computation

MDPI

Article

Solution of Coupled Systems of Reaction–Diffusion Equations Using Explicit Numerical Methods with Outstanding Stability Properties

Husniddin Khayrullaev [1], Andicha Zain [2] and Endre Kovács [1,*]

[1] Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary; hxayrullayev@mail.ru
[2] Institute of Mathematics, University of Miskolc, 3515 Miskolc, Hungary
* Correspondence: kendre01@gmail.com or endre.kovacs@uni-miskolc.hu

Abstract: Recently, new and nontrivial analytical solutions that contain the Kummer functions have been found for an equation system of two diffusion–reaction equations. The equations are coupled by two different types of linear reaction terms which have explicit time-dependence. We first make some corrections to these solutions in the case of two different reaction terms. Then, we collect eight efficient explicit numerical schemes which are unconditionally stable if the reaction terms are missing, and apply them to the system of equations. We show that they severely outperform the standard explicit methods when low or medium accuracy is required. Using parameter sweeps, we thoroughly investigate how the performance of the methods depends on the coefficients and parameters such as the length of the examined time interval. We obtained that, similarly to the single-equation case, the leapfrog–hopscotch method is usually the most efficient to solve these problems.

Keywords: PDE systems; self-similar solution; diffusion–reaction equations; explicit numerical methods; hopscotch methods

check for updates

Academic Editor: Shan Zhao

Received: 15 May 2025
Revised: 27 May 2025
Accepted: 29 May 2025
Published: 1 June 2025

Citation: Khayrullaev, H.; Zain, A.; Kovács, E. Solution of Coupled Systems of Reaction–Diffusion Equations Using Explicit Numerical Methods with Outstanding Stability Properties. *Computation* **2025**, *13*, 129. https://doi.org/10.3390/computation13060129

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Diffusion and diffusion–reaction equations are among the most widespread partial differential equations (PDEs) in science and technology. Therefore, analytical solutions have been sought and found not only in historical research [1,2], but recently as well. For example, Simpson et al. [3] constructed analytical solutions of coupled linear reaction–diffusion equations where the space domain grows in time. Escorcia and Suazo [4] very recently showed how to construct solutions explicitly via similarity transformations and a Ricatti system for coupled reaction–diffusion equations where the coefficients depend on time. The solutions of a reaction–diffusion equation where the reaction term explicitly depends on time were examined by Covei [5]. Mátyás and Barna found nontrivial solutions [6] a few years ago to regular and irregular diffusion equations containing Kummer or Whittaker functions using similarity transformations. Last year, they extended their investigations to systems of diffusion–reaction equations [7] as well.

These solutions not only provide insight into the diffusion process itself, but make the investigation of the properties and performance of numerical algorithms possible. There are a large number of numerical techniques that can be used to solve these and similar equations. Typical methods can be classified as explicit or implicit algorithms, and both categories have advantages and disadvantages. Explicit algorithms usually have a seriously confined stability domain [8], which means that when the time step size exceeds a certain

threshold, sometimes referred to as the CFL (Courant–Friedrichs–Lewy) limit, the solution sooner or later explodes, even in the case of the simple linear diffusion equation. On the other hand, the execution of one time step of the explicit scheme is relatively quick, even for larger systems, and, in principle, the calculations are easily parallelizable. This characterization also refers to the method of lines approach, where the time integration is based on the explicit Runge–Kutta (RK) scheme, or the multistep Adams–Bashforth (AB) scheme. There are explicit methods with better qualitative properties, such as the so-called exact and nonstandard finite difference (NSFD) algorithms [9], which have been proposed for single reaction–diffusion equations [10–12] and for cross-diffusion equations [13,14]. However, the favorable properties of these approaches manifest themselves typically only for short time step sizes.

Implicit methods have much better stability properties, and the time step size is limited generally by the accuracy requirements, with some exceptions [15]. The largest drawback of the implicit methods is the need to solve a system of algebraic equations in each time step, which can be time- and memory-consuming if the number of mesh elements is large, which is frequent in the case of two- and three-space-dimensional problems. Still, many scientists consider them to be superior and frequently employ them [16] for these and other related equations. The Crank–Nicolson method is maybe the most widely applied implicit approach [17,18]. For instance, Manaa and Sabawi [19] used the Crank–Nicolson method, as well as the explicit Euler time integration. Their finding was that the former is more accurate and does not have significant stability problems, but the latter has shorter running times, as expected. Pargaei and Kumar [20] applied the Haar wavelet method for systems of degenerate reaction–diffusion PDEs with nonlinear sources. Anjuman et al. proposed a shifted airfoil collocation method [21] for nonlinear drift-diffusion–reaction equations. Rufai et al. [22] developed an implicit hybrid block algorithm with Newton iterations for nonlinear reaction–diffusion equations where the coefficients depend on the time variable. The so-called proper orthogonal decomposition method was combined with the unconditionally positive finite difference scheme by Ndou et al. [23] for advection–diffusion–reaction equations, which can be linear and nonlinear. A Legendre–Gauss–Lobatto spectral collocation approach was used by Heidari et al. [24]. An implicit numerical algorithm preserving the non-negativity of the solutions was presented by Kolev et al. [25].

Certainly, it is not true that either the whole system is treated explicitly or fully implicitly. Another typical approach is that one term, most frequently the diffusion term, is treated implicitly and the reaction term is treated explicitly. These implicit–explicit, or IMEX, methods [26] try to balance stability and computational cost. For example, Calvo et al. [27] combined L-stable linearly implicit RK schemes with explicit ones, all of them with high order, for advection–reaction–diffusion equations. A reaction–diffusion equation system of the Lotka–Volterra type in two space dimensions is solved by Settanni and Sgura [28]. They also treated the diffusion term implicitly, but the reaction term was handled not only by traditional explicit schemes, but the alternating-direction implicit schemes as well. Other similar examples are the IMEX RK-type methods recently developed by Yadav et al. [29] for stiff convection–diffusion–reaction systems, where the diffusion, as well as the reaction terms, are handled implicitly.

Semi-implicit or semi-explicit schemes, developed to solve ODE systems, can also be applied after the spatial semi-discretization of the PDE. Time-efficient variants of these schemes based on the BDF (backward differentiation formula) and AB schemes were developed by Beuken et al. [30]. A variable step size controller was introduced in the paper of Fedoseev et al. [31] to make the performance of semi-implicit composition integration schemes better. Methods utilizing the Gauss–Legendre quadrature, the generalized Padé approximation, and explicit RK methods were developed by Ji and Xing [32].

A further step is treating the diffusion term exactly [33], which can be advised in very stiff cases. For example, semilinear and nonlinear advection–reaction–diffusion equations were examined by Jiang and Zhang [34] using Krylov implicit integration factor methods, where the effect of the linear terms was calculated exactly. On the other hand, it was reported [35] that, although tiny time step sizes below the CFL limit are used, explicit methods can beat implicit schemes in the case of large systems.

Nevertheless, these achievements do not mean that the above-mentioned shortages of the explicit and implicit methods are finally fixed. Due to the reasons explained so far, it seems logical to invest time to construct and test explicit methods with enhanced stability. A first-order member of this family is the so-called unconditionally positive finite difference (UPFD) method, proposed by Chen-Charpentier and Kojouharov [36], and further examined by other scholars [37,38]. The UPFD scheme has very good qualitative properties, but, on the other hand, the error decreases much slower with the time step size than in the case of other first-order methods. Other, more accurate methods also exist with unconditional stability for the diffusion equation. Al-Bayati et al. [39] compared the performance of the Saulyev-type alternating-direction explicit (ADE) method, the alternating-direction implicit (ADI) method, and the odd–even hopscotch method to solve a system of diffusion–reaction equations. They found that the implicit method is more accurate but slower than the explicit methods. Nonlinear PDEs were also solved by Pourghanbar et al. [40] using the ADE scheme, which was faster than using major implicit solvers.

Our research group proposed novel explicit methods in the last few years to solve diffusion and diffusion–reaction equations. We proved analytically and presented in our previous work that these algorithms are unconditionally stable for the single linear diffusion equation. The performance of these schemes was extensively examined for the diffusion equation in which the diffusion coefficient is a constant [41] and depends on time [42] or space [43], or both at the same time [44]. Furthermore, they were tested for different kinds of reaction–diffusion equations, for example, with a space- and time-dependent coefficient of the reaction term [45]. It turned out that the new methods can frequently outperform the conventional methods, either explicit RK types or implicit solvers. We are now continuing these investigations with the inclusion of equation systems. This is the first time that these methods are applied to mutually coupled reaction–diffusion equations.

This paper has the following structure. After the presentation of the studied problem in Section 2, we display the analytical solution constructed in paper [7], analyze the mistake in it, and propose a way to correct it. The examined numerical methods and their properties are described briefly in Section 3. Two case studies using ten numerical methods are displayed in Section 4. Section 5 presents parameter sweeps to discover how the numerical errors of the methods depend on some parameters of the problems to be solved, such as the coefficients of the reaction term and the duration of the simulated time interval. Finally, the conclusions are presented at the end of this paper.

2. The Equations and the Analytical Solutions

It is well known that the simplest form of the diffusion equation (also called the heat equation) is the following:

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2}, \tag{1}$$

where $x, t \in \mathbb{R}$ are the space and time variables, the $u(x,t)$ concentration function meets the required smoothness conditions, and the non-negative number D is called the diffusivity.

2.1. Coupled Diffusion Equations

The authors of paper [7] examine the following coupled system of PDEs, numbered as Equation (19) in [7]:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= D\frac{\partial^2 u(x,t)}{\partial x^2} + F(u(x,t), w(x,t)), \\ \frac{\partial w(x,t)}{\partial t} &= D\frac{\partial^2 w(x,t)}{\partial x^2} + G(u(x,t), w(x,t)), \end{aligned} \tag{2}$$

where F and G are real functions. The authors of [7] introduced the so-called reduced variable $\eta = \frac{x}{\sqrt{t}}$, and applied the self-similar Ansatz

$$u(x,t) = t^{-\alpha} f(\eta) \text{ and } w(x,t) = t^{-\gamma} g(\eta), \tag{3}$$

where $f(\eta)$ and $g(\eta)$ are the shape functions, while α and γ are independent real parameters. They constructed analytical solutions for two different versions of the reaction term. The first one, which will be called simple coupling, has the following form (Equation (20) in [7]):

$$F(u,w) = \frac{a \cdot w}{t}, \quad G(u,w) = \frac{b \cdot u}{t}, \tag{4}$$

where $a, b \in \mathbb{R}$. The transformation (3) is supposed to yield the ODEs

$$\begin{aligned} -\alpha f(\eta) - \frac{1}{2}\eta \frac{df}{d\eta} &= D\frac{d^2 f}{d\eta^2} + a \cdot g(\eta), \\ -\gamma g(\eta) - \frac{1}{2}\eta \frac{dg}{d\eta} &= D\frac{d^2 g}{d\eta^2} + b \cdot f(\eta), \end{aligned} \tag{5}$$

which, according to Maple 12 and Equations (21) and (22) in [7], have the solution

$$f(\eta) = \eta \cdot e^{-\frac{\eta^2}{4D}} \begin{pmatrix} c_1 M\left[p-q, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_2 M\left[p+q, \frac{3}{2}, \frac{\eta^2}{4D}\right] \\ +c_3 U\left[p-q, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_4 U\left[p+q, \frac{3}{2}, \frac{\eta^2}{4D}\right] \end{pmatrix}, \tag{6}$$

and

$$g(\eta) = \frac{\eta}{2a} e^{-\frac{\eta^2}{4D}} \left\{ \begin{array}{c} \left(c_1 M\left[p-q, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_3 U\left[p-q, \frac{3}{2}, \frac{\eta^2}{4D}\right]\right)(\gamma - \alpha + 2q) \\ +\left(c_2 M\left[p+q, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_4 U\left[p+q, \frac{3}{2}, \frac{\eta^2}{4D}\right]\right)(\gamma - \alpha - 2q) \end{array} \right\}. \tag{7}$$

The abbreviations $p = 1 - \frac{\gamma}{2} - \frac{\alpha}{2}$ and $q = \sqrt{4ab - 2\alpha\gamma + \gamma^2 + \alpha^2}/2$ were introduced. It is clear that if $a \cdot b \geq 0$, then q is real regardless of the values of α and γ.

The other type of reaction term (Equation (26) in [7]) is the following:

$$F(u,w) = -\frac{k}{t}(u(x,t) - w(x,t)), \quad G(u,w) = \frac{k}{t}(u(x,t) - w(x,t)), \tag{8}$$

where $k \in \mathbb{R}$ and k is usually positive. This case will be called time-decaying isomerization based on [7], where Equation (27) claims that the transformation yields the following ODE system:

$$\begin{aligned} -\alpha f(\eta) - \frac{1}{2}\eta\frac{df}{d\eta} &= D\frac{d^2f}{d\eta^2} - k(f(\eta) - g(\eta)), \\ -\gamma g(\eta) - \frac{1}{2}\eta\frac{dg}{d\eta} &= D\frac{d^2g}{d\eta^2} + k(f(\eta) - g(\eta)), \end{aligned} \tag{9}$$

which has the following solution (Equations (28) and (29) in [7]):

$$f(\eta) = \eta \cdot e^{-\frac{\eta^2}{4D}} \left(\begin{array}{c} c_1 M\left[p - q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_2 M\left[p + q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] \\ +c_3 U\left[p - q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_4 U\left[p + q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] \end{array} \right) \tag{10}$$

$$g(\eta) = \frac{\eta}{2k} e^{-\frac{\eta^2}{4D}} \left\{ \begin{array}{c} \left(c_1 M\left[p - q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_3 U\left[p - q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right]\right)(\gamma - \alpha + 2q') \\ +\left(c_2 M\left[p + q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right] + c_4 U\left[p + q' + k, \frac{3}{2}, \frac{\eta^2}{4D}\right]\right)(\gamma - \alpha - 2q') \end{array} \right\}. \tag{11}$$

where $q' = \sqrt{4k^2 - 2\alpha\gamma + \gamma^2 + \alpha^2}/2$.

