# TESTING AND IMPROVING A NON-CONVENTIONAL UNCONDITIONALLY POSITIVE FINITE DIFFERENCE METHOD

**Mahmoud Saleh**
*PhD student, Institute of Physics and Electric Engineering, University of Miskolc*
*3515 Miskolc, Miskolc-Egyetemváros, e-mail: mhmodsalh84@gmail.com*

**Endre Kovács**
*associate professor, Institute of Physics and Electric Engineering, University of Miskolc*
*3515 Miskolc, Miskolc-Egyetemváros, e-mail: kendre01@gmail.com*

**Gábor Pszota**
*associate professor, Institute of Physics and Electric Engineering, University of Miskolc*
*3515 Miskolc, Miskolc-Egyetemváros, e-mail: pszotag@gmail.com*

***Abstract***

*In this article, we applied the Unconditionally Positive Finite Difference (UPFD) method of Chen-Charpentier and Kojouharov to a diffusion-type partial differential equation (PDE), the so-called heat equation. Stability and convergence properties have been verified numerically for specific initial and boundary conditions. We also elaborated and tested the UPFD method with adaptive step size control. We have demonstrated that this version can be more efficient by reducing the time needed for solving the heat equation in one dimension.*

***Keywords:*** *explicit numerical methods, heat conduction equation, parabolic PDEs, adaptive time step size, UPFD method*

## 1. Introduction

Heat conduction plays a significant role in many applications of mechanical and electrical engineering systems and other aspects of life, and one does not need to go very far to see some application areas of heat transfer. For instance, heat transfer equipment such as heat exchangers, boilers, condensers, radiators, heaters, furnaces, refrigerators, and solar collectors are designed primarily on the basis of heat transfer analysis [1]. The governing equations of conductive heat transfer has been formulated by Fourier. We investigate this parabolic partial differential equation in the following, one-dimensional form:

$$\frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2}, \tag{1}$$

where $t$ [s] is the time, $x$ [m] is the space coordinate, $T = T(x, t)$ [K] is the temperature and $D$ [m$^2$/s] is the constant of thermal diffusivity. Although in the special case of homogeneous media, where $D$ is a constant, the general analytic solution of this equation is well-known, in general, it is necessary to use numerical methods to obtain the solution. Following the well-known method of lines, one approximates the space derivatives by finite differences and after this, the spatial coordinates are no longer present explicitly as independent variables. The remaining independent variable is the time, which means that one has a system of first order ordinary differential equations. This ODE system approximates the original PDE better if the space step size is smaller. However, this yields larger number of equations in the

ODE system and therefore slower simulation. Moreover, the system of equations often shows high degree of stiffness which means that very small time step sizes are required to avoid instability thus the user cannot change the time step size freely considering the accuracy requirements only [2]. Since the rate of change of the *T* functions and even the degree of stiffness during the course of the simulation can vary, using a fixed but small time step implies a heavy computational burden. A dynamically adaptive time step algorithm can automatically adjust the time step according to the accuracy requirements and the degree of stiffness, resulting in a more optimal use of computational resources [3]. Successful adaptive methods may lead to substantial savings in computational work for a given accuracy [4].

The aim of this paper is to test numerically the convergence and consistency properties of a new class of finite difference method, the so-called Unconditionally Positive Finite Difference (UPFD) method proposed by Chen-Charpentier and Kojouharov [5]. Also, we elaborated an adaptive algorithm based on UPFD method. All the schemes used in this paper were coded using MATLAB.

## 2. Unconditionally Finite Difference Method

In 2013, Chen-Charpentier and Kojouharov introduced a new class of Finite Difference methods for advection–diffusion reaction equations that guarantees the positivity of the solutions, independent of the time step and mesh size [5]. For simplicity, they considered the one-dimensional advection-diffusion equation with linear decay:

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} - D\frac{\partial^2 c}{\partial x^2} = -Kc, \qquad (x,t) \in [0, x_{max}] \times [0, t_{max}] \tag{2}$$

for the unknown concentration function $c = c\ (x,\ t)$, with appropriate boundary and initial conditions. The parameters $u$, *D* and *K* are positive constants.

