mathematics

MDPI

Article

Exploring the Performance of Some Efficient Explicit Numerical Methods with Good Stability Properties for Huxley's Equation

Husniddin Khayrullaev, Issa Omle and Endre Kovács *

Institute of Physics and Electrical Engineering, University of Miskolc, 3515 Miskolc, Hungary; hxayrullayev@mail.ru (H.K.); issa.omle@uni-miskolc.hu (I.O.)
* Correspondence: kendre01@gmail.com or endre.kovacs@uni-miskolc.hu

Abstract: Four explicit numerical schemes are collected, which are stable and efficient for the diffusion equation. Using these diffusion solvers, several new methods are constructed for the nonlinear Huxley's equation. Then, based on many successive numerical case studies in one and two space dimensions, the least performing methods are gradually dropped out to keep only the best ones. During the tests, not only one but all the relevant time step sizes are considered, and for them, running-time measurements are performed. A major aspect is computational efficiency, which means that an acceptable solution is produced in the shortest possible time. Parameter sweeps are executed for the coefficient of the nonlinear term, the stiffness ratio, and the length of the examined time interval as well. We obtained that usually, the leapfrog–hopscotch method with Strang-type operator-splitting is the most efficient and reliable, but the method based on the Dufort–Frankel scheme can also be very efficient.

Keywords: nonlinear PDEs; diffusion–reaction equations; Huxley's equation; explicit numerical methods; stiff equations

MSC: 35K20; 35K57; 65M06l

check for updates

Academic Editor: Huaizhong Zhao

Received: 10 November 2024
Revised: 1 January 2025
Accepted: 7 January 2025
Published: 9 January 2025

Citation: Khayrullaev, H.; Omle, I.; Kovács, E. Exploring the Performance of Some Efficient Explicit Numerical Methods with Good Stability Properties for Huxley's Equation. *Mathematics* **2025**, *13*, 207. https://doi.org/10.3390/math13020207

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

Nonlinear reaction–diffusion partial differential equations (PDEs) have an important role in modeling different phenomena in many branches of science. We examine Huxley's equation [1] in this paper assuming the following format:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta u^2 (1 - u). \tag{1}$$

where α and β are real values that are not negative. We always assume that the initial u function and the boundary conditions are given, and they will be explained later in this work. If $\beta = 0$, then one has the linear diffusion equation, which is more often called the heat equation and used to model Fourier-type heat conduction as well, mostly in solid media. One can observe that Equation (1) resembles Fisher's equation:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u + \beta u (1 - u), \tag{2}$$

but Equation (1) is more nonlinear than Equation (2) since the nonlinear component is simply quadratic here. On the other hand, a somewhat expanded version of (1), commonly also referred to as the Huxley equation [2], is

$$\frac{\partial u}{\partial t} = \alpha \, \nabla^2 u + \beta u(1-u)(u-\gamma), \tag{3}$$

where γ is in the unit interval. In this work, we always take $\gamma = 0$ to recover Equation (1). Equation (2) is also commonly referred to as the FitzHugh–Nagumo equation [3], but this phrase is also used to refer to more generic equations. For instance, it was shown [1] that equations like (1) or (3) that contain a cubic response term may be better to use to represent how a new, beneficial recessive gene spreads in a population that reproduces sexually than Equation (2) with a quadratic reaction term. In biological tissues, where the equation explains the passage of electrical impulses along nerve fibers, they are also appropriate [4] for simulating the propagation of neural pulses.

Several clever methods are proposed to solve these and similar PDEs, such as the shifted Chebyshev spectral collocation method [5], semi-analytical methods [6,7], the homotopy perturbation method [8], different versions of physics-informed neural networks (PINNs) [9,10], the weighted average method [11], and exact as well as nonstandard finite difference schemes (NSFD) [12]. Most of these approaches are constructed and tested for specific circumstances where the media are homogeneous, and/or the number of space dimensions is small, and the spatial mesh is uniform and consists of not very many nodes. There are some counterexamples, e.g., when a partially implicit scheme [13], coarse-grain models [14], or the asymptotic theory [15] were used in heterogeneous media, but these can be considered as exceptions. Nevertheless, under the mentioned specific circumstances, the weak points of the proposed algorithms do not always manifest. Moreover, some of these algorithms were developed aiming, most importantly, high accuracy. However, in most applications, there are non-negligible inaccuracies in the input data, and the models themselves contain simplifications that can cause even a few percent deviation from the measurements. Due to these limitations, the main goal should not be to press down the relative errors below, e.g., 10^{-8}, but to develop methods which are, on the one hand, fast in higher dimensions as well, and, on the other hand, robust, i.e., reliable for all possible parameter combinations.

Most explicit numerical algorithms have a severely constrained stability region [16]. If the time step size is greater than a certain threshold, often called the Courant–Friedrichs–Lewy (CFL) limit, the solution is predicted to explode even for the linear diffusion or heat equation. The heterogeneity in the physical media implies a space-dependent α coefficient [17], which results in a high stiffness ratio and a relatively small CFL limit for the explicit approaches. This is especially well-known when one uses the method of lines, where the PDE is first spatially discretized, and an ODE solver, such as a Runge–Kutta (RK) scheme, solves the resulting system of ODEs. This is true for low-order explicit schemes, such as the FTCS (forward-time central space) or semi-explicit [18] methods, as well as higher order techniques, including the standard fourth-order RK method and Strong-Stability-Preserving Runge–Kutta Methods [19,20].

Although step size restrictions may occasionally be necessary, implicit approaches, such as the technique called shifted airfoil collocation, proposed by Anjuman et al. [21] for nonlinear drift-diffusion–reaction equations, offer superior stability qualities [22]. The most restricting problem is that implicit methods need the solution of a system of nonlinear algebraic equations, which can take a lot of time and memory if the system is big, which is typical in two or three space dimensions. Despite this, implicit techniques are frequently used to solve these and similar equations [23], for example, backward implicit and Crank–

Nicolson (CrN) schemes with and without linearization [24]. Rufai et al. proposed a novel hybrid block method for the FitzHugh–Nagumo equation with time-dependent coefficients [25]. Their method is also implicit, and the Newton method is used to handle the nonlinearity. Furthermore, in the case of high stiffness, the diffusion term is occasionally even considered exactly [26]. Manaa and Sabawi [27] solved Equation (3) using the CrN and explicit (Euler) methods. They found that although the explicit technique requires shorter execution times, the CrN method is more accurate and does not have significant stability problems, as was to be predicted. A finite-difference numerical scheme with good qualitative properties was proposed by Macías-Díaz for the Burgers–Huxley equation [28]. However, his algorithm only has favorable features for short time step sizes below the CFL number since he employed the explicit Euler time discretization for the diffusion factor. This also applies to most NSFD algorithms, which were applied, for example, for the Fisher and Nagumo equations [3,29,30] and for cross-diffusion equations [31,32]. Our research group constructs and tests explicit numerical methods possessing excellent stability properties. A couple of these methods have been known for a long time, but they have been used for nonlinear PDEs containing a diffusion term only a few times, much less than they deserve. For example, the Dufort–Frankel scheme was successfully applied for the two-dimensional Sine–Gordon equation [33]. The nonlinear term is treated by a predictor–corrector-type operator-splitting. Gasparin et al. also applied the Dufort–Frankel scheme to nonlinear moisture transfer in porous materials [34]. The odd–even hopscotch method was utilized to solve the Frank–Kamenetskii equation by Harley [35]. These algorithms always showed a good performance; nevertheless, their investigations were usually not continued. Furthermore, in most scientific works, where the performance of numerical methods is tested, the parameters, such as the constants α, β and the space and time step size Δx are fixed, and only a small number of numerical methods are compared. Our goal in this work is to extensively investigate the performance of some methods for Huxley's equation under a large number of parameter combinations, for example, for less-stiff and very stiff systems. Hence, in Section 2, the spatial discretization of Equation (1) is performed not only for the most basic one-dimensional system with constant α but also for the most general case. In Section 3, we review four numerical methods that were proven to be very efficient when applied to the diffusion equation. All of these diffusion solvers are explicit and unconditionally stable in the linear case. In Section 4, we construct numerous ways to handle the nonlinear reaction term, as well as briefly present methods that will be used for comparison purposes. With these, we gain more than 40 different numerical methods to test for Huxley's equation. In Section 5, we perform the first round of the numerical experiments, which constitutes five concrete case studies. Based on the comparison of the errors for several time step sizes, we exclude most of the methods from further investigation and keep only 13. In the second round of numerical experiments in Section 6, three case studies are performed in large two-dimensional systems with running time measurement to choose the top seven most efficient methods. Only these seven methods are used in Section 7 to examine how their performance depends on some parameters, such as the strength of the nonlinear reaction term and the CFL limit of the system. Section 8 provides a summary of the findings and suggests methods that should be used under specific circumstances.

We do not know any other work in which so many methods are tested for such a large number of parameter combinations for any nonlinear PDE. The best few among these tested methods are very efficient and reliable, and they have never been applied to Huxley's equation before. These are the main novelties in our work.

2. The Studied Equation and the Spatial Discretization

We start by discretizing the pure diffusion or heat equation because the reaction term is entirely local. The first step is the simplest and most standard discretization by equidistant nodes $x_j = x_1 + (j-1)\Delta x\,,\ j = 1, \ldots, N\,,\ (N-1)\Delta x = L$ of the space interval $x \in [x_1,\ x_N = x_1 + L]$. The second spatial derivatives are subjected to the well-known central difference approximation, leading to the well-known ODE system

$$\frac{du_i}{dt} = \frac{\alpha}{\Delta x^2}(u_{i-1} - 2u_i + u_{i+1}) \tag{4}$$

for the nodes with index $i = 1, \ldots, N-1$. The M matrix, which is tridiagonal in the one-dimensional example, contains only a few of nonzero elements as follows:

$$m_{\mathrm{ii}} = -\frac{2\alpha}{\Delta x^2}\ (1 < i < N),\ \ m_{\mathrm{i,i+1}} = \frac{\alpha}{\Delta x^2}\ (1 \le i < N),\ m_{\mathrm{i,i-1}} = \frac{\alpha}{\Delta x^2}\ (1 < i \le N) \tag{5}$$

The boundary conditions (BCs) specify the first and last row. For example, the first and last nodes' time development will be provided directly in the case of Dirichlet BCs; thus, $m_{1,1} = 1,\ m_{N,N} = 1,\ m_{1,i} = 0\ (i > 1),\ m_{N,i} = 0\ (i < N)$. The matrix form of the ODE system (4) can be reformulated as follows:

$$\frac{dU}{dt} = MU\,, \tag{6}$$

where $U = (u_1, \ldots, u_N)^{\mathrm{T}}$. Our goal includes performing experiments in general cases where the geometrical and material characteristics of the simulated system are different in different spatial regions. It is necessary to adapt space discretization and subsequent techniques to reflect this generality level. If the media property α is dependent on the space variable while we remain in 1D, the PDE

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

can be used. We can discretize the parameter α and, simultaneously, $\partial u/\partial x$ to obtain

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

The average diffusivity between cell i and its (left) neighbor may be denoted by $\alpha_{i,i-1}$. In practice, it can be approximated by its value between these two nodes. With this, we have

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

This is a generalized version of Equation (4). We can build a highly adaptable resistance–capacitance model by taking cells into account rather than nodes. The volume of these cells can be expressed as $V_i = \Delta x$ in the case of a one-dimensional equidistant mesh. A cell's capacity is equal to its volume in the case of the most basic diffusion: $C_i = V_i = \Delta x$. Between two cells, the resistance may be calculated simply as $R_{i,\,i+1} = R_{i+1,i} = \frac{\Delta x}{\alpha_{i,\,i+1}},\ i = 1, \ldots, N-1$. A generalized ODE system for the time derivative of each cell variable u in one space dimension can be obtained using these:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i,i-1}C_i} + \frac{u_{i+1} - u_i}{R_{i,i+1}C_i}. \tag{7}$$

It is easy to see that this RC model with Equation (7) is indeed a generalization of the original model based on equidistant nodes, in which Equation (4) expressed the time development of the node variables. Hence, if a numerical method works for the general RC model, it will work for the special case as well. Furthermore, Equation (7) can be expressed in the same matrix form as Equation (6), but with entries that depend on the resistances and capacities.

We will also work in two space dimensions, where the numbering of the cells begins along the x-directional (horizontal) rows starting from 1 to N_x and then along the x-axis again from $N_x + 1$ to $2N_x$, and so on. Now, an ODE system (7) can be straightforwardly generalized to two space dimensions as follows:

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{R_{i,i-1}C_i} + \frac{u_{i+1} - u_i}{R_{i,i+1}C_i} + \frac{u_{i-Nx} - u_i}{R_{i,i-Nx}C_i} + \frac{u_{i+Nx} - u_i}{R_{i,i+Nx}C_i} = \sum_{j \neq i} \frac{u_j - u_i}{R_{i,j}C_i} \,. \tag{8}$$

The last equation is valid since the resistances are infinity for non-adjacent cells. Additional information about the discretization using this resistance–capacitance model can be found, for example, in [36].

In 2D numerical experiments, we usually take zero-Neumann BCs into account. The RC model will implement these in a straightforward manner: the matrix components that are responsible for conduction across the borders vanish when those resistances are regarded as infinite. In this instance, the conserved quantity is represented by the system matrix M's zero eigenvalue. All other eigenvalues M are negative due to the Second Law of Thermodynamics. The eigenvalues with the (nonzero) least and highest absolute values, respectively, are indicated by λ_{MIN} and λ_{MAX}. The standard definition of the problem's stiffness ratio is $SR = \lambda_{MAX}/\lambda_{MIN}$. In addition to it, the CFL limit $CFL = |2/\lambda_{MAX}|$ provides an accurate estimate of the maximum time step size that can be used for the semi-discretized linear diffusion problem (4), (7), or (8) using the explicit Euler (first-order explicit RK) scheme. Similar threshold time step sizes can be obtained for higher-order explicit RK schemes, such as for the fourth-order RK schemes: $CFL_{RK4} \approx 1.392 \cdot CFL$. This CFL threshold and the stiffness ratio will be utilized in this paper to describe how difficult it is to solve the problem. We must emphasize again that the unconditionally stable schemes employed in this work have no time step size restrictions for the linear diffusion problem due to stability considerations.

The simplest uniform discretization

$$t^n = t^{\text{ini}} + nh \,,\; t \in \left[t^{\text{ini}}, t^{\text{fin}}\right] \,,\; n = 1, \ldots, T,\; hT = t^{\text{fin}} - t^{\text{ini}}.$$

will always be used for the time variable.

3. The Examined Diffusion Solver Methods

We provide essential details regarding the algorithms that solve an equation without the nonlinear term. Since we mostly employ the more general forms in this study, only they are required.

For the 1D equidistant mesh, such as in Equation (1), the standard mesh ratio of $r = \frac{\alpha \Delta t}{\Delta x^2}$ is used in many publications and textbooks. In the case of the general mesh, the following notations are going to be used:

$$r_i = h \sum_{j \neq i} \frac{1}{C_i R_{ij}} \;\text{ and }\; A_i^n = h \sum_{j \neq i} \frac{u_j^n}{C_i R_{ij}}. \tag{9}$$

While the second quantity reflects the status and influence of the neighbors of cell i, the first quantity is the generalization of the mesh ratio.

1. The first method is called the CCL algorithm, which is the abbreviation for Constant–Constant–Linear neighbor. It is a one-step but three-stage method recently published by our group [37], where the constant-neighbor (CNe) formula is applied in the first and second stages. The first stage is a predictor with a $h/3$-sized time step:

$$u_i^{\mathrm{C}} = u_i^n e^{-\,r_i/3} + \frac{A_i^n}{r_i^n}\left(1 - e^{-\,r_i/3}\right).$$

Then, the first corrector stage comes

$$u_i^{\mathrm{CC}} = u_i^n e^{-\,2r_i/3} + \frac{A_i^{\mathrm{C}}}{r_i^n}\left(1 - e^{-\,2r_i/3}\right).$$

Finally, a full time step is taken with the linear-neighbor (LNe) formula in the third stage:

$$u_i^{n+1} = u_i^n e^{-\,r_i^n} + \left(A_i^n - \frac{A_i^{\mathrm{CC}} - A_i^n}{2r_i^n/3}\right)\frac{1 - e^{-\,r_i^n}}{r_i^n} + \frac{A_i^{\mathrm{CC}} - A_i^n}{2r_i^n/3}.$$

During the calculations, the A_i quantities must be refreshed after each stage: $A_i^{\mathrm{C}} = \frac{h}{C_i}\left(\frac{u_{i-1}^{\mathrm{C}}}{R_{i,\,i-1}^n} + \frac{u_{i+1}^{\mathrm{C}}}{R_{i,\,i+1}^n}\right)$ and $A_i^{\mathrm{CC}} = \frac{h}{C_i}\left(\frac{u_{i-1}^{\mathrm{CC}}}{R_{i,\,i-1}^n} + \frac{u_{i+1}^{\mathrm{CC}}}{R_{i,\,i+1}^n}\right)$, respectively. The CCL method has third-order temporal accuracy, and, similarly to the following three methods, it is proven to be unconditionally stable for the heat equation.

