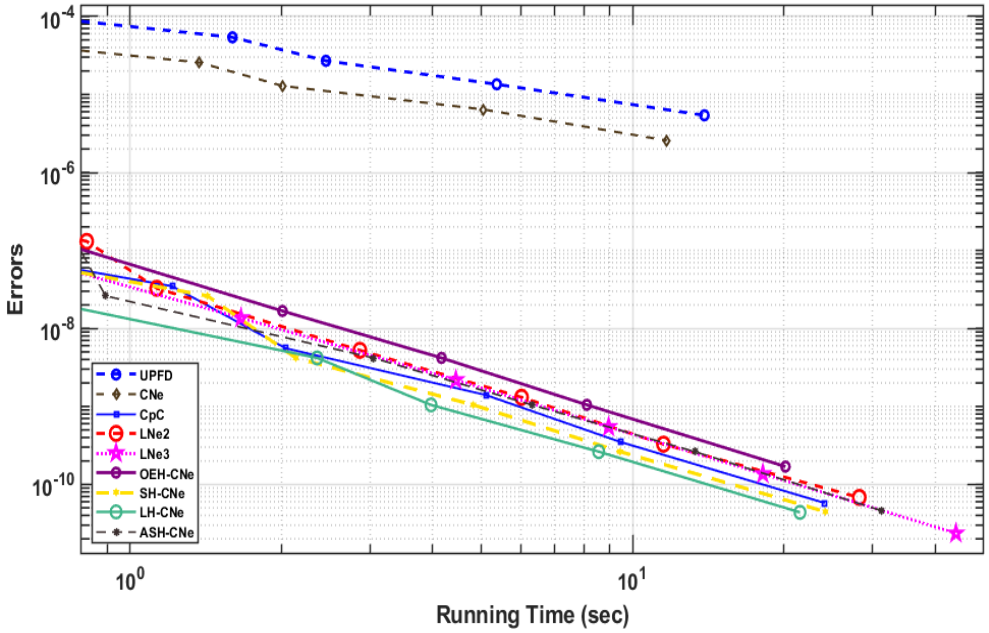

Figure 4. The L_∞ errors as a function of the running times for the (mildly stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

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

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

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

In Figures 5 and 6, the L_∞ error is presented as a function of the effective time step size h_{EFF} and the total running time, respectively.

We can see that in this case, the LH-CNe method outperforms all other examined positivity preserving methods provided that not only the accuracy, but also the speed, is taken into account. In Table 2, the data related to this numerical experiment is reported.

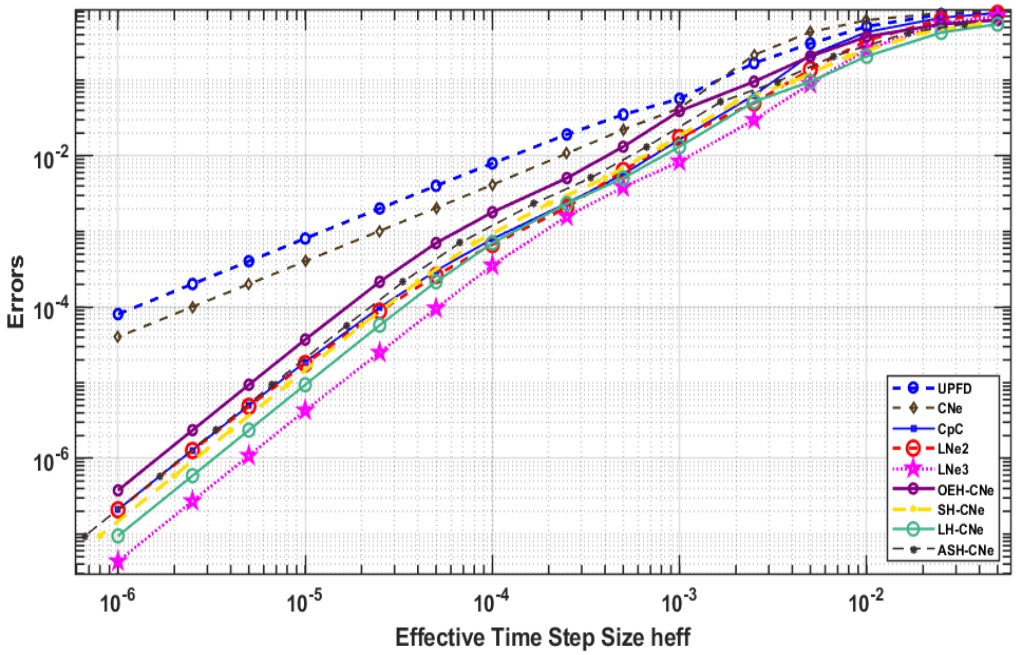

Figure 5. The L_∞ errors as a function of the effective time step size (h_{eff}) for the (very Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

