Multidiszciplináris tudományok, 10. kötet. (2020) 4 sz. pp. 339-348 **https://doi.org/10.35925/j.multi.2020.4.37**

CONSTRUCTION AND INVESTIGATION OF NEW NUMERICAL ALGORITHMS FOR THE HEAT EQUATION

Part 2

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

Ádám Nagy
Teaching Assistant, Institute of Physics and Electric Engineering, University of Miskolc
3515 Miskolc, Miskolc-Egyetemváros, e-mail: fizadam@uni-miskolc.hu

Endre Kovács
Associate Professor, Institute of Physics and Electric Engineering, University of Miskolc
3515 Miskolc, Miskolc-Egyetemváros, e-mail: fizendre@uni-miskolc.hu

Abstract

This paper is the second part of a paper-series in which we create and examine new numerical methods for solving the heat conduction equation. Now we present numerical test results of the new algorithms which have been constructed using the known, but non-conventional UPFD and odd-even hopscotch methods in Part 1. Here all studied systems have one space dimension and the physical properties of the heat conducting media are uniform. We also examine different possibilities of treating heat sources.

Keywords: *explicit numerical methods, heat equation, parabolic PDEs, hopscotch method, UPFD*

1. Introduction and the description of the used numerical methods

This paper is the second part of a longer paper-series on our new methods to simulate heat conduction phenomena. In the first part we used the unconditionally positive finite-difference (UPFD) method of Chen-Charpentier et al. [1], and the odd-even hopscotch algorithm of Gordon [2], and Gourlay [3], [4] to construct new schemes and then we analytically examined them. Now, in Part 2 we numerically investigate the performance of these methods compared to the original UPFD and odd-even hopscotch methods in the simplest, materially homogeneous system, where heat conduction is modelled by the heat equation:

$$\frac{\partial u}{\partial t} = \alpha \Delta u + q \,, \tag{1}$$

where $u = u(x,t)$ is the unknown temperature, a function of the space variable *x* and the time *t*, α is the thermal diffusivity (considered to be a constant in this part), and $q = q(x)$ is the intensity of external or internal heat sources, respectively. As we explained in Part 1, the space derivatives are always discretized by the usual most standard difference formula. Following the logic of Chen-Charpentier et al. for the time discretization and inserting the heat source term q_i we obtain

$$\frac{u_i^{n+1}-u_i^{n}}{\Delta t}=\alpha\frac{u_{i-1}^{n}-2u_i^{n+1}+u_{i+1}^{n}}{\Delta x^2}+q_{i}\,.$$

With the notations $r=\frac{2\alpha h}{\Delta x^2}>0$ and $h=\Delta t$ it can be rearranged into the following explicit form:

Algorithm 1a (UPFD method, source term inside).

$$u_i^{n+1}=\frac{u_i^{n}+\frac{r}{2}\left(u_{i-1}^{n}+u_{i+1}^{n}\right)+q_i h}{1+r}\,.$$

It is also possible to place the $q_i h$ term outside of the numerator, by which we obtain

Algorithm 1b (UPFD method, source term outside).

$$u_i^{n+1}=\frac{u_i^{n}+\frac{r}{2}\left(u_{i-1}^{n}+u_{i+1}^{n}\right)}{1+r}+q_i h$$

It is easy to see that if the time step size *h* tends to zero, $1+r\to 1$ thus the two versions tend to each other. So when the UPFD (or implicit Euler) formulas are used, which contain fractions, there are two possibilities to treat the source term and it cannot be decided *a priori* which one is more effective. So we can write two versions of the successive displacement UPFD (where the already obtained u_{i-1}^{n+1} values are used to calculate u_i^{n+1}) and the odd-even hopscotch method as well.

Algorithm 2a (successive displacement UPFD method, source term inside).

$$u_i^{n+1}=\frac{u_i^{n}+\frac{r}{2}\left(u_{i-1}^{n+1}+u_{i+1}^{n}\right)+q_i h}{1+r}\,.$$

Algorithm 2b (successive displacement UPFD method, source term outside).

$$u_i^{n+1}=\frac{u_i^{n}+\frac{r}{2}\left(u_{i-1}^{n+1}+u_{i+1}^{n}\right)}{1+r}+q_i h\,.$$

Algorithm 3a (original odd-even Hopscotch, source term inside).