We divide the space interval $[0, x_{max}]$ into subdivisions $x_0 < x_1 <, \dots, < x_N$. While $x_i = (i-1)\Delta x, i = 1, \dots N, \Delta x = ({}^{x_{max}}/_{N} - 1)$ and of course $x_N = x_{max}$. We also divide the time interval $[0, t_{max}]$ using equal time steps of size $\Delta t$. Let $c_i^n$ be the approximation of the unknown concentration function at time step *n* and node *i*. The UPFD scheme can be written:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} + u\frac{c_i^{n+1} - c_{i-1}^n}{\Delta x} - D\frac{c_{i+1}^n - 2c_i^{n+1} + c_{i-1}^n}{\Delta x^2} = -Kc_i^{n+1}. \tag{3}$$

Applying this scheme to Eq. (1) the terms $u$ and $K$ will be cancelled out, thus one can write

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} - D\frac{T_{i+1}^n - 2T_i^{n+1} + T_{i-1}^n}{\Delta x^2} = 0, \tag{4}$$

where $T_i^n$ is the numerically calculated value of the temperature at time step *n* and node *i*. Although this scheme seems to be implicit at the first glance, it can be reordered into the following explicit form:

$$T_i^{n+1} = \frac{\widehat{D}(T_{i+1}^n + T_{i-1}^n) + \frac{T_i^n}{\Delta t}}{\frac{1}{\Delta t} + 2\widehat{D}}, \tag{5}$$

where $\widehat{D} = D/\Delta x^2$.

Eq. (5) represents the scheme that we will use to find the approximated solution of our problem described above in Eq. (1).

## 2.1. Convergence

The *global* error or the so-called pointwise error at node *i* can be defined in the following way:

$$e_i^n = T_i^n - \hat{T}_i^n. \tag{6}$$

where $\hat{T}_i^n$ is the analytical solution corresponds to $T_i^n$. The maximum global error is the largest absolute value of $e_i^n$.

The local error refers to the error incurred over a single step. To be more specific, if one solves the same differential equation only for one-time step, using the $T_i^{n-1}$ values as initial conditions, then one obtains the local error as the difference between the local analytical solution $\hat{T}_i^{n*}$ and the local numerical solution:

$$\varepsilon_i^n = T_i^n - \hat{T}_i^{n*}. \tag{7}$$

By definition, the scheme is said to be consistent (convergent) at time step $t^n$ and node $i$ if the local (global) error tends to zero as $\Delta t$ and $\Delta x$ tend to zero [6]. In other words:

$$\varepsilon_i^n \to 0 \;\; (e_i^n \to 0) \; as \;\; \Delta t \to 0 \;\; and \;\; \Delta x \to 0 \tag{8}$$

In this paper our task is not the analytical examination of the convergence and stability properties, but to examine numerically the behaviour of the global error as a function of the time step size, since it hasn’t been done in the original paper of Chen-Charpentier and Kojouharov. To verify the convergence of the method, we chose to solve the heat conduction equation with the common Dirichlet boundary conditions. We consider the heat conduction Eq. (1) with *D*=1:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \;\; (x,t) \in [0,\pi] \times [0, 0.3] \tag{9}$$

Subjected to the following initial and boundary conditions:

$$T(x,0) = 10\,sin(x) + 77\,sin(10x)\,, x \in [0,\pi]. \tag{10}$$

$$T(0,t) = T(\pi,t) = 0,\;\; t \in [0, 0.3]. \tag{11}$$

The analytical solution of this problem is of the form:

$$T(x,t) = 10\,sin(x)\,e^{-t} + 77\,sin(10x)\,e^{-100t}. \tag{1}$$

First, we examined the behaviour of the maximum global error for four different, fixed space steps. It means that the space domain [0, $\pi$] was discretized into *N* nodes, were *N* was 25, 50, 100 and 200. Also, for each case we repeated the numerical experiment with many fixed time step sizes. For each of those time steps, we computed the maximum global error at the end of time interval $[0, 0.3]$. If we choose smaller final times instead of 0.3, the solution would be too close to the original initial function. On the other hand, for larger final times the terms in the initial function would decay very much and their contribution in the final solution and the errors would become negligible. Between these two extremes, 0.3 seemed to be a good compromise. Figure 1. shows that the error decreases with the first power of the time step size $\Delta t$ until the step size reaches a certain value, and then the error stops to decrease and tends to a constant, no matter how small the time step is. This remaining error is due to the space discretization, which is underpinned by the fact that, as one can see in the figure 1, the smaller the space step size, the smaller this remaining error is. In Figure 2. we present the temperature as a function of time for the centre node $n \approx N/2$ for different time and space step sizes. One can see that inaccuracy here means that the temperature variable changes too slowly compared to the exact solution and no unphysical oscillations are present like in the case of many explicit methods for larger time step sizes. This "laziness" is the price one have to pay for unconditional stability and positivity preservation.

We would like to point out that in the definition of consistency and convergence, the condition $\Delta t \to 0 \ and \ \Delta x \to 0$ contains some ambiguity, as the rate by which $\Delta t$ and $\Delta x \to 0$ is not fixed. If the errors tend to zero in any case, we talk about unconditional consistency/convergence, but if there is a constrain on the relative rate of convergence, the method is called conditionally consistent/convergent. The original article [5] refers to the conditional consistency of the UPFD method, thus we performed a second series of numerical experiments to investigate this property. For this purpose, we tend to zero with $\Delta x$ and $\Delta t$ together in such a way that the formula:

$$\Delta t = \Delta x^b, \tag{2}$$

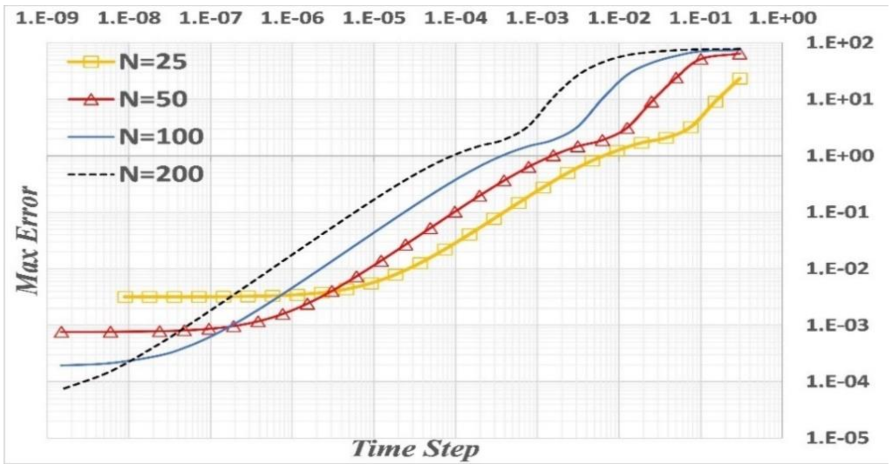

***Figure 1.*** *The maximum global error as a function of the time step size calculated at the end of the time interval for different number of nodes (logarithmic scale)*