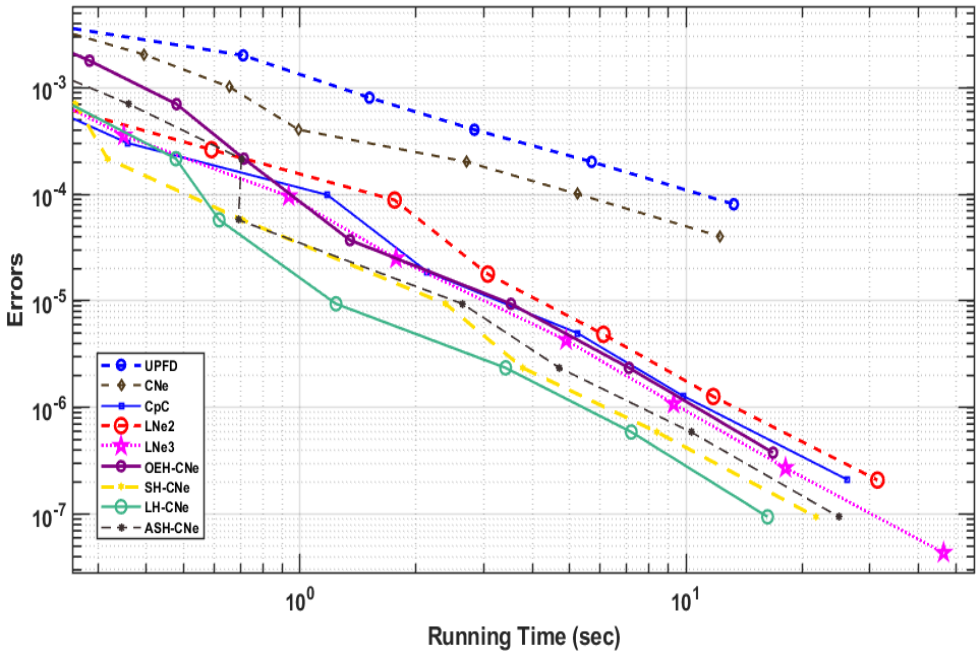

Figure 6. The L_∞ errors as a function of the running times for the (very Stiff) system, in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

Table 2. Comparison the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods for the very stiff system of 2500 cells.

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

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

Figures 7 and 8 show ARE errors as a function of h_{MAX} and stiffness ratio, respectively. We note the stiffness ratio affected on the accuracy of methods when they increased, so the accuracy becomes worse compared to the cases of small stiff ratios.