Stage 1. Take a time step with the explicit Euler method for every other nodes, e.g. where *n*+*i* is odd:

$$u_i^{n+1}=u_i^{n}+r\left(\frac{u_{i-1}^{n}+u_{i+1}^{n}}{2}-u_i^{n}\right)+q_i h\,.$$

Stage 2. Take a time step with the implicit Euler method for the remaining nodes, e.g. where *n*+*i* is even:

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

Algorithm 3b (original odd-even Hopscotch, source term outside). Stage 1 is the same as in Algorithm 3a, but the second stage is different:

Stage 2. Take a time step with the implicit Euler method for the remaining nodes, e.g. where *n+i* is even:

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

In case of the odd-even hopscotch, where the explicit Euler formulas are used in both stages, there are no different possibilities:

Algorithm 4 (explicit+explicit odd-even hopscotch). Stage 1 is the same as in Algorithm 3a, but the second stage is different:

Stage 2. Take a time step with the explicit Euler method again for the remaining nodes,:

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

In the case of the next algorithm, the UPDF method is used at the first stage, so we also have two possibilities.

Algorithm 5a (UPFD+Explicit Euler odd-even hopscotch).

Stage 1. Take a time step with the UPDF method for every other nodes, e.g. where *n+i* is odd:

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

Stage 2. Take a time step with the explicit Euler method for the remaining nodes, e.g. where *n+i* is even:

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

Algorithm 5b (UPFD+Explicit Euler odd-even hopscotch).

Stage 1. Take a time step with the UPDF method for every other nodes, e.g. where *n+i* is odd:

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

Stage 2. Same as in Algorithm 5a.

In the last algorithm UPFD formulas are used in both stages, thus we have four possibilities.

Algorithm 6a-a (UPFD+UPFD odd-even hopscotch, inside-inside).
Stage 1. Same as in Algorithm 5a.
Stage 2. Same as in Algorithm 3a.
Algorithm 6b-b (UPFD+UPFD odd-even hopscotch, outside-outside).
Stage 1. Same as in Algorithm 5b.
Stage 2. Same as in Algorithm 3b.
Algorithm 6a-b (UPFD+UPFD odd-even hopscotch, outside-outside). Stage 1 is the same as in Algorithm 6a-a while Stage 2 is the same as in Algorithm 6b-b.
Algorithm 6b-a (UPFD+UPFD odd-even hopscotch, outside-outside). Stage 1 is the same as in Algorithm 6b-b while Stage 2 is the same as in Algorithm 6a-a.

2. Numerical examples

In this section, we numerically investigate the behaviour of these methods. We solve PDE (1) for $\alpha = 1$ on the interval $x \in [0, \pi]$ with different initial and boundary conditions. We perform equidistant discretization of the space variable by setting

$$x_j = j\Delta x,\ j = 0,...,N-1, \quad \Delta x = \pi / (N-1)\ ,$$

where the number of nodes is N=100. The solutions are examined and compared at final time $t_{\text{fin}} = 0.3$. We define the (global) error as the average of the absolute value of the difference between the reference temperature $u_{\text{i}}^{\text{ref}}$ and the temperature $u_{\text{i}}^{\text{num}}$ obtained by our numerical methods at t_{fin}, the end of the examined time interval:

$$Error = \frac{1}{N}\sum_{i=1}^{N}\left|u_{\text{i}}^{\text{ref}}(t_{\text{fin}}) - u_{\text{i}}^{\text{num}}(t_{\text{fin}})\right|.$$

The reference solution is either the analytical solution of the PDE or a numerical solution obtained by applying a very accurate time integrator to the spatially discretized PDE.

2.1. Case study 1

In this case the initial condition function is identically zero:

$$u(x, t=0) \equiv 0.$$

The simplest zero Dirichlet boundary conditions are used

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

while the intensity of the heat source term is the following function:

$$q(x,t) = q_1 \sin(x) + q_2 \sin(2x),$$

where we set $q_1 = 1,\ q_2 = 2$. It is easy to check that the analytical solution of this problem is

$$u(x,t) = q_1 \sin(x)\left(1 - \mathrm{e}^{-t}\right) + \frac{q_2}{4}\sin(2x)\left(1 - \mathrm{e}^{-4t}\right).$$