However, after performing the transformation, we discovered that the ODE systems (5) and (9) above cannot be the transformed versions of the original equations with the reaction terms (4) and (8) after the general Ansatz (3) is used. The reason for this is that α and γ are independent parameters and γ can be eliminated from the first equation of (5) and (9), and α can be eliminated from the second equation of (5) and (9) only if they are equal. The authors probably accidentally substituted γ with α and α with γ in the first and second equations, respectively; therefore, their solutions are valid only in the $\alpha = \gamma$ case. This does not mean that the solutions in [7] are not valid for the examined equations, only that they contain only one free parameter and their form is simpler than given in [7]. We note that the paper [7] presented analytical results for some other equations which we did not check.

Our first goal in this paper is to save these valuable solutions in the general case. For this, we slightly change the PDE systems which are to be solved.

2.2. The Modified Equations and Their Solution for Simple Coupling

The first system of equations will be

$$\begin{aligned} \frac{\partial u}{\partial t} &= D\frac{\partial^2 u}{\partial x^2} + a \cdot w \cdot t^A, \\ \frac{\partial w}{\partial t} &= D\frac{\partial^2 w}{\partial x^2} + b \cdot u \cdot t^{-A-2}. \end{aligned} \tag{12}$$

where A is an arbitrary real parameter. If now we perform the transformation (3), we obtain

$$\begin{aligned} -\alpha \cdot t^{-\alpha-1} f(\eta) - \frac{1}{2} t^{-\alpha} \frac{x}{t^{3/2}} \frac{df}{d\eta} &= D \cdot t^{-\alpha} \frac{1}{t} \frac{d^2 f}{d\eta^2} + a \cdot t^{-\gamma} \cdot g(\eta) \cdot t^A, \\ -\gamma \cdot t^{-\gamma-1} g(\eta) - \frac{1}{2} t^{-\gamma} \frac{x}{t^{3/2}} \frac{dg}{d\eta} &= D \cdot t^{-\gamma} \frac{1}{t} \frac{d^2 g}{d\eta^2} + b \cdot t^{-\alpha} \cdot f(\eta) \cdot t^{-A-2}. \end{aligned}$$

Multiplying the first equation by $t^{\alpha+1}$ and the second one by $t^{\gamma+1}$, we obtain

$$-\alpha f(\eta) - \frac{1}{2}\eta\frac{df}{d\eta} = D\frac{d^2f}{d\eta^2} + a\cdot t^{A+\alpha+1-\gamma}\cdot g(\eta),$$
$$-\gamma g(\eta) - \frac{1}{2}\eta\frac{dg}{d\eta} = D\frac{d^2g}{d\eta^2} + b\cdot t^{\gamma-1-A-\alpha}\cdot f(\eta).$$

This gives back ODE system (5) if the exponents of t in the last terms are zero, i.e., $\gamma = A+\alpha+1$, where α is still arbitrary. One can easily see that we obtain the symmetric case for $A=-1$. However, now (6) and (7) are valid solutions for arbitrary values of A. This means that the solution of equation system (12) is the following:

$$u(x,t) = \frac{x}{t^{\alpha+1/2}}\cdot e^{-\frac{x^2}{4Dt}}\begin{pmatrix} c_1M\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_2M\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right] \\ +c_3U\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_4U\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right]\end{pmatrix} \tag{13}$$

and

$$w(x,t) = \frac{x}{2a\cdot t^{A+\alpha+3/2}}\cdot e^{-\frac{x^2}{4Dt}}\left\{\begin{array}{c}\left(c_1M\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_3U\left[p-q,\frac{3}{2},\frac{x^2}{4Dt}\right]\right)(\gamma-\alpha+2q) \\ +\left(c_2M\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_4U\left[p+q,\frac{3}{2},\frac{x^2}{4Dt}\right]\right)(\gamma-\alpha-2q)\end{array}\right\}. \tag{14}$$

2.3. The Time-Decaying Isomerization

On the other hand, let us start with the modified PDE system:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\frac{\partial^2u}{\partial x^2} - \frac{k}{t}\cdot\left(u - w\cdot t^B\right),\\ \frac{\partial w}{\partial t} &= D\frac{\partial^2w}{\partial x^2} - \frac{k}{t}\cdot\left(w - u\cdot t^{-B}\right).\end{aligned} \tag{15}$$

Now, the transformation (3) yields

$$-\alpha\cdot t^{-\alpha-1}f(\eta) - \frac{1}{2}t^{-\alpha}\frac{x}{t^{3/2}}\frac{df}{d\eta} = D\cdot t^{-\alpha}\frac{1}{t}\frac{d^2f}{d\eta^2} - \frac{k}{t}\cdot\left(t^{-\alpha}\cdot f(\eta) - t^{-\gamma}\cdot g(\eta)\cdot t^B\right),$$
$$-\gamma\cdot t^{-\gamma-1}g(\eta) - \frac{1}{2}t^{-\gamma}\frac{x}{t^{3/2}}\frac{dg}{d\eta} = D\cdot t^{-\gamma}\frac{1}{t}\frac{d^2g}{d\eta^2} - \frac{k}{t}\cdot\left(t^{-\gamma}\cdot g(\eta) - t^{-\alpha}\cdot f(\eta)\cdot t^{-B}\right).$$

In the same way as above, we obtain

$$-\alpha f(\eta) - \frac{1}{2}\eta\frac{df}{d\eta} = D\frac{d^2f}{d\eta^2} - k\left(f(\eta) - t^{-\gamma+\alpha+B}\cdot g(\eta)\right)$$
$$-\gamma g(\eta) - \frac{1}{2}\eta\frac{dg}{d\eta} = D\frac{d^2g}{d\eta^2} - k\cdot\left(g(\eta) - t^{-\alpha-B+\gamma}\cdot f(\eta)\right).$$

This is equivalent to the ODE system (9) for an arbitrary α only if $\gamma = B+\alpha$, thus the symmetric case is obtained for $B=0$. Now, (10) and (11) are valid solutions for arbitrary values of B, hence the solution of equation system (15) is the following:

$$u(x,t) = \frac{x}{t^{\alpha+1/2}}\cdot e^{-\frac{x^2}{4Dt}}\begin{pmatrix} c_1M\left[p-q'+k,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_2M\left[p+q'+k,\frac{3}{2},\frac{x^2}{4Dt}\right] \\ +c_3U\left[p-q'+k,\frac{3}{2},\frac{x^2}{4Dt}\right] + c_4U\left[p+q'+k,\frac{3}{2},\frac{x^2}{4Dt}\right]\end{pmatrix} \tag{16}$$

and

$$w(x,t) = \frac{x}{2k \cdot t^{B+\alpha+1/2}} \cdot e^{-\frac{x^2}{4Dt}} \left\{ \begin{array}{c} \left(c_1 M\left[p-q'+k, \frac{3}{2}, \frac{x^2}{4Dt}\right] + c_3 U\left[p-q'+k, \frac{3}{2}, \frac{x^2}{4Dt}\right]\right)(\gamma-\alpha+2q') \\ +\left(c_2 M\left[p+q'+k, \frac{3}{2}, \frac{x^2}{4Dt}\right] + c_4 U\left[p+q'+k, \frac{3}{2}, \frac{x^2}{4Dt}\right]\right)(\gamma-\alpha-2q') \end{array} \right\}. \quad (17)$$

In the rest of this paper, we will numerically reproduce the solutions (13) and (14) and (16) and (17) for several parameter sets.

3. The Numerical Methods and Their Properties

3.1. The Spatial and Temporal Discretization

We solve the equations in the one-dimensional space interval $x \in [x_0\,,\ x_0 + L_x]$. This is discretized by creating $N_x \in \mathbb{N}\backslash\{0\}$ nodes equidistantly as usual: $x_i = i\Delta x\,,\ i = 1, \ldots, N_x\,.$ The time interval is $t \in \left[t^{\text{in}}\,,\ t^{\text{fin}}\right]$, where $t^{\text{fin}} = t^{\text{in}} + \text{TIME}$, and it is also discretized uniformly, $t^n = t^{\text{in}} + n \cdot \Delta t\,,\ n \in \{0,\ 1,\ 2, \ldots,\ T\}$, and the numerical solution at x_i and t^n is denoted by u_i^n. We define the mesh ratio by the widely applied formula $r = \dfrac{\alpha \cdot \Delta t}{\Delta x^2}$.

3.2. The Algorithms

We now provide essential details regarding the algorithms which solve the system of equations. The formulas of the algorithms are given for a general reaction term, denoted by R_u and R_w for the first, and the last member of the system of equations, respectively. Then, the specific forms of the reaction terms are given.

1. First, the FTCS formula is applied to Equation (18), which is based on the explicit Euler time discretization

$$u_i^{n+1} = u_i^n + r\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \Delta t \cdot R_u \text{ and } \ w_i^{n+1} = w_i^n + r\left(w_{i+1}^n - 2w_i^n + w_{i-1}^n\right) + \Delta t \cdot R_w \quad (18)$$

where

$$\begin{array}{ll} R_u = a \cdot w_i^n \cdot t^A \text{ and } R_w = b \cdot u_i^n \cdot t^{(-A-2)} & \text{for simple coupling,} \\ R_u = -\frac{k \cdot (u_i^n - w_i^n \cdot t^B)}{t} \text{ and } R_w = \frac{k \cdot (u_i^n \cdot t^{-B} - w_i^n)}{t} & \text{for isomerization.} \end{array} \quad (19)$$

2. The Adams–Bashforth method is an explicit multistep numerical technique. We use its two-step second order variant, which predicts u_i^{n+1} using information from the previous two points, making it more accurate than single-step methods like Euler's method. Hence, this method is not a self-starter, and usually an RK method is utilized to start the calculation. In our case, the first step is performed using Euler's method in the same way as above. Now, we have u and w values for two time levels, and the values in the old time levels will be denoted by $u2_i^n$ and $w2_i^n$. Now, the formulas of the Adams–Bashforth method are the following:

$$\begin{array}{l} fu_i^{n+1} = -2 \cdot r \cdot u_i^n + r\left(u_{i+1}^n + u_{i-1}^n\right) + \Delta t \cdot R_u \text{ and } fw_i^{n+1} = -2 \cdot r \cdot w_i^n + r\left(w_{i+1}^n + w_{i-1}^n\right) + \Delta t \cdot R_w\,, \text{ where} \\ R_u = a \cdot w_i^n \cdot t^A \text{ and} R_w = b \cdot u_i^n \cdot t^{(-A-2)},\ R_u = -\dfrac{k \cdot (u_i^n - w_i^n \cdot t^B)}{t} \text{ and } R_w = \dfrac{k \cdot (u_i^n \cdot t^{-B} - w_i^n)}{t} \\ fuold_i^{n+1} = -2 \cdot r \cdot u2_i^n + r\left(u2_{i+1}^n + u2_{i-1}^n\right) + \Delta t \cdot R_u \text{ and } fwold_i^{n+1} = -2 \cdot r \cdot w2_i^n + r\left(w2_{i+1}^n + w2_{i-1}^n\right) + \Delta t \cdot R_w\,, \\ \text{where} \\ R_u = a \cdot w2_i^n \cdot t^A \text{ and } R_w = b \cdot u2_i^n \cdot t^{(-A-2)},\ R_u = -\dfrac{k \cdot \left(u2_i^n - w2_i^n \cdot t^B\right)}{t} \text{ and } R_w = \dfrac{k \cdot \left(u2_i^n \cdot t^{-B} - w2_i^n\right)}{t}. \end{array} \quad (20)$$

Finally, the updated values are calculated as follows:

$$u_i^{n+1} = u_i^n + \frac{(3 \cdot fu_i^{n+1} - fuold_i^{n+1})}{2} \text{ and } w_i^{n+1} = w_i^n + \frac{(3 \cdot fw_i^{n+1} - fwold_i^{n+1})}{2}.$$

3. The alternating-direction explicit (ADE) scheme is a known [46,47] but non-conventional method for which the condition of consistency is also known. We include it here for comparison purposes. In a one-dimensional equidistant mesh, one splits the calculation, i.e., one first sweeps the mesh from left to right, and then vice versa, independently. This means that the following equations are solved from left to right and from right to left, respectively:

$$uright_i^{n+1} = \frac{(1-r)u_i^n + r\left(u_{i+1}^n + uright_{i-1}^{n+1}\right) + \Delta t \cdot R_u}{1+r} \text{ and } uleft_i^{n+1} = \frac{(1-r)u_i^n + r\left(uleft_{i+1}^{n+1} + u_{i-1}^n\right) + \Delta t \cdot R_u}{1+r}$$

$$wright_i^{n+1} = \frac{(1-r)w_i^n + r\left(w_{i+1}^n + wright_{i-1}^{n+1}\right) + \Delta t \cdot R_w}{1+r} \text{ and}$$

$$wleft_i^{n+1} = \frac{(1-r)w_i^n + r\left(wleft_{i+1}^{n+1} + w_{i-1}^n\right) + \Delta t \cdot R_w}{1+r},$$

and (19) is used to calculate the reaction terms. The final values are the simple averages of the two half-sided terms:

$$u_i^{n+1} = (uright_i^{n+1} + uleft_i^{n+1})/2, \text{ and } w_i^{n+1} = (wright_i^{n+1} + wleft_i^{n+1})/2.$$

We note that for non-uniform meshes, the ADE method loses its fully explicit character, and matrix calculations would be necessary.

4. The Dufort–Frankel (DF) method is a classic example of explicit and stable methods [48] (p. 313) and is second-order in time. We adapted it to the case of Equation (2), where the following formula must be used

$$u_i^{n+1} = \frac{(1-2\cdot r)u_i^{n-1} + 2\cdot r\left(u_{i+1}^n + u_{i-1}^n\right) + 2\cdot \Delta t \cdot R_u}{1+2\cdot r}, \; w_i^{n+1} = \frac{(1-2\cdot r)w_i^{n-1} + 2\cdot r\left(w_{i+1}^n + w_{i-1}^n\right) + 2\cdot \Delta t \cdot R_w}{1+2\cdot r}, \quad (21)$$

and in (19), the latest u_i^n and w_i^n values are used. As can be seen, it is a two-step, one-stage method (the formula includes u_i^{n-1}). Since it is not self-starting, the calculation u_i^1 must be performed by another method. We use the UPFD formula for this purpose

$$u_i^1 = \frac{u_i^0 + r\left(u_{i+1}^0 + u_{i-1}^0\right) + \Delta t \cdot R_u}{1+2\cdot r}, \; w_i^1 = \frac{w_i^0 + r\left(w_{i+1}^0 + w_{i-1}^0\right) + \Delta t \cdot R_w}{1+2\cdot r} \quad (22)$$

where, of course, the initial values are used in Equation (19) for the reaction terms.