2. The Dufort–Frankel (DF) method is a classic example of the explicit and stable methods [38] (p. 313), which is second-order in time. We adapted it to the general case, where the following formula must be used:

$$u_i^{\mathrm{n+1}} = \frac{(1 - r_i)u_i^{n-1} + 2A_i}{1 + r_i}. \tag{10}$$

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 so-called UPFD (unconditional finite difference) formula for this purpose:

$$u_i^1 = \frac{u_i^0 + A_i}{1 + 2r_i}. \tag{11}$$

3. The original odd–even hopscotch (OOEH) method was discovered more than 50 years ago [39]. It requires a special spatial and temporal structure. In essence, the mesh must be bipartite, i.e., it is divided into two parts, the so-called odd and even nodes (or cells), where the closest neighbor of the even cells is odd, and vice versa. First, the FTCS formula is applied to the odd cells, followed by the BTCS (Backward-time Central-space) formula, which is based on implicit Euler time discretization. After every time step, the odd and even labels are switched, as it is illustrated in Figure 1. The equations used are as follows:

$$\text{First stage}: \; u_i^{n+1} = (1 - r_i)u_i^n + A_i. \tag{12}$$

.

$$\text{Second stage}: u_i^{n+1} = \frac{u_i^n + A_i^{\mathrm{new}}}{1 + r_i}, \tag{13}$$

where the new concentration data are used to calculate A_i^{new} in the same manner as A_i in Equation (12), essentially making the implicit formula explicit.

4. The recently invented leapfrog–hopscotch (LH) approach [40] also requires the odd–even space structure. Moreover, it has a structure made up of many full time steps and two half time steps. 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 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, as shown in Figure 1.

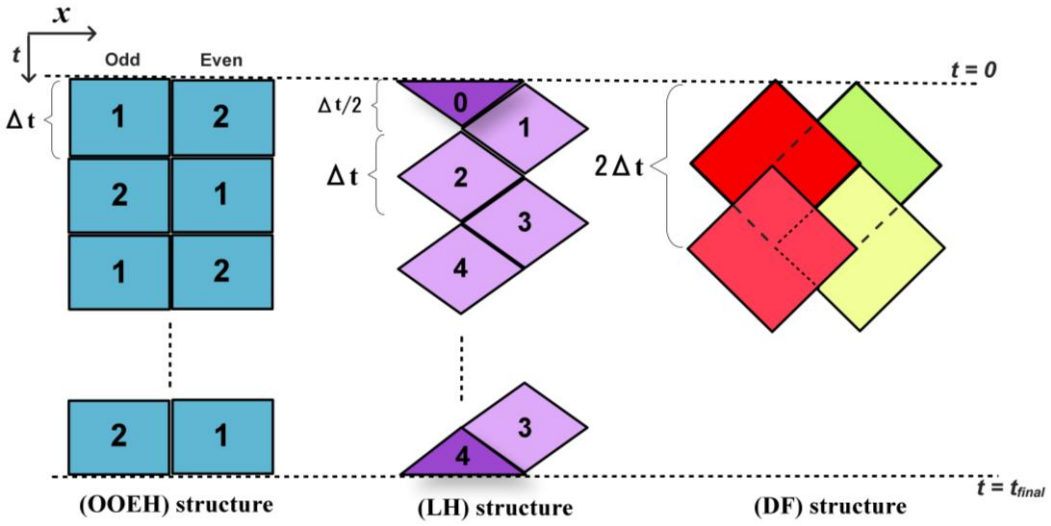


Figure 1. The structures of original odd–even hopscotch (OOEH), leapfrog–hopscotch (LH), and Dufort–Frankel (DF) methods.

Since the first stage's time step is halved, the following general formula is applied:

$$u_i^{\frac{1}{2}} = \frac{u_i^0 + A_i^0/2}{1 + r_i/2}, \tag{14}$$

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

$$u_i^1 = \frac{(1 - r_i/2)u_i^0 + A_i^{\frac{1}{2}}}{1 + r_i/2}, \tag{15}$$

Full time steps, in the same manner, are then taken for the odd and even nodes in turn, where always the most recently obtained values of the neighbors are used to calculate the new A_i values. Lastly, the computations for the odd nodes must be closed using a half-length time step.

$$u_i^T = \frac{(1 - r_i/4)u_i^{T-1/2} + A_i^T/2}{1 + r_i/4}, \tag{16}$$

According to most of the numerical experiments in our previous works, the LH method is the most efficient among the explicit methods that are unconditionally stable to the linear diffusion equation.

The favorable properties of these algorithms are formulated in the following theorems, which were proved in the original papers.

Theorem 1. *The CCL, OOEH, DF, and LH schemes are unconditionally stable when applied to the spatially discretized linear diffusion equation* $\frac{\partial u}{\partial t} = \alpha \nabla^2 u$*, which means that the solution is bounded for arbitrary values of* α *and the time step size.*

Theorem 2. *The OOEH, DF, and LH schemes have second-order while the CCL method has third-order convergence when applied to the spatially discretized linear diffusion equation* $\frac{\partial u}{\partial t} = \alpha \nabla^2 u$.

4. The Incorporation of the Nonlinear Term and Further Methods Used for Comparison

4.1. Operator-Splitting Treatments of the Nonlinear Term

We consider the impact of the Huxley and diffusion terms separately during a time step. We obtain a concentration value u_i^{diff} after taking into account the diffusion term completely via the diffusion solvers mentioned in the previous subsection, which we briefly represent by $p = u_i^{\text{diff}}$ for brevity. The form $\beta u^2 - \beta u^3$ can be used to express the local reaction term. We perform a selective substitution of u by the p values in each of the 12 potential ways:

$$\begin{array}{ccc} 1.\ \beta up - \beta u^2 p & 2.\ \beta up - \beta u^3 & 3.\beta u^2 - \beta u^2 p \\ 4.\ \beta p^2 - \beta u^3 & 5.\ \beta u^2 - \beta u^3 & 6.\ \beta p^2 - \beta u^2 p \\ 7.\ \beta u^2 - \beta up^2 & 8.\ \beta p^2 - \beta up^2 & 9.\ \beta up - \beta up^2. \\ 10.\ \beta u^2 - \beta p^3 & 11.\ \beta up - \beta p^3 & 12.\ \beta p^2 - \beta p^3 \end{array} \tag{17}$$

Each of the obtained expressions is then inserted into the right-hand side of a simple ODE to approximate the time development of u due to the reaction term during the actual time step. In this way, we obtained 12 initial value problems (IVPs), such as

$$\begin{cases} \frac{du}{dt} = \beta pu - \beta pu^2, \\ u(t=0) = u_i^{\text{diff}} = p \ , \quad t \in [0,\ h] \end{cases}$$

Then, we attempt to solve these IVPs in an analytical way with Maple software. The solution is found in nine cases and not found in three cases, i.e., 4, 5, and 12. During the preliminary numerical tests, we observed that there are five cases (6, 7, 9, 10, and 11) where the scheme does not behave well: the error is large, sometimes due to instabilities. Hence, only the remaining four cases are displayed here in Table 1 and will be used in the remaining part of this paper.

Table 1. The solutions of the remaining four treatment cases.

Treatment	Substitution	Solution
t_1	$\beta up - \beta u^2 p$	$u_i^{n+1} = \frac{u_i^{\text{diff}}}{u_i^{\text{diff}} + \left(1 - u_i^{\text{diff}}\right) e^{-u_i^{\text{diff}} \beta h}}$ (18)
t_2	$\beta up - \beta u^3$	$u_i^{n+1} = \frac{u_i^{\text{diff}}}{\sqrt{u_i^{\text{diff}} + \left(1 - u_i^{\text{diff}}\right) e^{-2u_i^{\text{diff}} \beta h}}}$ (19)
t_3	$\beta u^2 - \beta u^2 p$	$u_i^{n+1} = \frac{u_i^{\text{diff}}}{\left(\left(u_i^{\text{diff}}\right)^2 \cdot \beta h\right) - \left(u_i^{\text{diff}} \cdot \beta h\right) + 1}$ (20)
t_8	$\beta p^2 - \beta up^2$	$u_i^{n+1} = 1 - \left(1 - u_i^{\text{diff}}\right) e^{-\left(u_i^{\text{diff}}\right)^2 \beta h}$ (21)

When simple operator-splitting is used, a time step begins with the diffusion being treated using one of the algorithms specified in Section 3. Then, as a substitutional step, one of the four procedures (18)–(21) is performed to assess the increment due to the nonlinear term. However, we also apply Strang-splitting, which entails executing one of the processes (18)–(21) once prior to and again subsequent to the computation of the diffusion effect, both with a halved time step size. This indicates that $h \to h/2$ needs to be substituted

in (18)–(21) in the first and last stages, and in the first stage, u_i^n stands instead of u_i^{diff} in (18)–(21).

Unfortunately, during the preliminary test, we were not able to obtain useful methods when these operator-splitting approaches were combined with the DF scheme. Therefore, only the CCL, OOEH, LH, and (as we will describe later) the CrN diffusion solvers will be combined with the operator-splitting approach.

4.2. Further Treatments of the Nonlinear Term

The following treatments are inspired by our previous work [41], where they were elaborated and applied to the linear reaction (convection) term and the nonlinear radiation term. For further details of the logic of their derivation, the reader should consult that paper. Since the CCL method is a three-stage method with completely different formulas than the OOEH, DF, and LH methods, we were not able to combine it with the following treatments.

1. Inside treatment.

In this case, the Huxley term is evaluated at the beginning of the actual time step to obtain $\beta h \cdot \left(u_i^n\right)^2\left(1-u_i^n\right)$ and inserted during the space and time discretization of the equation. It is multiplied by the time step size h to assess the increment due to the Huxley term. In this approach, the nonlinear term turns up in the numerator of the formulas, as listed below.

- LH method:

$$\text{First stage}: \; u_i^{n+1} = \frac{u_i^n + \beta h \cdot \left(u_i^n\right)^2\left(1-u_i^n\right)/2 + A_i/2}{1 + r_i/2}. \tag{22}$$

$$\text{Second stage}: \; u_i^{n+1} = \frac{(1-r_i/2)\cdot u_i^n + 0.5 \cdot \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1-u_i^n\right) + A_i}{1 + r_i/2}. \tag{23}$$

- DF scheme:

$$\text{First, UPFD step}: \; u_i^{n+1} = \frac{u_i^n + A_i + \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1-u_i^n\right)}{1 + 2r_i}. \tag{24}$$

$$\text{All other steps}: \; u_i^{n+1} = \frac{(1-r_i)u_i^{n-1} + 2A_i + \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1-u_i^n\right)}{1 + r_i}. \tag{25}$$

- OOEH:

$$\text{First stage}: u_i^{n+1} = (1-r_i)u_i^n + A_i + \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1-u_i^n\right). \tag{26}$$

$$\text{Second stage}: \; u_i^{n+1} = \frac{u_i^n + A_i + \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1-u_i^n\right)}{1 + r_i}. \tag{27}$$

2. Pseudo-implicit treatment (*PI*).

This case is similar to the previous one, but one of the u_i-s is evaluated at the end of the time step; thus, $\beta h \cdot \left(u_i^n\right)^2\left(1-u_i^{n+1}\right)$ is inserted instead of $\beta h \cdot \left(u_i^n\right)^2\left(1-u_i^n\right)$. When we rearrange the formula to express u_i^{n+1}, the nonlinear term appears in the denominator as well. The goal of this trick is to enhance stability.

- LH method:

$$\text{First stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + \beta h \cdot u_i^n/2\right) + A_i/2}{1 + r_i/2 + \beta h \cdot \left(u_i^n\right)^2/2}. \tag{28}$$

$$\text{Second stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 - r_i/2 + \beta h \cdot u_i^n\right) + A_i}{1 + r_i/2 + \beta h \cdot \left(u_i^n\right)^2}. \tag{29}$$

- The Dufort–Frankel (DF):

$$\text{First step}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + \beta h \cdot u_i^n\right) + A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2}. \tag{30}$$

$$\text{All other steps}: \ u_i^{n+1} = \frac{(1 - r_i)u_i^{n-1} + \beta h \cdot \left(u_i^n\right)^2 + 2A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2}. \tag{31}$$

- OOEH:

$$\text{First stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 - r_i + \beta h \cdot u_i^n\right) + A_i}{1 + \beta h \cdot \left(u_i^n\right)^2}. \tag{32}$$

$$\text{Second stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + \beta h \cdot u_i^n\right) + A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2}. \tag{33}$$

3. Mixed treatment

Mixed treatment means that simply the average of the previous two expressions $\beta h \cdot \left(u_i^n\right)^2\left(1 - u_i^n\right)$ and $\beta h \cdot \left(u_i^n\right)^2\left(1 - u_i^{n+1}\right)$ is inserted during the discretization. This treatment was proved to be successful in our previous work [41] for the convection term. In this case, we have the following formulas.

- LH method:

$$\text{First stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + 0.5 \cdot \beta h \cdot u_i^n\right) \cdot \left(1 - u_i^n/2\right) + A_i/2}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2/4} \tag{34}$$

$$\text{Second stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 - r_i/2 + \beta h \cdot u_i^n\right) \cdot \left(1 - u_i^n/2\right) + A_i}{1 + r_i/2 + \beta h \cdot \left(u_i^n\right)^2/2}. \tag{35}$$

- The Dufort–Frankel (DF):

$$\text{First step}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + \beta h \cdot u_i^n\right) \cdot \left(1 - u_i^n/2\right) + A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2/2}. \tag{36}$$

$$\text{All other steps}: \ u_i^{n+1} = \frac{(1 - r_i)u_i^{n-1} + 2\beta h \cdot \left(u_i^n\right)^2 \cdot \left(1 - u_i^n/2\right) + 2A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2}. \tag{37}$$

- OOEH:

$$\text{First stage}: \ u_i^{n+1} = \frac{u_i^n \cdot (1 - r_i) + \beta h \cdot u_i^n \cdot \left(1 - u_i^n/2\right) + A_i}{1 + \beta h \cdot \left(u_i^n\right)^2/2}. \tag{38}$$

$$\text{Second stage}: \ u_i^{n+1} = \frac{u_i^n \cdot \left(1 + \beta h \cdot u_i^n\right) \cdot \left(1 - u_i^n/2\right) + A_i}{1 + r_i + \beta h \cdot \left(u_i^n\right)^2/2}. \tag{39}$$

We note that these treatments will be subjected to intensive numerical tests, but the analytical investigation of the constructed methods is out of the scope of this paper.

4.3. Methods Used for Comparison Purposes

We employ the so-called classical version of the fourth-order Runge–Kutta (RK4) method [42] (p. 737) for comparison's sake when the running time is measured for large-sized systems. Applying it in the most standard way to our system that is spatially discretized, we have

$$k_i^1 = A_i^n - r_i^n u_i^n + \beta h \cdot \left(u_i^n\right)^2 \cdot \left(1 - u_i^n\right),$$
$$\text{then } A_i^1 = \frac{h}{C_i}\left(\frac{u_{i-1}^n + k_{i-1}^1/2}{R_{i,\,i-1}} + \frac{u_{i+1}^n + k_{i+1}^1/2}{R_{i,\,i+1}}\right) \text{ and } u_{TEMP}(i) = u_i^n + k_i^1/2.$$
$$k_i^2 = A_i^1 - r_i^n u_{TEMP}(i) + \beta h \cdot \left(u_{TEMP}(i)\right)^2 \cdot \left(1 - u_{TEMP}(i)\right), \text{ then}$$
$$A_i^2 = \frac{h}{C_i}\left(\frac{u_{i-1}^n + k_{i-1}^2/2}{R_{i,\,i-1}} + \frac{u_{i+1}^n + k_{i+1}^2/2}{R_{i,\,i+1}}\right) \text{ and}$$
$$u_{TEMP}(i) = u_i^n + k_i^2/2.$$
$$k_i^3 = A_i^2 - r_i^n u_{TEMP}(i) + \beta h \cdot \left(u_{TEMP}(i)\right)^2 \cdot \left(1 - u_{TEMP}(i)\right), \text{then}$$
$$A_i^3 = \frac{h}{C_i}\left(\frac{u_{i-1}^n + k_{i-1}^3}{R_{i,\,i-1}} + \frac{u_{i+1}^n + k_{i+1}^3}{R_{i,\,i+1}}\right) \text{and } u_{TEMP}(i) = u_i^n + k_i^3.$$

And, finally,

$$k_i^4 = A_i^3 - r_i^n u_{TEMP}(i) + \beta h \cdot \left(u_{TEMP}(i)\right)^2 \cdot \left(1 - u_{TEMP}(i)\right), \text{and}$$
$$u_i^{n+1} = u_i^n + \left(k_i^1 + 2k_i^2 + 2k_i^3 + k_i^4\right)/6.$$

Along with the RK4, the two-step Adams–Bashforth [43] method (AB2) is also tested in Section 5. As far as we know, the performance of the explicit and stable methods has never been compared to the standard explicit multistep methods.