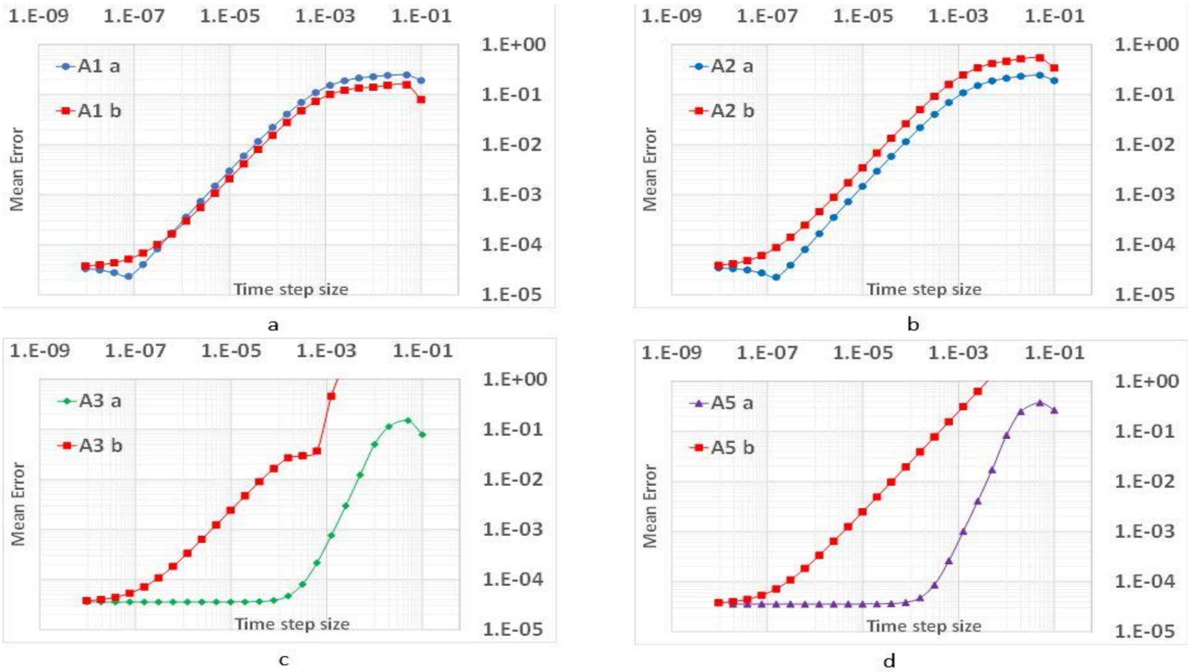

Figure 1. *The error as a function of the time step size for the algorithms with different treatments of the source term. A1a means Algorithm 1a (UPFD method, source term inside), etc.*

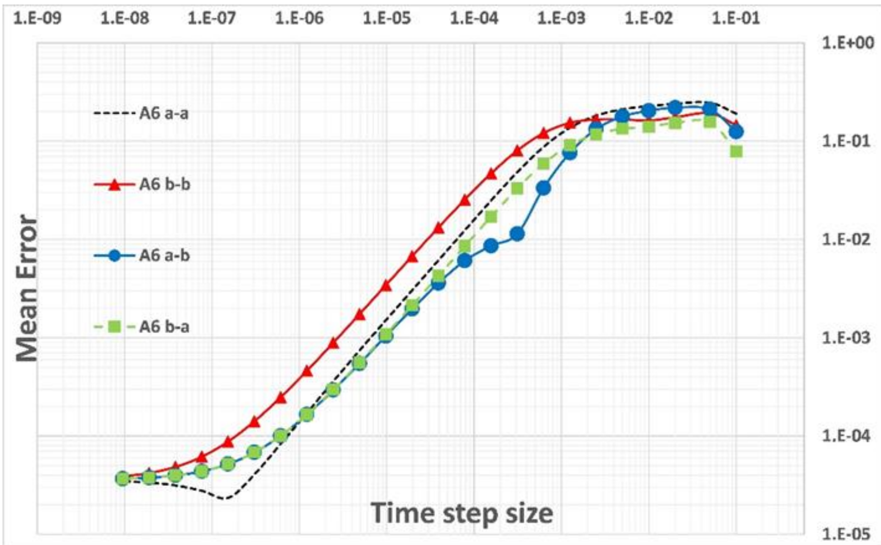

Figure 2. *The error as a function of the time step size for the four different treatments of the source term. A6a-a means Algorithm 6a (UPFD+UPFD odd-even hopscotch, inside-inside), etc.*

