

## NEW EXPLICIT ALGORITHM BASED ON THE ASYMMETRIC HOPSCOTCH STRUCTURE TO SOLVE THE HEAT CONDUCTION EQUATION

Issa Omle

PhD student, University of Miskolc, Institute of Physics and Electrical Engineering  
3515 Miskolc, Miskolc-Egyetemváros, e-mail: [issa.j.omle@gmail.com](mailto:issa.j.omle@gmail.com)

### Abstract

In this paper we will consider a new four-stage structure inspired by the well-known odd-even hopscotch method to construct new schemes for the numerical solution of the two-dimensional heat or diffusion equation. In this structure the first and the last time step are halved stage and therefore the time steps are shifted compared to each other for odd and even cells. We insert 10 concrete formulas into this structure to obtain  $10^4$  different combinations. First we test all of these in case of small systems with random parameters, and then examine the competitiveness of the best algorithms by testing them in case of large systems against popular solvers. We select the top 5 combinations, and demonstrate that these new methods are indeed effective if the goal is to produce results with acceptable accuracy in very short time.

**Keywords:** Odd-even hopscotch methods, Heat equation, Diffusion equation, Stiff equations, Unconditional stability

### 1. Introduction

The investigation of heat transport processes is crucial from the point of view of engineering. In this paper, we are dealing with the numerical solution of the heat conduction equation which is mathematically equivalent to the diffusion equation. The most widely used form of this equation is the following:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \quad (1a)$$

However, if the medium in which the diffusion takes place is not homogeneous, we can use a more general form:

$$c \rho \frac{\partial u}{\partial t} = \nabla (k \nabla u) \quad (1b)$$

Where, in the case of conductive heat transfer,  $u = u(\vec{r}, t)$  denotes the temperature,  $\alpha = k / (c \rho) > 0$  is the thermal diffusivity,  $c = c(\vec{r}, t)$ ,  $k = k(\vec{r}, t)$ , and  $\rho = \rho(\vec{r}, t)$  are the specific heat, the heat conductivity and the mass density, respectively. The quantities  $\alpha$ ,  $c$ ,  $k$ , and  $\rho$  are nonnegative, which has an important role in the stability of our numerical methods.

It is well known that equation (1a,b) is solved analytically as well, but these solutions are valid only for certain systems, mostly with simple geometrical shapes, homogeneous material properties and specific initial and boundary conditions, which are not really relevant to most applications. Although there are counter-examples, for example Zoppou and Knight have found analytical solutions in case of special fixed forms of spatially varying coefficients (Zoppou and Knight, 1999) for the two- and three-dimensional advection-diffusion equation. However, for space-dependent coefficients in general, one needs to use numerical calculations, therefore finding effective numerical methods are still important. In this paper we continue some previous work (Kovács and Gilicz, 2018; Kovács, 2020a; Kovács, 2020b; Saleh et al., 2020a; Saleh et al., 2020b; Saleh et al., 2020c) on developing such numerical methods and techniques to solve these equations which are explicit (thus easily parallelizable) and unconditionally stable at the same time.

One of the most well-known example of this kind of method is the odd-even hopscotch (OEH) algorithm, presented by Gordon (Gordon, 1965), and Gourlay (Gourlay, 1970; Gourlay and McGuire, 1971; Gourlay, 1971). However, it was shown recently that for stiff systems, this method can produce extremely large errors for large time step sizes (Saleh et al., 2020c). In our previous paper we modified the underlying space and time-structure and obtained shifted-hopscotch methods with much higher accuracy for stiff systems. In this report we continue that work by trying another modification. We construct and test the new algorithms in a very similar way as in that previous paper and for the sake of brevity we don't repeat too much details if not necessary.

## 2. The new structure

We present the new method only for the more general equation (1b). For this purpose, a mesh is constructed consisting of cells with heat capacity  $C_i$ , while the thermal resistance between cells  $i$  and  $j$  is  $R_{ij}$ . The behaviour of the temperature of these cells can be given by the following ordinary differential equation:

$$\frac{du_i}{dt} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j} C_i} \quad (2)$$

