*Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi*
*Sayı 8, S. 284-292, 2, 2024*
*© Telif hakkı IJANSER'e aittir*
***Araştırma Makalesi***

IJANSER

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

*International Journal of Advanced Natural Sciences and Engineering Researches*
*Volume 8, pp. 284-292, 2, 2024*
*Copyright © 2024 IJANSER*
***Research Article***

# Numerical performance of some positivity preserving methods for the diffusion equation where the diffusion coefficient depends on both time and space

**Husniddin Khayrullaev [1], Endre Kovács [1]**

[1] *Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary H. K.:* hxayrullayev@mail.ru
*Email of corresponding author:* kendre01@gmail.com , endre.kovacs@uni-miskolc.hu

*(Received: 07 March 2024, Accepted: 12 March 2024)*

(4th International Conference on Innovative Academic Studies ICIAS 2024, March 12-13, 2024)

**ATIF/REFERENCE:** Khayrullaev, H. & Kovács, E. (2024). Numerical performance of some positivity preserving methods for the diffusion equation where the diffusion coefficient depends on both time and space. *International Journal of Advanced Natural Sciences and Engineering Researches, 8(2), 284-292.*

***Abstract*** – The transient diffusion equation is solved, where the diffusion coefficient itself depends simultaneously on space and time. A nontrivial analytical solution containing the Whittaker functions is reproduced by 15 explicit numerical time integrators, most of which unconditionally preserve the positivity of the solutions. The accuracy of the methods is extensively examined, and it is found that these algorithms give very good results even in those cases where the standard explicit Runge-Kutta methods are hopeless due to the extreme stiffness of the problem.

*Keywords – Diffusion, Heat Conduction Unconditionally Stable Numerical Methods, Positivity Preserving Schemes.*

## I. INTRODUCTION AND THE STUDIED PROBLEM

We examine the linear partial differential equation (PDE) in one space dimension

$$\frac{\partial u(x,t)}{\partial t}=\frac{\partial}{\partial x}\left(\alpha(x,t)\frac{\partial u(x,t)}{\partial x}\right),\ u(x,t=0)=u^0(x),$$

(1)

where $x,t\in\mathbb{R}$ are the space and time variable, $u:\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$; $(x,t)\mapsto u(x,t)$ is the unknown function meaning concentration in the case of particle diffusion and temperature in the case of heat conduction. We consider $u^0$ as a given function and $\alpha(x,t)\in\mathbb{R}$ is the diffusion coefficient or diffusivity. The diffusivity has spatial and temporal dependence, which is justified by the fact that in several engineering problems, the properties of the materials widely change [1] because of natural or artificial inhomogeneities. Some new types of analytic solutions were found [2], [3] and plenty of numerical algorithms [4] were applied for different types of diffusion equations. In this work, we assume a specific type of diffusivity function, namely

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

where *D* is a positive constant. We constructed analytical solutions for this case in our previous paper [5], which contain the Whittaker functions, and therefore highly nontrivial. We set $D=1$ and with this, we have the following form of the solution:

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

The current work can be considered as the continuation of that previous work [5], where we applied several numerical schemes to solve Eq. (1) and the function (2) was used as the reference solution. Now we focus on only the positivity preserving methods and include the very recently published CLQ type family [6] which will be described later.

A large number of methods are proposed to solve Eq. (1) and similar kinds of equations [7] [8]. Both the common explicit and the implicit methods have at least one serious disadvantage. The widely used explicit algorithms, such as the Runge-Kutta types, must be used with a rather short time step size, because they become unstable above the so-called mesh Fourier number or CFL (Courant–Friedrichs–Lewy) limit. On the other hand, when one uses the implicit methods, a system of algebraic equations is required to be solved at each time step. This is nontrivial to be parallelized and, especially in multiple dimensions of space, the calculations can be very time-consuming. In real-life problems, explicit algorithms can be more efficient even with a short time step size [9]. The explicit and the implicit methods can be combined to obtain semi-explicit or semi-implicit algorithms [10], [11], [12], [13], but they do not overcome the above-mentioned problems.

Moreover, the true solution of the heat or diffusion equation always follows the maximum and minimum principles [14] (p. 87) reflecting the Second law of thermodynamics. However, most finite difference or finite element methods do not necessarily produce solutions with this property. That is why unconditionally positive schemes [15],[16],[17],[18] are investigated by some scholars.

In this work we collect several explicit methods, most of which fulfil the maximum and minimum principles. We test these methods by performing a sweep for the parameter *m* to explore how the performance of the methods is changing with this parameter. We write our own numerical codes in MATLAB to perform the numerical simulations.