We note that all analytical solutions in this paper-series are constructed by the authors, but similar solutions can be found in standard textbooks [5, p. 223] and many webpages. In Fig. 1 we present the errors as a function of the time step size for Algorithms 1, 2, 3 and 5, where there are two possibilities for the treatment of the heat source term, explained in the previous section. In Fig. 2 the results are presented in the four versions of Algorithm 6. Based on these figures, we choose the version of the source-treatment that produces the smallest error in each case. These curves are shown in Fig. 3 together with the error function of Algorithm 4. One can see that the error tends to the same nonzero residual value in all cases. This small non-vanishing error is due to the discretization of the space variable. More precisely, it is the truncation error

$$\varepsilon_i = -\frac{\Delta x^2}{12} f''''(\eta_i)$$

of the central difference formula [5, p. 7], which has been accumulated during the subsequent time steps.

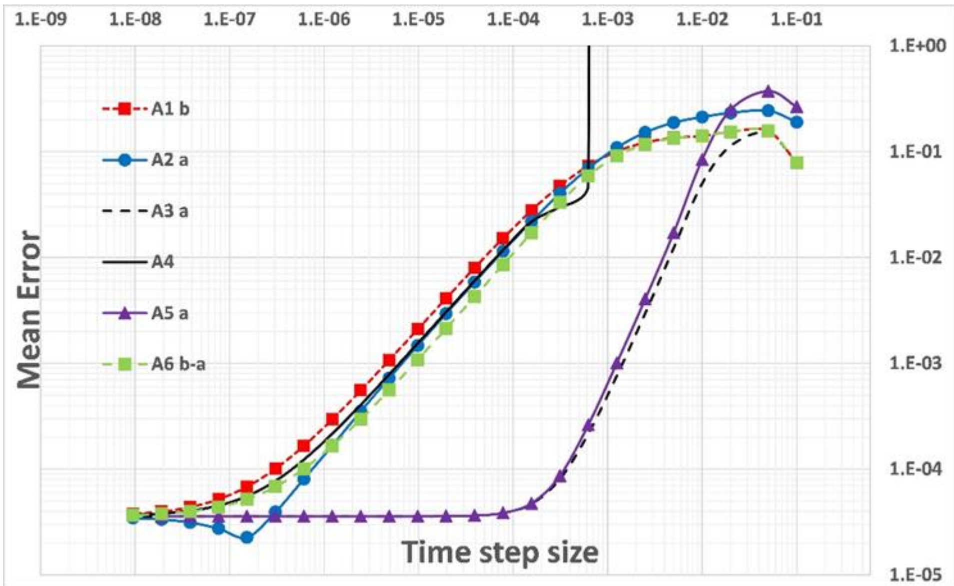

Figure 3. *The error as a function of the time step size for the six different algorithms*

2.2. Case study 2

The initial condition is the following function:

$$u(x,t=0) = 10\sin(x) + 77\sin(10x),$$

and the boundary conditions are zero Dirichlet again:

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

while the external heat source term is zero: $q \equiv 0$. It is easy to check that the analytical solution of this problem is

$$u(x,t) = 10\sin(x)\mathrm{e}^{-t} + 77\sin(2x)\mathrm{e}^{-4t}.$$

The behaviour of the error functions are presented in Fig. 4.