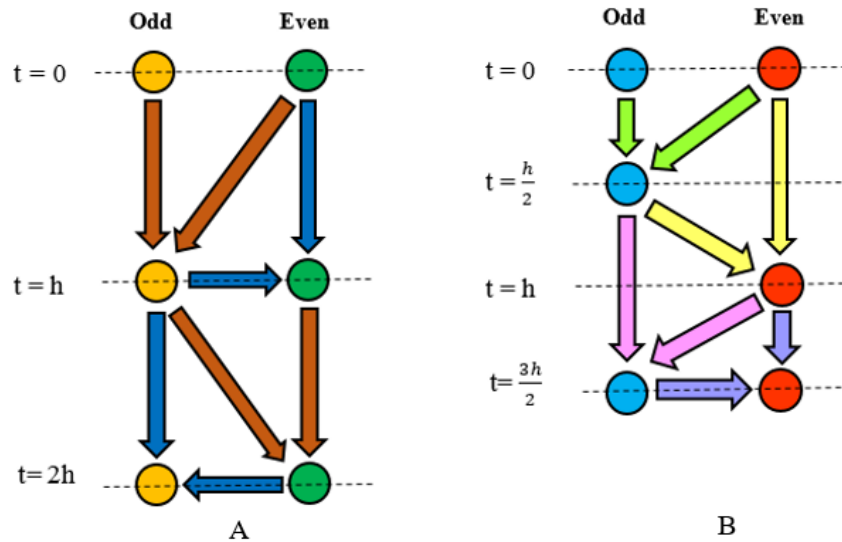
As a result of the discretization, we have  $i=1, 2, \dots, N$  differential equations for each cell, where the total number of differential equations (cells) denoted by  $N$ .

The time discretization of hyperbolic partial differential equations is typically the evolution of a system of ordinary differential equations obtained by spatial discretization of the original problem. Methods for this time evolution include multistep, multistage, or multiderivative methods, as well as a combination of these approaches. The time step constraint is mainly a result of the absolute stability requirement, as well as additional conditions that mimic physical properties of the solution, such as positivity or total variation stability.

More details about this construction, as well as on the formulas in the case of Eq. (1a) can be found in (Kovács, 2020b; Saleh et al., 2020a; Saleh et al., 2020c). Differential equation system (2) has been solved for the following time interval  $[t_0 = 0, t_{FIN} = 1]$ .

The new OEH-type method uses the same space-structure as the original one. More concretely, we must define odd and even nodes in the above mentioned mesh, which divide the whole mesh into two similar subsets. The new methods apply a wide range of different formulas instead of the well-known explicit Euler formula at the first stage and the implicit Euler formula at the second stage. The time

structure is also modified as follows. We take a half-sized time step for the odd nodes (green arrows in Figure 1 b), then we take a full time step for the even nodes using the already calculated values (yellow arrows), then a full time step for the odd cells comes (pink arrows) and finally, the calculation of the values is closed by a half-size time step for the even nodes (purple arrows). Thus, we have a structure which includes 4 stages, which correspond to 4 partial time steps, which altogether span  $h + \frac{h}{2} = \frac{3}{2}h$  time steps for odd and even cells, too. During the implementation of the original OEH, one must be sure that  $t_{FIN} = 2hk, k \in \mathbb{N}$ . In the case of the new structure,  $t_{FIN} = \frac{3}{2}hk, k \in \mathbb{N}$  must hold. Similarly to the original OEH, the new methods use the latest available values of the neighbours in each stage, so they are completely explicit and it is not necessary to store the previous values.



**Figure 1.** The stencil of the odd-even hopscotch algorithm. (A) The original OEH algorithm. The brown arrows and blue arrows indicate the first stage and second stage, respectively. (B) The new OEH algorithm. The green, yellow, purple and blue arrows indicate Stages 1, 2, 3 and 4, respectively

The first (1-D) type of formula we use now is the adapted version of the well-known theta-method:

$$u_i^{n+1} = \frac{(1 - 2r\theta)u_i^n + r(u_{i-1}^m + u_{i+1}^m)}{1 + 2r(1 - \theta)} \quad (3)$$

where  $m = n, n + \frac{1}{2}$ , or  $n + 1$  at the first, middle and last stages, respectively, see Fig. 1. (b) as well.

Equation system (2) can be written into a condensed matrix-form:

$$\frac{d\vec{u}}{dt} = M\vec{u} \quad (4)$$

The matrix  $M$  is tridiagonal in the one-dimensional case of Equation (3) with the following elements:

$$m_{ii} = -\frac{2\alpha}{\Delta x^2} \quad (1 < i < N), \quad m_{i,i+1} = \frac{\alpha}{\Delta x^2} \quad (1 \leq i < N), \quad m_{i,i-1} = \frac{\alpha}{\Delta x^2} \quad (1 < i \leq N) \quad (5a)$$

In the general case of Equation (2), the nonzero elements of the matrix can be given as:

$$m_{ij} = \frac{1}{R_{i,j} C_i}, \quad m_{ii} = -\sum_{i \neq j} m_{ij} \quad (5b)$$