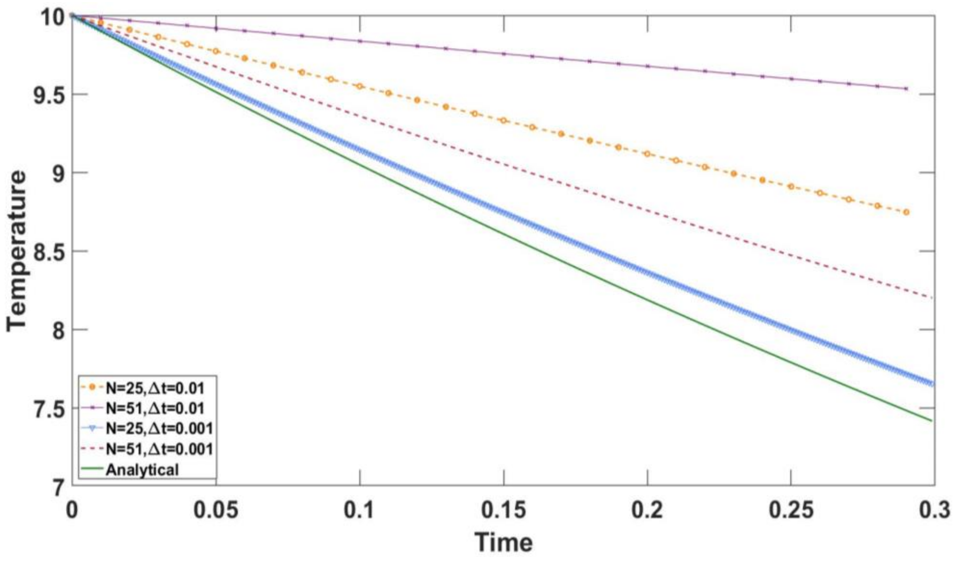

***Figure 2.*** *The temperature as a function of time for centre node for different time step sizes*

where $b$ is constant. The constant $b$ was given different values. For each value of $b$, we investigated the maximum error at different time steps at the end of time the interval. The results are presented in Figure 3. We found that when $b$ is larger than 2, e.g. $b = 2.1$, the error tends to zero when both $\Delta t$ *and* $\Delta x$ tend to zero. On the other hand, when $b \leq 2$ the error tends to a nonzero constant. Thus, the scheme converges only if the time step size goes to zero faster than the second power of the space step.

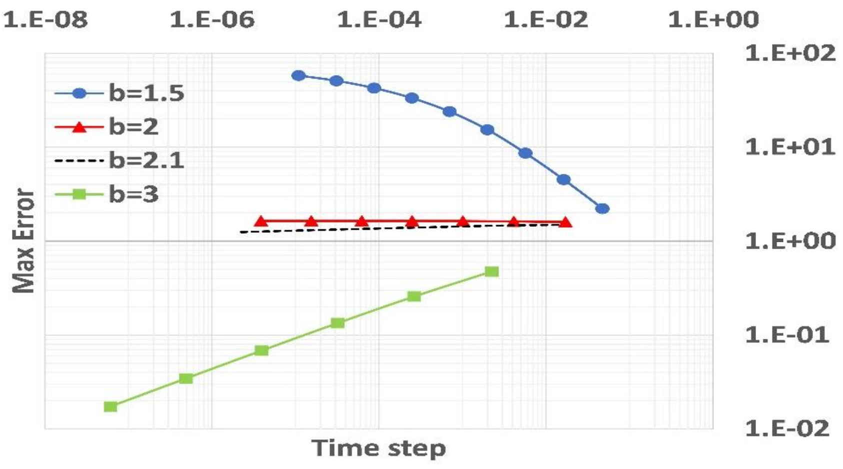

***Figure 3.*** *The max Error vs time step size calculated at the end of the time interval for different values of b (logarithmic scale). We have to emphasize that for any time step there is a corresponding space step determined by formula (11). For b=2.1, the error tends to zero, albeit very slowly.*

We note that for large values of the space and time step sizes the error can slightly increase with decreasing $\Delta t$ and $\Delta x$. This phenomenon does not contradict with the convergence of the method, as the convergence and consistency statements apply when $\Delta t$ and $\Delta x$ tend to zero and they do not state anything for large values of $\Delta t$ and $\Delta x$.