5. The original odd–even hopscotch (OOEH) method was discovered more than 50 years ago [49]. It requires a special spatial and temporal structure as shown in Figure 1. In essence, the mesh must be divided into two parts, the so-called odd and even nodes (or cells), where the closest neighbors of the even cells are odd, and vice versa. First, the FTCS formula given in Equation (18) is applied, but only to the odd cells. Then, it is followed by the BTCS (backward-time central-space) formula, which is based on implicit Euler time discretization with the equations

$$u_i^{n+1} = \frac{u_i^n + r\cdot\left(u_{i+1}^{n+1} + u_{i-1}^{n+1}\right) + \Delta t \cdot R_u}{1+2\cdot r} \quad w_i^{n+1} = \frac{w_i^n + r\cdot\left(w_{i+1}^{n+1} + w_{i-1}^{n+1}\right) + \Delta t \cdot R_w}{1+2\cdot r} \quad (23)$$

and with the application of (19). Since the values of the neighbors u_{i+1}^{n+1}, etc., are already calculated, this is an explicit method. After every time step, the odd and even labels are switched, as is illustrated in Figure 1.

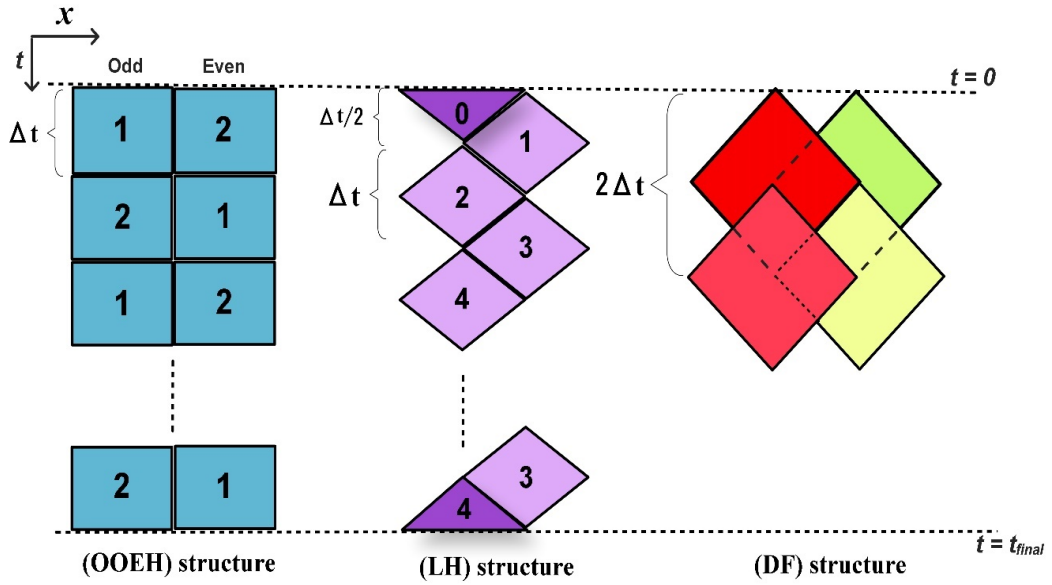


Figure 1. The structures of original odd–even hopscotch (OOEH), leapfrog–hopscotch (LH), and Dufort–Frankel (DF) methods.

6. The CpC method [50] is the organization of the so-called CNe scheme into a two-stage algorithm. The first stage is a fractional time step with half time step size employing the CNe formula:

$$u_i^{\text{pred}} = u_i^n \cdot e^{-r} + \frac{u_{i+1}^n + u_{i-1}^n + \Delta t \cdot R_u/r}{2}\left(1 - e^{-r}\right),\ w_i^{\text{pred}} = w_i^n \cdot e^{-r} + \frac{w_{i+1}^n + w_{i-1}^n + \Delta t \cdot R_w/r}{2}\left(1 - e^{-r}\right) \tag{24}$$

where Equation (19) is used for the reaction terms. These predictor values are used at the second stage, which is a full time step size corrector step; thus, the final values are as follows:

$$\begin{gathered} u_i^{n+1} = u_i^n \cdot e^{-2\cdot r} + \frac{u_{i+1}^{\text{pred}} + u_{i-1}^{\text{pred}} + \Delta t \cdot R_u/r}{2}\left(1 - e^{-2r}\right),\ w_i^{n+1} = u_i^n \cdot e^{-2\cdot r} + \frac{w_{i+1}^{\text{pred}} + w_{i-1}^{\text{pred}} + \Delta t \cdot R_w/r}{2}\left(1 - e^{-2r}\right) \\ R_u = a \cdot w_i^{\text{pred}} \cdot t^A \text{ and } R_w = b \cdot u_i^{\text{pred}} \cdot t^{(-A-2)},\ R_u = -\tfrac{k}{t}\left(u_i^{\text{pred}} - w_i^{\text{pred}} \cdot t^B\right) \text{ and } R_w = \tfrac{k}{t}\left(u_i^{\text{pred}} \cdot t^{-B} - w_i^{\text{pred}}\right) \end{gathered} \tag{25}$$

7. The next method is called the CCL algorithm, which is an abbreviation of Constant–Constant–Linear neighbor. It is a one-step but three-stage method, recently published by our group [51], with third-order accuracy. The CNe formula is applied in the first and second stages and the so-called linear-neighbor (LNe) formula in the last stage. The first stage is a predictor with a $\Delta t/3$-sized time step

$$\begin{gathered} au0_i = \frac{u_{i+1}^n + u_{i-1}^n + \Delta t \cdot R_u/r}{2},\ aw0_i = \frac{w_{i+1}^n + w_{i-1}^n + \Delta t \cdot R_w/r}{2} \\ u_i^C = u_i^n \cdot e^{-2r/3} + \left(1 - e^{-2r/3}\right) au0_i,\ w_i^C = w_i^n \cdot e^{-2r/3} + \left(1 - e^{-2r/3}\right) aw0_i\ , \end{gathered} \tag{26}$$

where Equation (19) is used again for the R terms. Then, the first corrector stage is performed:

$$\begin{aligned}
&au1_i = \frac{u_{i+1}^C + u_{i-1}^C + \Delta t \cdot R_u/r}{2},\ aw1_i = \frac{w_{i+1}^C + w_{i-1}^C + \Delta t \cdot R_w/r}{2}\\
&R_u = a \cdot w_i^C \cdot t^A \text{ and } R_w = b \cdot u_i^C \cdot t^{(-A-2)}\\
&R_u = -\frac{k \cdot \left(u_i^C - w_i^C \cdot t^B\right)}{t} \text{ and } R_w = \frac{k \cdot \left(u_i^C \cdot t^{-B} - w_i^C\right)}{t}\\
&u_i^{CC} = u_i^n \cdot e^{-4r/3} + \left(1 - e^{-4r/3}\right) au1_i,\ w_i^{CC} = w_i^n \cdot e^{-4 \cdot r/3} + \left(1 - e^{-4r/3}\right) aw1_i
\end{aligned} \tag{27}$$

The last stage is as follows:

$$\begin{aligned}
&au2_i = \frac{u_{i+1}^{CC} + u_{i-1}^{CC} + \Delta t \cdot R_u/r}{2},\ aw2_i = \frac{w_{i+1}^{CC} + w_{i-1}^{CC} + \Delta t \cdot R_w/r}{2}\\
&R_u = a \cdot w_i^{CC} \cdot t^A \text{ and } R_w = b \cdot u_i^{CC} \cdot t^{(-A-2)},\\
&R_u = -\frac{k \cdot (u_i^{CC} - w_i^{CC} \cdot t^B)}{t} \text{ and } R_w = \frac{k \cdot (u_i^{CC} \cdot t^{-B} - w_i^{CC})}{t}\\
&u_i^{n+1} = u_i^n \cdot e^{-2 \cdot r} + \left(1 - e^{-2 \cdot r}\right) au0_i + \frac{3 \cdot (au2_i - au0_i)}{2} \cdot \left(1 - \frac{e^{-2 \cdot r}}{2 \cdot r}\right)\\
&w_i^{n+1} = w_i^n \cdot e^{-2 \cdot r} + \left(1 - e^{-2 \cdot r}\right) aw0_i + \frac{3 \cdot (aw2_i - aw0_i)}{2} \cdot \left(1 - \frac{e^{-2 \cdot r}}{2 \cdot r}\right)
\end{aligned} \tag{28}$$

8. The four-stage CLQ2 (Constant–Linear–Quadratic–Quadratic) neighbor method [52] is a bit more complicated than the previous ones, since it has four stages and it employs the quadratic-neighbor approximation as well. Its first stage utilizes the CNe formula

$$\begin{aligned}
au_i^C &= \frac{u_{i+1}^n + u_{i-1}^n + \Delta t \cdot R_u/r}{2} & aw_i^C &= \frac{w_{i+1}^n + w_{i-1}^n + \Delta t \cdot R_w/r}{2}\\
u_i^C &= u_i^n \cdot e^{-2 \cdot r} + au_i^C \cdot (1 - e^{-2 \cdot r}), & w_i^C &= w_i^n \cdot e^{-2 \cdot r} + aw_i^C \cdot (1 - e^{-2 \cdot r})
\end{aligned} \tag{29}$$

where Equation (19) holds again. After this, the LNe formula is applied in the second stage with not only a full time step, but simultaneously a half-length time step as well, to obtain the sets u_i^{CL}, w_i^{CL} and $u_i^{CL^{1/2}}$, $w_i^{CL^{1/2}}$, respectively:

$$\begin{aligned}
&au_i^{CL} = \frac{u_{i+1}^C + u1_{i-1}^C + \Delta t \cdot R_u/r}{2} \quad \text{and} \quad aw_i^{CL} = \frac{w_{i+1}^C + w_{i-1}^C + \Delta t \cdot R_w/r}{2}\\
&R_u = a \cdot w_i^C \cdot t^A \text{ and } R_w = b \cdot u_i^C \cdot t^{(-A-2)}, \text{ or}\\
&R_u = -\frac{k \cdot (u_i^C - w_i^C \cdot t^B)}{t} \text{ and } R_w = \frac{k \cdot (u_i^C \cdot t^{-B} - w_i^C)}{t}\\
&u_i^{CL} = u_i^n \cdot e^{-2 \cdot r} + au_i^{CL} \cdot (1 - e^{-2 \cdot r}) + (au_i^{CL} - au_i^C) \cdot \left(1 - \frac{e^{-2 \cdot r}}{2 \cdot r}\right)\\
&u_i^{CLmid} = u_i^n \cdot e^{-r} + au_i^{CL} \cdot (1 - e^{-r}) + \frac{au_i^{CL} - au_i^C}{2} \cdot \left(1 - \frac{e^{-r}}{r}\right)\\
&w_i^{CL} = w_i^n \cdot e^{-2 \cdot r} + aw_i^{CL} \cdot (1 - e^{-2 \cdot r}) + (aw_i^{CL} - aw_i^C) \cdot \left(1 - \frac{e^{-2 \cdot r}}{2 \cdot r}\right)\\
&w_i^{CLmid} = w_i^n \cdot e^{-r} + aw_i^{CL} \cdot (1 - e^{-r}) + \frac{aw_i^{CL} - aw_i^C}{2} \cdot \left(1 - \frac{e^{-r}}{r}\right)
\end{aligned} \tag{30}$$

The third stage contains the calculations below:

$$
\begin{aligned}
&au_i^{CLQ} = \frac{u_{i+1}^{CL} + u_{i+1}^{CL} + \Delta t \cdot R_u/r}{2},\ aw_i^{CLQ} = \frac{w_{i+1}^{CL} + w_{i-1}^{CL} + \Delta t \cdot R_w/r}{2}\\
&au_{i,mid}^{CLQ} = \frac{u_{i+1}^{CLmid} + u_{i+1}^{CLmid} + \Delta t \cdot R_{au1}/r}{2},\ aw_{i,mid}^{CLQ} = \frac{w_{i+1}^{CLmid} + w_{i-1}^{CLmid}) + \Delta t \cdot R_{aw1}/r}{2}\\
&R_u = a \cdot w_i^{CL} \cdot t^A \text{ and } R_w = b \cdot u_i^{CL} \cdot t^{(-A-2)} \quad \text{or} \quad R_u = a \cdot w_i^{CL} \cdot t^A \text{ and } R_w = b \cdot u_i^{CL} \cdot t^{(-A-2)}\\
&R_{au1} = -\frac{k \cdot (u_i^{CL} - w_i^{CL} \cdot t^A)}{t} \text{ and } R_{aw1} = \frac{k \cdot (u_i^{CL} \cdot t^{-A} - w_i^{CL})}{t}\\
&su = 4au_{i,mid}^{CLQ} - au_i^{CLQ} - 3 \cdot au_i^C,\ gu = 2 \cdot (au_i^{CLQ} - 2au_{i,mid}^{CLQ} + au_i^C)\\
&sw = 4aw_{i,mid}^{CLQ} - aw_i^{CLQ} - 3 \cdot aw_i^C,\ gw = 2 \cdot (aw_i^{CLQ} - 2aw_{i,mid}^{CLQ} + aw_i^C)\\
&u_i^{CLQ} = u_i^n \cdot e^{-2 \cdot r} + (1 - e^{-2 \cdot r}) \cdot (gu/2/r^2 - su/2/r + au_i^C) + gu - gu/r + su\\
&w_i^{CLQ} = w_i^n \cdot e^{-2 \cdot r} + (1 - e^{-2 \cdot r}) \cdot (gw/2/r^2 - sw/2/r + aw_i^C) + gw - gw/r + sw\\
&u_{i,\,mid}^{CLQ} = u_i^n \cdot e^{-r} + (1 - e^{-r}) \cdot (gu/2/r^2 - su/2/r + au_i^C) + gu/4 - gu/2/r + su/2\\
&w_{i,\,mid}^{CLQ} = w_i^n \cdot e^{-r} + (1 - e^{-r}) \cdot (gw/2/r^2 - sw/2/r + aw_i^C) + gw/4 - gw/2/r + sw/2
\end{aligned}
\tag{31}
$$

There is no need to mention that the su, sw, gu, and gw auxiliary quantities also depend on the space and time level, but since they are used only immediately after their evaluation, they can be defined as scalar variables instead of arrays. We add the fourth stage by using the last four values u_i^{CLQ}, …, $w_{i,\,mid}^{CLQ}$. Using these, one can calculate au_i^{CLQQ},…, $aw_{i,mid}^{CLQQ}$ values in the same manner as the previous formulas. Then, one repeats the calculations of the third stage, including those of su, …, gw, and finally obtains the u_i^{CLQQ} and w_i^{CLQQ} sets as final results at the end of the actual time step. With these, one has four stages altogether, which constitute the CLQ2 algorithm [52].