## II. THE DISCRETIZATION AND THE NUMERICAL ALGORITHMS

### 2.1. The spatial and temporal discretization

The time variable is discretized uniformly, i.e. $t \in [t^0, t^{\text{fin}}]$, and $t^n = t^0 + nh$, $n = 1,...,T$, $hT = t^{\text{fin}} - t^0$. An equidistant spatial grid $x_j = x_0 + j\Delta x$, $j = 0,...,N$, $N\Delta x = L$ is constructed on the interval $x \in [x_0, x_N = x_0 + L] \subset \mathbb{R}$. We reproduce analytical solution (2), thus that formula will be used to prescribe the Dirichlet boundary conditions.

We simultaneously discretize the function α and $\partial u/\partial x$ in Eq. (1). The central difference formula is used to obtain

$$\left.\frac{\partial u}{\partial t}\right|_{x_i,t^n} = \frac{1}{\Delta x}\left[\alpha\left(x_i + \frac{\Delta x}{2}, t^n\right)\frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} + \alpha\left(x_i - \frac{\Delta x}{2}, t^n\right)\frac{u(x_i - \Delta x) - u(x_i)}{\Delta x}\right].$$

If we move from node to cell variables, $u_i$ will be the approximation of the temperature of the cell i, by its value at the center of the cell. Moreover, the diffusivity between two cells will be estimated by its value at the border of the appropriate cells. This yields

$$\frac{du_i}{dt} = \frac{1}{\Delta x}\left(\alpha_{i,i+1}^n \frac{u_{i+1} - u_i}{\Delta x} + \alpha_{i,i-1}^n \frac{u_{i-1} - u_i}{\Delta x}\right)$$

Let us introduce the capacity of the cell as $C_i = \Delta x$. The resistances can be introduced as follows:

$$R_{i,i+1}^n = \frac{\Delta x}{\alpha_{i,i+1}^n} = \frac{\Delta x}{D\left(x_{i,i+1}/\sqrt{t^n}\right)^m}, \quad i = 1,...,N-1,$$

which depend on time as well. Now we obtain the following system of ordinary differential equations:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i,i-1}^n C_i} + \frac{u_{i+1} - u_i}{R_{i,i+1}^n C_i} \tag{3}$$

which has the matrix form

$$\frac{d\vec{u}}{dt} = M\vec{u}\,. \tag{4}$$

The $N \times N$ dimensional system matrix $M$ depends on the time. More details about this manner of discretization can be found e.g. in [19]. The matrix $M$ have always negative eigenvalues. The smallest (largest) absolute value eigenvalues are denoted by $\lambda_{\text{MIN}}$ ($\lambda_{\text{MAX}}$). Now, the CFL limit can be calculated as $h\text{MAX} = |2/\lambda_{\text{MAX}}|$. It is valid for the Explicit Euler method, but for higher order RK methods, this limit is only a little larger. The stiffness ratio of the system can also be calculated as $SR = \lambda_{\text{MAX}}/\lambda_{\text{MIN}}$.

### 2.2. The description of the 15 numerical algorithms

All the tested algorithms are already published before and one can find more details about them (e.g. analytical proofs) in the given references. Let us briefly present immediately the formulas which are applied for Eq. (3). We introduce the following two quantities:

$$r_i^n = \frac{h}{C_i}\left(\frac{1}{R_{i,i-1}^n} + \frac{1}{R_{i,i+1}^n}\right) \text{ and } A_i^n = \frac{h}{C_i}\left(\frac{u_{i-1}^n}{R_{i,i-1}^n} + \frac{u_{i+1}^n}{R_{i,i+1}^n}\right),\ i = 1,...,N-1,\ n = 0,...,T \tag{5}$$

The term $r_i^n$ is similar to the usual mesh-ratio $r = \frac{Dh}{\Delta x^2}$, which is frequently used in the case of constant diffusivity. On the other hand, $A_i^n$ summarize information about the neighbors of the cell. The methods are the following:

1. The so-called UPFD (unconditionally positive finite difference) scheme has one stage only with the formula

$$u_i^{n+1} = \frac{u_i^n + A_i^n}{1 + r_i^n} \tag{6}$$

This method was proposed by Chen-Charpentier and Kojouharov [15] for the more general case of the diffusion-advection-reaction equation.