## 3. Adaptive time step size for UPFD method

In order to improve the time needed to execute the code, an adaptive time step controller is a good solution. The core idea of this controller is to increase the time step or decrease it when it is necessary. In order to be able to do this, we need an estimation of the local error at the end of each time step. For this purpose, we used the UPFD method with one full time step and, at the same time, with two halved time steps to compute the numerical values at each node [7]. It means that, at the end of the *n*th time step, we have two different solution: $T(h)$ and $T\left(\frac{h}{2}+\frac{h}{2}\right)$. Of course, the second one is more accurate, so the first one is used only to make the following average local error-estimation:

$$\frac{1}{N} sum \left|T\left(\frac{h}{2}+\frac{h}{2}\right) - T(h)\right|. \qquad (14)$$

After calculating this, the code compares this mean local error to some tolerance value (set by the user). Based on that comparison the code decides if it has to increase the time step, keep it or decrease it. The algorithm used to decide the size of the time step is presented in Figure 4.

When we applied this adaptive time step method for the problem (8-11) we obtained that the adaptive controller does not fulfil the expectations, but in fact it makes the code slightly slower.

```
If (error_rate<0.8*tolerance)
    Accept the solution and increase the time step for the next iteration.
    $\Delta t_{new} = 1.1 \times \Delta t_{old}$
elseif (error_rate >= 0.8*tolerance && error_rate < 1*tolerance)
    Accept the solution and keep the time step for the next iteration.
    $\Delta t_{new} = \Delta t_{old}$
elseif (error_rate >= 1*tolerance && error_rate < 2*tolerance)
    Accept the solution and decrease the time step for the next iteration.
    $\Delta t_{new} = 0.5 \times \Delta t_{old}$
else
    Reject the solution, decrease the time step and repeat the iteration.
    $\Delta t_{new} = 0.1 \times \Delta t_{old}$
end
```

***Figure 4.*** *The algorithm used to decide the size of time-step*

The reason for this is that any produced error is absorbed by the edges of the system because of the zero Dirichlet boundary conditions, thus the time step controller mostly wastes the CPU time. That is why we decided to apply Neumann boundary conditions and use following problem:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}\ ,\ (x,t) \in [0,\pi] \times [0,4]. \tag{3}$$

$$T(x,0) = 10\,sin(x) + 77\,sin(10x)\,, x \in [0,\pi]. \tag{4}$$

$$\frac{\partial T(0,t)}{\partial x} = \frac{\partial T(\pi,t)}{\partial x} = 0. \tag{5}$$

Physically Eq. (17) is equivalent with the thermal isolation of the system at the ends. The reference solution of the discretized ODE system is produced by the well-established MATLAB ODE solver `ode45` with extremely small tolerance. The code was also written in MATLAB.

We compared the UPFD method with Adaptive-UPFD method, and found that for any concrete level of accuracy the adaptive-UPFD method is faster than the original UPFD method, i.e. shorter time (CPU time) is necessary to execute the code. **Figure** Figure 5. shows the result clearly, as one can see that the continuous line is closer to the bottom left corner of the diagram where an ideal code would take place. We obtained similar results for different values of the parameters of the algorithm (i.e. when numbers like 0.8 and 0.5 in Figure 4. are replaced by other values), albeit the distance of the curves was slightly smaller.