9. The pseudo-implicit (PI) two-stage method was developed in [53] (Algorithm 5 there, applied to the pure diffusion Equation (1)). A half time step is taken to obtain the predictor values and then a full time step for the corrector values. The first-stage formulas are the following:

$$
u_i^{\text{pred}} = \frac{u_i^n + r/2\left(u_{i+1}^n + u_{i-1}^n\right) + \Delta t \cdot R_u/2}{1 + r} \qquad w_i^{\text{pred}} = \frac{w_i^n + r/2\left(w_{i+1}^n + w_{i-1}^n\right) + \Delta t \cdot R_w/2}{1 + r} \tag{32}
$$

The second stage is as follows

$$
u_i^{n+1} = \frac{(1 - r)u_i^n + r\left(u_{i+1}^{\text{pred}} + u_{i-1}^{\text{pred}}\right) + \Delta t \cdot R_u}{1 + r} \qquad w_i^{n+1} = \frac{(1 - r)w_i^n + r\left(w_{i+1}^{\text{pred}} + w_{i-1}^{\text{pred}}\right) + \Delta t \cdot R_w}{1 + r} \tag{33}
$$

and (19) is used in both stages, but in the second stage, the predictor u and w values are substituted.

10. The recently invented leapfrog–hopscotch (LH) approach [42] also requires the odd–even space structure. Moreover, it has a structure made up of many full time steps and two half time steps, as can be seen in Figure 1. Using the initial values, the calculation begins by taking a half-sized time step for the odd nodes. Full time steps are then taken strictly alternately for the even and odd nodes (marked by light purple diamonds in Figure 1) until the last time step is reached, which should be halved for odd nodes to reach the same final time point as the even nodes. Since the first stage's time step is halved, the following general formula is applied:

$$
u_i^{odd} = \frac{u_i^n + r/2\left(u_{i+1}^n + u_{i-1}^n\right) + \Delta t \cdot R_u/2}{1 + r} \text{ and } w_i^{odd} = \frac{w_i^n + r/2\left(w_{i+1}^n + w_{i-1}^n\right) + \Delta t \cdot R_w/2}{1 + r} \tag{34}
$$

Next, for the even nodes, a full time step is made using

$$
u_i^{even} = \frac{(1 - r)u_i^n + r \cdot \left(u_{i+1}^n + u_{i-1}^n\right) + \Delta t \cdot R_u}{1 + r} \qquad w_i^{even} = \frac{(1 - r)w_i^n + r \cdot \left(w_{i+1}^n + w_{i-1}^n\right) + \Delta t \cdot R_w}{1 + r} \tag{35}
$$

When the calculations are performed, the latest available values of the left and right neighbors are always used. This is not true for the reaction terms, where the values at the

beginning of the actual time step are substituted to Equation (19) even if newer values are available for u_i.

3.3. Properties of the Algorithms

The methods have well-established properties when they are applied to the simple Equation (1). First of all, they have the following theoretical order of temporal accuracy. The FTCS is first, the AB2, ADE, DF, OOEH, CpC, CCL, PI and LH methods are second, the CCL method is third, and finally the CLQ2 scheme is fourth in order of time step size. With the exception of the FTCS and AB2, the methods are unconditionally stable for the simple diffusion Equation (1), i.e., the previously mentioned CFL limit does not hold for them. We need to stress again that among explicit methods, unconditional stability is not the rule but the exception.

We note that the CpC, CCL, CLQ2, PI, and LH methods are recently invented by our research group, and the verifications, analytical proofs, etc., are typically presented in those original papers.

Furthermore, the CpC and the CLQ2 methods have an extra property that the calculated u_i^{n+1} values are the convex combination of the old u_i^n, u_{i-1}^n, u_{i+1}^n, etc., values in the case of Equation (1), thus the maximum and minimum principles are trivially fulfilled and unphysical oscillations cannot appear, even for extremely large time step sizes. Based on this, the positivity-preserving property can be easily proved for the CpC scheme when applied to the system of Equation (12) with simple coupling for non-negative coupling coefficients.

Proposition 1. *Assume that the CpC method is applied to Equation (12). If the initial u and w values are non-negative, than the new values are also non-negative for arbitrary* $A \in \mathbb{R}$ *and* $D,\ a,\ b,\ \Delta t\ \geq 0$.

Proof. In the Formulas (24) and (25), the coefficient of each term u_i^n, u_{i-1}^n, u_{i+1}^n and R_u, R_w lies in the unit interval. Since the u_j^n values are non-negative, $j = 1, \ldots, N_x$, the reaction terms R_u, R_w are also non-negative, which implies the statement. □

Remark 1. *We note that the goal of this work is the numerical investigation of the ten listed methods. A detailed analytical examination of their properties when applied to the studied system of equations is out of scope of this work.*

4. Verification in 1D

The L2 error norm is determined by taking the square root of the sum of the squared differences between the numerical solution and the analytical (reference) solution at each point:

$$\begin{aligned} Error(L_2 u) &= \left(\sum_{i=1}^{N_x} \left(u_i^n - u^{ref}(x_1, t_n) \right)^2 \right)^{1/2}, \\ Error(L_2 w) &= \left(\sum_{i=1}^{N_x} \left(w_i^n - w^{ref}(x_1, t_n) \right)^2 \right)^{1/2}. \end{aligned} \tag{36}$$

We perform two case studies, one for the simple coupling and one for isomerization using the analytical solutions (13), (14), (16), and (17), with different parameter values. We always use the same coefficients $c_1 = 1,\ c_2 = 0,\ c_3 = 0,\ c_4 = 0$ in each case study, including the rest of this paper.

Case Study 1: Diffusion with simple coupling

In this experiment, we used the analytical solutions (13) and (14) with the following parameters: $x_0 = -2$, $L_x = 4$, $N_x = 1000$, $t^{\text{in}} = 0.04$, TIME $= 0.1$, $a = 4$, $b = 9$, $A = -0.6$, $\alpha = 0.87$.

First, we present the graphs of the initial function $u(x, t = 0.04)$, the reference analytical solution $u^{exact}(x,\ t^{fin} = 0.14)$, and two corresponding numerical solutions for $\Delta t = 10^{-4}$ in Figure 2. Then, in Figure 3, we display the errors calculated by Formula (36) as a function of the time step size for both u and w for each method. We consider the verification successful, since for decreasing Δt, the errors are decreasing as expected based on our similar experiments conducted for single diffusion-type equations.

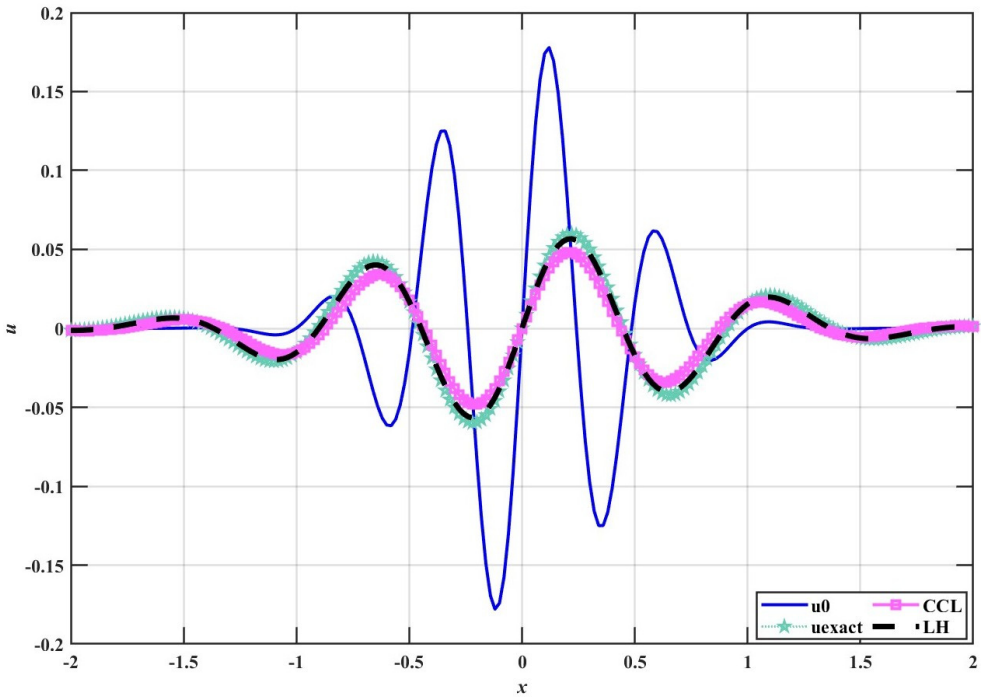


Figure 2. The concentration of u in terms of x in the case of the initial function u^0; for u^{exact} analytical solution, CLL and LH are the numerical methods.

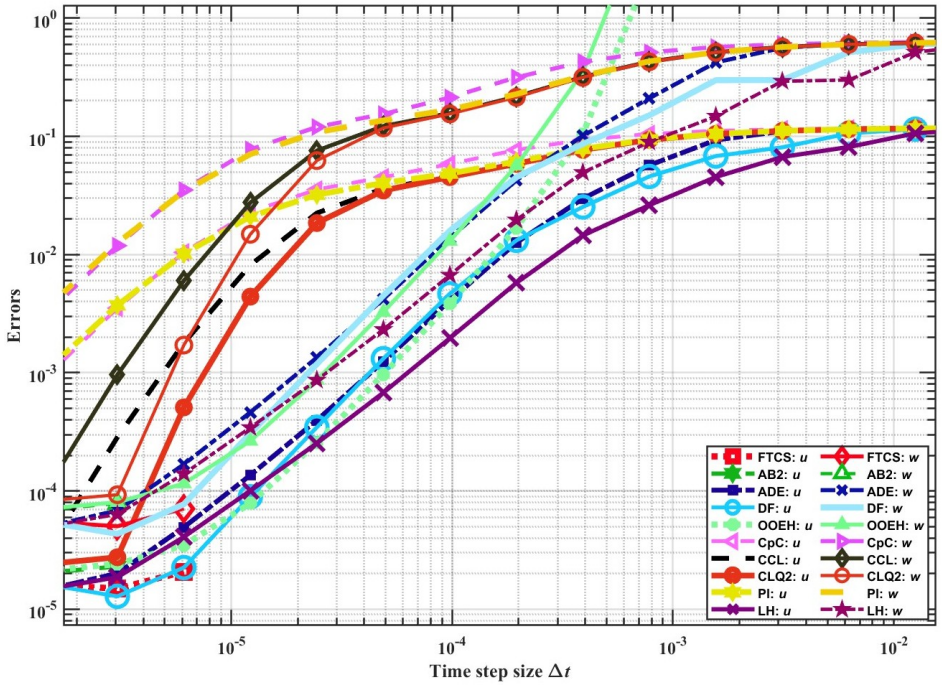


Figure 3. L_2 errors as a function of the time step size Δt for the simple coupling case.

Case study 2: Isomerization for large coupling coefficient

In this experiment, we used the analytical solutions (16) and (17) with the following parameters:

$x_0 = -2$, $L_x = 4$, $N_x = 800$, $t^{\text{in}} = 0.04$, TIME $= 0.24$, $k = 3$, $B = 0.26$, $\alpha = 0.5$. We present the graphs of the initial function $u(x, t = 0.04)$. The reference analytical solution $u^{exact}(x,\ t^{fin})$, and two corresponding numerical solutions with $\Delta t = 1.25 \cdot 10^{-4}$ in Figure 4. Then, in Figure 5, we display the errors as a function of the time step size for both u and w for each method.

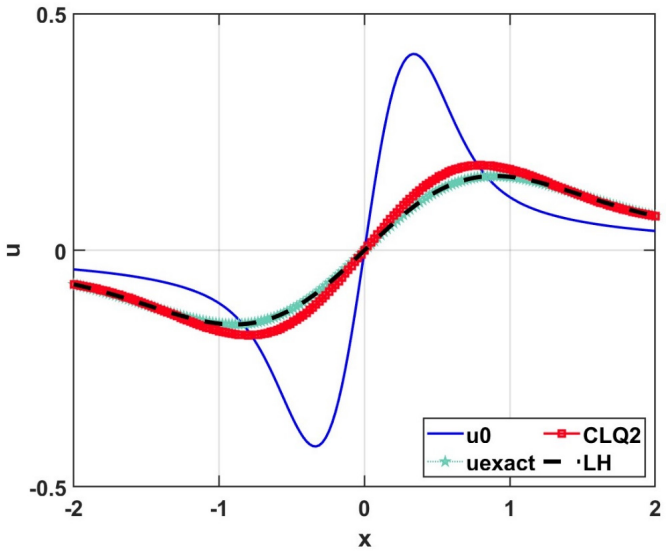


Figure 4. The concentration of u in terms of x in the case of the initial function u^0, for u^{exact} analytical solution, and the numerical solutions by the CLQ2 and LH schemes.