2. The constant neighbour (CNe) method [20], [21] is a one-stage method again, which contains the $r_i$ quantities in the exponents:

$$u_i^{n+1} = u_i^n \cdot e^{-r_i^n} + \frac{A_i^n}{r_i^n}\left(1 - e^{-r_i^n}\right) \tag{7}$$

3. The two-stage CpC method [22] employs the CNe formula two times. The first one is a fractional (halved) sized stage, where the substitution $h \to h/2$ must be performed in (5) and then (7) is applied. These predictor values makes possible the calculation of the new values of the $A$ quantities:

$$A_i^{\text{new}} = \frac{h}{C_i}\left(\frac{u_{i-1}^{\text{pred}}}{R_{i,i-1}^n} + \frac{u_{i+1}^{\text{pred}}}{R_{i,i+1}^n}\right). \tag{8}$$

At the second stage, these $A_i^{\text{new}}$ values are substituted into (7) in the full-length corrector stage. We stress that the resistances are updated only when a new time step starts. This principle will be valid in the case of the other multi-stage methods.

4. The first stage of the linear-neighbor (LNe) algorithm [21] is a full-length predictor time step with the CNe formula (7) to calculate the $u_i^{\text{pred}}$ values. Using them new $A_i^{\text{new}}$ values are calculated as in (8). The corrector step is performed with the formula:

$$u_i^{n+1} = u_i^n e^{-r_i^n} + \left(A_i^n - \frac{A_i^{\text{new}} - A_i^n}{r_i^n}\right)\frac{1 - e^{-r_i^n}}{r_i^n} + \frac{A_i^{\text{new}} - A_i^n}{r_i^n} \tag{9}$$

5-6. Using the corrector values obtained by (9), we are able to recalculate $A_i^{\text{new}}$, then repeat (9) to obtain the refreshed and usually more accurate corrector values. This three stage-algorithm is called the LNe3 method [21]. Repeating (8) with the new values and the corrector step (9) again, a 4-stage formula abbreviated as LNe4 is obtained.

7. The CLL method [23] is a modification of LNe3 to achieve third order convergence in time. It uses fractional-sized time steps in the first two stages with $h_1 = \frac{2}{3}h$. In the first stage, the CNe formula is employed with this reduced time step size to calculate new predictor values denoted by $u_i^{\text{C}}$. These are the basis of the calculation of the $A_i^{\text{C}}$ values such as in Eq. (8). In the second stage, we use formulas similar to (9), but with the reduced time step size to obtain the first corrector values:

$$u_i^{\text{CL}} = u_i^n e^{-2r_i^n/3} + \left(A_i^n - \frac{A_i^{\text{C}} - A_i^n}{2r_i^n/3}\right)\frac{1 - e^{-2r_i^n/3}}{r_i^n} + \frac{A_i^{\text{C}} - A_i^n}{r_i^n}$$

The third, full time step size stage starts with the calculation of the new $A_i^{\text{CL}}$ using the just obtained $u_i^{\text{CL}}$ values, and then the LNe formula is employed again:

$$u_i^{n+1} = u_i^n e^{-r_i^n} + \left(A_i^n - \frac{A_i^{\text{CL}} - A_i^n}{2r_i^n/3}\right)\frac{1 - e^{-r_i^n}}{r_i^n} + \frac{A_i^{\text{CL}} - A_i^n}{2r_i^n/3}. \tag{10}$$

8. The CCL method [24] is similar to the CLL, but the length of the first fractional time step is only $h_1 = h/3$. Apart from this, the first stage is the same as in the CLL method. At the second stage, however, the CNe formula is used again with $\frac{2}{3}h$ length to obtain the values denoted by $u_i^{\text{CC}}$. These are used to calculate $A_i^{\text{CC}}$ in the third stage, which is a full-time step with the LNe formula (10), but with the $A_i^{\text{CC}}$ quantities instead of the $A_i^{\text{CL}}$ ones.

9. The two-stage pseudo-implicit (PI) method [25] has a half time step size predictor stage and then a full time step size corrector stage. They employ the formulas:

$$\text{Stage 1: } u_i^{\text{pred}} = \frac{u_i^n + A_i^n/2}{1 + r_i^n/2}, \qquad \text{Stage 2: } u_i^{n+1} = \frac{(1 - r_i/2)u_i^n + A_i^{\text{new}}}{1 + r_i/2},$$

where $A_i^{\text{new}}$ is calculated as in (8).

10. The three-stage Constant-Linear-Quadratic neighbour (CLQ) algorithm [6] utilizes the same first and second stages as the LNe method. The only difference is that the second stage has to be performed with not only a full, but a half time step size as well using the LNe formula. Let us denote the obtained values by $u_i^{\text{L}}$

and $u_i^{\mathrm{L}\frac{1}{2}}$, respectively. Using them we can calculate $A_i^{\mathrm{pred,L}}$ and $A_i^{\mathrm{pred,L}\frac{1}{2}}$ such as above, and then the quantities $S_i = 4A_i^{\mathrm{pred,L}\frac{1}{2}} - A_i^{\mathrm{pred,L}} - 3A_i$ and $W_i = 2\left(A_i^{\mathrm{pred,L}} - 2A_i^{\mathrm{pred,L}\frac{1}{2}} + A_i\right)$, where $A_i$ is calculated at the beginning of the first stage. The final values of *u* at the end of the time step are calculated by

$$u_i^{\mathrm{Q}} = e^{-r_i}u_i^n + \frac{1-e^{-r_i}}{r_i}\left(\frac{2W_i}{r_i^2} - \frac{S_i}{r_i} + A_i\right) + \frac{W_i\left(1-2/r_i\right)+S_i}{r_i}$$