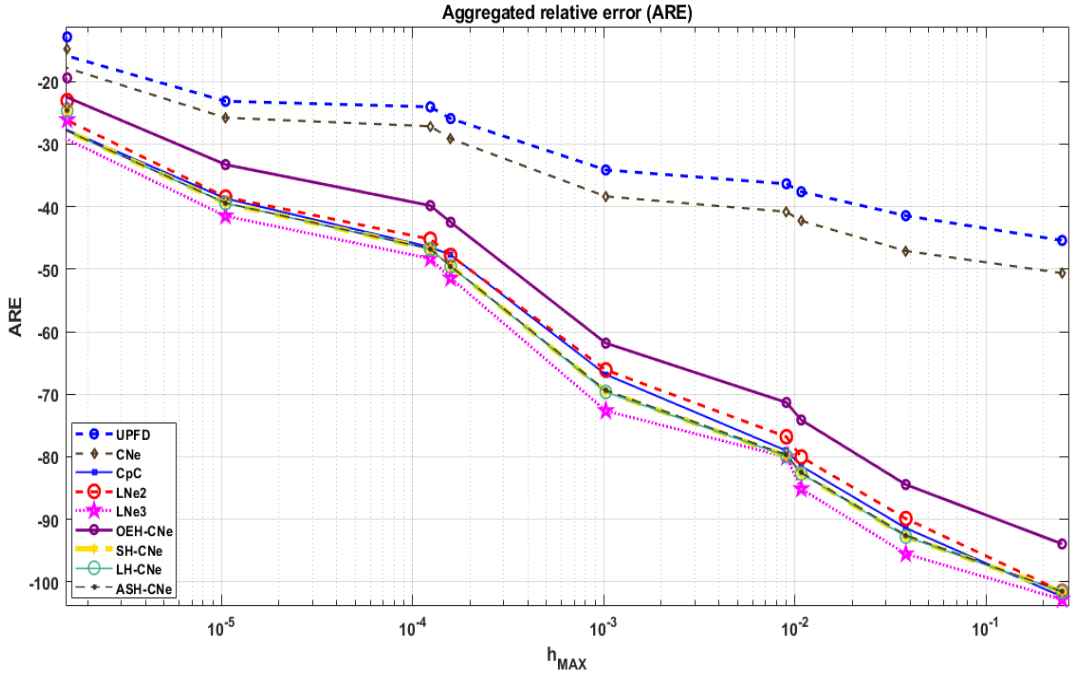

Figure 7. The (ARE) errors as a function of h_{MAX} in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

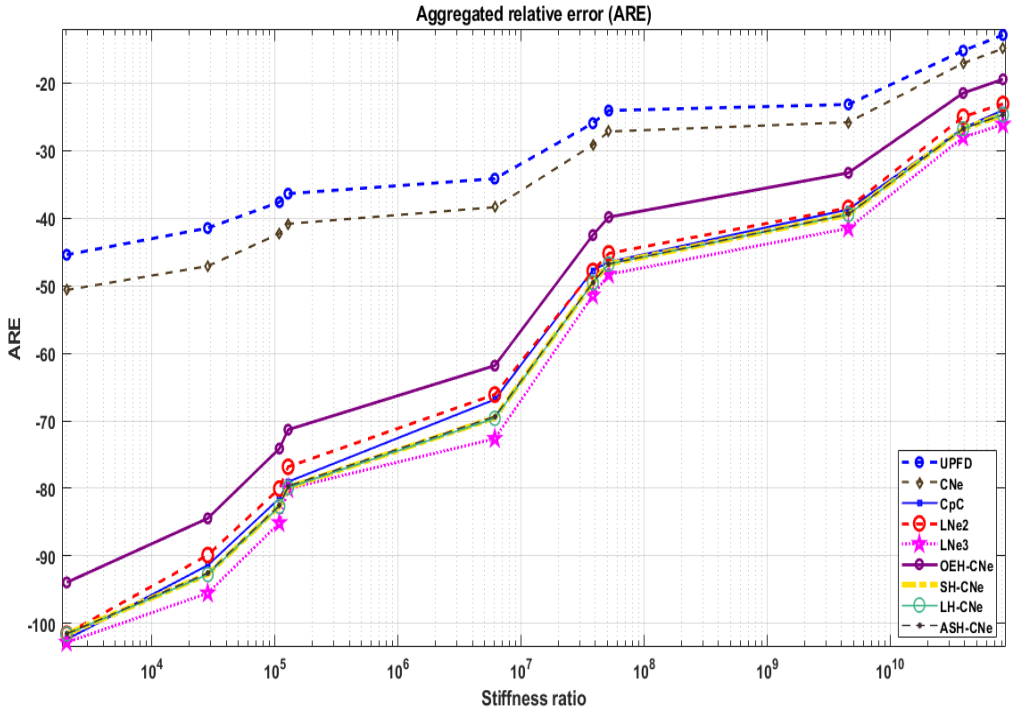

Figure 8. The (ARE) errors as a function of Stiff Ratio in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

We summarize the ARE error quantities, defined in Equation (22), for both case studies in the following table:

Table 3. ARE (average relative error) quantities of different explicit stable algorithms.

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

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

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

$$u(x,z,t=0) = \sin(\pi x)\sin(k\pi z), \tag{23}$$

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

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

The analytical solution of this problem is obviously

$$u(x,t) = \sin(\pi x)\sin(k\pi z)\,\mathrm{e}^{-\left(1+k^2\right)\pi^2 t}. \tag{24}$$

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

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

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

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

Then we examined the aggregated errors as a function of this anisotropy coefficient AC. Of course, the initial condition function in (23) and the exact solution must be also adjusted with recalculating the wave number $k = 2^{AC-1}$. In Figure 9, ARE errors are presented as a function of the anisotropy coefficient AC. We note the relative advantage of LNe3 method increased whereas relative disadvantage of other methods is also increased. In Table 4, we give h_{MAX} and the stiffness ratio for some AC values.