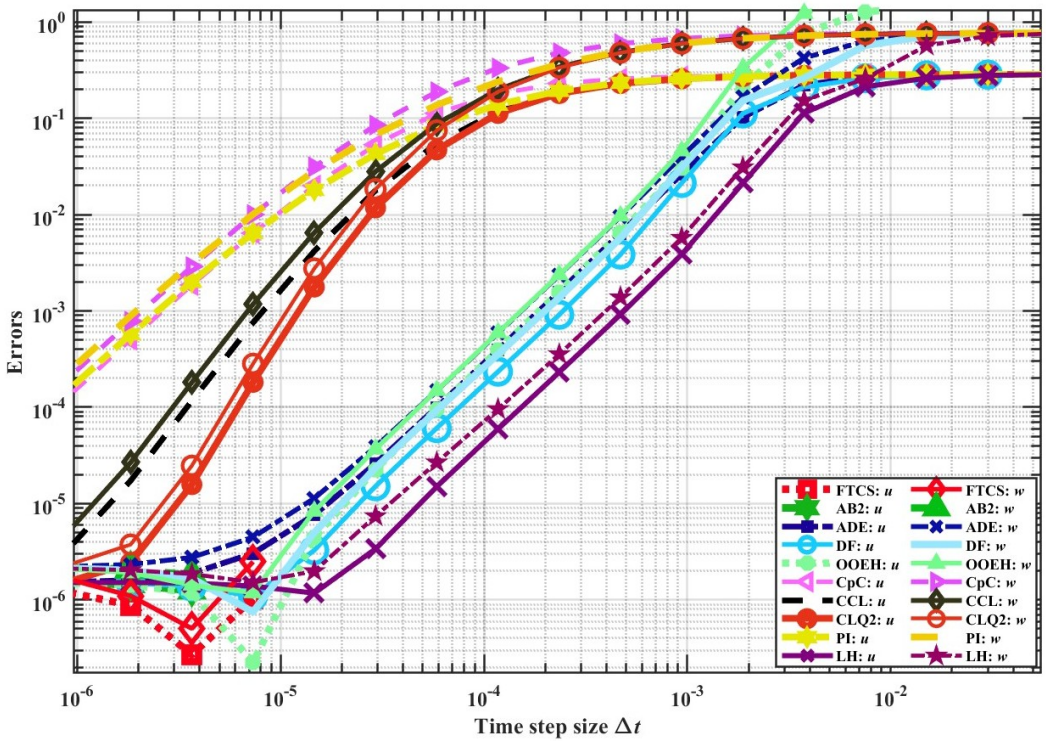


Figure 5. Errors as a function of the time step size Δt for the simple coupling case.

5. Parameter Sweeps for Further Testing of the Methods

In this section, we conduct parameter sweep experiments to explore how the performance of the methods depends on the parameters. In the case of simple coupling and isomerization,

the analytical solutions (13), (14), (16), and (17) will be used for reference purposes, and we fix all the parameters except those we sweep. So unless it is stated explicitly otherwise, the PDEs are discretized using $x_0 = 1$, $L_x = 0.8$, $N_x = 100$, $t^{\text{in}} = 0.1$, TIME $= 0.1$, while the coefficients are $a = 3$, $b = 2a$, $k = 0.3$, $A = -0.9$, $\alpha = 0.6$ in each simulation.

We calculate the average of the logarithm of the maximum or L_∞ errors, i.e., the following formula is used:

$$AgErr(L_\infty u) = \frac{1}{S}\sum_{s=1}^{S} \log\left(\max_i \left|u_i^n - u^{ref}(x_1, t_n)\right|\right), \tag{37}$$

which will be called the aggregated error. Our goal with this aggregated error *AgErr* is to assess the overall accuracy of the methods for small, medium, and large time step sizes. Note that when the *AgErr* is negative with a large absolute value, the method is very accurate. In this section, we calculated the error for $S = 9$ different time step sizes for all the examined methods. Since the FTCS and AB2 methods produce "*NaN*" (not-a-number) values, the aggregated errors cannot be calculated for them; therefore, we proceed only with the remaining eight methods with good stability properties.

Case Study 3: Errors as a function of parameter *a* for the simple coupling case

The following series is used for the values of the coefficient: $a \in \{0.1, 0.3, 1, 3, 6, 10, 20, 30, 100, 300\}$; $b = 2a$ holds, which means that the two coupling coefficients a and b grow together. The aggregated errors are displayed in Table 1 and Figure 6.

Table 1. The aggregated errors (*AgErr*) for the values of *a* for simple coupling in Case Study 3.

Numerical Methods	0.1	0.3	1	3	6	10	20	30	100	300
ADEu	−20.45	−18.70	−12.14	−8.56	−2.21	4.33	24.69	38.61	166.61	420.26
ADEw	−16.91	−16.05	−10.02	−6.51	−0.16	6.36	26.69	40.61	168.60	422.25
DFu	−20.29	−16.63	−13.09	−11.74	−5.26	−0.79	13.52	23.05	87.91	215.24
DFw	−16.92	−14.68	−10.84	−9.73	−3.22	1.22	15.56	25.08	89.93	217.23
OOEHu	−18.71	−17.06	−10.68	−4.49	6.10	18.11	50.65	75.71	267.83	676.56
OOEHw	−15.08	−14.04	−8.44	−2.19	8.58	20.86	53.98	79.51	273.61	684.90
CpCu	−9.40	−6.12	−4.96	−2.95	−1.26	−1.36	0.29	3.41	33.03	123.56
CpCw	−6.54	−3.79	−2.69	−0.84	0.80	0.69	2.31	5.44	35.03	125.56
CCLu	−10.27	−7.05	−6.38	−4.30	−1.89	−1.14	3.81	9.31	60.57	208.27
CCLw	−7.42	−4.71	−4.17	−2.22	0.16	0.90	5.82	11.33	62.57	210.27
CLQ2u	−10.46	−7.23	−6.61	−4.51	−1.96	−1.05	4.33	10.23	65.33	228.88
CLQ2w	−7.61	−4.90	−4.41	−2.43	0.09	0.98	6.33	12.25	67.33	230.88
PIu	−9.80	−6.58	−5.78	−3.67	−1.68	−1.38	1.97	6.32	47.37	165.17
PIw	−6.95	−4.24	−3.57	−1.57	0.37	0.66	3.98	8.34	49.37	167.16
LHu	−23.69	−19.97	−17.01	−15.08	−6.31	−0.86	17.25	29.56	113.61	278.26
LHw	−20.43	−18.05	−14.78	−12.98	−4.29	1.14	19.28	31.58	115.61	280.25

Case Study 4: Errors as a function of the *b* parameter, simple coupling case

The series used for the values of the coefficient is $b \in \{0.1, 0.3, 1, 3, 10, 30, 100, 300\}$, while $a = 3$ is kept fixed, thus only the second is increased gradually. The aggregated errors are displayed in Table 2 and Figure 7.

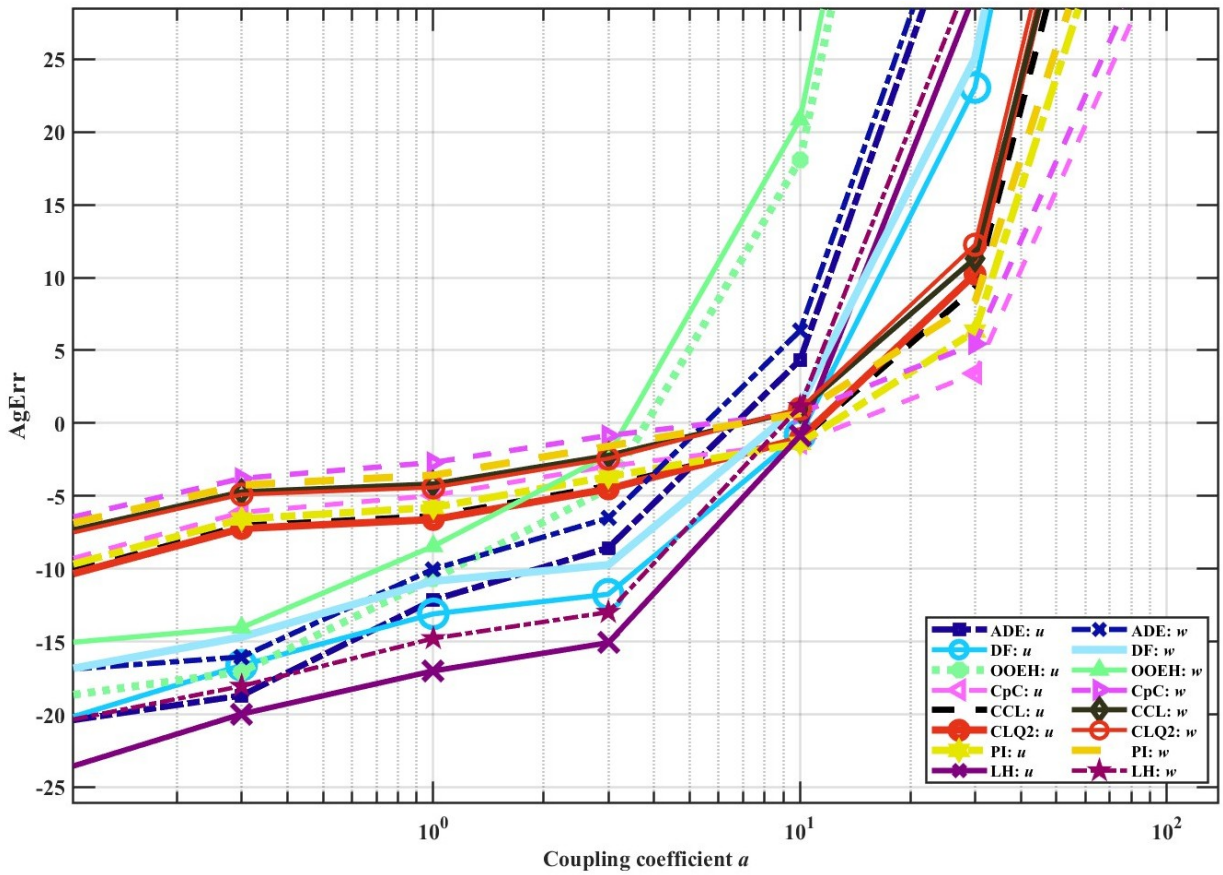


Figure 6. Aggregated errors as a function of the number of a for the 8 methods in Case Study 3.

Table 2. The aggregated errors (*AgErr*) for the values of b for simple coupling in Case Study 4.

Numerical Methods	0.1	0.3	1	3	10	30	100	300
ADEu	−16.03	−12.90	−9.03	−8.85	−5.46	−3.17	4.56	12.58
ADEw	−21.48	−16.69	−10.41	−8.08	−2.41	2.00	12.07	22.22
DFu	−12.74	−9.49	−12.40	−8.31	−8.02	−6.01	−1.46	5.60
DFw	−18.70	−13.19	−13.79	−7.64	−4.96	−0.83	6.06	15.30
OOEHu	−14.57	−11.11	−7.42	−6.09	−1.32	5.14	21.09	42.46
OOEHw	−19.50	−14.86	−8.60	−4.92	2.02	10.82	29.48	53.50
CpCu	−2.93	−0.99	−2.88	−1.18	−3.12	−4.28	−5.87	−7.54
CpCw	−8.62	−4.53	−4.21	−0.44	−0.02	0.92	1.67	2.09
CCLu	−3.89	−2.15	−3.82	−1.92	−3.74	−4.51	−4.72	−4.03
CCLw	−9.59	−5.68	−5.20	−1.19	−0.65	0.68	2.80	5.59
CLQ2u	−4.08	−2.35	−4.01	−2.08	−3.87	−4.53	−4.52	−3.52
CLQ2w	−9.77	−5.88	−5.40	−1.35	−0.78	0.66	3.01	6.10
PIu	−3.42	−1.64	−3.36	−1.53	−3.51	−4.56	−5.53	−5.84
PIw	−9.11	−5.17	−4.73	−0.80	−0.42	0.64	2.00	3.78
LHu	−16.09	−12.95	−16.06	−10.53	−9.53	−6.72	−0.28	9.53
LHw	−22.09	−16.70	−17.45	−9.87	−6.48	−1.55	7.23	19.21

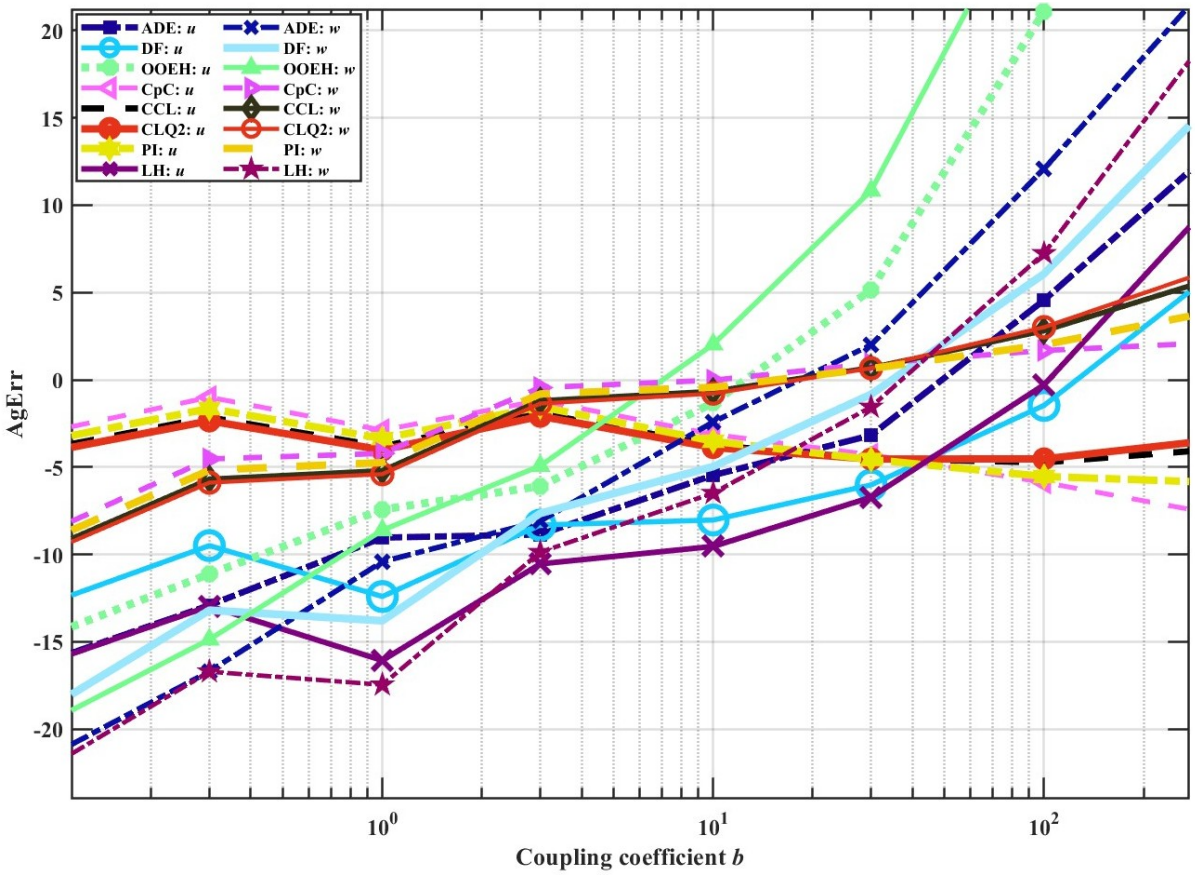


Figure 7. Aggregated errors as a function of the coefficient b in Case Study 4.