Moreover,  $r = \frac{\alpha h}{\Delta x^2} = -\frac{m_{ii} h}{2} > 0$ ,  $0 < i < N - 1$  is the usual mesh ratio,  $m_{ii} = \frac{2\alpha}{\Delta x^2}$  and  $\theta \in [0, 1]$ . For  $\theta = 0, \frac{1}{2}$ , and 1 we obtain the implicit Euler, the Crank-Nicolson and the explicit Euler respectively (or, more concretely, the forward-time central-space, FTCS) schemes, respectively (Gordon, 1965). If  $\theta > 0$ , the original theta-method is implicit. Now, in our asymmetric hopscotch scheme, the neighbours are always taken into account at the same, latest time level, thus we insert  $u_{i-1}^m$  and  $u_{i+1}^m$  into (3), The general-mesh form of this theta-formula is the following:

$$u_i^{n+1} = \frac{(1 - r_i \theta) u_i^n + 2A_i^{\text{new}}}{1 + r_i(1 - \theta)} \quad (6)$$

where:

$$r_i = \frac{h}{\tau_i}, \quad A_i = h \sum_{j \neq i} m_{ij} u_j^n = h \sum_{j \in \text{neighbours}(i)} \frac{u_j^n}{C_i R_{ij}}, \quad (7)$$

and

$$A_i^{\text{new}} = h \sum_{j \neq i} m_{ij} u_j^m = h \sum_{j \in \text{neighbours}(i)} \frac{u_j^m}{C_i R_{ij}}$$

One should be aware that  $r_i = 2r$  for the special, equidistant case.

The other formula we use is the following CNe method, which is introduced in our papers (Kovács, 2020a; Kovács, 2020b; Nagy et al., 2021) and now briefly restated here.

$$u_i^{n+1} = u_i^n \cdot e^{-2r} + \frac{u_{i-1}^{n+1/2} + u_{i+1}^{n+1/2}}{2} (1 - e^{-2r}) \quad (8)$$

For more information see (Kovács, 2020a; Kovács, 2020b; Saleh et al., 2020a; Saleh et al., 2020b; Saleh et al., 2020c). Similarly, the generalized CNe formula is

$$u_i^{n+1} = u_i^n \cdot e^{-r_i} + \frac{A_i}{r_i} (1 - e^{-r_i}) \quad (9)$$

And of course, for halved time steps  $r_i$  and  $A_i$  must be divided by 2.

We will use a compact notation of the individual combinations, where 4 data are given in a bracket, the numbers are the values of the parameter  $\theta$ , while the letter ‘C’ is for the CNe constant neighbour method. For example (1/5, 2/3, C, 0) means the following 4-stage algorithm.

**Example of Algorithm: A2** (1/5, 2/3, C, 0) general form;

Stage 1. Take a half time step with the (16) formula with  $\theta=1/4$  for odd cells:

$$u_i^{n+1} = \frac{\left(1 - \frac{r_i}{10}\right)u_i^n + A_{i,\text{half}}}{1 + \frac{r_i}{2}\left(1 - \frac{1}{5}\right)}, \quad A_{i,\text{half}} = \frac{h}{2} \sum_{j \neq i} \frac{u_j^m}{C_i R_{ij}}$$

Stage 2. Take a full time step with the (16) formula with  $\theta=1/2$  for even cells:

$$u_i^{n+1} = \frac{\left(1 - \frac{2r_i}{3}\right)u_i^n + A_i}{1 + r_i\left(1 - \frac{2}{3}\right)}, \quad A_i = h \sum_{j \neq i} \frac{u_j^m}{C_i R_{ij}}$$

Stage 3. Take a full time step with the (17) formula for odd cells:

$$u_i^{n+1} = u_i^n \cdot e^{-r_i} + \frac{A_i}{r_i} \left(1 - e^{-r_i}\right), \quad A_i = h \sum_{j \neq i} \frac{u_j^m}{C_i R_{ij}}$$

Stage 4. Take a half time step with the (16) formula with  $\theta=0$  for odd cells:

$$u_i^{n+1} = \frac{u_i^n + A_{i,\text{half}}}{1 + \frac{r_i}{2}}, \quad A_{i,\text{half}} = \frac{h}{2} \sum_{j \neq i} \frac{u_j^m}{C_i R_{ij}}$$

All other combinations can be constructed in this manner easily.

### 3. Results

#### 3.1. Preliminary tests

We will apply the 9 different values for parameter theta:  $\theta \in \{0, 1/5, 1/4, 1/3, 1/2, 2/3, 3/4, 4/5, 1\}$ , with the CNe formula and so we have 10 different formulas and we want to apply all of these into the shifted-hopscotch structure in all combination.

There are 4 stages in the structure, we have  $10^4=10000$  different algorithm-combinations. We have written a code to systematically construct and test all these combinations.

We solve Eq. (2) with randomly generated initial conditions  $u_i(0) = rand$ , where rand is a random number with a uniform distribution in the interval (0, 1), generated by the MATLAB for each cell. We also generate different random values for the heat capacities and for the thermal resistances, but with a log-uniform distribution:

$$C_i = 10^{(\alpha_c - \beta_c \times rand)}, \quad R_{x,i} = 10^{(\alpha_{R_x} - \beta_{R_x} \times rand)}, \quad R_{z,i} = 10^{(\alpha_{R_z} - \beta_{R_z} \times rand)} \quad (10)$$

where the coefficients  $\alpha_c, \dots, \beta_{R_z}$  in the exponents will be concretized later.