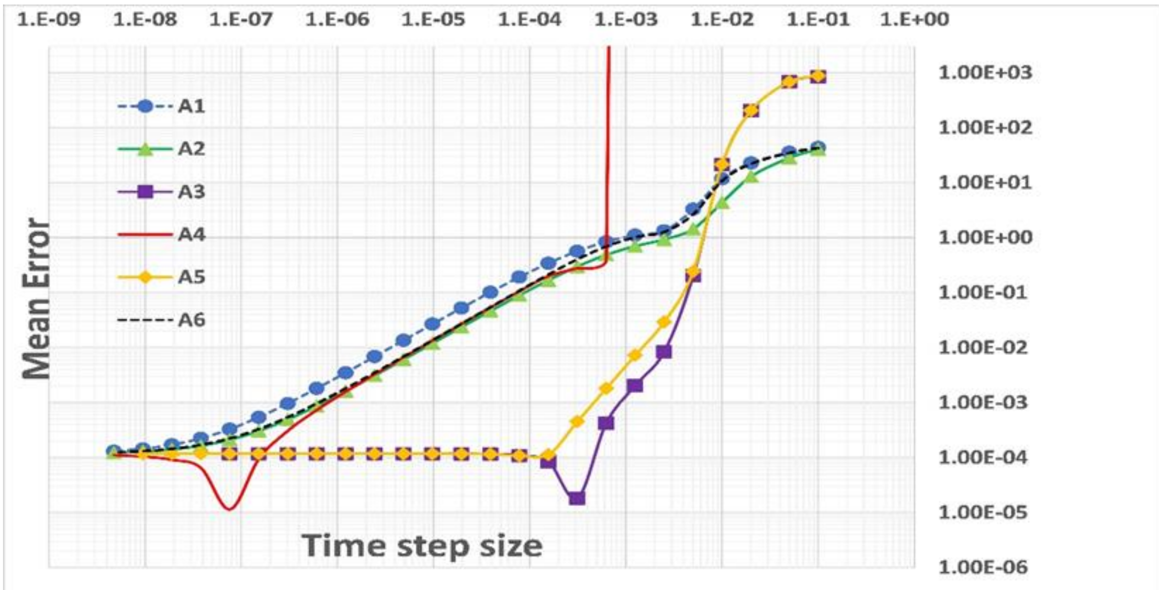

Figure 4. *The error as a function of the time step size for the six different algorithms.*

2.3. Case study 3

The simulation starts from the following initial condition:

$$u(x,t=0)=10\cos(2x) ,$$

and now we consider periodic boundary conditions

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

while the external heat source term is zero everywhere. The analytical solution of this problem is the following function:

$$u(x,t)=10\cos(2x)\mathrm{e}^{-4t} .$$

The errors are presented in Fig 5.

2.4. Case study 4

The initial function is a quadratic function:

$$u(x,t=0)=x^2 ,$$

while the external heat source term is zero. Now there is a straightforward analytical solution

$$u(x,t)=x^2+2t ,$$

if we suppose the following time-dependent Dirichlet boundary conditions

$$u(x=0,t)=2t, \quad u(x=\pi,t)=1+2t .$$

We note that if the function is a polynomial with degree less than four, the central difference formula gives the exact value of the second derivative, e.g.

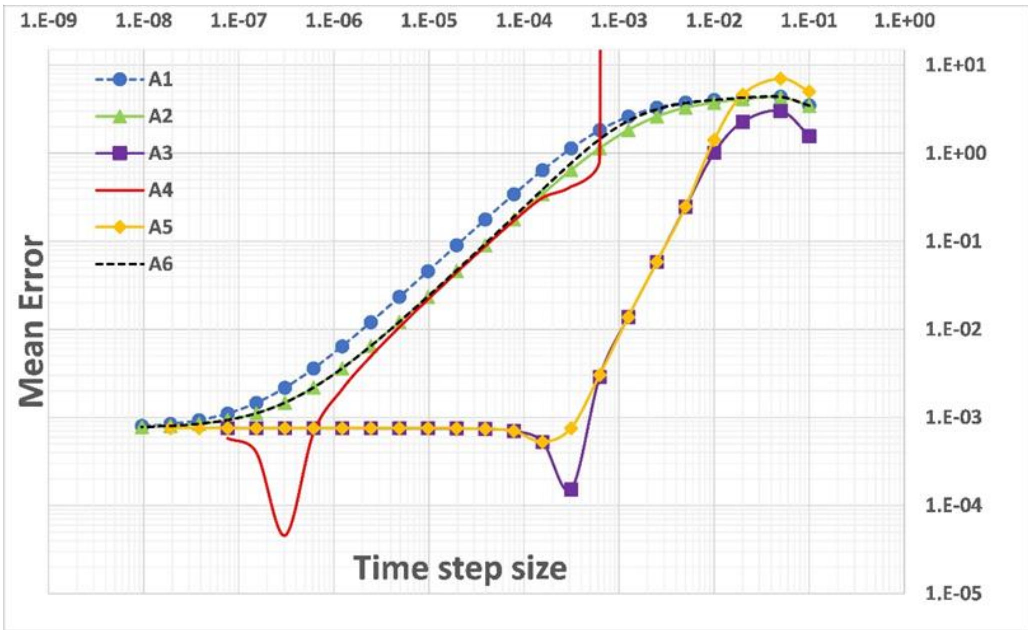

Figure 5. *The error as a function of the time step size for the six different algorithms.*

$$\frac{\partial^2}{\partial x^2} x^2 \approx \frac{(x-\Delta x)^2 + (x+\Delta x)^2 - 2x^2}{\Delta x^2} = 2 .$$