Case Study 5: Errors as a function of the A parameter, simple coupling case

The following series is used for the values of the coefficient: $A \in \{-10, -3, -1, 0, 1, 3, 10\}$. The aggregated errors are displayed in Table 3 and Figure 8.

Table 3. The aggregated errors (*AgErr*) for the values of A for simple coupling in Case Study 5.

Numerical Methods	−10	−3	−1	0	1	3	10
ADEu	−8.43	−10.59	−8.10	−13.86	−22.27	−38.65	−88.89
ADEw	−67.24	−23.25	−6.75	−5.48	−6.57	−8.04	−9.96
DFu	−8.09	−7.60	−10.85	−18.43	−26.66	−43.72	−97.15
DFw	−65.49	−19.45	−9.50	−9.45	−10.23	−12.91	−14.55
OOEHu	−9.87	−6.78	−4.01	−9.94	−17.95	−32.74	−80.60
OOEHw	−67.03	−18.38	−2.38	−1.55	−2.83	−4.04	−5.52
CpCu	−2.64	−0.58	−2.25	−9.92	−19.80	−37.71	−92.86
CpCw	−60.87	−13.13	−0.90	−0.85	−2.57	−4.43	−6.22
CCLu	−3.47	−1.45	−3.59	−10.94	−20.54	−37.94	−92.41
CCLw	−61.37	−13.91	−2.24	−2.05	−3.74	−5.56	−7.96
CLQ2u	−3.64	−1.62	−3.84	−11.10	−20.69	−38.05	−92.39
CLQ2w	−61.51	−14.06	−2.49	−2.22	−3.90	−5.73	−8.15
PIu	−3.11	−1.02	−2.96	−10.54	−20.24	−37.70	−91.99
PIw	−61.07	−13.50	−1.60	−1.62	−3.36	−5.17	−7.40
LHu	−10.00	−10.08	−14.72	−19.77	−27.89	−44.40	−100.32
LHw	−67.32	−21.82	−13.37	−11.13	−12.03	−14.53	−20.20

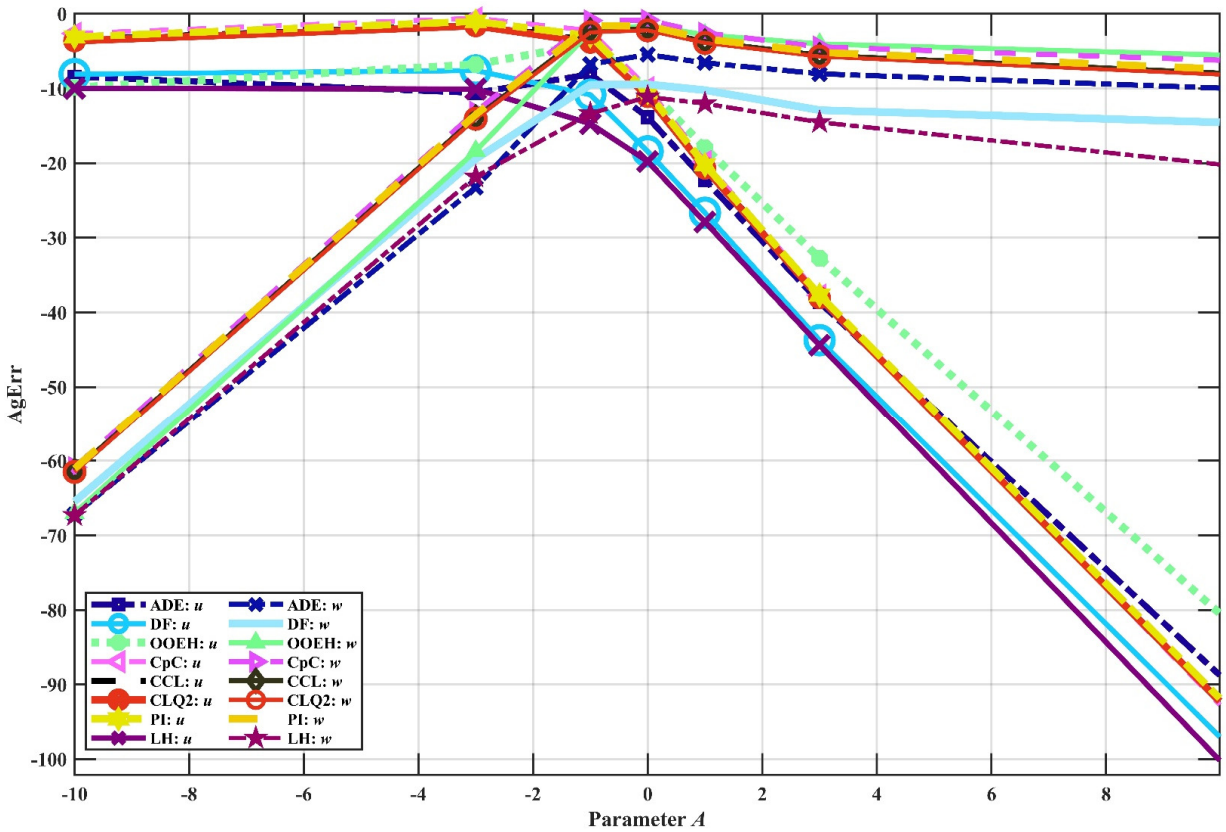


Figure 8. Aggregated errors as a function of the parameter value of A in Case Study 5.

Case Study 6: Errors as a function of the TIME parameter, simple coupling case

The following series is used for the values of the coefficient: $TIME \in \{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300\}$. The aggregated errors are displayed in Table 4 and Figure 9.

Table 4. The aggregated errors (*AgErr*) for the values of A for simple coupling in Case Study 6.

Numerical Methods	0.0001	0.0003	0.001	0.003	0.01	0.03	0.1	0.3	1	3	10	30	100	300
ADE*u*	−52.56	−47.12	−40.45	−32.62	−22.69	−13.64	−8.56	−7.20	−9.39	−9.53	−8.55	−7.80	−6.43	−5.91
ADE*w*	−50.28	−44.85	−38.18	−30.35	−20.43	−11.45	−6.51	−5.31	−7.30	−7.54	−6.51	−5.48	−4.14	−3.61
DF*u*	−48.82	−42.92	−36.21	−29.69	−22.20	−15.01	−11.74	−6.28	−4.83	−4.69	−2.64	−2.06	−2.67	−3.25
DF*w*	−46.49	−40.58	−33.91	−27.42	−19.95	−12.82	−9.73	−4.41	−3.13	−3.20	−0.91	−0.11	−0.96	−1.54
OOEH*u*	−53.20	−48.50	−41.81	−34.40	−24.77	−14.54	−4.49	6.83	18.73	34.10	52.17	63.97	73.20	79.25
OOEH*w*	−51.96	−46.83	−39.85	−32.11	−22.22	−11.89	−2.19	8.97	20.59	35.56	53.21	64.58	73.34	78.96
CpC*u*	−47.82	−39.68	−30.37	−21.53	−12.55	−5.79	−2.95	−0.40	−8.05	−4.40	−5.53	−4.64	−4.50	−4.52
CpC*w*	−45.48	−37.32	−27.99	−19.24	−10.27	−3.56	−0.84	1.52	−5.94	−3.30	−2.81	−2.26	−2.21	−2.23
CCL*u*	−51.52	−44.14	−35.15	−26.33	−16.16	−7.85	−4.30	−0.74	−8.23	−4.99	−6.37	−5.54	−5.59	−5.73
CCL*w*	−49.20	−41.80	−32.78	−24.04	−13.90	−5.63	−2.22	1.15	−6.20	−4.01	−3.75	−3.27	−3.37	−3.49
CLQ2*u*	−52.89	−45.88	−36.93	−28.14	−17.98	−8.68	−4.51	−0.77	−8.43	−5.01	−6.37	−5.54	−5.59	−5.73
CLQ2*w*	−50.56	−43.52	−34.56	−25.83	−15.70	−6.46	−2.43	1.12	−6.21	−4.02	−3.75	−3.27	−3.37	−3.50
PI*u*	−47.69	−39.67	−30.54	−21.83	−13.03	−6.41	−3.67	−0.66	−7.59	−4.95	−6.24	−5.41	−5.41	−5.43
PI*w*	−45.35	−37.31	−28.15	−19.55	−10.76	−4.18	−1.57	1.23	−5.98	−3.93	−3.65	−3.15	−3.20	−3.21
LH*u*	−50.12	−44.52	−38.13	−31.76	−24.23	−17.10	−15.08	−9.11	−8.07	−8.73	−6.69	−4.02	−3.19	−3.73
LH*w*	−47.79	−42.19	−35.85	−29.50	−22.01	−14.92	−12.98	−7.29	−6.41	−7.14	−5.11	−2.26	−1.40	−1.85

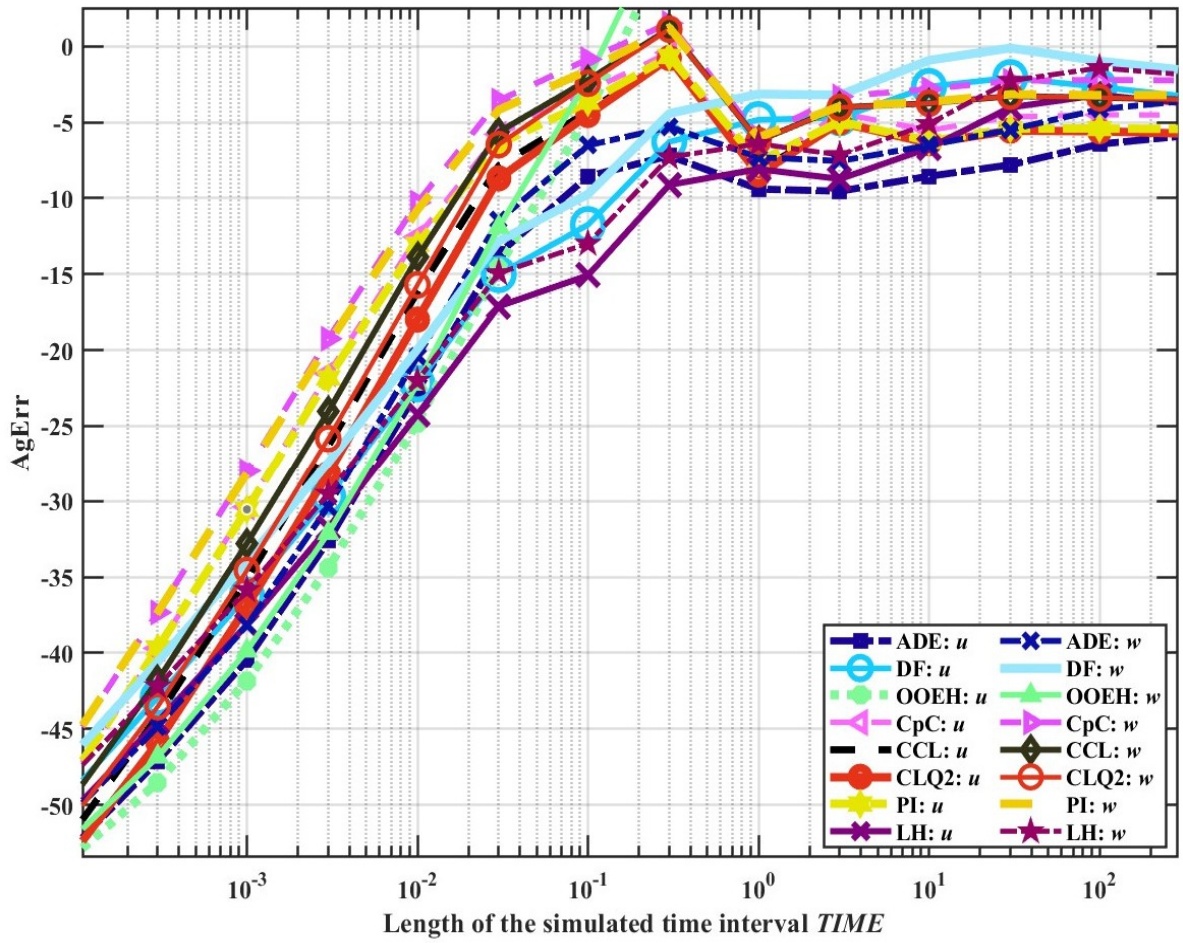


Figure 9. Aggregated errors as a function of the duration of the simulation *TIME* in Case Study 6.