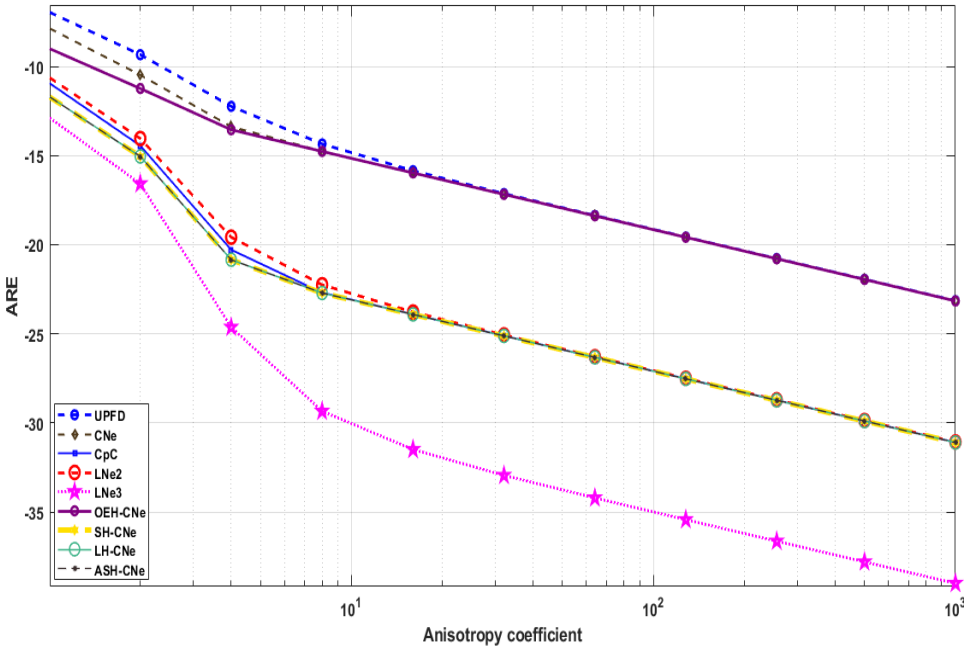

Figure 9. The (ARE) errors as a function of anisotropy coefficient *AC* in the case of the UPFD, CNe, CpC, LNe2, LNe3, the OEH-CNe, the SH-CNe, the LH-CNe and the ASH-CNe methods.

Table 4. h_{MAX} and stiffness ratio quantities of different AC values.

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

5. Summary and Conclusions

In this work, we conducted systematic tests of nine explicit numerical algorithms which introduced in previous papers to solve the heat equation. All of the methods preserve the positivity of the solutions, thus stable regardless of the time step size and the stiffness of the system. First, we examined two-dimensional stiff systems containing 2500 cells each with completely discontinuous random parameters and discontinuous initial conditions. We observed that the 3-stage LNe3

method produced the most accurate results for a given time step size, but the LH-CNe method (and sometimes the SH-CNe method) requires the least CPU time to reach any prescribed accuracy. The increasing stiffness ratio and the decreasing CFL limit decreased the accuracy of methods, and the advantage of the best methods compared to the worst are decreased.

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

We can conclude that if there is a possibility to construct the OEH structure (because the mesh is rectangular) than the hopscotch-CNe methods, especially the LH-CNe is the most effective among the positivity preserving methods, except when the anisotropy is strong. However, if an unstructured mesh is given, the LNe3 method is the most effective. For a very anisotropic system, for example a thin and wide layer, we also propose the LNe3 method.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] I.F. Barna, L. Mátyás, General self-similar solutions of diffusion equation and related constructions, (2021). http://arxiv.org/abs/2104.09128.

[2] N.A. Mbroh, J.B. Munyakazi, A robust numerical scheme for singularly perturbed parabolic reaction-diffusion problems via the method of lines, Int. J. Computer Math. 99 (2022), 1139–1158.

[3] I. Ali, S. Haq, K.S. Nisar, S.U. Arifeen, Numerical study of 1D and 2D advection-diffusion-reaction equations using Lucas and Fibonacci polynomials, Arab. J. Math. 10 (2021) 513–526.

[4] A.K. Verma, S. Kayenat, An efficient Mickens' type NSFD scheme for the generalized Burgers Huxley equation, J. Differ. Equ. Appl. 26 (2020), 1213–1246.