We will also extensively test an implicit method, which is the widely used Crank–Nicolson (CrN) scheme. It is well known that applying Equation (6) leads to the matrix equation

$$\underbrace{\left(I - \frac{h}{2}M\right)}_{A} U^{n+1} = \underbrace{\left(I + \frac{h}{2}M\right)}_{B} U^n\,,$$

where I is the unit matrix of size $N \times N$. Since A and B are time-independent in the current work, one can spare running time by calculating $Y = A^{-1}B$ prior to the first time step. In this way, only a single matrix multiplication must be performed in each time step: $U^{n+1} = Y\,U^n$. The CrN method will be combined with the simple and Strang operator-splitting treatments of the nonlinear term listed in Section 4.1.

The next set of methods belongs to the family of the nonstandard finite difference methods. They are developed to solve nonlinear PDE (1) in one space dimension with $\alpha = 1$ and using an equidistant mesh. Four nonstandard finite difference techniques using the following formulas were proposed in paper [3]:

$$\text{NSFD1}: \; u_i^{n+1} = \frac{(1-2H)u_i^n + Hu_{i\pm1}^n + \beta\phi_2\left((1+\gamma)\left(u_{i-1}^n\right)^2 + \left(u_{i-1}^n\right)^3/2\right)}{1 + \beta\gamma\phi_2 + 3/2\beta\phi_2\left(u_{i-1}^n\right)^2},$$
$$\text{NSFD2}: \; u_i^{n+1} = \frac{(1-2H)u_i^n + Hu_{i\pm1}^n}{1 + \beta\gamma\phi_2 - \beta(1+\gamma)\phi_2 u_i^n + \beta\phi_2\left(u_i^n\right)^2},$$
$$\text{NSFD3}: \; u_i^{n+1} = \frac{(1-2H)u_i^n + Hu_{i\pm1}^n + \beta\phi_2\left(u_i^n\right)^3 + (1+\gamma)\left(u_i^n\right)^2}{1 + \beta\gamma\phi_2 + 2\beta\phi_2\left(u_i^n\right)^2},$$
$$\text{NSFD4}: \; u_i^{n+1} = \frac{(1-2H)u_i^n + Hu_{i\pm1}^n + \beta\phi_2\left((1+\gamma)\left(u_i^n\right)^2 + \left(u_i^n\right)^3/2\right)}{1 + \beta\gamma\phi_2 + 3/2\beta\phi_2\left(u_i^n\right)^2},$$

where $\phi_2 = \frac{e^{\beta\Delta t}-1}{\beta}$, $\psi_1 = \frac{1-e^{-\beta\Delta x}}{\beta}$, $\psi_2 = \frac{e^{\beta\Delta x}-1}{\beta}$, and $H = \frac{\phi_2}{\psi_1\,\psi_2}$. The positivity and boundedness of these schemes have been analytically shown. These are not, however, unconditional; the most crucial condition is $H \leq \frac{1}{2}$, which is similar to the typical CFL limit for the explicit Euler method.

When we perform running time measurements in large 2D systems, we also employ ode15s for comparison purposes. This solver is included in MATLAB, where 15s means a one- to five-order numerical differentiation formula with variable step and variable order (VSVO) that was designed for stiff problems.

5. First Round of Numerical Experiments: Verification and Selection Based on Errors

The following formula is used to compare the numerical solutions u_j^{num} generated by the solver under examination with the analytical reference solution u_j^{ref} at the final time t^{fin} in order to determine the maximum numerical errors:

$$\text{Error}(L_\infty) = \max_{1\leq j\leq N}\left|u_j^{\text{ref}}\left(t^{\text{fin}}\right) - u_j^{\text{num}}\left(t^{\text{fin}}\right)\right|. \tag{40}$$

The average error is also very similar:

$$\text{Error}(L_1) = \frac{1}{N}\sum_{1\leq j\leq N}\left|u_j^{\text{ref}}(t_{\text{fin}}) - u_j^{\text{num}}(t_{\text{fin}})\right|. \tag{41}$$

We refer to the third kind of error as the energy error as it has an energy dimension in the context of the heat conduction equation:

$$\text{Error}(Energy) = \sum_{1\leq j\leq N} C_j\left|u_j^{\text{ref}}\left(t^{\text{fin}}\right) - u_j^{\text{num}}\left(t^{\text{fin}}\right)\right|. \tag{42}$$

As we already mentioned, our goal is to assess the performance of the methods not only for one concrete time step size but for many. Hence, starting from a very large time step size, we calculate the above-defined errors for a series of decreasing time step sizes. This series of time step sizes usually starts from $T/4$ and has S elements, where the next element is obtained by taking half of the previous one. Then, we compute the three kinds of aggregated error, $AgErr(L_\infty)$, $AgErr(L_1)$, and $AgErr(Energy)$, as the mean of the logarithm of these errors as follows:

$$\begin{aligned} AgErr(L_\infty) &= \frac{1}{S}\sum_{s=1}^{S}\log(\text{Error}(L_\infty)),\\ AgErr(L_1) &= \frac{1}{S}\sum_{s=1}^{S}\log(\text{Error}(L_1))\\ AgErr(Energy) &= \frac{1}{S}\sum_{s=1}^{S}\log(\text{Error}(Energy)). \end{aligned} \tag{43}$$

Finally, the average of the three different $AgErr$ errors is determined:

$$AgErr = \frac{1}{3}(AgErr(L_\infty) + AgErr(L_1) + AgErr(Energy)), \tag{44}$$

and this $Agerr$ quantity is used to evaluate the overall accuracy for the small, medium, and large time step sizes for many possible combinations. It should be noted that the method is very accurate when $AgErr$ is negative and has a big absolute value. In the case of instabilities, the u values can be too large for MATLAB to handle, and '*NaN*' symbols appear instead of numbers. In these cases, the error is substituted by 10^{10} as a penalty

for unstable behavior. This ensures that the aggregated errors are numbers that can be comparable to other errors.

5.1. Experiment 1: One Space Dimension Using an Exact Solution

In this and the following experiment, PDE (1) with $\alpha = 1$ will be solved on the domain $(x,\ t) \in [x_1,\ x_N] \times [t^{\text{ini}},\ t^{\text{fin}}]$. The following analytical solution [1] (p. 34) serves as the reference for validation.

$$u^{\text{exact}}(x,t) = \frac{e^{\sqrt{\frac{\beta}{2}}x + \frac{\beta}{2}t}}{e^{\sqrt{\frac{\beta}{2}}x + \frac{\beta}{2}t} + c} \tag{45}$$

By evaluating this function at the time and space locations that correspond to the initial time $u(x,\ 0)$ and the boundaries $u(x_1,t), u(x_N,t)$, respectively, the initial and Dirichlet boundary conditions are determined.

In Experiment 1, $x_1 = 0,\ x_{N=500} = 2,\ t^{\text{ini}} = 0.01,\ t^{\text{fin}} = 0.1$, and the space step size $\Delta x = 0.004$. The nonlinear coefficient is $\beta = 2$, and $c = 19$ in Equation (43). Figures 2–4 show the maximum errors as a function of time step size for different method combinations. The aggregated errors for different diffusion solvers and treatments of the Huxley term are shown in Figure 5 and Table 2. Since the errors go down with the time step size, the methods can be considered verified.

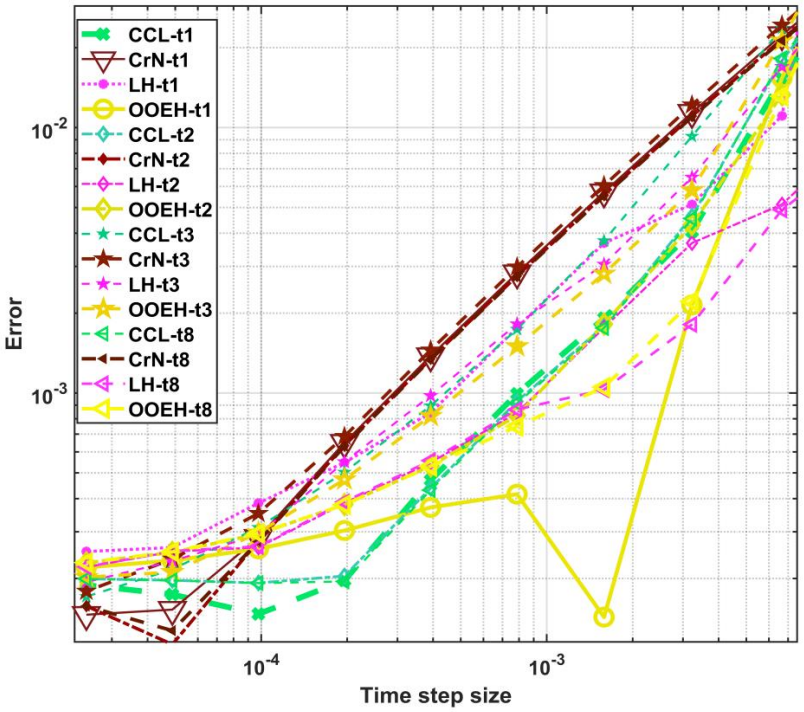


Figure 2. Maximum errors as a function of time step size h for different operator-splitting treatments in the case of Experiment 1.

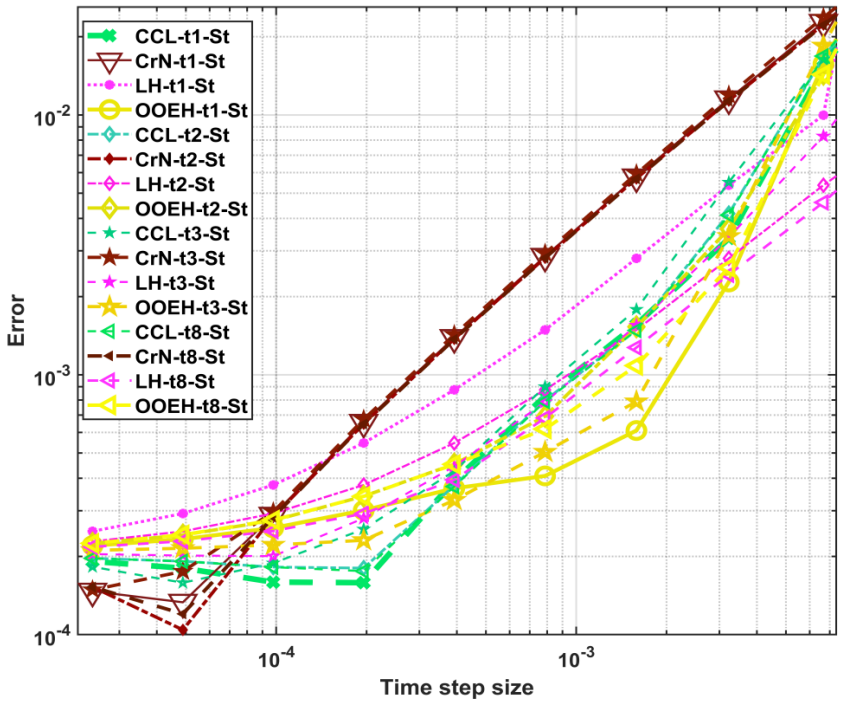


Figure 3. Maximum errors as a function of time step size for different operator-splitting treatments with Strang-splitting in the case of Experiment 1.

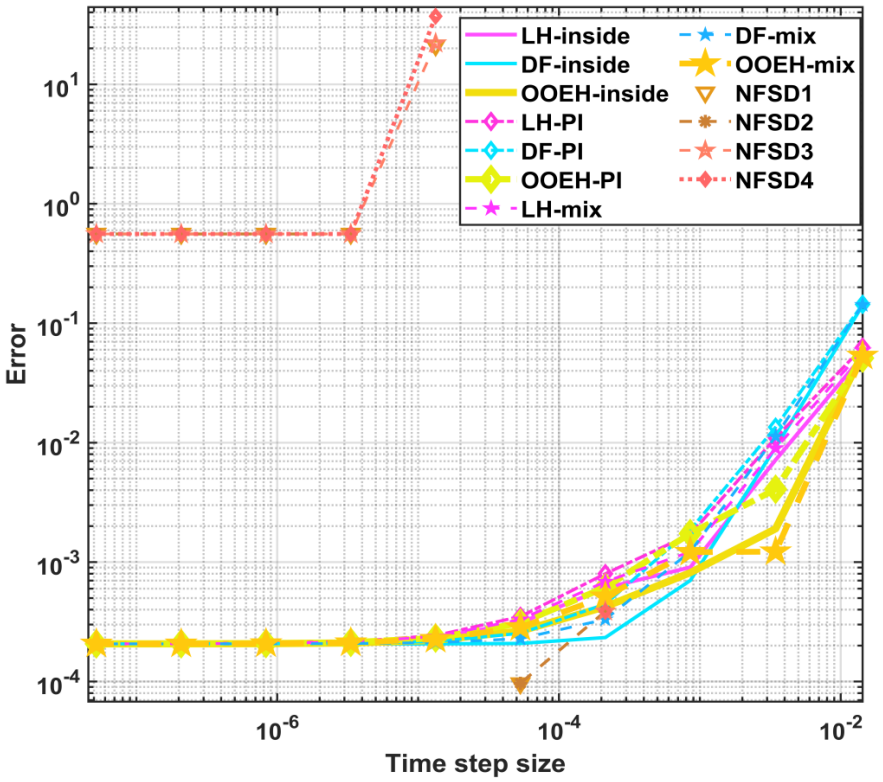


Figure 4. Maximum errors as a function of time step size for the remaining methods in the case of Experiment 1.

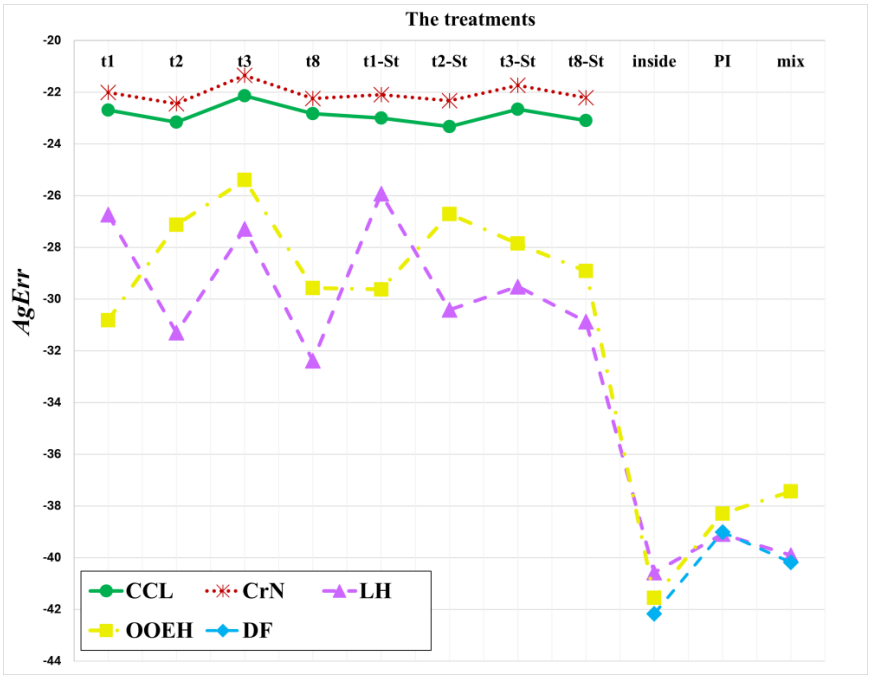


Figure 5. The aggregated errors (*AgErr*) for different diffusion solvers and treatments of the Huxley term in the case of Experiment 1.