11. The above obtained $u_i^{\mathrm{Q}}$ values can be used to add one more stage, with which we have a four-stage method called the CLQ2 method. To make it possible, the midpoint values must be calculated at the third stage:

$$u_i^{\mathrm{Q}\frac{1}{2}} = e^{-r_i/2}u_i^n + \frac{1-e^{-r_i/2}}{r_i}\left(\frac{2W_i}{r_i^2} - \frac{S_i}{r_i} + A_i\right) + \frac{W_i}{4r_i} - \frac{W_i}{r_i^2} + \frac{S_i}{2r_i}$$

Now, in Stage 4, one repeats the calculations of the third stage, but uses $A_i^{\mathrm{pred},Q}$, and $A_i^{\mathrm{pred},Q\frac{1}{2}}$ to obtain the new values of S and W, etc.

12-13. The iteration of point 11 may be further repeated. In this manner, we obtain the CLQ3 scheme (5 stages altogether), as well as the CLQ4 scheme (6 stages altogether) [6].

Let us now focus on the hopscotch schemes, where the odd and even nodes are treated differently. These algorithms work properly only if in each step, the latest available *u* values of the left and right neighbours are used. The CNe formula will be applied everywhere to maintain the positivity of the results.

14. OEH-CNe (odd-even hopscotch CNe) method: full-length time steps are made, first for the odd nodes only and then for the even nodes. In the next step, and in all of the coming steps, the roles are interchanged: first the even and then the odd nodes are treated, then vice versa, etc.

15. Leapfrog-hopscotch-CNe (LH-CNe, [19]): We start with a half-sized time step for only the nodes, then full time steps for the even and odd nodes are coming alternately. A half-sized time step closes the calculation for the odd nodes to reach the final time of the simulation.

The order of convergence is one for the UPFD and the CNe algorithms, two for the PI, LH-CNe, CpC, and LNe-LNe5 algorithms, three for the CCL, CLL and CLQ schemes, and four for the CLQ2-4 methods. All of them are unconditionally stable for the diffusion equation, thus the CFL restriction does not apply in their case, which is exceptional. From the point of view of preserving the positivity, there are three categories. In the case of the UPFD, CNe, CpC, LNe, LNe3-4, OEH-CNe, and LH-CNe schemes, it is analytically proven that the Maximum and Minimum principles hold for arbitrary mesh, i.e. for arbitrary values of the $C_\mathrm{i}$ and $R_\mathrm{i}$ quantities. This is a consequence of the fact that the new $u_i^{n+1}$ values are the convex combination of the $u_i^n$, $u_{i-1}^n$, $u_{i+1}^n$ etc. values. For the CLQ-CLQ4 family, this is proven only for the simplest, one-dimensional equidistant case, but all numerical experiments support the validity of the principle for the general case as well. On the other hand, we know that the CCL, CLL and PI schemes do not always fulfil the principle. However, these violation of the principle appears only for extremely large time step sizes and for vary stiff cases. Thus, in practice, they can be considered as “almost positivity preserving” methods. We will see that the error of the methods is strictly limited even in the case of extremely stiff problems.

## III. NUMERICAL RESULTS FOR DIFFERENT PARAMETER VALUES

We investigate how the numerical error depends on the time step size *h* and the parameter *m*. First, we do the simulation with all methods for a very large and fixed *h* and calculate the usual $L_\infty$ or maximum error at $t^{\mathrm{fin}}$:

$$Error = \max_{1<i<N}\left|u_\mathrm{i}^{\mathrm{analytic}}(t^{\mathrm{fin}}) - u_\mathrm{i}^{\mathrm{num}}(t^{\mathrm{fin}})\right|$$

Then this procedure is repeated with smaller time step sizes until very small error values are obtained. We used *Nh* =17 different time step sizes for all the examined methods. Note that some data about the running times can be found in our previous papers. In the first experiment, the used parameter values are the following:
$m = 2.1,\ \alpha = 0.1,\ N = 100, x_0 = 0.25,\ \Delta x = 0.01,\ t^0 = 0.1,\ t^{\text{fin}} = 0.2$ . The value of *c* is always set to be a normalization constant, i.e. the largest absolute values of the functions are always unity. After this first experiment, we changed *m* to *m*=20. The errors as a function of the time step size are displayed in Fig. 1 and 2. for the first and second experiment, respectively. One sees that the accuracy difference between the algorithms became much smaller for this large value of *m*.