In this case, the explicit Euler, the implicit Euler and the original odd-even Hopscotch method provides the exact values in theory, thus only the round-off errors remain. Therefore the errors are presented only for five algorithms in Fig. 6.

2.5. Case study 5

The initial condition is set to:

$$u(x, t=0) = 100\cos(x) ,$$

but now we consider zero Neumann boundary conditions, which means there is thermal isolation at the border of the system. Here the external heat source term is the following function of space:

$$q(x) = 2x - x^2 .$$

For the reference solution, we use the MATLAB routine `ode45` to solve the ODE system obtained by the spatial discretization of the problem. Because of this, there is no residual error due to space discretization and the errors are decreasing for decreasing time step size h until the round-off errors appear. From Fig. 7, where the behaviour of the errors are presented, it is clear that Algorithms 3 and 5 converge much faster than the others.

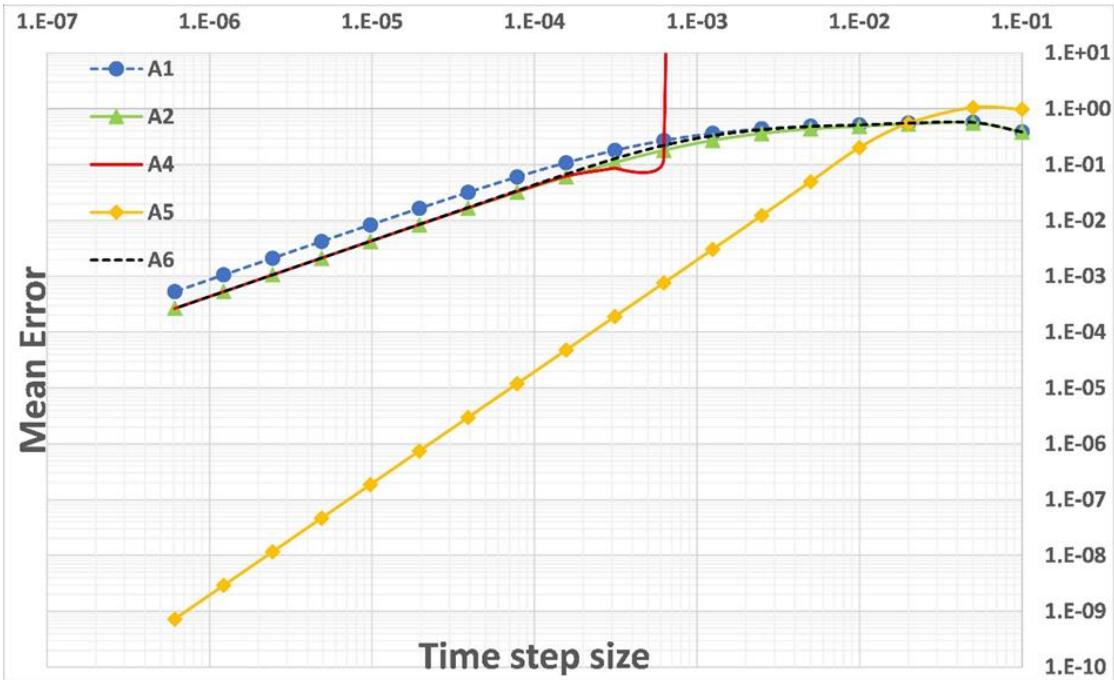

Figure 6. *The error as a function of the time step size for five different algorithms.*