Table 2. The aggregated error (*AgErr*) values for different treatment of the methods in the case of Experiment 1.

Methods	Treatment of the Nonlinear Term										
	t_1	t_2	t_3	t_8	t_1-St	t_2-St	t_3-St	t_8-St	Inside	PI	mix
	AgErr										
CCL	−22.70	−23.16	−22.13	−22.83	−23.00	−23.33	−22.66	−23.10	-	-	-
CrN	−22.00	−22.44	−21.34	−22.24	−22.10	−22.33	−21.73	−22.21	-	-	-
LH	−26.73	−31.29	−27.29	−32.38	−25.92	−30.42	−29.51	−30.87	−40.58	−39.08	−39.90
OOEH	−30.81	−27.13	−25.40	−29.60	−29.62	−26.71	−27.85	−28.91	−41.55	−38.29	−37.44
DF	-	-	-	-	-	-	-	-	−42.17	−39.01	−40.18

5.2. Experiment 2: One Space Dimension Using an Exact Solution, Larger Final Time

Here, PDE (1) is discretized on the $x_1 = 0$, $x_{N=400} = 2$, $t^{\text{ini}} = 0.01$, and $t^{\text{fin}} = 0.9$ domain using the space step size $\Delta x = 0.005$, $S = 10$. The nonlinear coefficient is $\beta = 3$ and $c = 179$ in Equation (43). The aggregated errors for different diffusion solvers and treatments of the Huxley term are shown in Figure 6 and Table 3. The maximum errors as a function of time step size for different method combinations are presented in the Supplementary Materials (see Figures S1–S3).

The main conclusion of the first two experiments is that the operator-splitting treatments yield larger errors than the other (inside, PI, and mixed) treatments, with the exception of OOEH–inside. The NSFD schemes sometimes give very small errors, but they are completely unreliable. Since the systems are not stiff here, the implicit CrN method has no advantage compared to the other methods.

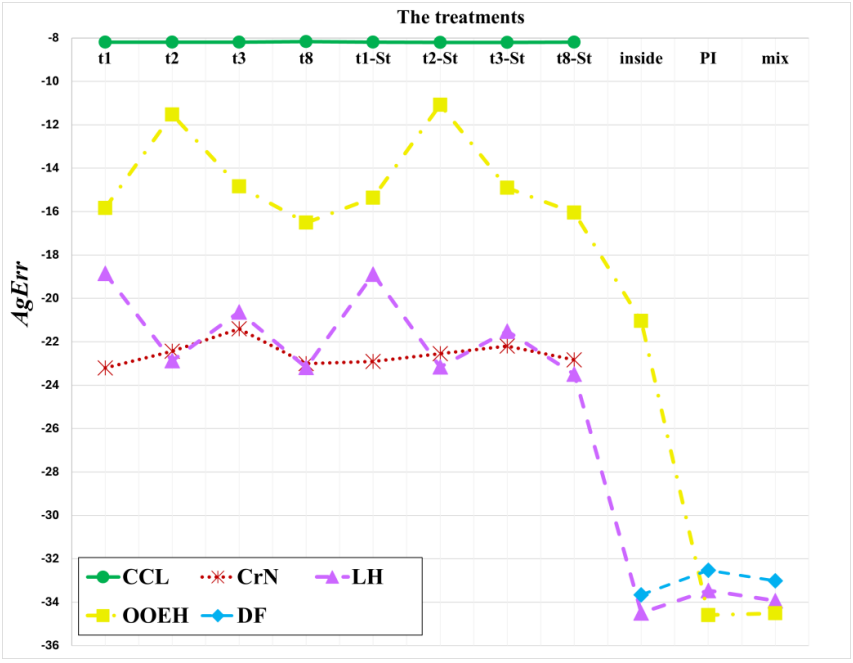


Figure 6. Aggregated errors (*AgErr*) for different diffusion solvers and treatments of the Huxley term in the case of Experiment 2.

Table 3. The aggregated error (*AgErr*) values for different treatment of the methods in the case of Experiment 2.

Methods	**Treatment of the Nonlinear Term**										
	t_1	t_2	t_3	t_8	t_1-St	t_2-St	t_3-St	t_8-St	Inside	PI	mix
	AgErr										
CCL	−8.19	−8.19	−8.19	−8.17	−8.20	−8.20	−8.20	−8.19	-	-	-
CrN	−23.21	−22.43	−21.40	−23.00	−22.91	−22.54	−22.19	−22.83	-	-	-
LH	−18.84	−22.87	−20.62	−23.18	−18.88	−23.16	−21.51	−23.49	−34.49	−33.46	−33.93
OOEH	−15.84	−11.53	−14.84	−16.51	−15.36	−11.07	−14.90	−16.05	−21.05	−34.59	−34.51
DF	-	-	-	-	-	-	-	-	−33.66	−32.53	−33.01

5.3. Experiment 3: Stiff 2D System

Employing the capacity–resistivity model described in Section 2, we have two space dimensions from this point. Using a log-uniform distribution, we produce random values for the resistances and capacities as follows:

$$C_i = 10^{(a_C - b_C \cdot rand)}, \quad R_{x,i} = 10^{(a_{Rx} - b_{Rx} \cdot rand)}, \quad R_{y,i} = 10^{(a_{Ry} - b_{Ry} \cdot rand)}. \tag{46}$$

We create highly varied test problems by changing the a and b values. The variable *rand* is MATLAB-generated random numbers within the unit interval. From this point, the reference solution is provided by the ode15s solver of MATLAB with a very stringent tolerance (10^{-11}) to ensure accuracy. The used mesh size and final time are $N_x = 31$, $N_y = 13$, $t^{\text{fin}} = 0.3$, while $\beta = 8.3$ and $a_C = 3$, $b_C = 6$, $a_{Rx} = 1$, $b_{Rx} = 2$, $a_{Rz} = 1$, $b_{Rz} = 2$. These variables give $SR = 9.1 \cdot 10^8$ and $CFL = 1.7 \cdot 10^{-5}$. The initial concentration function is the sum of a smooth function and some random numbers:

$$u_i^0 = \frac{1}{\sqrt{(i+4)}} + 0.2 \times rand.$$

Figures 7 and 8 show the maximum errors as a function of time step size for different method combinations. The maximum errors for different operator-splitting treatments with Strang-splitting are presented in the Supplementary Materials (Figure S4). The aggregated errors for different diffusion solvers and treatments of the Huxley term are presented in Figure 9 and Table 4.

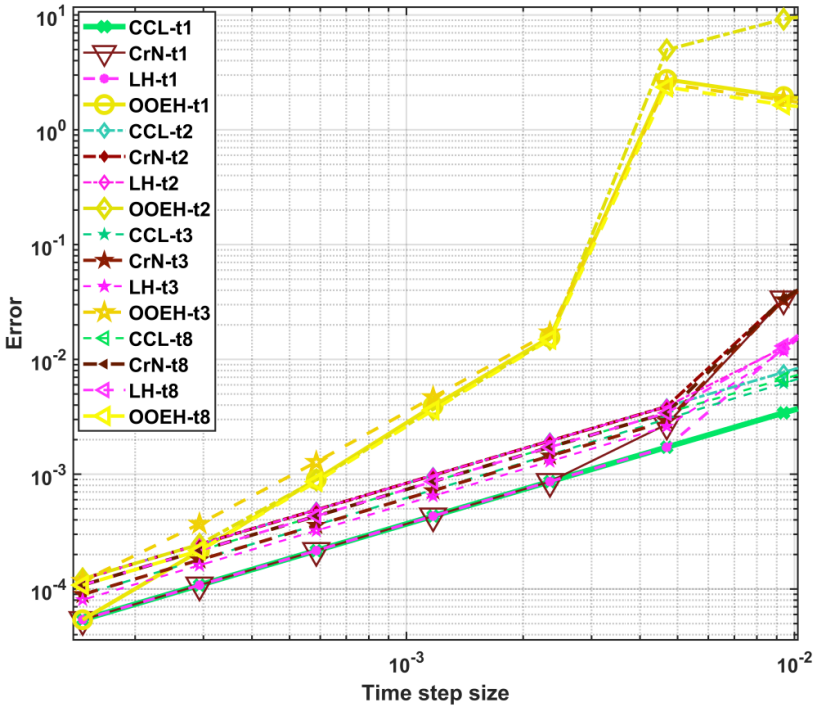


Figure 7. Maximum errors as a function of time step size for different operator-splitting treatments in the case of Experiment 3.

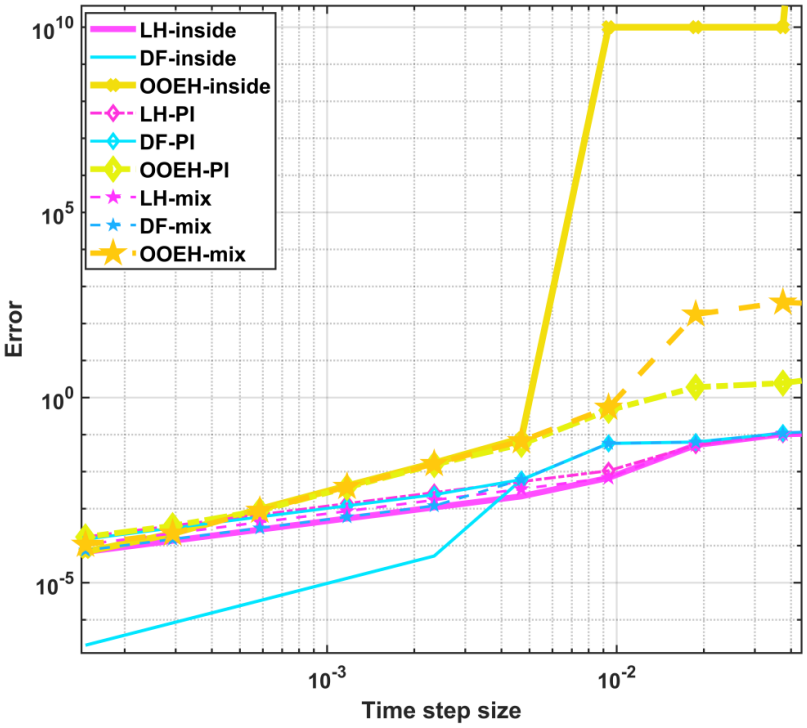


Figure 8. Maximum errors as a function of time step size for the remaining methods in the case of Experiment 3.

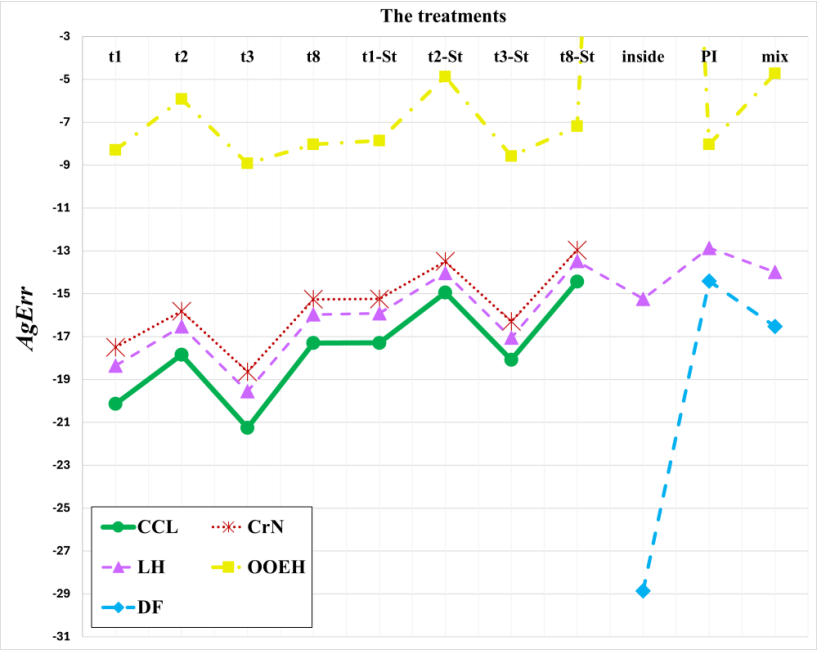


Figure 9. The aggregated errors (*AgErr*) for different diffusion solvers and treatments of the Huxley term in the case of Experiment 3.

Table 4. The aggregated error (*AgErr*) values for different treatment of the methods in the case of Experiment 3.

Methods	Treatment of the Nonlinear Term										
	t_1	t_2	t_3	t_8	t_1-St	t_2-St	t_3-St	t_8-St	Inside	PI	mix
	AgErr										
CCL	−20.14	−17.84	−21.25	−17.30	−17.29	−14.94	−18.08	−14.43	-	-	-
CrN	−17.49	−15.81	−18.64	−15.26	−15.24	−13.49	−16.29	−12.96	-	-	-
LH	−18.35	−16.52	−19.55	−15.97	−15.91	−14.03	−17.05	−13.48	−15.24	−12.86	−13.99
OOEH	−8.29	−5.92	−8.92	−8.03	−7.86	−4.87	−8.58	−7.19	52.60	−8.04	−4.72
DF	-	-	-	-	-	-	-	-	−28.86	−14.41	−16.53

One can see that the DF–inside combination shows the best performance since its convergence rate is the highest. On the other hand, the operator-splitting treatments behave much better than before, especially the t1, t3, and t3-Strang treatments.

5.4. Experiment 4: Non-Stiff System with Strong Nonlinearity

The used mesh size and final time are $N_x = 21$, $N_y = 11$, and $t^{\text{fin}} = 0.9$, while $\beta = 29$, and $a_C = 0$, $b_C = 0$, $a_{Rx} = 0$, $b_{Rx} = 0$, $a_{Rz} = 0$, and $b_{Rz} = 0$. These variables give $SR = 3.5 \cdot 10^2$ and $CFL = 0.25$. The random component in the initial concentration function is slightly larger than before:

$$u_i^0 = \frac{1}{\sqrt{(i+6)}} + 0.5 \times rand$$

Figure 10 shows the maximum errors as a function of time step size for different operator-splitting treatments. These errors for different operator-splitting treatments with Strang-splitting and for the remaining methods are presented in the Supplementary Materials (see Figures S5 and S6). The aggregated errors for different diffusion solvers and

treatments of the Huxley term are shown in Figure 11 and Table 5. The results show that in this case, the treatment of the nonlinear term is much more important than that of the diffusion term. This is logical since the coefficient β is large, while the problem is easy to solve from the point of view of diffusion. The large value of β has another consequence: the "force" acting on u towards its maximum value 1 is large. Since the final time is also quite large, the values of u are getting very close to 1. When this happens, the importance of the diffusion process decreases, which further reinforces the previously mentioned effect and decreases the calculation errors below 10^{-10} for short time step sizes. The DF–inside method, which was the most accurate in the previous experiment, is unstable here for almost all time step sizes, and that is why its error is large.

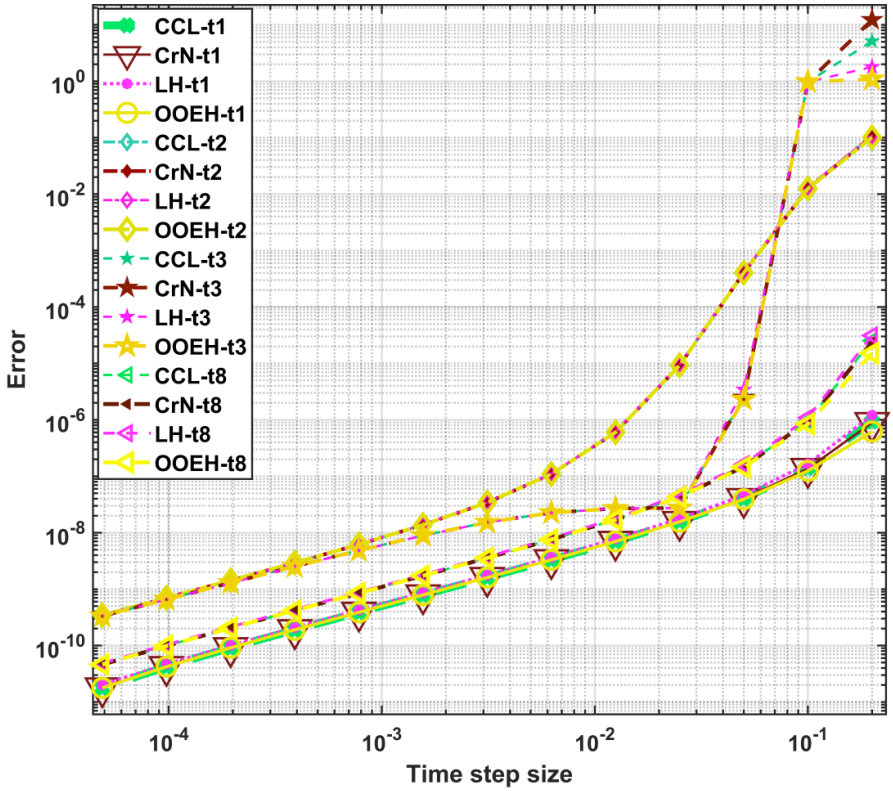


Figure 10. Maximum errors as a function of time step size for different operator-splitting treatments in the case of Experiment 4.