Now, we perform a parameter sweep for *m*. To be able to characterize the performance of a method with one number, we calculate the aggregated maximum error:

$$\text{AgE}(L_\infty) = \frac{1}{Nh}\sum_{nh=1}^{Nh} \log\left(\text{Error}(L_\infty)\right) \qquad (11)$$

The average absolute error can also be calculated:

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

and based on this, $\text{AgE}(L_1)$ is calculated as in (11). The third type of error gives the misplaced energy in the case of heat conduction, therefore we call it energy error:

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

The simple average of the three types of errors $\text{AgE} = \left(\text{AgE}(L_\infty) + \text{AgE}(L_1) + \text{AgE}(Energy)\right)/3$ will be used to assess the overall accuracy of the methods. It is easy to see that if a method has negative and larger absolute value AgE, it is more accurate. For the parameter sweep, the $m \in \{2.1,\ 3,\ 5,\ 7,\ 10,\ 12,\ 14.5, 17,\ 20\}$ values of the coefficient are used. The AgE values as a function of *m* are displayed in Fig. 3. Table 1 shows the numerical value of these errors as well as *SR* and *hMAX* values at the initial and at the final time moments.

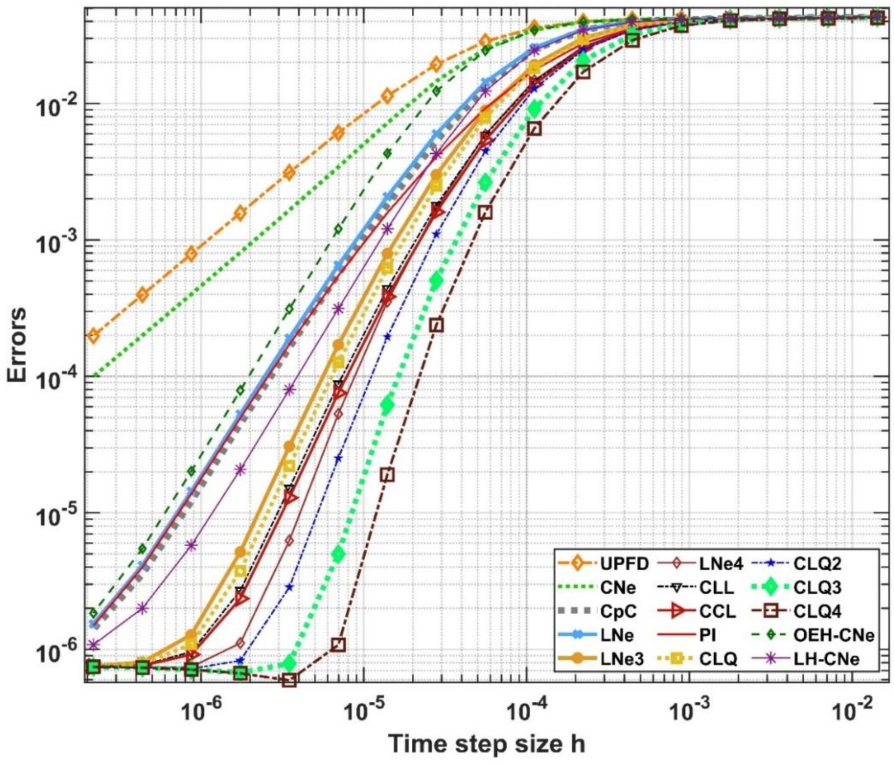

Figure 1 The maximum error as a function of the time step size for m=2.1

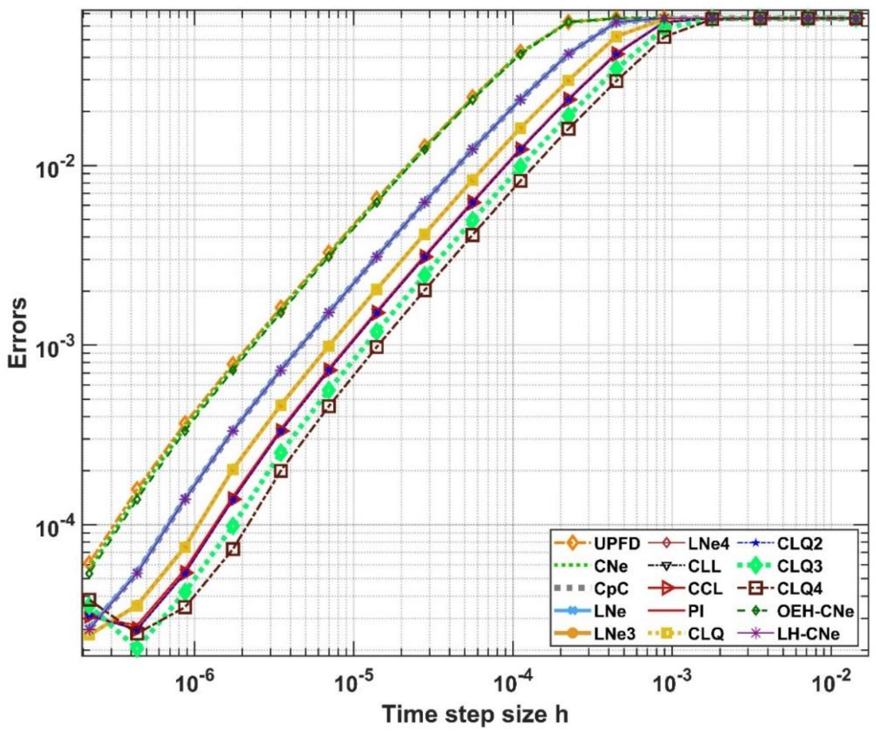

Figure 2 The maximum error as a function of the time step size for m=20

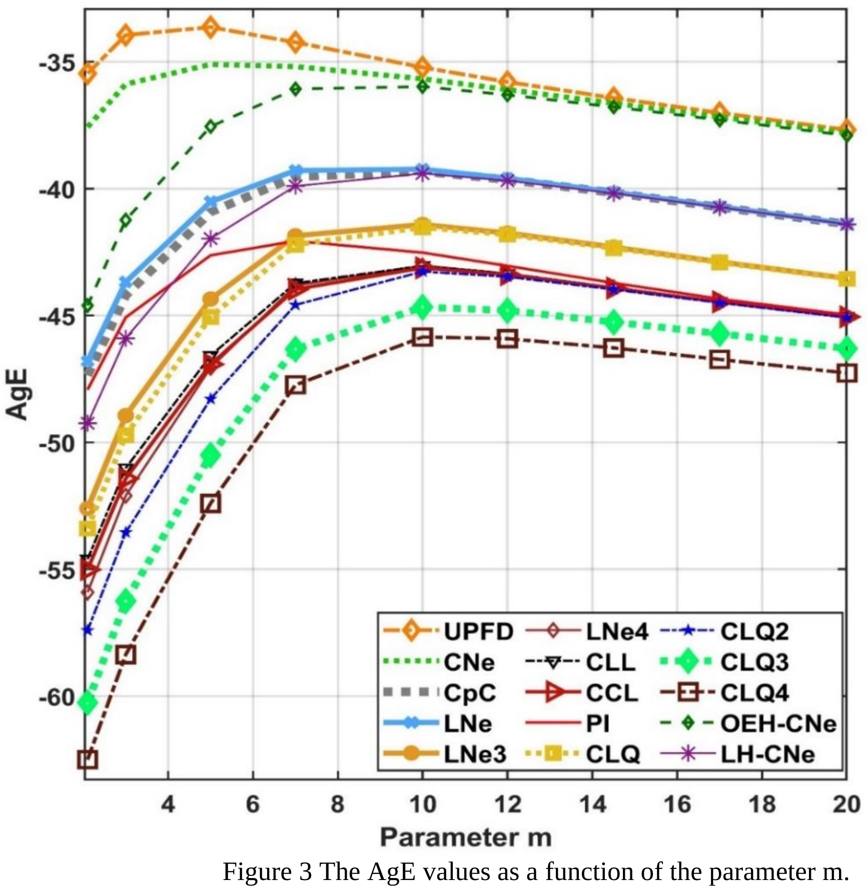

Figure 3 The AgE values as a function of the parameter m.

Table 1 The exponents and the stiffness ratios for the cases, as well as the AgE values for the methods