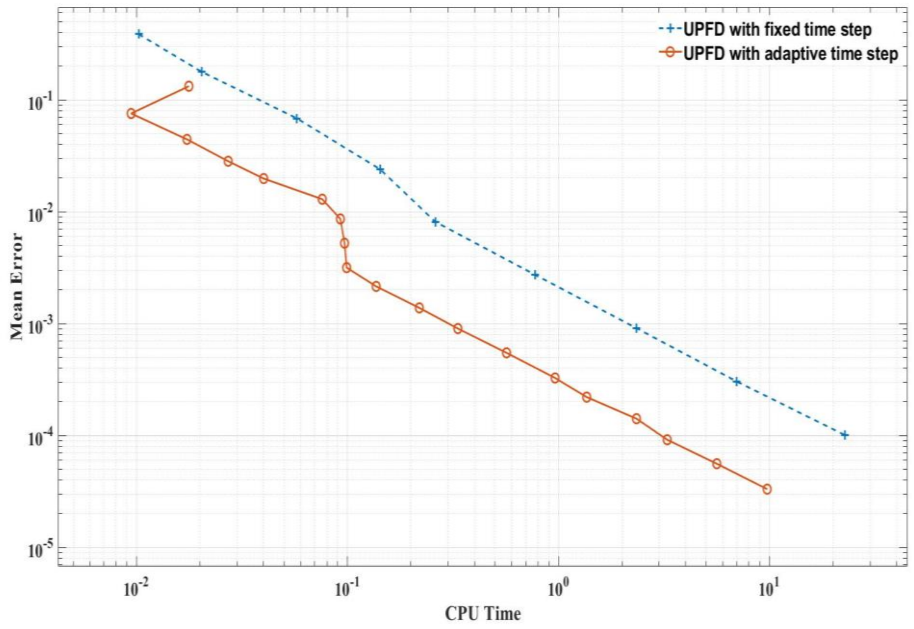

***Figure 5.*** *The mean error vs CPU time calculated at the end of the time interval for each order of accuracy (Logarithmic Scale).*

## 4. Summary

1. For any fixed mesh, the UPFD method was found to be a convergent time-integrator with order 1.
2. The UPFD method is conditionally consistent and convergent if the space step size $\Delta x$ goes to zero, more precisely, the time step size $\Delta t$ has to go to zero faster than the second power of $\Delta x$ to achieve that the error goes to zero as well.
3. In case of Neumann boundary conditions, supposing the same level of accuracy, the adaptive-UPFD method needs less time to solve the problem than the original UPFD method. However, this is not true in the case of Dirichlet boundary conditions.
4. Although the adaptive time step size controller implemented in this paper can, in some cases, significantly reduce the running time for a given accuracy, it demands extra calculations to obtain the necessary error-estimation. Adaptive step sizes controlling is more effective when the original method itself produces two different solution at the end of each time step, which is only possible if the algorithm consists of at least two stages like the (explicit) trapezoidal rule. These methods are called embedded methods, and the most popular examples are the Bogacki-Shampine-Runge-Kutta, the Runge-Kutta-Fehlberg and the Runge-Kutta-Cash-Karp methods [7]. However, these explicit methods are only conditionally stable and definitely not unconditionally positive. In our future work we try to construct explicit, unconditionally stable embedded methods for diffusion and heat conduction problems.

## References

[1] Çengel, Y. A., Ghajar, A. J.: *Heat and mass transfer fundamental and applications*, vol. 5, New York: McGraw-Hill Education, 2015.

[2] Alberdi Celaya, E., Anza Aguirrezabala, J. J., Chatzipantelidis, P.: *Implementation of an Adaptive BDF2 Formula and comparison with the MATLAB Ode15s*, Procedia Computer Science, vol. 29, p. 1014–1026, 2014. **https://doi.org/10.1016/j.procs.2014.05.091**

[3] Alikhani, J., Shoghli, B., Bhowmik, U. K., Massoudieh, A.: *An adaptive time-step backward differentiation algorithm to solve stiff ordinary differential rquations: Application to solve activated sludge models*, American Journal of Computational Mathematics, vol. 6, pp. 298-312, 2016. **https://doi.org/10.4236/ajcm.2016.64031**

[4] Eriksson, K., Johnson, C.: *Adaptive finite element methods for parabolic problems*, SIAM Journal on Numerical Analysis, vol. 28, pp. 43-77, 1991. **https://doi.org/10.1137/0728003**

[5] Chen-Charpentier, B. M., Kojouharov, H. V: *An unconditionally positivity preserving scheme for advection–diffusion reaction equations*, Mathematical and Computer Modelling, vol. 57, pp. 2177-2185, 2013. **https://doi.org/10.1016/j.mcm.2011.05.005**

[6] Causon, D., Mingham, C.: *Introductory finite difference methods for PDEs*, Manchester Metropolitan University, 2010.

[7] Atkinson, K., Han, W., Stewart, D.: *Numerical solution of ordinary differential equations*, Iowa: A John Wiley & Sons, 2009. **https://doi.org/10.1002/9781118164495**