Table 5. The aggregated error (*AgErr*) values for different treatment of the methods in the case of Experiment 4.

Methods	**Treatment of the Nonlinear Term**										
	t_1	**t_2**	**t_3**	**t_8**	**t_1-St**	**t_2-St**	**t_3-St**	**t_8-St**	**Inside**	**PI**	**mix**
	AgErr										
CCL	−113.64	−81.17	−85.91	−107.26	−117.42	−90.35	−96.61	−112.29	-	-	-
CrN	−113.61	−81.18	−85.694	−107.28	−117.37	−90.36	−96.60	−112.31	-	-	-
LH	−113.03	−81.11	−85.61	−106.92	−117.76	−90.95	−98.35	−112.68	−78.67	−84.96	−101.67
OOEH	−113.89	−81.21	−86.00	−107.51	−117.71	−90.41	−96.56	−112.56	−78.22	−84.61	−101.21
DF	-	-	-	-	-	-	-	-	201.37	−79.19	−66.52

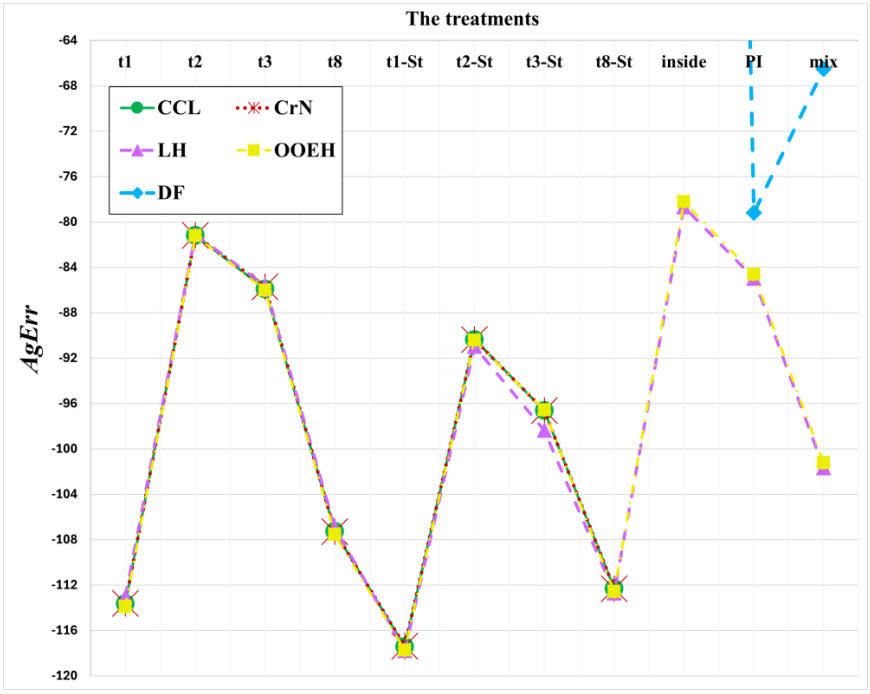


Figure 11. The aggregated errors (*AgErr*) for different diffusion solvers and treatments of the Huxley term in the case of Experiment 4.

5.5. Experiment 5: Medium Stiffness and Nonlinear Coefficient

The used mesh size and final time are $N_x = 17$, $N_y = 12$, and $t^{\text{fin}} = 0.5$, while $\beta = 13$, and $a_C = 2$, $b_C = 4$, $a_{Rx} = 2$, $b_{Rx} = 4$, $a_{Rz} = 2$, and $b_{Rz} = 4$. These variables give $SR = 2.4 \cdot 10^6$ and $CFL = 3.7 \cdot 10^{-4}$. The initial concentration function is:

$$u_i^0 = \frac{1}{\sqrt{(i+11)}} + 0.7 \times rand.$$

Figure 12 shows the maximum errors as a function of time step size for different operator-splitting treatments. The maximum errors as a function of time step size for different operator-splitting treatments with Strang-splitting and for the remaining methods are presented in the Supplementary Materials (see Figures S7 and S8). The aggregated errors for different diffusion solvers and treatments of the Huxley are presented in Figure 13 and Table 6.

Table 6. The Aggregated error (*AgErr*) values for different treatment of the methods in the case of Experiment 5.

Methods	**Treatment of the Nonlinear Term**										
	t_1	**t_2**	**t_3**	**t_8**	**t_1-St**	**t_2-St**	**t_3-St**	**t_8-St**	**Inside**	**PI**	**mix**
	AgErr										
CCL	−34.56	−27.81	−29.44	−29.93	−37.50	−31.79	−34.38	−33.48	-	-	-
CrN	−33.99	−27.56	−29.16	−29.66	−36.41	−31.10	−34.06	−32.76	-	-	-
LH	−33.89	−27.61	−30.01	−29.53	−37.65	−31.92	−34.96	−33.51	−30.16	−25.10	−28.19
OOEH	−27.14	−22.48	−24.41	−24.74	−28.32	−24.82	−26.87	−26.81	82.12	−20.16	−13.03
DF	-	-	-	-	-	-	-	-	−34.94	−25.22	−28.72

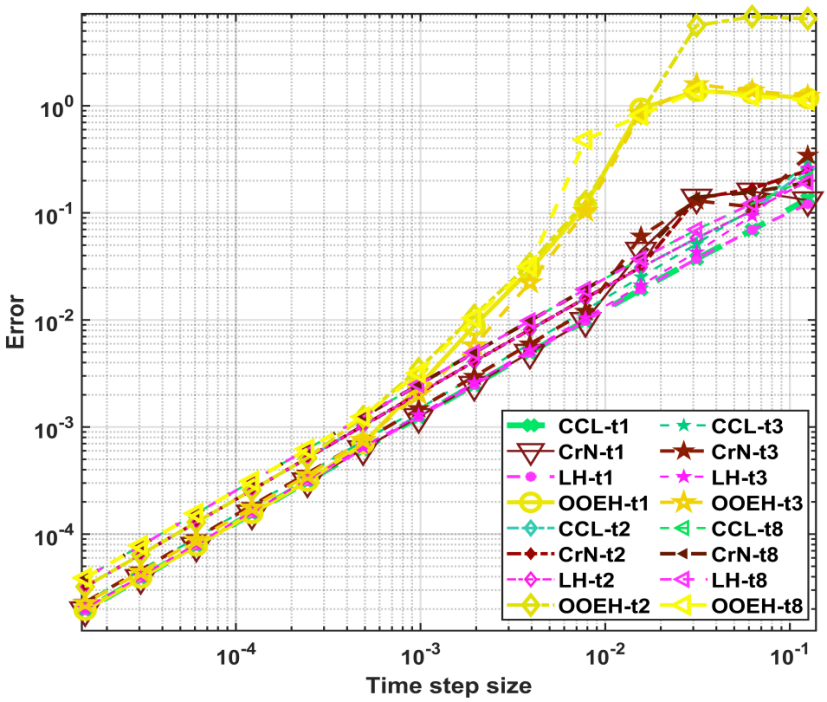


Figure 12. Maximum errors as a function of time step size for different operator-splitting treatments in the case of Experiment 5.

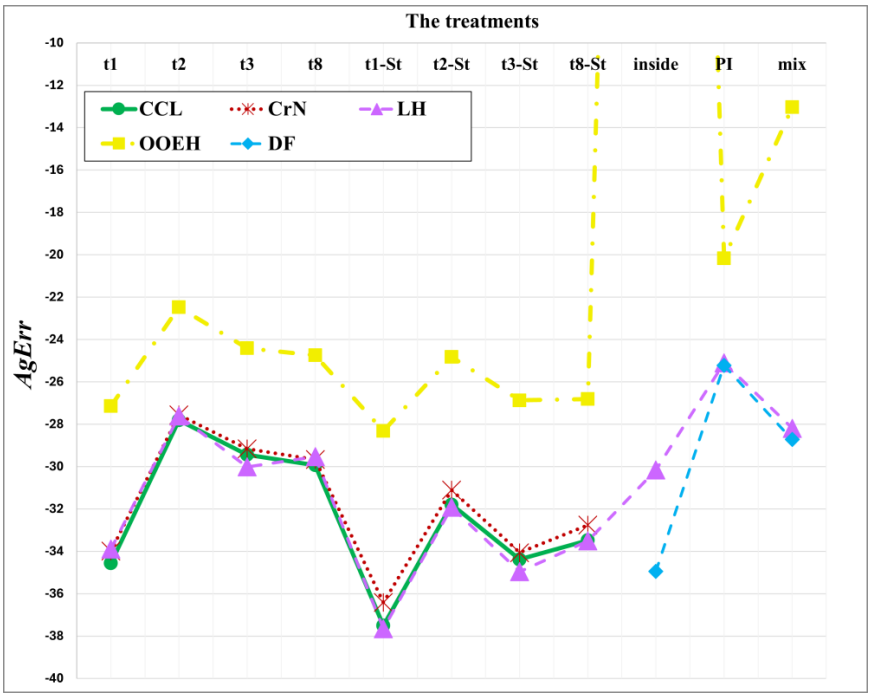


Figure 13. The aggregated errors (*AgErr*) for different diffusion solvers and treatments of the Huxley term in the case of Experiment 5.

5.6. The Summary of the Five Experiments

The simple sum of the aggregated errors for different diffusion solvers and treatments of the Huxley term are shown in Figure 14 and Table 7. Based on these results, we select the following 15 methods for further examination: LH-t1, LH-t1-Strang, LH-t3-Strang, LH-t8-Strang, LH–inside, LH-mix, DF-PI, DF–mixed, OOEH-PI, OOEH-t8-Strang, CCL-t1, CCL-t1-Strang, CCL-t8-Strang, CrN-t1, and CrN-t1-Strang. We note that since simple

operator-splitting is calculated faster than Strang-splitting, we are slightly biased towards the simple splitting to give them a further chance in the next section, where running time will be measured.

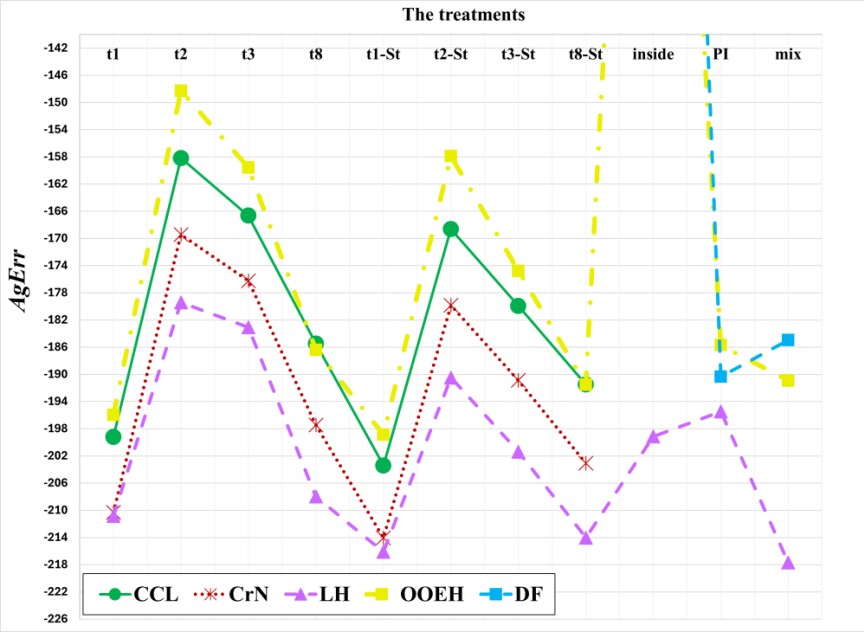


Figure 14. The aggregated errors (AgErr) for different diffusion solvers and treatments of the Huxley term.

Table 7. The aggregated errors (*AgErr*) for different treatment of the methods.

Methods	Treatment of the Nonlinear Term										
	t_1	t_2	t_3	t_8	t_1-St	t_2-St	t_3-St	t_8-St	Inside	PI	mix
	AgErr										
CCL	−199.23	−158.18	−166.65	−185.49	−203.41	−168.61	−179.93	−191.49	-	-	-
CrN	−210.30	−169.42	−176.24	−197.44	−214.03	−179.82	−190.87	−203.07	-	-	-
LH	−210.84	−179.40	−183.08	−207.98	−216.12	−190.48	−201.38	−214.03	−199.14	−195.46	−217.68
OOEH	−195.97	−148.27	−159.57	−186.39	−198.87	−157.89	−174.76	−191.52	−6.09	−185.69	−190.91
DF	-	-	-	-	-	-	-	-	61.74	−190.36	−184.96

6. Second Round of Numerical Experiments: Selection by the Performance with Running Time Measurements in 2D

6.1. Experiment 6: Non-Stiff System

In this section, the systems are much larger to make running time measurements meaningful. Here, the used system size is $N_x = 201$ and $N_y = 60$. In order to create a system with a relatively low stiffness ratio, we set $C_i = R_{x,i} = R_{y,i} = 0.01$, which is equivalent to a uniform mesh with constant α. These parameters yield $CFL = 2.5 \cdot 10^{-5}$, $SR = 3.99 \cdot 10^4$. Meanwhile, the final time is $t^{\text{fin}} = 0.3$, while the coefficient of the reaction term is $\beta = 9$. The initial concentration is given by the continuous function:

$$u_i^0 = 0.9 \cdot \cos\left(\frac{\pi}{2}\frac{x}{N_x}\right) \cdot \cos\left(\frac{\pi}{2}\frac{y}{N_y}\right).$$

The tic-toc function in MATLAB is used to measure the running times. All the running times are measured on a desktop computer with an Intel Core i7-13700 (24 CPUs) and 64 GB

RAM is used. The calculations are completed more than one time; then, the average running times are determined to reduce the impact of the random fluctuations. The obtained maximum errors are shown as a function of the running times in Figure 15. One can see from the figure the LH-t1, LH-t1-Strang, and LH-t8-Strang methods are more efficient than other methods when low or medium accuracy is required. The solvers RK4 and ode15s provide extremely high accuracy but with longer computation time. The aggregated errors for different methods in Experiment 6 are presented in the Supplementary Materials (see Table S1).

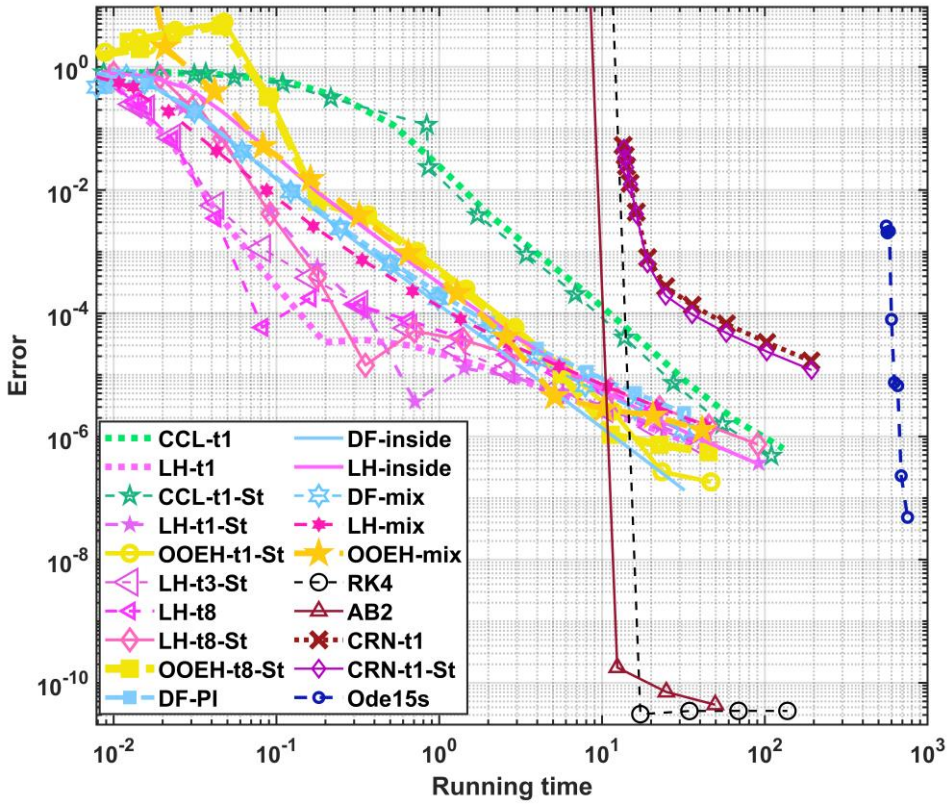


Figure 15. Maximum errors as a function of the running time for Experiment 6; non-stiff system.