Case Study 7: Errors as a function of the TIME parameter, isomerization case

The following series is used for the values of the coefficient: $TIME \in \{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300\}$. The aggregated errors are displayed in Table 5 and Figure 10.

Table 5. The aggregated errors (*AgErr*) for the values of *TIME* for isomerization in Case Study 7.

Numerical Method	0.0001	0.0003	0.001	0.003	0.01	0.03	0.1	0.3	1	3	10	30	100	300
ADE*u*	−68.44	−62.54	−55.36	−47.34	−37.54	−28.60	−20.34	−17.65	−15.43	−14.64	−12.99	−11.14	−9.63	−9.00
ADE*w*	−81.73	−76.07	−68.39	−60.20	−50.42	−41.13	−31.54	−26.54	−24.12	−22.05	−21.29	−20.08	−19.59	−19.58
DF*u*	−62.42	−55.65	−47.85	−40.71	−32.86	−25.84	−19.77	−14.45	−9.78	−7.44	−3.44	−1.75	−0.93	−0.64
DF*w*	−71.05	−63.66	−55.17	−47.40	−39.11	−32.38	−29.13	−21.52	−19.26	−17.15	−12.74	−11.96	−11.40	−11.55
OOEH*u*	−69.91	−64.44	−57.03	−49.07	−39.32	−29.95	−19.48	−15.66	−9.79	−5.81	0.64	5.71	10.40	14.62
OOEH*w*	−82.76	−76.75	−68.90	−60.71	−50.28	−40.29	−29.67	−22.47	−16.36	−8.60	1.33	10.16	19.05	27.14
CpC*u*	−61.06	−51.54	−40.12	−30.35	−21.57	−14.87	−8.71	−4.57	−2.22	−1.46	−1.23	−1.19	−1.19	−1.19
CpC*w*	−69.20	−58.62	−46.87	−37.18	−28.23	−21.67	−16.29	−13.01	−11.39	−11.62	−12.87	−14.84	−16.25	−17.81
CCL*u*	−66.24	−57.47	−46.83	−36.66	−25.40	−17.05	−9.91	−5.33	−2.84	−2.13	−2.01	−2.02	−2.05	−2.06
CCL*w*	−76.31	−66.46	−54.48	−43.21	−31.79	−23.54	−17.17	−13.63	−12.20	−12.97	−14.14	−15.73	−16.92	−18.29
CLQ2*u*	−67.91	−59.61	−49.04	−38.79	−27.70	−17.91	−10.12	−5.39	−2.86	−2.14	−2.01	−2.03	−2.05	−2.07
CLQ2*w*	−78.83	−69.35	−57.48	−46.18	−33.82	−24.36	−17.36	−13.67	−12.22	−13.00	−14.15	−15.74	−16.92	−18.29
PI*u*	−60.84	−51.36	−40.12	−30.60	−22.03	−15.48	−9.38	−5.17	−2.78	−2.10	−1.98	−2.00	−2.03	−2.04
PI*w*	−68.88	−58.42	−46.89	−37.36	−28.55	−22.06	−16.69	−13.46	−12.12	−12.90	−14.13	−15.70	−16.85	−18.04
LH*u*	−63.78	−57.23	−49.73	−42.79	−35.23	−28.59	−22.32	−17.01	−13.13	−11.22	−7.89	−3.91	−1.63	−0.94
LH*w*	−73.22	−66.26	−58.30	−50.91	−42.77	−35.93	−30.48	−24.46	−22.75	−20.90	−16.49	−13.42	−11.13	−10.88

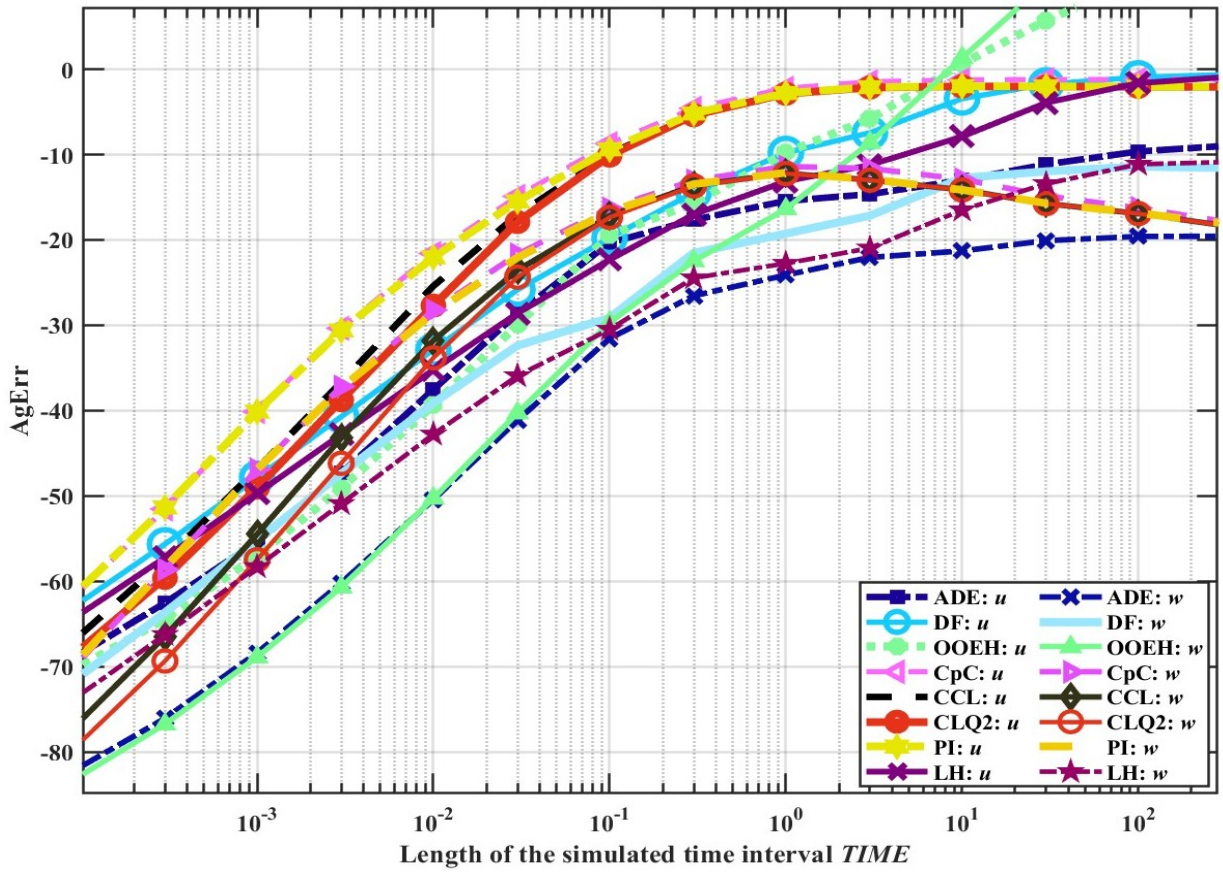


Figure 10. Aggregated errors as a function of the duration of the simulation *TIME* in Case Study 7.

Case Study 8: Errors as a function of the *k* parameter, isomerization case

The following series is used for the values of the coefficient: $k \in \{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 60, 100, 170, 300\}$. The used coefficients are $a = 3$, $b = 2a$, $A = -0.9$ and $\alpha = 0.6$. The aggregated errors are displayed in Table 6 and Figure 11.

Table 6. The aggregated errors (*AgErr*) for the values of *k* for isomerization in Case Study 8.

Numerical Method	0.01	0.03	0.1	0,3	1	3	10	30	60	100	170	300
ADE*u*	−18.3	−18.5	−19.1	−20.34	−22.07	−22.84	−22.48	−18.89	−5.39	21.37	79.39	198.79
ADE*w*	−42.81	−38.67	−34.48	−31.54	−29.95	−29.51	−28.8	−24.59	−10.66	16.07	74.48	193.67
DF*u*	−19.43	−19.80	−20.07	−19.77	−18.52	−16.88	−11.14	18.85	57.48	101.26	165.45	262.00
DF*w*	−38.57	−34.60	−31.13	−29.13	−25.89	−25.22	−18.42	12.13	51.26	95.02	159.50	256.10
OOEH*u*	−16.86	−17.07	−17.77	−19.48	−23.34	−24.15	−22.45	−12.48	3.91	23.42	63.60	142.68
OOEH*w*	−41.34	−37.16	−32.86	−29.67	−27.69	−27.19	−25.72	−16.33	−0.23	19.04	59.13	137.85
CpC*u*	−10.17	−10.88	−12.10	−8.71	−6.85	−6.71	−7.56	−8.790	−9.56	−7.14	6.08	48.02
CpC*w*	−27.72	−23.57	−19.35	−16.29	−14.62	−14.63	−15.94	−17.71	−18.17	−15.27	−1.55	40.72
CCL*u*	−11.09	−11.82	−13.69	−9.91	−7.99	−8.02	−9.14	−10.56	−11.43	−3.54	16.33	65.85
CCL*w*	−28.57	−24.42	−20.20	−17.17	−15.63	−15.94	−17.77	−19.58	−20.00	−11.37	8.91	58.87
CLQ2*u*	−11.28	−12.02	−13.93	−10.12	−8.20	−8.24	−9.37	−10.79	−11.69	−12.30	−12.20	3.89
CLQ2*w*	−28.75	−24.60	−20.38	−17.36	−15.83	−16.17	−18.03	−19.80	−20.28	−20.57	−19.75	−2.89
PI*u*	−10.59	−11.32	−13.07	−9.38	−7.45	−7.42	−8.51	−9.97	−10.91	−11.57	−11.88	−1.10
PI*w*	−28.11	−23.96	−19.70	−16.69	−15.10	−15.31	−17.04	−19.04	−19.61	−19.90	−19.58	−7.98
LH*u*	−22.80	−23.13	−23.41	−22.32	−20.46	−18.91	−19.46	−19.56	−19.67	−19.20	−16.47	−6.27
LH*w*	−41.86	−37.82	−33.82	−30.48	−27.98	−27.00	−27.90	−27.81	−26.83	−25.96	−22.92	−12.67

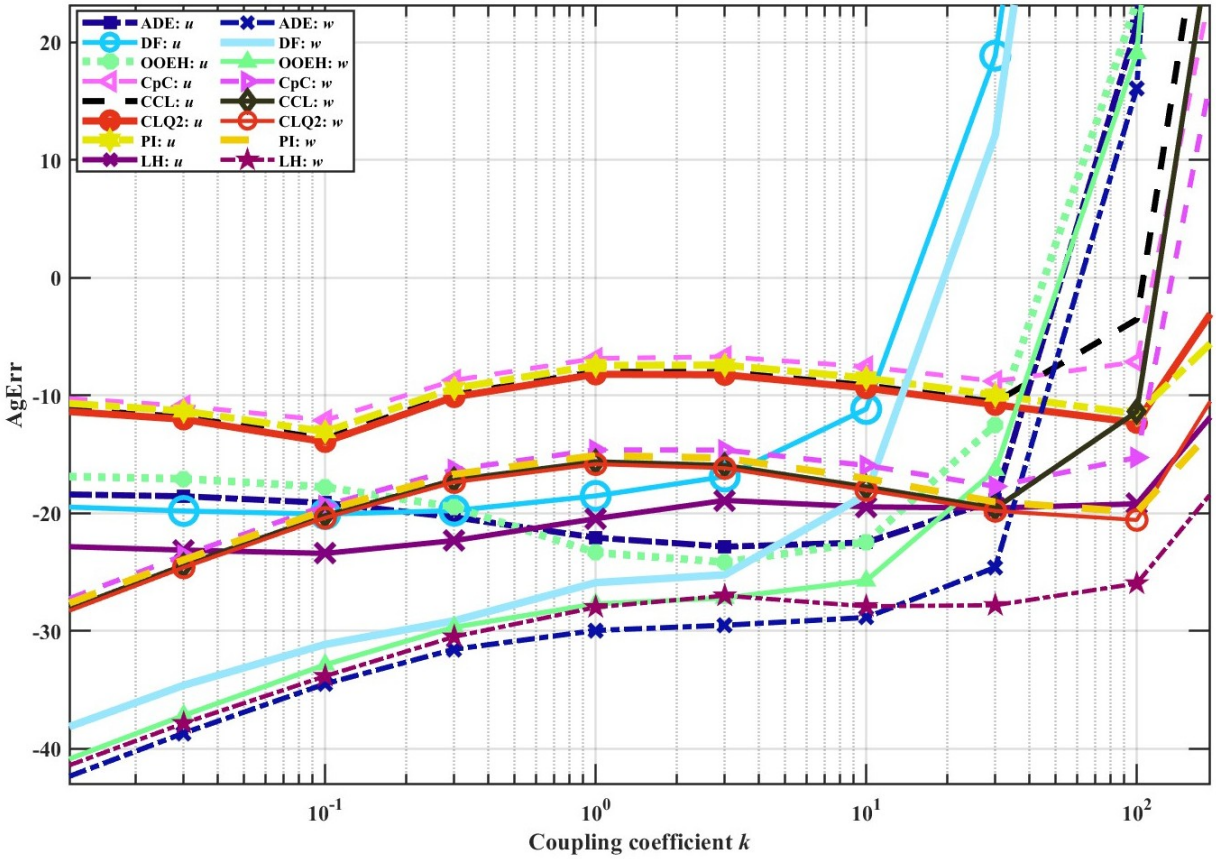


Figure 11. Aggregated errors as a function of the coupling coefficient k in Case Study 8.

6. Conclusions and Summary

We studied coupled sets of reaction–diffusion equations in one space dimension with two different, time-dependent reaction terms. We recited a set of recently published analytical solutions, which are highly nontrivial since they contain the Kummer functions. We showed that there is a mistake in these solutions, which means that they are valid only in the simple case, when $\alpha = \gamma$, as they contain one less free parameter. We modified the time-dependent terms in the original PDEs, and now they contain a free parameter in the exponent apart from the coupling coefficient. Therefore, the solutions are now valid for a more general set of reaction–diffusion equations and in the general case when $\alpha \neq \gamma$ as well.