We use zero Neumann boundary conditions, i.e. the system is thermally isolated. This is implemented naturally at the level of Eq. (2) since it is enough to omit those terms of the sum which have infinite resistivity in the denominator due to the isolated border. This implies that the system matrix  $M$  has one zero eigenvalue, belongs to the uniform distribution of temperatures, all other eigenvalues must be negative. Let us denote by  $\lambda_{\min}$  ( $\lambda_{\max}$ ) the (nonzero) smallest (largest) absolute value eigenvalues of matrix  $M$ . The stiffness ratio of the system can be defined as  $\lambda_{\max} / \lambda_{\min}$ . The maximum possible time step size for the FTCS (explicit Euler) scheme (from the point of view of stability) can be exactly calculated as  $h_{\max}^{\text{FTCS}} = |2 / \lambda_{\max}|$ , above which the solution are expected to blow up. We are going to use these two numbers to characterize the “difficulty level” of the problem.

The parameters  $\alpha_C, \beta_C, \alpha_{R_x}, \beta_{R_x}, \alpha_{R_z}, \beta_{R_z}$  of the distribution of the mesh-cells data have been chosen to construct test problems with various stiffness ratios and  $h_{\max}^{\text{FTCS}}$ , for example  $\alpha_C = 1, 2, \text{ or } 3, \beta_C = 2, 4, \text{ or } 6, \alpha_{R_x} = -1, 1, \beta_{R_x} = -3, 3, \alpha_{R_z} = 0, 0, \beta_{R_z} = -2, 2$ . The (pseudo-)random number, rand, is generated by MATLAB for each quantity with a uniform distribution in the unit interval (0, 1). We also generate different random values for the initial conditions  $u_i(0) = \text{rand}$ .

We performed the procedure in case of 2 different systems (stiff and non-stiff system) with  $N_x \times N_z = 2 \times 6$ . After these tests, the few best combinations are chosen, and we continue the work only with them, in the current test we get the best 5 combinations as follows:

$$(1/5, 2/3, C, 0), \quad (1/5, 1/5, 3/4, C), \quad (1/5, 1/4, 3/4, C), \quad (C, 1/4, 2/3, C), \quad (C, 1/2, 1/2, C),$$

The numerical error is calculated by comparing our numerical solutions  $u_j^{\text{num}}$  with the reference solution  $u_j^{\text{ref}}$  at final time  $t_{\text{fin}}$ . We used the following three types of (global) error. The first one was the maximum of the absolute differences:

$$\text{Error}(L_\infty) = \max_{0 \leq j \leq N} |u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}})| \quad (11)$$

The second one is the average absolute error:

$$\text{Error}(L_1) = \frac{1}{N} \sum_{0 \leq j \leq N} |u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}})| \quad (12)$$

### 3.2. Verification

We consider a nontrivial analytical solution of Eq. (1) (Gourlay, 1970). It is given on the whole real number line for positive values of  $t$  as follows

$$u^{\text{anal}} = \frac{x}{t^{5/2}} \left( 1 - \frac{x^2}{6\alpha t} \right) e^{-\frac{x^2}{4\alpha t}} \quad (13)$$

We reproduce these solutions only in finite space and time intervals  $x \in [x_1, x_2]$  and  $t \in [t_0, t_{\text{fin}}]$ , where  $x_1 = -5, x_2 = 5, t_0 = 0.5, t_{\text{fin}} = 0.6$ . The space interval is discretized by creating nodes as follows:

$x_j = x_1 + j \Delta x$ ,  $j = 0, \dots, 1000$ ,  $\Delta x = 0.01$ . We use the analytical solution to gain the prescribed Dirichlet boundary conditions:

$$u_2(x = x_{1,2}, t) = \frac{x_{1,2}}{t^{5/2}} \left( 1 - \frac{x_{1,2}^2}{6\alpha t} \right) e^{-\frac{x_{1,2}^2}{4\alpha t}} \quad (14)$$

In Figure 2 the  $L_\infty$  errors as a function of the effective time step size  $h_{\text{EFF}}$  are presented for the top 5 algorithms and a first-order original CNe method. We note that very similar curves have been obtained for the  $u_1$  solution, as well as for other space and time intervals. We found that the new methods are convergent and the order of convergence is at least one. In fact, the methods behave as second order methods for large time step sizes.