6.2. Experiment 7: Moderately Stiff System

In this case, the system size and the final time used are $N_x = 101$, $N_y = 120$, $t^{\text{fin}} = 0.8$, and $\beta = 21$, while the exponents are $a_C = a_{Rx} = a_{Ry} = 0$ and $b_C = b_{Rx} = b_{Ry} = 2$. These variables give $SR = 8.6 \cdot 10^5$ and $CFL = 7.2 \cdot 10^{-5}$. This indicates that the system is moderately stiff only. The initial concentrations contain random numbers in the unit interval as follows:

$$u_i^0 = 0.3 \cdot \cos\left(\frac{x}{N_x}\frac{\pi}{2}\right) \cdot \cos\left(\frac{y}{N_y}\frac{\pi}{2}\right) + 0.6 \cdot rand.$$

In Figure 16, we show the errors as a function of running times. The aggregated errors for different treatments of the methods in the case of Experiment 7 are presented in the Supplementary Materials (see Table S2). One can see when the stiffness increases, the LH-t3-Strang method is the most efficient, closely followed by LH-t1, CCL-t1-Strang, LH-t1-Strang, LH-t8-Strang, OOEH-t3-Strang, and DF-mix.

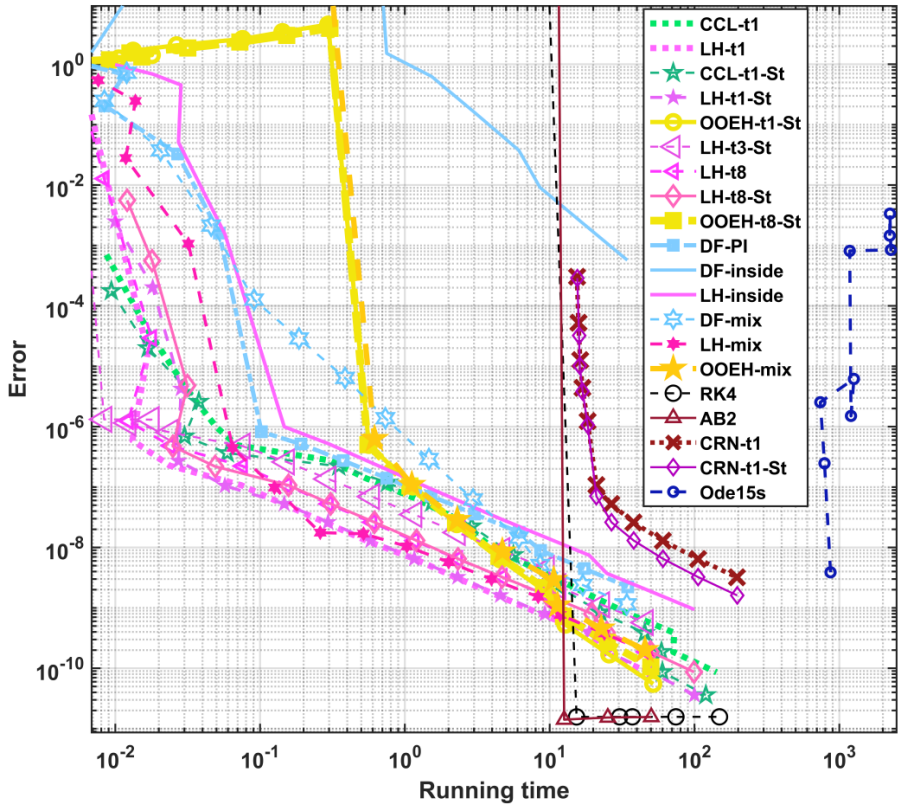


Figure 16. Maximum errors as a function of the running time for Experiment 7; moderately stiff system.

6.3. Experiment 8: Very Stiff System

Now, the used system size is $N_x = 101$, $N_y = 100$, and the exponents are $a_C = 3$, $b_C = 6$, $a_{Rx} = 3$, $b_{Rx} = 6$, $a_{Ry} = 2$, and $b_{Ry} = 4$. These parameters provide the values $SR = 1.3 \cdot 10^{10}$ and $CFL = 1.2 \cdot 10^{-5}$, which means that the system is rather very stiff. The final time is $t^{\text{fin}} = 0.7$, and the coefficient of the reaction term is $\beta = 7$. The initial concentrations also contain random numbers in the unit interval:

$$u_i^0 = 0.2 \cdot \cos\left(\frac{x}{N_x}\frac{\pi}{2}\right) \cdot \cos\left(\frac{y}{N_y}\frac{\pi}{2}\right) + 0.7 \cdot rand.$$

In Figure 17, we present the maximum errors as a function of the running times. The aggregated errors for different treatments of the methods in the case of Experiment 8 are presented in the Supplementary Materials (see Table S3).

Based on these and the previous experiments, we choose the top seven methods to include the following combinations: LH-t1, LH-t1-Strang, LH-t3-Strang, LH-t8-Strang, CCL-t1-Strang, OOEH-t8-Strang and DF–mixed.

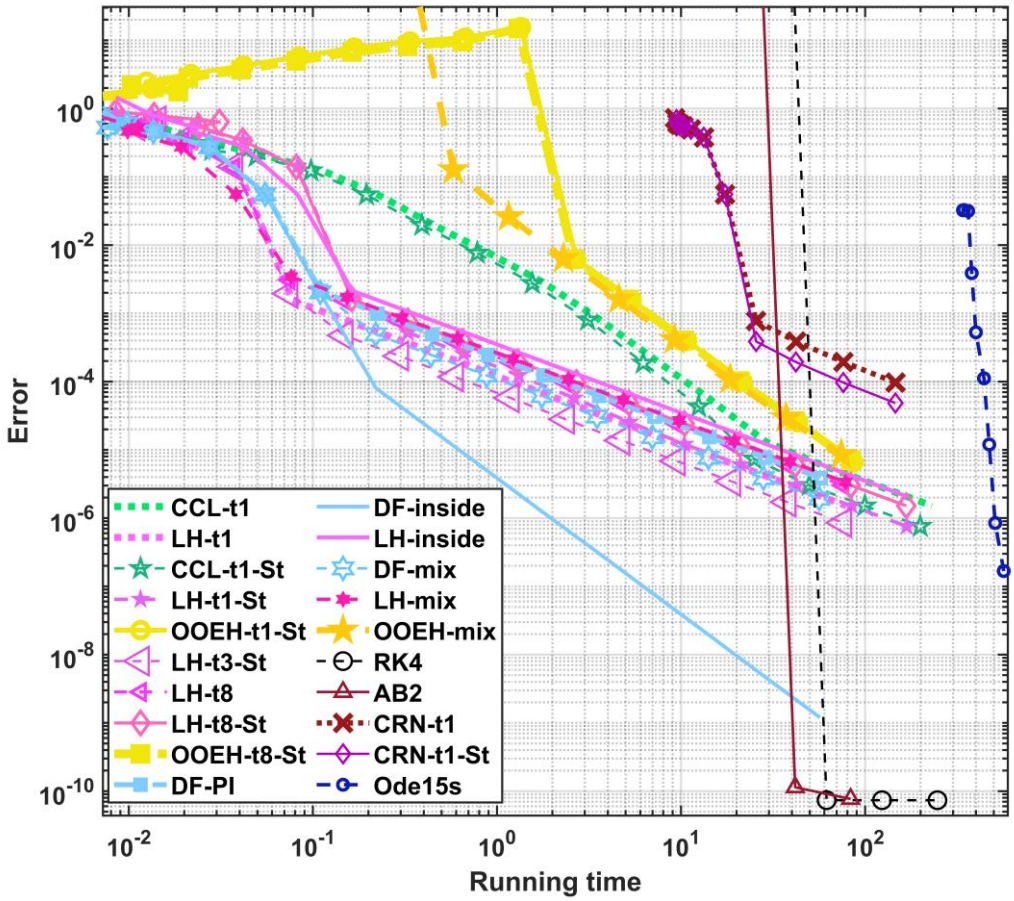


Figure 17. Maximum errors as a function of the running time for Experiment 8; very stiff system.

7. Third Round of Numerical Experiments: Parameter Sweep for the Top Seven Methods

7.1. Experiment 9: Sweep for the Size of Capacities and Resistances

The used system size and final time are $N_x = N_y = 31$ and $t^{\text{fin}} = 0.8$, while $\beta = 3.7$ and $S = 17$. Let us simulate a system that consists of two homogeneous parts. In the second half, the material properties are different, which is reflected by smaller capacities and the resistances. We assume

$$\begin{cases} C_i = R_{x,i} = R_{y,i} = 1 & \text{if } i \le N/2 \\ C_i = R_{x,i} = R_{y,i} = \varepsilon & \text{if } i > N/2\,, \end{cases}$$

where $\varepsilon \in \{1, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001\}$. This means we first run the simulation using the first value of ε, register the errors, decrease ε to 0.3 without changing any other parameters, run the code again, etc. The initial concentration function is a smooth function as follows:

$$u_i^0 = \left(\cos\left(\frac{2\pi x}{N_x}\right) \cdot \cos\left(\frac{\pi y}{N_y}\right)\right).$$

If the capacities and resistances are decreasing, the physical process of diffusion becomes faster. To follow these faster changes, a shorter time step size should be used due to the decreasing CFL limit. However, we keep the set of the used time step sizes fixed; therefore, we experience an increase in the errors with decreasing ε. The interesting fact is that this increase is very different for the different methods: the worst is for the OOEH method and much less for the other methods. In Figure 18 and Table 8, we present the aggregated errors as a function of the ε in the case of Experiment 9 for the top seven

methods. One can see that with the exception of the original hopscotch method, the algorithms perform well even in the case where the CFL limits for the explicit RK methods are extremely low.

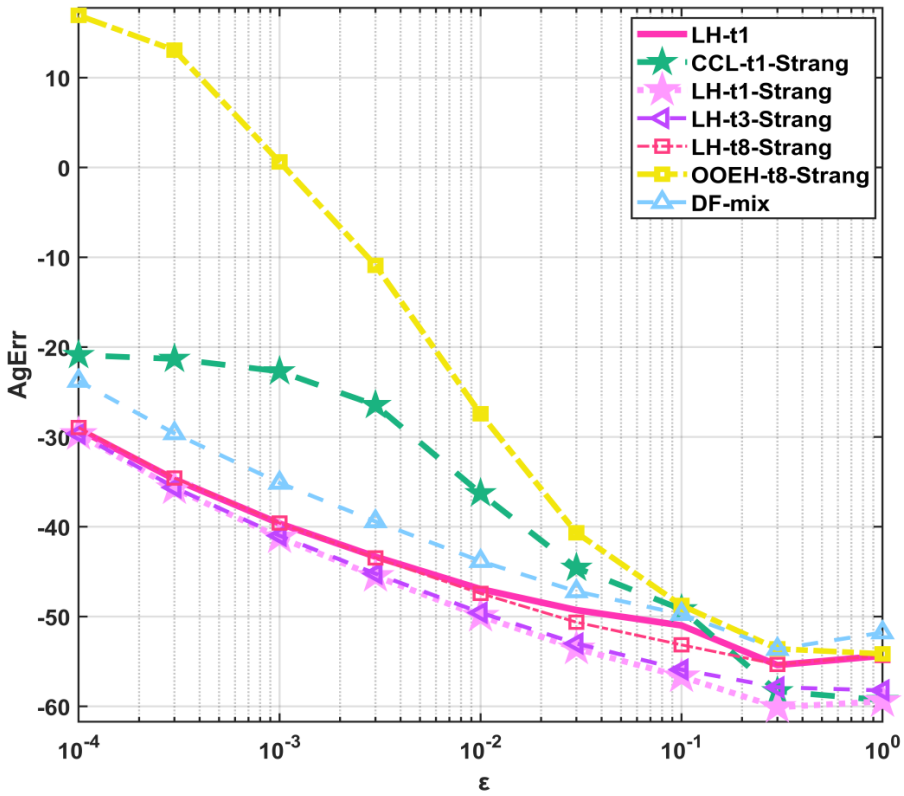


Figure 18. Aggregated errors (AgErr) as a function of the ε in the case of Experiment 9 for the top 7 methods.

Table 8. Aggregated errors (*AgErr*) for different (ε) values of the top 7 methods.

Methods	**(ε) Values**								
	1	0.3	0.1	0.03	0.01	0.003	0.001	0.0003	0.0001
	CFL limit								
	2.5×10^{-3}	2.3×10^{-2}	2.5×10^{-1}	2.3×10^{-4}	2.5×10^{-5}	2.3×10^{-6}	2.5×10^{-7}	2.3×10^{-8}	2.5×10^{-9}
	AgErr								
LH-t1	−54.37	−55.40	−50.99	−49.28	−46.94	−43.33	−39.64	−34.71	−29.14
CCL-t1-St	−59.30	−58.32	−49.19	−44.57	−36.27	−26.45	−22.70	−21.31	−20.89
LH-t1-St	−59.44	−60.09	−56.68	−53.58	−49.93	−45.50	−41.19	−35.74	−29.70
LH-t3-St	−58.22	−57.87	−55.90	−53.00	−49.56	−45.21	−41.00	−35.63	−29.64
LH-t8-St	−54.33	−55.33	−53.15	−50.63	−47.42	−43.46	−39.59	−34.57	−28.94
OOEH-t3-St	−54.15	−53.59	−48.76	−40.67	−27.42	−10.89	0.62	13.06	16.93
DF-mix	−51.78	−53.62	−49.72	−47.20	−43.83	−39.40	−35.12	−29.60	−23.77

7.2. Experiment 10: Sweep for the Stiffness Ratio

Now, in this section, we can add a new parameter γ to generate a wide scale of random values for the C and R quantities as follows:

$$C_i = 10^{\gamma(1-2rand)}, \quad R_{x,i} = 10^{\gamma(1-2rand)}, \quad R_{y,i} = 10^{\gamma(1-2rand)},$$

where $\gamma \in \{0, 0.33, 0.66, 1, 1.33, 1.66, 2, 2.33, 2.66, 3, 3.33\}$. The used final time is $t^{\text{fin}} = 0.4$, the system size is $N_x = N_y = 21$, and $S = 13$, while $\beta = 3.7$. The initial concentration function is as follows: $u_i^0 = \frac{1}{i+1} + 0.4 \cdot rand$. The aggregated errors as a function of the stiffness ratio in the case of Experiment 10 for the top seven methods are shown in Figure 19 and Table 9.

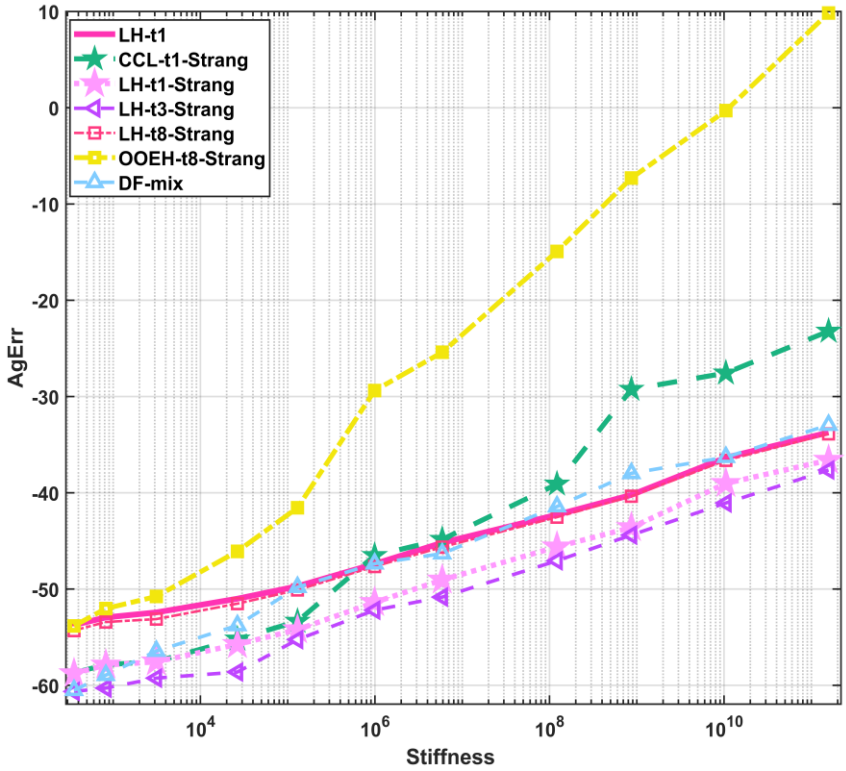


Figure 19. Aggregated errors (*AgErr*) as a function of the stiffness ratio (*SR*) in the case of Experiment 10 for the top 7 methods.