| $m$ | 2.1 | 3 | 5 | 7 | 10 | 12 | 14.5 | 17 | 20 |
|---|---|---|---|---|---|---|---|---|---|
| SR(i) | $3.1\times10^{5}$ | $1.1\times10^{6}$ | $1.5\times10^{7}$ | $2.2\times10^{8}$ | $1.2\times10^{10}$ | $1.9\times10^{11}$ | $5.6\times10^{12}$ | $1.6\times10^{14}$ | $9.4\times10^{15}$ |
| SR(f) | $1.5\times10^{5}$ | $3.7\times10^{5}$ | $2.7\times10^{6}$ | $2\times10^{7}$ | $4\times10^{9}$ | $3\times10^{9}$ | $8.1\times10^{10}$ | $3\times10^{12}$ | $2.5\times10^{14}$ |
| hMAX(i) | $3.1\times10^{-6}$ | $9.3\times10^{-7}$ | $6.4\times10^{-8}$ | $4.3\times10^{-9}$ | $7.6\times10^{-11}$ | $5.1\times10^{-12}$ | $1.7\times10^{-13}$ | $6\times10^{-15}$ | $1\times10^{-16}$ |
| hMAX(f) | $6.4\times10^{-6}$ | $2.6\times10^{-6}$ | $3.6\times10^{-7}$ | $4.9\times10^{-8}$ | $2.4\times10^{-9}$ | $3.3\times10^{-10}$ | $2.7\times10^{-11}$ | $2.2\times10^{-12}$ | $1.1\times10^{-13}$ |
| UPFD | -35.46 | -33.94 | -33.63 | -34.23 | -35.22 | -35.79 | -36.42 | -37.01 | -37.68 |
| CNe | -37.57 | -35.88 | -35.10 | -35.19 | -35.67 | -36.09 | -36.63 | -37.17 | -37.81 |
| CpC | -47.36 | -44.20 | -40.92 | -39.53 | -39.32 | -39.64 | -40.16 | -40.72 | -41.40 |
| LNe | -46.79 | -43.67 | -40.51 | -39.28 | -39.23 | -39.59 | -40.13 | -40.71 | -41.39 |
| LNe3 | -52.60 | -48.94 | -44.35 | -41.85 | -41.41 | -41.75 | -42.30 | -42.88 | -43.53 |
| LNe4 | -55.90 | -52.09 | -47.05 | -43.81 | -43.04 | -43.35 | -43.90 | -44.45 | -45.05 |
| CLL | -54.60 | -51.01 | -46.58 | -43.72 | -43.04 | -43.34 | -43.89 | -44.45 | -45.04 |
| CCL | -55.01 | -51.42 | -46.92 | -43.96 | -43.12 | -43.39 | -43.92 | -44.46 | -45.05 |
| PI | -47.92 | -45.07 | -42.62 | -42.06 | -42.52 | -43.04 | -43.71 | -44.34 | -44.97 |
| CLQ | -53.40 | -49.71 | -45.05 | -42.21 | -41.52 | -41.81 | -42.34 | -42.91 | -43.55 |
| CLQ2 | -57.41 | -53.54 | -48.28 | -44.58 | -43.26 | -43.47 | -43.97 | -44.50 | -45.08 |
| CLQ3 | -60.25 | -56.25 | -50.50 | -46.32 | -44.67 | -44.80 | -45.26 | -45.71 | -46.30 |
| CLQ4 | -62.50 | -58.36 | -52.40 | -47.73 | -45.84 | -45.91 | -46.27 | -46.73 | -47.26 |
| OEH-CNe | -44.60 | -41.23 | -37.55 | -36.07 | -35.98 | -36.30 | -36.78 | -37.29 | -37.90 |
| LH-CNe | -49.25 | -45.89 | -41.96 | -39.90 | -39.40 | -39.68 | -40.18 | -40.73 | -41.40 |

## IV. DISCUSSION AND SUMMARY

The transient diffusion equation has been investigated in which the diffusion coefficient depends on the space and time coordinates at the same time. A recent nontrivial analytical solution containing a Whittaker function has been used as the reference solution. This solution has a parameter *m*, and increasing the value of *m* yields increasing stiffness ratio and decreasing CFL limit. For large values of *m*, the CFL limit is extremely small. Moreover, because of the time-dependence of the diffusivity, the CFL limit is changing in time, so using the traditional explicit algorithms is very risky, if not impossible.

The examined 15 explicit numerical algorithms reproduced accurately the reference solution. We observed that the difference in the accuracy between the methods are decreasing with increasing *m*, which can be a sign of the order reduction. It reinforces the common experience that higher order methods have significant advantages only if the problem is not very stiff, otherwise, lower order methods are enough to use. The relative strength of the schemes are only slightly changing with increasing *m*, with the exception of the PI method, since it is remarkably improving and overtakes the LH-CNe, the LNe3 and even the third order CLQ method. The other exception is the second-order OEH-CNe method, because its performance is seriously declining with increasing *m*, and become the same as that of the first-order methods.

## REFERENCES

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

[2] L. Mátyás and I. F. Barna, 'General Self-Similar Solutions of Diffusion Equation and Related Constructions', *Romanian J. Phys.*, vol. 67, p. 101, 2022.

[3] I. F. Barna and L. Mátyás, 'Advanced Analytic Self-Similar Solutions of Regular and Irregular Diffusion Equations', *Mathematics*, vol. 10, no. 18, p. 3281, Sep. 2022, doi: 10.3390/math10183281.