We then reproduced the analytical solutions by 10 explicit numerical methods including the most standard FTCS and Adams–Bashforth methods. The rest of the methods are unconditionally stable for the diffusion equation without the reaction term, and some of them have been constructed by our research group in the past few years. First, we verified the algorithms by showing that the errors properly decrease with the time step size for both types of reaction term. Then, we performed sweeps for some key parameters to study how the methods behave as a function of these parameters.

The first conclusion is that the methods are not unconditionally stable for the studied diffusion–reaction equation system. If the coefficients of the reaction term are large, the errors start to grow as time elapses. However, the methods that are stable for the diffusion equation are still more stable for the case when a reaction term is present than the traditional FTCS and AB2 schemes. The least stable among these methods is the OOEH method and the most stable is the CpC method, which has the best qualitative properties. Nevertheless, the error is usually the smallest for the LH method, but the DF, OOEH, and ADE schemes

also perform well if they are not close to their stability limits. The OOEH scheme is the best if the simulated time interval is very short, but as time elapses, its error drastically increases.

After this study's publication, we plan to analytically investigate the properties of the methods when applied to this system of equations. We also wish to repeat some numerical experiments to the other terms in the analytical solutions, i.e., when the coefficients c_2, c_3, and c_4 are nonzero.

Author Contributions: Conceptualization, resources, methodology, and supervision: E.K. investigation and visualization: H.K. and A.Z.; software, writing—original draft preparation, review and editing: H.K., A.Z. and E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Two codes are available at the following link https://github.com/Drendre/Coupled-diffusion-reaction-PDEs-codes, accessed on 30 May 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jacobs, M.H. *Diffusion Processes*; Springer: Berlin/Heidelberg, Germany, 1935; ISBN 978-3-642-86414-8.
2. Lienhard, J.H., IV; Lienhard, J.H., V. *A Heat Transfer Textbook*, 4th ed.; Phlogiston Press: Cambridge, MA, USA, 2017; ISBN 978-0-9713835-2-4.
3. Simpson, M.J.; Sharp, J.A.; Morrow, L.C.; Baker, R.E. Exact Solutions of Coupled Multispecies Linear Reaction–Diffusion Equations on a Uniformly Growing Domain. *PLoS ONE* **2015**, *10*, e0138894. [CrossRef]
4. Escorcia, J.M.; Suazo, E. On explicit solutions for coupled reaction-diffusion and Burgers-type equations with variable coefficients through a Riccati system. *Discret. Contin. Dyn. Syst. S* **2025**. [CrossRef]
5. Covei, D.-P. A nonlinear diffusion equation of the Gurtin-MacCamy type: Existence, uniqueness, and numerical simulations. *arXiv* **2025**. [CrossRef]
6. Mátyás, L.; Barna, I.F. General Self-Similar Solutions of Diffusion Equation and Related Constructions. *Rom. J. Phys.* **2022**, *67*, 101.
7. Barna, I.F.; Mátyás, L. Diffusion Cascades and Mutually Coupled Diffusion Processes. *Mathematics* **2024**, *12*, 3298. [CrossRef]
8. Jejeniwa, O.A.; Gidey, H.H.; Appadu, A.R. Numerical Modeling of Pollutant Transport: Results and Optimal Parameters. *Symmetry* **2022**, *14*, 2616. [CrossRef]
9. Köroğlu, C.; Aydin, A. Exact and nonstandard finite difference schemes for the Burgers equation B(2,2). *Turk. J. Math.* **2021**, *45*, 647–660. [CrossRef]
10. Agbavon, K.M.; Appadu, A.R. Construction and analysis of some nonstandard finite difference methods for the FitzHugh–Nagumo equation. *Numer. Methods Partial. Differ. Equ.* **2020**, *36*, 1145–1169. [CrossRef]
11. Agbavon, K.M.; Appadu, A.R.; Khumalo, M. On the numerical solution of Fisher's equation with coefficient of diffusion term much smaller than coefficient of reaction term. *Adv. Differ. Equ.* **2019**, *2019*, 146. [CrossRef]
12. Appadu, A.R.; Inan, B.; Tijani, Y.O. Comparative study of some numerical methods for the Burgers-Huxley equation. *Symmetry* **2019**, *11*, 1333. [CrossRef]
13. Marc, E. Songolo a Positivity-Preserving Nonstandard Finite Difference Scheme for Parabolic System with Cross-Diffusion Equations and Nonlocal Initial Conditions. *Am. Sci. Res. J. Eng. Technol. Sci.* **2016**, *18*, 252–258.
14. Chapwanya, M.; Lubuma, J.M.S.; Mickens, R.E. Positivity-Preserving nonstandard finite difference schemes for Cross-Diffusion equations in biosciences. *Comput. Math. Appl.* **2014**, *68*, 1071–1082. [CrossRef]
15. Bansal, S.; Natesan, S. A novel Higher-Order efficient computational method for pricing European and Asian options. *Numer. Algorithms* **2024**. [CrossRef]
16. Rieth, Á.; Kovács, R.; Fülöp, T. Implicit numerical schemes for generalized heat conduction equations. *Int. J. Heat. Mass. Transf.* **2018**, *126*, 1177–1182. [CrossRef]
17. Britz, D.; Baronas, R.; Gaidamauskaitė, E.; Ivanauskas, F. Further comparisons of finite difference schemes for computational modelling of biosensors. *Nonlinear Anal. Model. Control* **2009**, *14*, 419–433. [CrossRef]
18. Mbroh, N.A.; Munyakazi, J.B. A robust numerical scheme for singularly perturbed parabolic Reaction-Diffusion problems via the method of lines. *Int. J. Comput. Math.* **2021**, *99*, 1139–1158. [CrossRef]
19. Manaa, S.; Sabawi, M. Numerical Solution and Stability Analysis of Huxley Equation. *AL-Rafidain J. Comput. Sci. Math.* **2005**, *2*, 85–97. [CrossRef]

20. Pargaei, M.; Kumar, B.V.R. A 3D Haar wavelet method for a coupled degenerate system of parabolic equations with nonlinear source coupled with Non-Linear ODEs. *Appl. Numer. Math.* **2023**, *185*, 141–164. [CrossRef]
21. Anjuman; Leung, A.Y.T.; Das, S. Two-Dimensional Time-Fractional Nonlinear Drift Reaction–Diffusion Equation Arising in Electrical Field. *Fractal Fract.* **2024**, *8*, 456. [CrossRef]
22. Rufai, M.A.; Kosti, A.A.; Anastassi, Z.A.; Carpentieri, B. A New Two-Step Hybrid Block Method for the FitzHugh–Nagumo Model Equation. *Mathematics* **2024**, *12*, 51. [CrossRef]
23. Ndou, N.; Dlamini, P.; Jacobs, B.A. Enhanced Unconditionally Positive Finite Difference Method for Advection–Diffusion–Reaction Equations. *Mathematics* **2022**, *10*, 2639. [CrossRef]
24. Heidari, M.; Ghovatmand, M.; Skandari, M.H.N.; Baleanu, D. Numerical Solution of Reaction–Diffusion Equations with Convergence Analysis. *J. Nonlinear Math. Phys.* **2023**, *30*, 384–399. [CrossRef]
25. Kolev, M.K.; Koleva, M.N.; Vulkov, L.G. An Unconditional Positivity-Preserving Difference Scheme for Models of Cancer Migration and Invasion. *Mathematics* **2022**, *10*, 131. [CrossRef]
26. Ascher, U.M.; Ruuth, S.J.; Wetton, B.T.R. Implicit-Explicit Methods for Time-Dependent Partial Differential Equations. *SIAM J. Numer. Anal.* **1995**, *32*, 797–823. [CrossRef]
27. Calvo, M.P.; de Frutos, J.; Novo, J. Linearly implicit Runge–Kutta methods for Advection–Reaction–Diffusion equations. *Appl. Numer. Math.* **2001**, *37*, 535–549. [CrossRef]
28. Settanni, G.; Sgura, I. Devising efficient numerical methods for oscillating patterns in Reaction-Diffusion systems. *J. Comput. Appl. Math.* **2016**, *292*, 674–693. [CrossRef]
29. Yadav, V.S.; Singh, A.; Maurya, V.; Rajpoot, M.K. New RK type Time-Integration methods for stiff Convection–Diffusion–Reaction systems. *Comput. Fluids* **2023**, *257*, 105865. [CrossRef]
30. Beuken, L.; Cheffert, O.; Tutueva, A.; Butusov, D.; Legat, V. Numerical Stability and Performance of Semi-Explicit and Semi-Implicit Predictor–Corrector Methods. *Mathematics* **2022**, *10*, 2015. [CrossRef]
31. Fedoseev, P.; Pesterev, D.; Karimov, A.; Butusov, D. New Step Size Control Algorithm for Semi-Implicit Composition ODE Solvers. *Algorithms* **2022**, *15*, 275. [CrossRef]
32. Ji, Y.; Xing, Y. Highly Accurate and Efficient Time Integration Methods with Unconditional Stability and Flexible Numerical Dissipation. *Mathematics* **2023**, *11*, 593. [CrossRef]
33. Chou, C.S.; Zhang, Y.T.; Zhao, R.; Nie, Q. Numerical methods for stiff Reaction-Diffusion systems. *Discret. Contin. Dyn. Syst.-Ser. B* **2007**, *7*, 515–525. [CrossRef]
34. Jiang, T.; Zhang, Y.T. Krylov implicit integration factor WENO methods for semilinear and fully nonlinear Advection-Diffusion-Reaction equations. *J. Comput. Phys.* **2013**, *253*, 368–388. [CrossRef]
35. Essongue, S.; Ledoux, Y.; Ballu, A. 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.* **2022**, *211*, 103825. [CrossRef]
36. Chen-Charpentier, B.M.; Kojouharov, H.V. An unconditionally positivity preserving scheme for Advection-Diffusion reaction equations. *Math. Comput. Model.* **2013**, *57*, 2177–2185. [CrossRef]
37. Appadu, A.R. Performance of UPFD scheme under some different regimes of advection, diffusion and reaction. *Int. J. Numer. Methods Heat. Fluid. Flow.* **2017**, *27*, 1412–1429. [CrossRef]
38. Savović, S.; Drljača, B.; Djordjevich, A. A comparative study of two different finite difference methods for solving Advection–Diffusion reaction equation for modeling exponential traveling wave in heat and mass transfer processes. *Ric. Mat.* **2022**, *71*, 245–252. [CrossRef]
39. Al-Bayati, A.; Manaa, S.; Al-Rozbayani, A. Comparison of Finite Difference Solution Methods for Reaction Diffusion System in Two Dimensions. *AL-Rafidain J. Comput. Sci. Math.* **2011**, *8*, 21–36. [CrossRef]
40. Pourghanbar, S.; Manafian, J.; Ranjbar, M.; Aliyeva, A.; Gasimov, Y.S. An efficient alternating direction explicit method for solving a nonlinear partial differential equation. *Math. Probl. Eng.* **2020**, *2020*, 47416. [CrossRef]
41. Nagy, Á.; Majár, J.; Kovács, E. Consistency and Convergence Properties of 20 Recent and Old Numerical Schemes for the Diffusion Equation. *Algorithms* **2022**, *15*, 425. [CrossRef]
42. Nagy, Á.; Omle, I.; Kareem, H.; Kovács, E.; Barna, I.F.; Bognar, G. Stable, Explicit, Leapfrog-Hopscotch Algorithms for the Diffusion Equation. *Computation* **2021**, *9*, 92. [CrossRef]
43. Saleh, M.; Kovács, E.; Barna, I.F.; Mátyás, L. New Analytical Results and Comparison of 14 Numerical Schemes for the Diffusion Equation with Space-Dependent Diffusion Coefficient. *Mathematics* **2022**, *10*, 2813. [CrossRef]
44. Saleh, M.; Kovács, E.; Barna, I.F. Analytical and Numerical Results for the Transient Diffusion Equation with Diffusion Coefficient Depending on Both Space and Time. *Algorithms* **2023**, *16*, 184. [CrossRef]
45. Askar, A.H.; Nagy, Á.; Barna, I.F.; Kovács, E. Analytical and Numerical Results for the Diffusion-Reaction Equation When the Reaction Coefficient Depends on Simultaneously the Space and Time Coordinates. *Compututation* **2023**, *11*, 127. [CrossRef]
46. Barakat, H.Z.; Clark, J.A. On the solution of the diffusion equations by numerical methods. *J. Heat. Transf.* **1966**, *88*, 421–427. [CrossRef]

47. Liu, H.; Leung, S. An Alternating Direction Explicit Method for Time Evolution Equations with Applications to Fractional Differential Equations. *Methods Appl. Anal.* **2020**, *26*, 249–268. [CrossRef]
48. Hirsch, C. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*; John Wiley & Sons: Hoboken, NJ, USA, 1988.
49. Gourlay, A.R.; McGuire, G.R. General Hopscotch Algorithm for the Numerical Solution of Partial Differential Equations. *IMA J. Appl. Math.* **1971**, *7*, 216–227. [CrossRef]
50. Kovács, E.; Nagy, Á.; Saleh, M. A set of new stable, explicit, second order schemes for the Non-Stationary heat conduction equation. *Mathematics* **2021**, *9*, 2284. [CrossRef]
51. Kovács, E.; Nagy, Á. A new stable, explicit, and generic Third-Order method for simulating conductive heat transfer. *Numer. Methods Partial. Differ. Equ.* **2023**, *39*, 1504–1528. [CrossRef]
52. Kovács, E.; Majár, J.; Saleh, M. Unconditionally Positive, Explicit, Fourth Order Method for the Diffusion- and Nagumo-Type Diffusion–Reaction Equations. *J. Sci. Comput.* **2024**, *98*, 39. [CrossRef]
53. Jalghaf, H.K.; Kovács, E.; Majár, J.; Nagy, Á.; Askar, A.H. Explicit stable finite difference methods for Diffusion-Reaction type equations. *Mathematics* **2021**, *9*, 3308. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.