Table 9. Stiffness ratio (*SR*), CFL limit (*CFL*), and aggregated error (*AgErr*) values for different values of gamma (γ) and different treatments of the top 7 methods.

Methods	**Gamma (γ) values**										
	0	0.33	0.66	1	1.33	1.66	2	2.33	2.66	3	3.33
	Stiffness ratio										
	356.12	824.31	3122.74	2.6×10^4	1.3×10^5	9.9×10^5	5.8×10^6	1.2×10^8	8.67×10^8	1×10^{10}	1.57×10^{11}
	CFL limit										
	0.25	0.11	0.043	0.008	0.0029	0.00077	0.00016	3.3×10^{-5}	1×10^{-5}	1.88×10^{-6}	3.83×10^{-7}
	AgErr Errors										
LH-t_1	−53.72	−52.94	−52.42	−50.99	−49.70	−47.38	−45.18	−42.28	−40.22	−36.34	−33.73
CCL-t_1-St	−58.69	−57.86	−57.51	−55.19	−53.35	−46.52	−44.84	−39.10	−29.25	−27.56	−23.21
LH-t_1-St	−58.78	−57.88	−57.55	−55.74	−54.21	−51.38	−49.05	−45.61	−43.60	−39.02	−36.61
LH-t_3-St	−60.63	−60.28	−59.25	−58.61	−55.25	−52.18	−50.83	−47.05	−44.38	−41.05	−37.63
LH-t_8-St	−54.30	−53.42	−53.11	−51.51	−50.09	−47.65	−45.63	−42.52	−40.36	−36.62	−33.87
OOEH-t_3-St	−53.83	−52.02	−50.77	−46.08	−41.55	−29.38	−25.40	−14.93	−7.30	−0.29	9.83
DF-mix	−60.50	−58.94	−56.44	−53.77	−49.79	−47.36	−46.31	−41.42	−37.97	−36.29	−32.96

This experiment reinforces that it is generally true that the performance of the OOEH algorithm declines much more severely with a decreasing CFL limit and increasing stiffness

than the other methods. From these two experiments, one could conclude that this statement, albeit to a much lower extent, is true for the CCL method. However, this would be a hasty judgement since, in Experiment 3, increased stiffness resulted in a relative advantage for the CCL method.

7.3. Experiment 11: Sweep for the Nonlinear Coefficient β

In this case, we use the coefficient of the Huxley term $\beta \in \{0, 0.15, 0.4, 1, 3, 8, 14, 27, 40, 64, 90, 125\}$, where the system size and final time are $N_x = N_y = 21$, $t^{\text{fin}} = 0.2$, and $S = 14$, while the exponents are $a_C = 1$, $b_C = 2$, $a_{Rx} = 1$, $b_{Rx} = 2$, $a_{Rz} = 1$, and $b_{Rz} = 2$. The initial concentration function is given as $u_i^0 = \frac{1}{(i+2)} + 0.3 \cdot rand$. In this experiment, however, a problem arises when the value of β is large. In these cases, the values of the u function are forced to approach unity very closely due to the Huxley term, which is always positive for u values in the unit interval. Hence, a method can be *accidentally* accurate if it yields the $u_i^n \equiv 1$ function, which is the only equilibrium point of the equation. This distorts the results and the evaluation of the performance of the methods. To avoid this, we change the boundary conditions in two points of the system to $u_1 = u_N = 0$. This means that we apply fixed Dirichlet boundary conditions, but only for these two cells; the other boundaries are still subjected to zero-Neumann b.c.-s. This is implemented by modifying the loop for the cells from *i = 1:N* to *i = 2:(N-1)*.

In Figure 20 and Table 10, we present the aggregated errors as a function of the third root of the β parameter for the top seven methods. One can see that the initially significant differences between the methods almost vanish with increasing β, and two groups of the methods appear. The group with the smaller error always uses the t1-Strang or t3-Strang treatments, and the rest of the treatments constitute the other group. Nevertheless, the difference between these two groups is quite small. One can also observe that the initial disadvantage of the OOEH method (which is due to the moderate stiffness of the problem) vanishes because, for large values of β, the treatment of the nonlinear term is much more important than that of the diffusion term.

Table 10. Aggregated errors (*AgErr*) for different β values of the top 7 methods.

Methods	**Nonlinear coefficient (β) values**											
	0	0.15	0.4	1	3	8	14	27	40	64	90	125
	AgErr											
LH-t1	−96.91	−86.67	−82.34	−76.83	−65.32	−52.89	−43.35	−32.93	−29.20	−30.15	−32.46	−31.94
CCL-t1-St	−110.79	−99.27	−92.40	−84.49	−71.00	−57.56	−48.05	−37.51	−33.78	−34.51	−36.98	−36.39
LH-t1-St	−96.91	−93.76	−89.49	−83.03	−70.55	−57.71	−48.19	−37.56	−33.87	−34.65	−37.15	−36.66
LH-t3-St	−60.06	−95.00	−91.58	−86.47	−73.94	−61.25	−50.83	−38.56	−34.48	−34.97	−37.36	−37.00
LH-t8-St	−96.91	−92.38	−87.17	−80.02	−66.69	−53.31	−43.72	−33.15	−29.43	−30.13	−32.54	−32.05
OOEH-t3-St	−70.26	−71.68	−71.64	−68.32	−60.09	−50.74	−42.82	−32.44	−29.03	−29.59	−31.98	−31.67
DF-mix	−88.96	−84.34	−80.97	−77.12	−69.43	−57.44	−46.28	−33.84	−29.50	−29.83	−31.86	−32.18

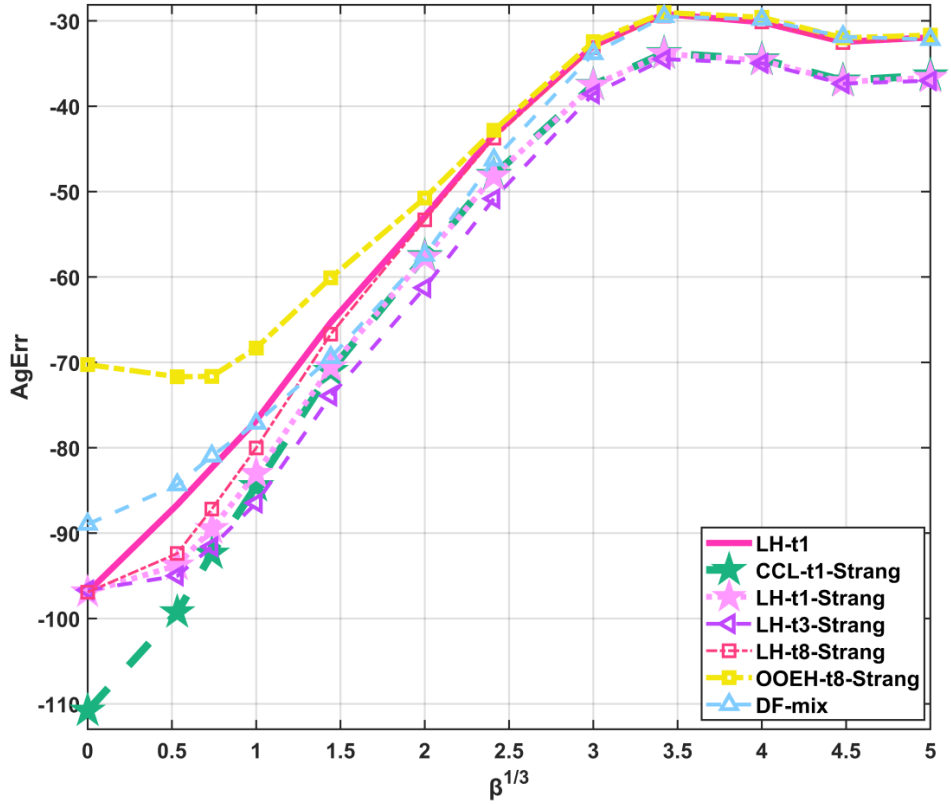


Figure 20. Aggregated errors (*AgErr*) as a function of parameter β in the case of Experiment 11 for the top 7 methods.

7.4. Experiment 12: Sweep for the Final Time T

In this case, we gradually increase the final time, i.e., $t^{\text{fin}} \in \{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100\}$, while the system with all of its parameters is kept fixed. The used system size is $N_x = N_y = 21$, $S = 13$, while $\beta = 4.3$. The exponents are $a_C = a_{Rx} = a_{Rz} = 1$, $b_C = b_{Rx} = b_{Rz} = 2$. The initial concentration function is given as:

$$u_i^0 = 0.4 \cdot \left(\cos\left(\frac{2\pi x}{N_x}\right) \cdot \cos\left(\frac{\pi y}{N_y}\right) + 1 \right) + 0.2 \cdot rand.$$

Since for large final times, the u values become very close to 1, as in the previous example for large values of β, we apply the same boundary conditions as in Experiment 11. Moreover, the series of time step sizes start from $T/4$; thus, the set of the used time step sizes is shifted with the final times. This choice, which is made to avoid extremely long running times, yields gradually increasing aggregated errors for most of the methods. Figure 21 and Table 11 show the aggregated errors as a function of the final time T in the case of Experiment 12 for the top seven methods. One can see that when T exceeds 1, there is a temporary decrease in the aggregated errors. This is because, by this time, the simulated system approached the stationary state very closely, and, therefore, the fine differences in the transient simulation by the different methods decay. The maximum errors as a function of time step size for T = 100 in the case of Experiment 13 for the top seven methods are presented in the Supplementary Materials (see Figure S9). One can conclude that the DF scheme has a relative advantage when the task is to approach accurately the stationary states.

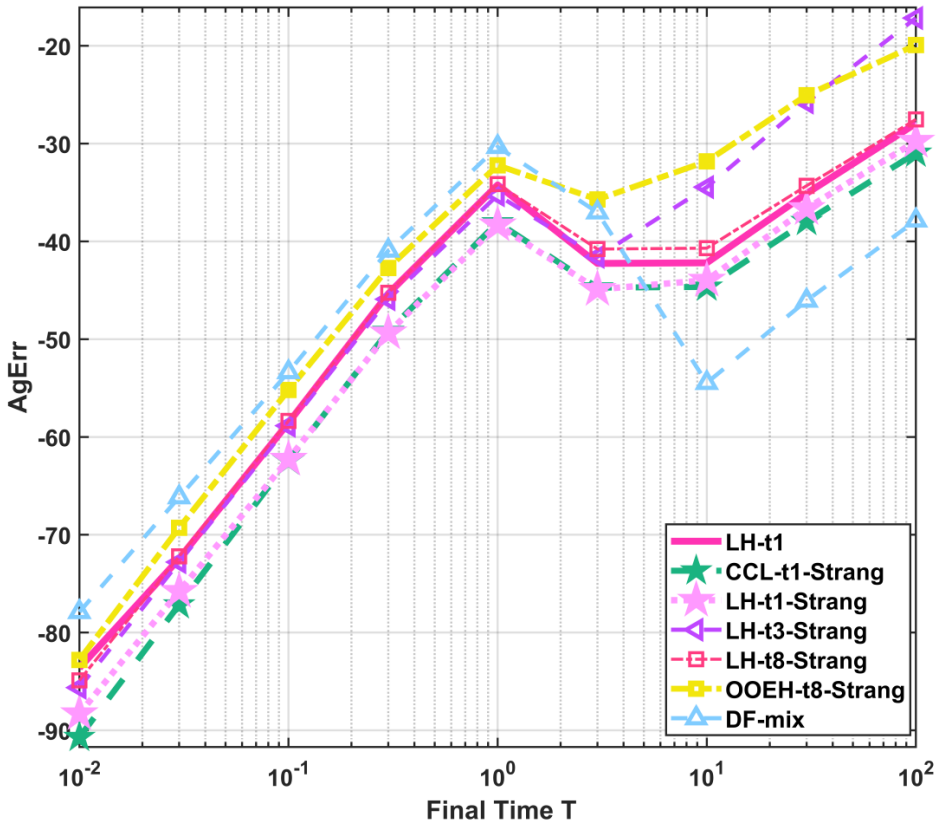


Figure 21. Aggregated errors (*AgErr*) as a function of the time in the case of Experiment 12 for the top 7 methods.

Table 11. Aggregated errors (*AgErr*) for different time values of the top 7 methods.

Methods	Time values								
	0.01	0.03	0.1	0.3	1	3	10	30	100
	AgErr								
LH-t1	−83.62	−72.45	−58.54	−45.34	−34.22	−42.25	−42.19	−35.09	−27.93
CCL-t1-St	−90.71	−77.15	−62.42	−49.17	−38.07	−44.72	−44.65	−37.94	−30.95
LH-t1-St	−88.29	−75.78	−62.27	−49.34	−38.31	−44.91	−43.96	−36.63	−29.75
LH-t3-St	−85.63	−72.79	−58.84	−45.91	−35.26	−41.57	−34.44	−25.78	−17.15
LH-t8-St	−84.891	−72.21	−58.36	−45.25	−34.13	−40.78	−40.69	−34.31	−27.51
OOEH-t3-St	−82.79	−69.28	−55.19	−42.71	−32.21	−35.71	−31.81	−25.05	−19.92
DF-mix	−77.89	−66.16	−53.36	−40.93	−30.33	−37.02	−54.47	−46.05	−37.85

7.5. Experiment 13: Sweep for the Initial Function Wavelength

The parameters of this experiment are the following: $N_x = N_y = 21$, $t^{\text{fin}} = 0.4$, $S = 15$, $\beta = 5.2$, $a_C = 1$, $b_C = 2$, $a_{Rx} = 1$, $b_{Rx} = 2$, $a_{Rz} = 1$, and $b_{Rz} = 2$. The initial concentration function is a sum of the components. Here, and for the rest of this paper, we return back to fully zero-Neumann b. c.-s. The first component with weight w is a smooth long-wave cosine function, while the second component with weight 1-w is the shortest possible wavelength function:

$$u_i^0 = w \cdot \left(\cos\left(\frac{2\pi x}{N_x}\right) \cdot \cos\left(\frac{\pi y}{N_y}\right) + 1 \right) + (1 - w) \cdot \left(0.5 \cdot (-1)^i + 1 \right).$$

where $w \in \{0,\ 0.05,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 0.95,\ 1\}$.

Figure 22 and Table 12 show the aggregated errors as a function of w in the case of Experiment 12 for the top seven methods. The maximum errors as a function of the time step size for $w = 1$ in the case of Experiment 13 for the top seven methods are presented in the Supplementary Materials (see Figure S10). This experiment shows that the accuracy of the methods only slightly depends on the initial function.

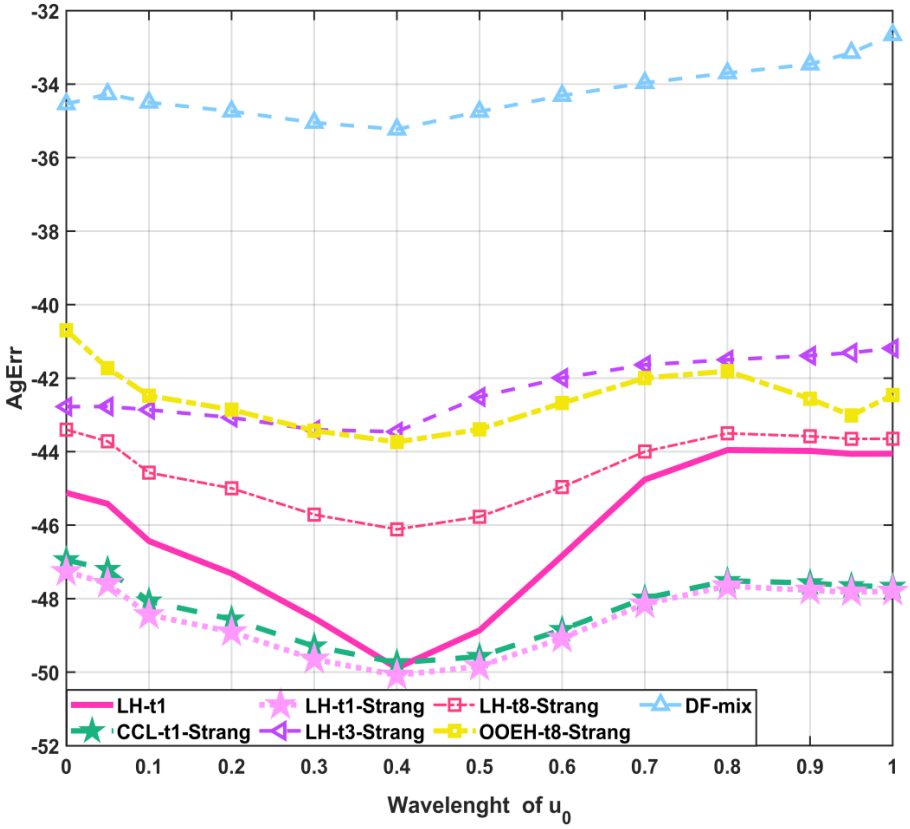


Figure 22. Aggregated errors (*AgErr*) as a function of the wavelength in the case of Experiment 13 for the top 7 methods.