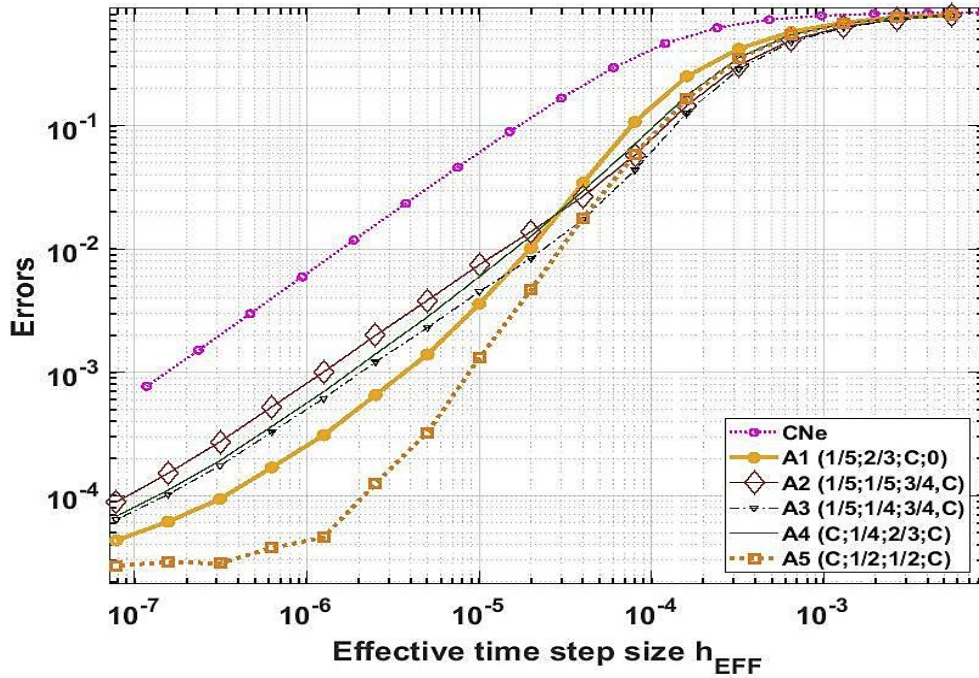


Figure 2. The  $L_\infty$  errors as a function of  $h_{\text{EFF}}$  for the  $u_2$  solutions of the heat equation for  $\alpha = 1$ .

### 3.3. Case study I and comparison with other solvers

The sizes of studied grid were fixed to  $N_x = 100$  and  $N_z = 100$ , thus the total cell number was 10000, while the final time was  $t_{\text{fin}} = 0.1$ . The exponents have been set to the following values:

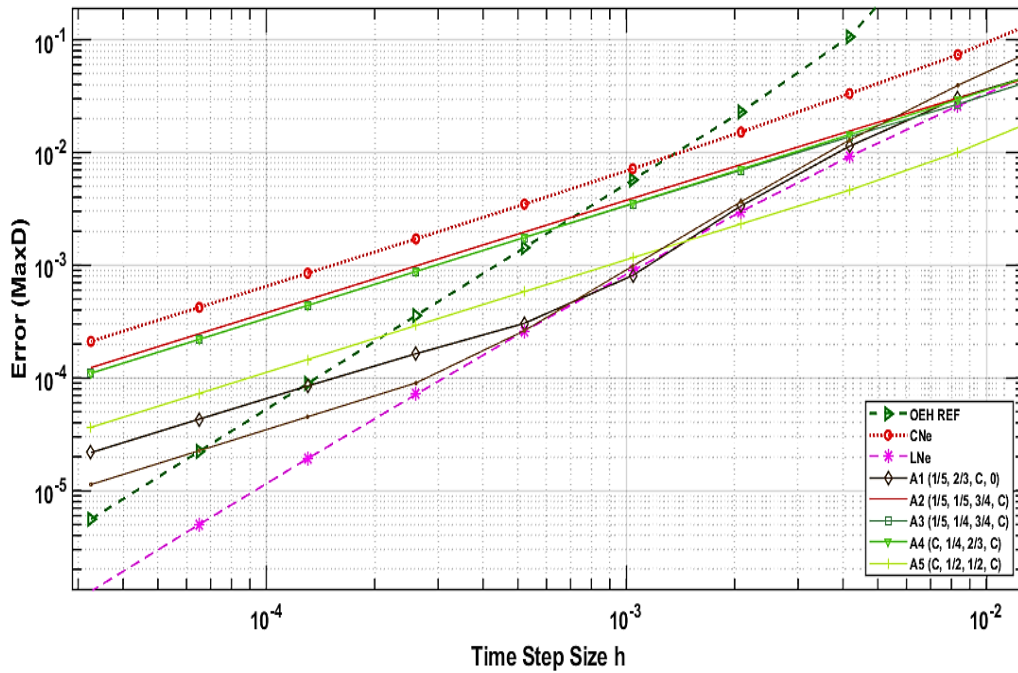
$$\alpha_C = 2, \beta_C = 4, \alpha_{R_x} = \alpha_{R_z} = 1, \beta_{R_x} = \beta_{R_z} = 2,$$

For that log-uniformly distributed capacities have given values between 0.01 and 100. The generated system can be characterized by its stiffness ratio and  $h_{\text{MAX}}^{\text{FTCS}}$  values, which are  $3.1 \times 10^8$  and  $7.3 \times 10^{-4}$ ,

respectively, thus we can say that this system is moderately stiff. The performance of new algorithms was compared with the following widely used MATLAB solvers:

- ode15s, a first to fifth order (implicit) numerical differentiation formulas with variable-step and variable order (VSVO), developed for solving stiff problems;
- ode23s, a second order modified (implicit) Rosenbrock formula;
- ode23t, applies (implicit) trapezoidal rule with using free interpolant;
- ode23tb, combines backward differentiation formula and trapezoidal rule;
- ode45, a fourth/fifth order explicit Runge-Kutta-Dormand-Prince formula;
- ode23, second/third order explicit Runge-Kutta-Bogacki-Shampine method;
- ode113, 1 to 13 order VSVO Adams-Bashforth-Moulton numerical solver.