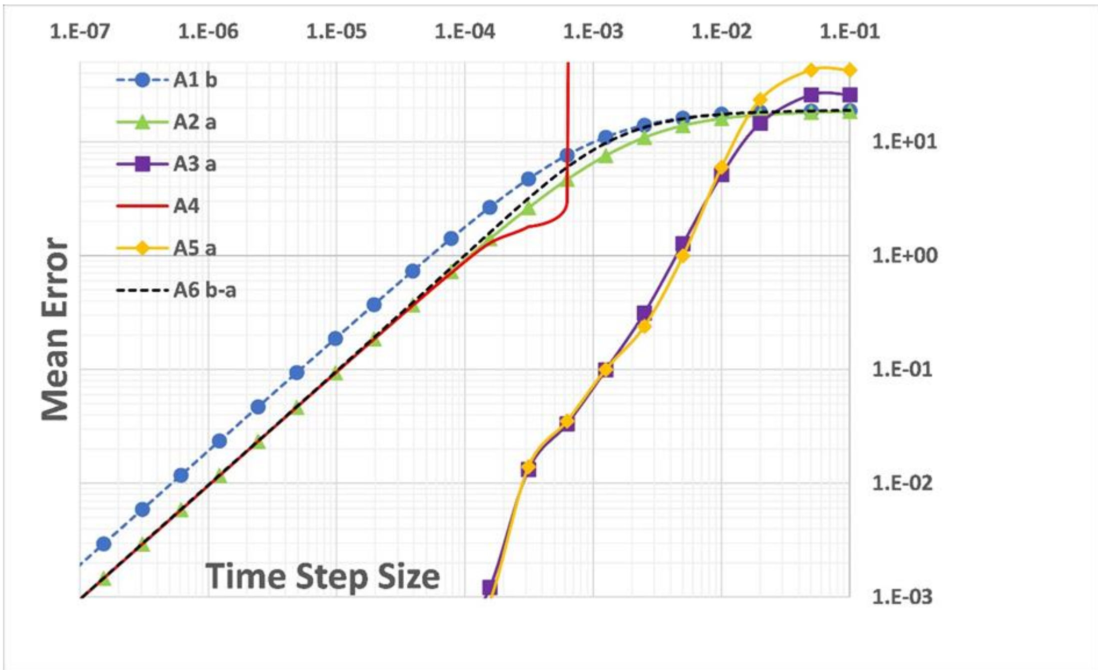

Figure 7. *The error as a function of the time step size for six different algorithms.*

3. Summary and conclusions

In this paper-series, we use two known, but non-conventional algorithms, namely the UPFD and the odd-even hopscotch methods to create new numerical algorithms for the solution of the heat equation. In Part 1 we defined and analysed the old and new methods such as convergence, stability and positivity in one space dimensional homogeneous media without the heat source term. In this (second) part, five numerical case studies are presented and all the six algorithms have been examined in each of them. The same equidistant grid has been used to discretize the one dimensional physical space in each cases. Different posibilities for handling the q source term have been tested in case of five algorithms.

According to the numerical results, Algorithms 1, 2, 4 and 6 are first order in time, while the remaining Algorithm 3 and 5 seem to be second order time integrators. All methods except A4 (explicit+explicit odd-even hopscotch) seem to be stable. However, although Algorithms 3 and 5 are the most accurate for medium and small time step sizes, they are often very inaccurate in case of large time step size. We note that in Part 3 we will present more case studies for larger space dimensions as well, and in the Summary of that paper a detailed comparison of the properties and performance of the six methods will be presented.

References

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

[2] Gordon, P.: *Nonsymmetric difference equations*, J. Soc. Ind. Appl. Math., vol. 13, no. 3, (1965) pp. 667–673. **https://doi.org/10.1137/0113044**

[3] Gourlay, A. R.: *Hopscotch: a fast second-order partial differential equation solver*, IMA J. Appl. Math., vol. 6, no. 4, (1970) pp. 375–390. **https://doi.org/10.1093/imamat/6.4.375**

[4] Gourlay, A. R.: *General hopscotch algorithm for the numerical solution of partial differential equations*, IMA J. Appl. Math., vol. 7, no. 2, (1971) pp. 216–227. **https://doi.org/10.1093/imamat/7.2.216**

[5] Le Dret, H., Lucquin, B.: *Partial differential equations: Modeling, analysis and numerical approximation*, in International Series of Numerical Mathematics, Switzerland: Springer, 2016, p. 395. **https://doi.org/10.1007/978-3-319-27067-8**

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