[4] M. Saleh, E. Kovács, I. F. Barna, and L. Mátyás, 'New Analytical Results and Comparison of 14 Numerical Schemes for the Diffusion Equation with Space-Dependent Diffusion Coefficient', *Mathematics*, vol. 10, no. 15, p. 2813, Aug. 2022, doi: 10.3390/math10152813.

[5] M. Saleh, E. Kovács, and I. F. Barna, 'Analytical and Numerical Results for the Transient Diffusion Equation with Diffusion Coefficient Depending on Both Space and Time', *Algorithms*, vol. 16, no. 4, p. 184, Mar. 2023, doi: 10.3390/a16040184.

[6] E. Kovács, J. Majár, and M. Saleh, 'Unconditionally Positive, Explicit, Fourth Order Method for the Diffusion- and Nagumo-Type Diffusion–Reaction Equations', *J. Sci. Comput. 2024 982*, vol. 98, no. 2, pp. 1–39, Jan. 2024, doi: 10.1007/S10915-023-02426-9.

[7] S. M. Savović and A. Djordjevich, 'Numerical solution of diffusion equation describing the flow of radon through concrete', *Appl. Radiat. Isot.*, vol. 66, no. 4, pp. 552–555, 2008, doi: 10.1016/j.apradiso.2007.08.018.

[8] O. A. Jejeniwa, H. H. Gidey, and A. R. Appadu, 'Numerical Modeling of Pollutant Transport: Results and Optimal Parameters', *Symmetry*, vol. 14, no. 12, p. 2616, Dec. 2022, doi: 10.3390/sym14122616.

[9] S. Essongue, Y. Ledoux, and A. Ballu, 'Speeding up mesoscale thermal simulations of powder bed additive manufacturing thanks to the forward Euler time-integration scheme: A critical assessment', *Finite Elem. Anal. Des.*, vol. 211, p. 103825, Nov. 2022, doi: 10.1016/j.finel.2022.103825.

[10] L. Beuken, O. Cheffert, A. Tutueva, D. Butusov, and V. Legat, 'Numerical Stability and Performance of Semi-Explicit and Semi-Implicit Predictor–Corrector Methods', *Mathematics*, vol. 10, no. 12, Jun. 2022, doi: 10.3390/math10122015.

[11] Y. Ji and Y. Xing, 'Highly Accurate and Efficient Time Integration Methods with Unconditional Stability and Flexible Numerical Dissipation', *Mathematics*, vol. 11, no. 3, p. 593, Jan. 2023, doi: 10.3390/math11030593.

[12] P. Fedoseev, D. Pesterev, A. Karimov, and D. Butusov, 'New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers', *Algorithms*, vol. 15, no. 8, p. 275, Aug. 2022, doi: 10.3390/a15080275.

[13] N. Ndou, P. Dlamini, and B. A. Jacobs, 'Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations', *Mathematics*, vol. 10, no. 15, p. 2639, Jul. 2022, doi: 10.3390/math10152639.

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

[15] B. M. Chen-Charpentier and H. V. Kojouharov, 'An unconditionally positivity preserving scheme for advection-diffusion reaction equations', *Math. Comput. Model.*, vol. 57, pp. 2177–2185, 2013, doi: 10.1016/j.mcm.2011.05.005.

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

[17] B. Drljača and S. Savović, 'Unconditionally positive finite difference and standard explicit finite difference schemes for power flow equation', *Univ. Thought - Publ. Nat. Sci.*, vol. 9, no. 2, pp. 75–78, 2019, doi: 10.5937/univtho9-23312.

[18] N. Ndou, P. Dlamini, and B. A. Jacobs, 'Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations', *Mathematics*, vol. 10, no. 15, Art. no. 15, Jan. 2022, doi: 10.3390/math10152639.

[19] Á. Nagy, I. Omle, H. Kareem, E. Kovács, I. F. Barna, and G. Bognar, 'Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation', *Computation*, vol. 9, no. 8, p. 92, 2021.

[20] E. Kovács, 'New Stable, Explicit, First Order Method to Solve the Heat Conduction Equation', *J. Comput. Appl. Mech.*, vol. 15, no. 1, pp. 3–13, 2020, doi: 10.32973/jcam.2020.001.

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

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

[23] E. Kovács, Á. Nagy, and M. Saleh, 'A New Stable, Explicit, Third-Order Method for Diffusion-Type Problems', *Adv. Theory Simul.*, 2022, doi: 10.1002/adts.202100600.

[24] E. Kovács and Á. Nagy, 'A new stable, explicit, and generic third-order method for simulating conductive heat transfer', *Numer. Methods Partial Differ. Equ.*, vol. 39, no. 2, pp. 1504–1528, Nov. 2023, doi: 10.1002/num.22943.

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