[5] S. Pourghanbar, J. Manafian, M. Ranjbar, A. Aliyeva, Y.S. Gasimov, An efficient alternating direction explicit

method for solving a nonlinear partial differential equation, Math. Probl. Eng. 2020 (2020), 9647416.

[6] A. Al-Bayati, S. Manaa, A. Al-Rozbayani, Comparison of finite difference solution methods for reaction diffusion system in two dimensions, AL-Rafidain J. Computer Sci. Math. 8 (2011), 21–36.

[7] A. Costa-Solé, E. Ruiz-Gironés, J. Sarrate, High-order hybridizable discontinuous galerkin formulation for one-phase flow through porous media, J. Sci. Comput. 87 (2021), 1–31.

[8] M. H. Holmes, Introduction to perturbation methods, vol. 20. Springer Science \& Business Media, 2012.

[9] B.M. Chen-Charpentier, H.V. Kojouharov, An unconditionally positivity preserving scheme for advection–diffusion reaction equations, Math. Computer Model. 57 (2013), 2177–2185.

[10] A.R. Appadu, Performance of UPFD scheme under some different regimes of advection, diffusion and reaction, Int. J. Numer. Methods Heat Fluid Flow. 27 (2017), 1412–1429.

[11] B. Drljača, S. Savović, Unconditionally positive finite difference and standard explicit finite difference schemes for power flow equation, Univ. Thought Publ. Nat. Sci. 9 (2019), 75–78.

[12] S. Savović, B. Drljača, A. Djordjevich, A comparative study of two different finite difference methods for solving advection–diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes, Ricerche Mat. (2021). https://doi.org/10.1007/s11587-021-00665-2.

[13] M.M. Khalsaraei, F. Khodadosti, Nonstandard finite difference schemes for differential equations, Sahand Commun. Math. Anal. 1 (2014), 47-54.

[14] H. Burchard, E. Deleersnijder, A. Meister, A high-order conservative Patankar-type discretisation for stiff systems of production--destruction equations, Appl. Numer. Math. 47 (2003), 1–30.

[15] D.T. Dimitrov, H.V Kojouharov, Positive and elementary stable nonstandard numerical methods with applications to predator--prey models, J. Comput. Appl. Math. 189 (2006), 98–108.

[16] E. Kovács, A. Gilicz, New stable method to solve heat conduction problems in extremely large systems, Des. Mach. Struct. 8 (2018), 30–38.

[17] E. Kovács, New stable, explicit, first order method to solve the heat conduction equation, J. Comput. Appl. Mech. 15 (2020), 3–13.

[18] E. Kovács, A class of new stable, explicit methods to solve the non-stationary heat equation, Numer. Methods Partial Differ. Equ. 37 (2020), 2469–2489.

[19] M. Saleh, Á. Nagy, E. Kovács, Construction and investigation of new numerical algorithms for the heat equation: Part 1, Multidiszcip. Tudományok. 10 (2020), 323–338.

[20] M. Saleh, Á. Nagy, E. Kovács, Construction and investigation of new numerical algorithms for the heat equation: Part 2, Multidiszcip. Tudományok. 10 (2020), 339–348.

[21] M. Saleh, Á. Nagy, E. Kovács, Construction and investigation of new numerical algorithms for the heat equation : Part 3, Multidiszcip. Tudományok. 10 (2020), 349–360.

[22] Á. Nagy, M. Saleh, I. Omle, H. Kareem, E. Kovács, New stable, explicit, shifted-hopscotch algorithms for the heat equation, Math. Comput. Appl. 26 (2021), 61.

[23] Á. Nagy, I. Omle, H. Kareem, E. Kovács, I.F. Barna, G. Bognar, Stable, explicit, Leapfrog-Hopscotch algorithms for the diffusion equation, Computation. 9 (2021), 92.

[24] O. Issa, New explicit algorithm based on the asymmetric hopscotch structure to solve the heat conduction equation, Multidiszcip. Tudományok, 11 (2021), 233–244.

[25] E. Kovács, Á. Nagy, M. Saleh, A set of new stable, explicit, second order schemes for the non-stationary heat conduction equation, Mathematics. 9 (2021), 2284.

[26] M. Saleh, Á. Nagy, E. Kovács, Construction and investigation of new numerical algorithms for the heat equation: Part 1, Multidiszcip. Tudományok. 10 (2020), 323–338.

[27] M. Munka, J. Pápay, 4D numerical modeling of petroleum reservoir recovery. Budapest: Akadémiai Kiadó, 2001.