In Figure 3, we present the error functions with time steps and in figure 4 we present the average error functions with the running time only for these top 5 combinations, for the first (moderately stiff) system.



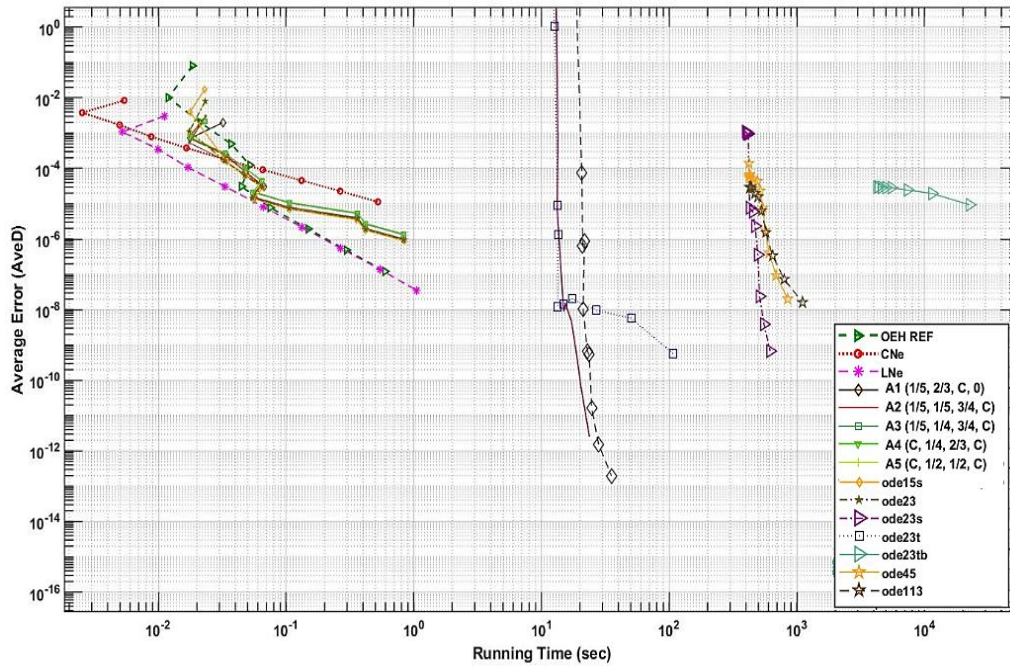
**Figure 3.** Errors as a function of the time step for the first (moderately stiff) system, in the case of the original (OEH REF) method, the original one stage CNe method, the new algorithms A1-A5

### 3.4. Case study II and Comparison with other Solvers

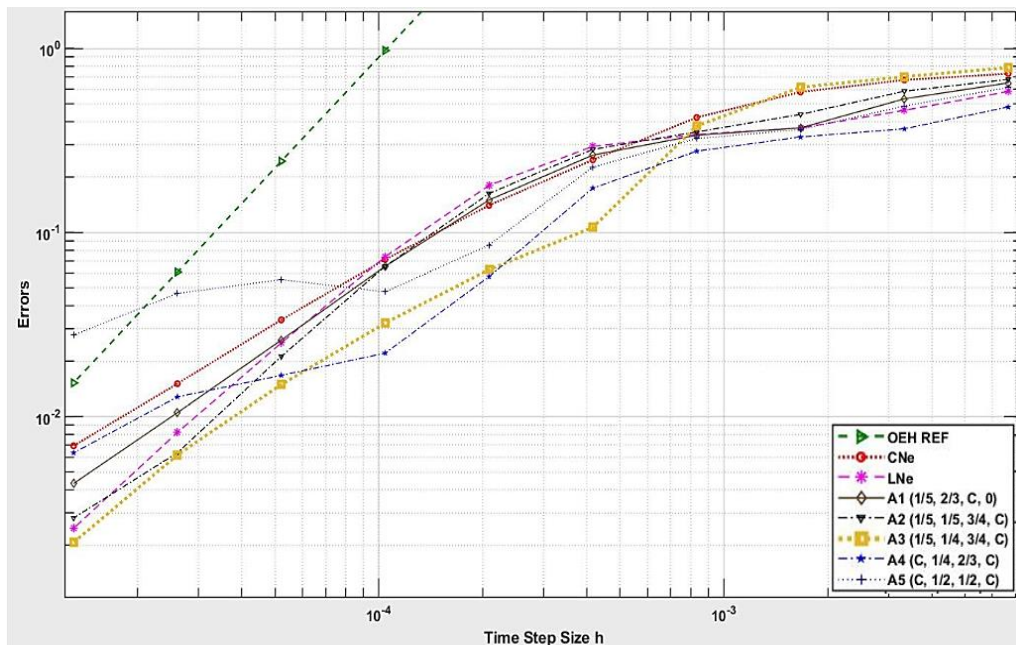
We tested our new algorithms and the conventional solvers for a harder problem as well. Thus, new values have been set for the  $\alpha$  and  $\beta$  exponents:

$$\alpha_C = 3, \beta_C = 6, \alpha_{Rx} = 3, \alpha_{Rz} = 1, \beta_{Rx} = \beta_{Rz} = 4$$





**Figure 4.** Average errors as a function of the running times for the first (moderately stiff) system, in the case of the original OEH method (OEH REF), one stage CNe method, the new algorithms A1-A5 and different MATLAB routines.



**Figure 5.** Errors as a function of the time step size for the second (very stiff) system, in the case of the original (OEH REF) method, the original one stage CNe method, the new algorithms A1-A5.

This means that the width of the distribution of the capacities and thermal resistances have been increased and the system has been acquired some anisotropy, since the resistances in the x direction are two orders of magnitude larger than in the z direction on average  $R_{x,i} \in [10^{-1}, 10^3]$  ,  $R_{z,i} \in [10^{-3}, 10^1]$  . With this modification we have gained a system with much higher stiffness ratio,  $2.5 \times 10^{11}$  while the maximum allowed time step size for the standard FTCS was  $h_{MAX}^{EE} = 1.6 \times 10^{-6}$  . All other parameters and circumstances remained the same as in Subsection 3.3.

In Figure 5, we present the error functions with time steps and in figure 6 we present the average error functions with the running time only for these top 5 combinations, for the second (very stiff) system.

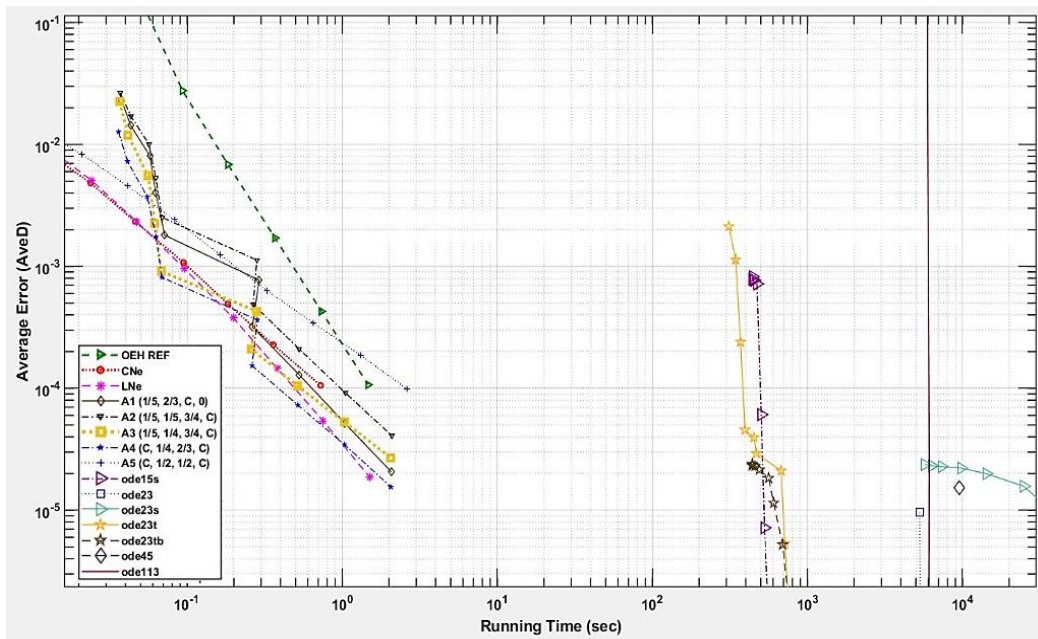


Figure 6. Average errors as a function of the running times for the second (very stiff) system, in the case of the original OEH method (OEH REF), one stage CNe method, the new algorithms A1-A5.

#### 4. Discussion and summary

In this paper, the numerical algorithms were tested to solve the non-stationary diffusion (or heat) equation, and these new algorithms are fully explicit time-integrators obtained by applying half and full time steps in the OEH structure. All these algorithms consist of four stages and one-step methods, meaning that when we want to calculate the new values of the unknown function u, we will use only the most recently calculated u values. The conventional theta-method was applied with 9 different values of  $\theta$  and the non-conventional CNe method to construct  $10^4$  combinations, and we choose the top 5 of them via numerical experiments. The concrete results of these 5 algorithms are shown in the case of two 2-dimensional stiff systems containing 10000 cells with highly inhomogeneous randomly generated parameters and discontinuous initial conditions. The last experiments show that the suggested methods are indeed competitive because they can give clear accurate results orders of magnitude faster than the well-

optimized MATLAB routines and also significantly more accurate for stiff systems than the original hopscotch method.

We obtained that the numerical order of the new algorithms is only one. We think that if numerical results must be produced in very short running time and only low accuracy requirements, the A3 (1/5, 1/4, 3/4, C) combination might be suggested, but, on the other hand, when higher accuracy is the goal, higher order methods than these should be applied.

## References

- [1] Zoppou, C., Knight, J. H. (1999). Analytical solution of a spatially variable coefficient advection-diffusion equation in up to three dimensions. *Appl. Math. Model.*, 23(9), 667-685. [https://doi.org/10.1016/S0307-904X\(99\)00005-0](https://doi.org/10.1016/S0307-904X(99)00005-0)
- [2] Kovács, E., Gilicz, A. (2018). New stable method to solve heat conduction problems in extremely large systems. *Des. Mach. Struct.*, 8(2), 30-38.
- [3] Kovács, E. (2020a). New stable, explicit, first order method to solve the heat conduction equation. *J. Comput. Appl. Mech.*, 15(1), 3-13. <https://doi.org/10.32973/jcam.2020.001>
- [4] Kovács, E. (2020b). A class of new stable, explicit methods to solve the non-stationary heat equation. *Numer. Methods Partial Differ. Equ.*, 37(3), 2469-2489. <https://doi.org/10.1002/num.22730>
- [5] Saleh, M., Nagy, Á., Kovács, E. (2020a). Construction and investigation of new numerical algorithms for the heat equation : Part 1. *Multidiszciplináris tudományok*, 10(4), 323-338. <https://doi.org/10.35925/j.multi.2020.4.36>
- [6] Saleh, M., Nagy, Á., Kovács, E. (2020b). Construction and investigation of new numerical algorithms for the heat equation : Part 2. *Multidiszciplináris tudományok*, 10(4), 339-348. <https://doi.org/10.35925/j.multi.2020.4.37>
- [7] Saleh, M., Nagy, Á., Kovács, E. (2020c). Construction and investigation of new numerical algorithms for the heat equation : Part 3. *Multidiszciplináris tudományok*, 10(4), 349-360. <https://doi.org/10.35925/j.multi.2020.4.38>
- [8] Gordon, P. (1965). Nonsymmetric difference equations. *J. Soc. Ind. Appl. Math.*, 13(3), 667-673. <https://doi.org/10.1137/0113044>
- [9] Gourlay, A. R. (1970). Hopscotch: a fast second-order partial differential equation solver. *IMA J. Appl. Math.*, 6(4), 375-390. <https://doi.org/10.1093/imamat/6.4.375>
- [10] Gourlay, A. R., McGuire, G. R. (1971). General hopscotch algorithm for the numerical solution of partial differential equations. *IMA J. Appl. Math.*, 7(2), 216-227. <https://doi.org/10.1093/imamat/7.2.216>
- [11] Gourlay, A. R. (1971). Some recent methods for the numerical solution of time-dependent partial differential equations. *Proc. R. Soc. London. A. Math. Phys. Sci.*, 323(1553), 219-235. <https://doi.org/10.1098/rspa.1971.0099>
- [12] Maritim, S., Rotich, J. K. (2018). Hybrid hopscotch method for solving two dimensional system of Burgers' Equation. *Int. J. Sci. Res.*, 8(8), 492-497.
- [13] Holmes, M. H. (2007). *Introduction to numerical methods in differential equations*. New York: Springer. <https://doi.org/10.1007/978-0-387-68121-4>
- [14] Chen-Charpentier, B. M., Kojouharov, H. V. (2013). An unconditionally positivity preserving scheme for advection-diffusion reaction equations. *Math. Comput. Model.*, 57, 2177-2185. <https://doi.org/10.1016/j.mcm.2011.05.005>

- 
- [15] Appadu, A. R. (2017). Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *International Journal of Numerical Methods for Heat and Fluid Flow*, 1412-1429. <https://doi.org/10.1108/HFF-01-2016-0038>
- [16] Munka, M., Pápay, J. (2001). *4D numerical modeling of petroleum reservoir recovery*. Budapest: Akadémiai Kiadó.
- [17] Appadu, A. R. (2016). Analysis of the unconditionally positive finite difference scheme for advection-diffusion-reaction equations with different regimes. *AIP Conference Proceedings*, 1738(1), 030005. <https://doi.org/10.1063/1.4951761>
- [18] Nagy, Á., Omle, I., Kareem, H., Kovács, E., Barna, I. F., Bognar, G. (2021). Stable, explicit, leapfrog-hopscotch algorithms for the diffusion equation. *Computation* 2021, 9, 92. <https://doi.org/10.3390/computation9080092>