Table 12. Aggregated errors (*AgErr*) for different wavelength values of the top 7 methods.

Methods	**Wavelengths**												
	0	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
	AgErr												
LH-t1	−45.10	−45.40	−46.40	−47.30	−48.53	−49.92	−48.81	−46.80	−44.70	−43.94	−43.90	−44.12	−44.00
CCL-t1-St	−46.94	−47.21	−48.07	−48.50	−49.30	−49.74	−49.60	−48.80	−47.99	−47.50	−47.60	−47.60	−47.70
LH-t1-St	−47.26	−47.59	−48.44	−48.90	−49.65	−50.08	−49.84	−49.07	−48.14	−47.65	−47.77	−47.82	−47.80
LH-t3-St	−42.77	−42.77	−42.86	−43.07	−43.40	−43.45	−42.51	−41.99	−41.64	−41.50	−41.38	−41.30	−41.20
LH-t8-St	−43.40	−43.72	−44.57	−44.99	−45.70	−46.11	−45.77	−44.96	−43.99	−43.50	−43.58	−43.65	−43.60
OOEH-t3-St	−40.70	−41.70	−42.48	−42.80	−43.40	−43.70	−43.40	−42.67	−41.99	−41.81	−42.56	−43.01	−42.40
DF-mix	−34.54	−34.30	−34.50	−34.70	−35.00	−35.23	−34.74	−34.30	−33.90	−33.70	−33.46	−33.20	−32.60

7.6. Experiment 14: Sweep for the Wavelength of the β Function

In reality, the reactions can be influenced by the properties of the media, such as temperature, which can depend on space. Therefore, in this subsection, we examine the behavior of the numerical methods under circumstances where the nonlinear coefficient β is a function of space. We construct a long-wave and a short-wave random function, and

the $\beta(x,\ y)$ function is the weighted average of these different wavelength contributions as follows

$$\beta(x,\ y) = 5 \cdot \left(w \cdot \left(\cos\left(\frac{2\pi x}{N_x}\right) \cdot \cos\left(\frac{\pi y}{N_y}\right) + 1 \right) + (1-w) \cdot rand \right),$$

where the sweep goes through the parameter values $w \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1\}$, and the used system size and final time are $N_x = N_y = 21$, $t^{\text{fin}} = 0.4$, and $S = 15$. The exponents of C and R are $a_C = 1$, $b_C = 2$, $a_{Rx} = 1$, $b_{Rx} = 2$, $a_{Rz} = 1$, and $b_{Rz} = 2$. The initial concentration function is as follows: $u_i^0 = 0.25 \cdot \left(\cos\left(\frac{2\pi x}{N_x}\right) \cdot \cos\left(\frac{\pi y}{N_y}\right) + 1\right) + 0.3 \cdot rand$. Figure 23 and Table 13 show the aggregated errors as a function of the wavelength of the β function in the case of Experiment 14 for the top seven methods. The general trend is that the accuracy of the methods is lower when the nonlinear coefficient changes slowly in space. This is particularly true for the DF–mixed method.

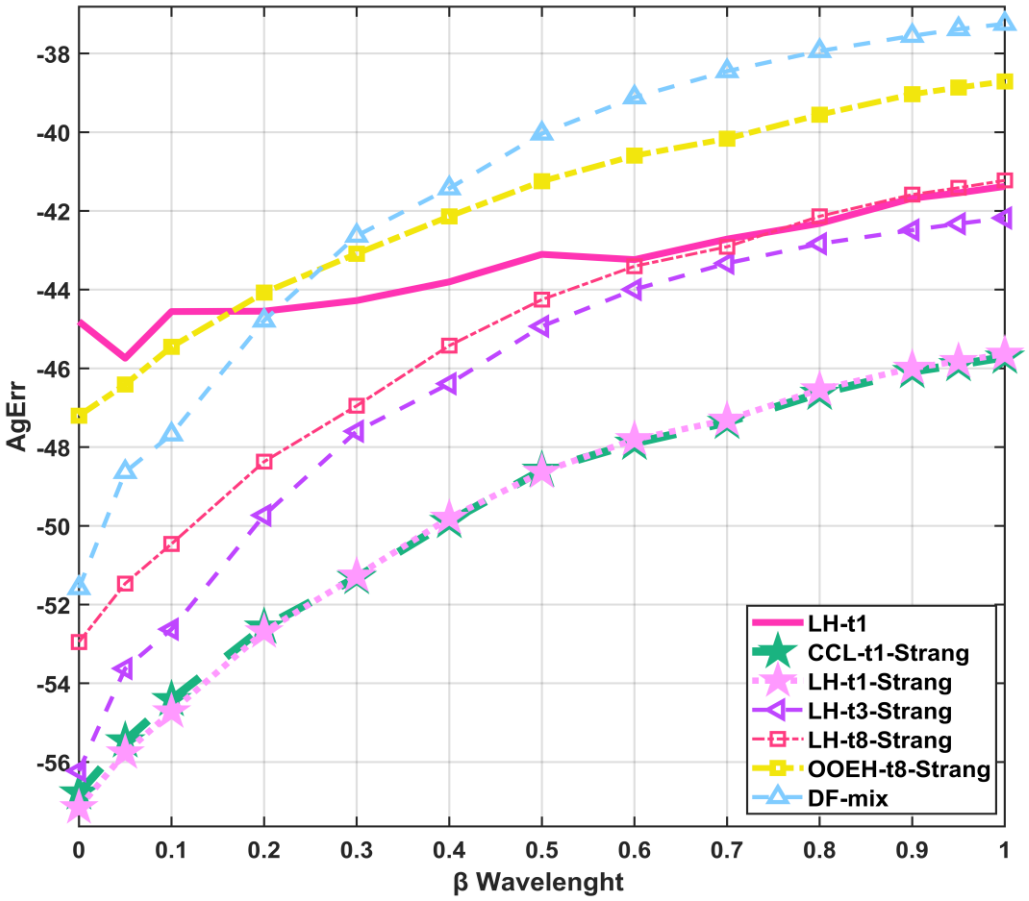


Figure 23. Aggregated errors (*AgErr*) as a function of the wavelength of the β function in the case of Experiment 14 for the top 7 methods.

Table 13. Aggregated errors (*AgErr*) for different values of the wavelength of the β function for the top 7 methods.

Methods	Beta (β) wavelengths												
	0	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
	AgErr												
LH-t1	−44.80	−45.74	−44.55	−44.54	−44.27	−43.80	−43.10	−43.23	−42.71	−42.32	−41.67	−41.54	−41.37
CCL-t1-St	−56.78	−55.46	−54.41	−52.57	−51.31	−49.88	−48.58	−47.90	−47.37	−46.63	−46.07	−45.90	−45.71
LH-t1-St	−57.15	−55.76	−54.72	−52.68	−51.27	−49.80	−48.63	−47.80	−47.30	−46.54	−45.99	−45.82	−45.63
LH-t3-St	−56.22	−53.62	−52.62	−49.73	−47.59	−46.38	−44.93	−43.99	−43.33	−42.83	−42.48	−42.32	−42.18
LH-t8-St	−52.95	−51.46	−50.45	−48.37	−46.94	−45.42	−44.25	−43.40	−42.91	−42.13	−41.58	−41.41	−41.22
OOEH-t3-St	−47.20	−46.40	−45.45	−44.07	−43.08	−42.13	−41.24	−40.59	−40.16	−39.55	−39.03	−38.86	−38.71
DF-mix	−51.58	−48.63	−47.67	−44.79	−42.63	−41.42	−40.03	−39.11	−38.45	−37.94	−37.55	−37.38	−37.25

8. Discussion and Conclusions

We performed extensive numerical tests to solve Huxley's equation. The goal was to explore the performance of some algorithms about which unconditional stability is analytically proven for the linear diffusion equation; thus, excellent stability properties for the nonlinear case were expected. For some, but not all, of the methods, these expectations are fulfilled well.

LH-t1-Strang: According to the numerical tests, the LH with the t1 treatment and Strang-splitting is generally the most efficient and reliable among the examined methods. It provides relatively accurate results quite quickly for all the examined sets of parameters, and no signs of instability are detected, even for extremely stiff systems of strong nonlinearity. Hence, non-strict time step size restrictions are necessary only to reach the desired accuracy, but not for stability considerations.

LH-t1: This combination is usually slightly less accurate than LH-t1 with Strang-splitting. It can be recommended when the diffusion part of the problem is much harder to solve than the reaction part, e.g., due to small β or large stiffness.

LH-t3-Strang: This combination can also give very accurate results, and sometimes it is the most efficient. However, in some cases, for example, if the simulated time interval is long, it is not so reliable.

LH-t8-Strang: Usually slightly less accurate and efficient than the LH-t1-Strang and LH-t3-Strang combinations. If, however, the geometry of the physical system is complicated such that the construction of a bipartite mesh is hardly feasible, the hopscotch-type methods are contraindicated.

OOEH-t8-Strang: The original odd–even hopscotch method can be proposed only for an equidistant mesh with a constant diffusion parameter. In these cases, this method is simpler to code than the LH scheme, and its accuracy is roughly the same.

CCL-t1-Strang: This combination is the opposite of the OOEH method from the point of view that it has a relative disadvantage in the case of a physically homogeneous system with an equidistant mesh. It is relatively slow but reliable for stiff problems. It should be used only if the geometry is complicated and the odd–even division of the cells faces difficulties.

DF–mixed: This combination is much faster and may be beneficially used for complicated geometries. Its accuracy strongly fluctuates; thus, it should be combined with an error estimator.

CrN: In the studied cases where the geometry is simple and the mesh is rectangular, the Crank–Nicolson method with operator-splitting has no advantage against the LH method. Its execution time strongly increases with the system size, so we advise using it

only when the number of nodes is small. However, in that case, the CrN method without operator-splitting, i.e., with standard Newton iterations, would be more accurate.

RK4: RK4 should be used if extreme accuracy is required; hence, the above-mentioned low-order methods are not favorable.

AB2: The performance of AB2 is quite similar to the RK4 method due to its similar conditional stability. It is a bit less accurate, but it is faster and requires less memory.

In the near future, we plan to compare the performance of our explicit methods with the following:

(a) Implicit methods (such as Crank–Nicolson) without operator-splitting;
(b) Semi-explicit, semi-implicit, and implicit–explicit schemes;
(c) Runge–Kutta–Chebyshev methods, which are explicit but have improved stability.

Our most important goal is, however, to extend these ideas to more complicated nonlinear diffusion–reaction PDEs (such as the FitzHugh–Nagumo equation) and systems of PDEs.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math13020207/s1, Figure S1: Maximum errors as a function of the time step size for different operator-splitting treatments in the case of Experiment 2; Figure S2: Maximum errors as a function of the time step size for different operator-splitting treatments with Strang-splitting in the case of Experiment 2; Figure S3: Maximum errors as a function of time step size for the remaining methods in the case of Experiment 2; Figure S4: Maximum errors as a function of the time step size for different operator-splitting treatments with Strang-splitting in the case of Experiment 3; Figure S5. Maximum errors as a function of the time step size for different operator-splitting treatments with Strang-splitting in the case of Experiment 4; Figure S6. Maximum errors as a function of time step size for the remaining methods in the case of Experiment 4; Figure S7. Maximum errors as a function of time step size for different operator-splitting treatments with Strang-splitting in the case of Experiment 5; Figure S8. Maximum errors as a function of the time step size for the remaining methods in the case of Experiment 5; Figure S9. Maximum errors as a function of the time step size in the case of Experiment 12 for the top 7 methods; Figure S10. Maximum errors as a function of the time step size in the case of Experiment 13 for the top 7 methods for w = 1; Table S1. The Aggregated error (AgErr) values for different treatment of the methods in case of Experiment 6; Table S2. The Aggregated error (AgErr) values for different treatment of the methods in case of Experiment 7; Table S3. The Aggregated errors (AgErr) for different treatments of the methods in case of Experiment 8.

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

Funding: This research was funded by the EKÖP-24-4-I, supported by the University Research Scholarship Program of the Ministry for Culture and Innovation from the source of the National Research, Development, and Innovation Fund.

Data Availability Statement: The data are available from the authors on reasonable request.

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

References

1. Bradshaw-Hajek, B. Reaction-Diffusion Equations for Population Genetics. Ph.D. Thesis, University of Wollongong, Wollongong, NSW, Australia, 2004.
2. Mohan, M.T.; Khan, A. On the generalized Burgers-Huxley equation: Existence, uniqueness, regularity, global attractors and numerical studies. *Discret. Contin. Dyn. Syst. B* **2021**, *26*, 3943–3988. [CrossRef]

3. 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]
4. Wang, X. Nerve propagation and wall in liquid crystals. *Phys. Lett. A* **1985**, *112*, 402–406. [CrossRef]
5. Sharma, S.; Prabhakar, N. Numerical Simulation of Generalized FitzHugh-Nagumo Equation by Shifted Chebyshev Spectral Collocation Method. *IAENG Int. J. Appl. Math.* **2024**, *54*, 1371–1382.
6. Lukonde, J.; Kasumo, C. On numerical and analytical solutions of the generalized Burgers-Fisher equation. *J. Innov. Appl. Math. Comput. Sci.* **2023**, *3*, 121–141.
7. Patel, Y.F.; Dhodiya, J.M. Efficient algorithm to study the class of Burger's Fisher equation. *Int. J. Appl. Nonlinear Sci.* **2022**, *3*, 242–266. [CrossRef]
8. Dhumal, M.; Sontakke, B. Solving Time-Fractional Fitzhugh–Nagumo Equation using Homotopy Perturbation Method. *Indian J. Sci. Technol.* **2024**, *17*, 1272–1282. [CrossRef]
9. Shi, J.; Yang, X.; Liu, X. A novel fractional physics-informed neural networks method for solving the time-fractional Huxley equation. *Neural Comput. Appl.* **2024**, *36*, 19097–19119. [CrossRef]
10. Bai, Y.; Chaolu, T.; Bilige, S. Solving Huxley equation using an improved PINN method. *Nonlinear Dyn.* **2021**, *105*, 3439–3450. [CrossRef]
11. Loyinmi, A.C.; Sanyaolu, M.D.; Gbodogbe, S. Exploring the Efficacy of the Weighted Average Method for Solving Nonlinear Partial Differential Equations: A Study on the Burger-Fisher Equation. *Educ. J. Sci. Math. Technol.* **2025**, *12*, 60–79. [CrossRef]
12. 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]
13. Jaglan, J.; Maurya, V.; Singh, A.; Yadav, V.S.; Rajpoot, M.K. Acoustic and soliton propagation using fully-discrete energy preserving partially implicit scheme in homogeneous and heterogeneous mediums. *Comput. Math. Appl.* **2024**, *174*, 379–396. [CrossRef]
14. March, N.G.; Carr, E.J.; Turner, I.W. Numerical Investigation into Coarse-Scale Models of Diffusion in Complex Heterogeneous Media. *Transp. Porous Media* **2021**, *139*, 467–489. [CrossRef]
15. Kondratenko, P.S.; Matveev, A.L.; Vasiliev, A.D. Numerical implementation of the asymptotic theory for classical diffusion in heterogeneous media. *Eur. Phys. J. B* **2021**, *94*, 50. [CrossRef]
16. Jejeniwa, O.A.; Gidey, H.H.; Appadu, A.R. Numerical Modeling of Pollutant Transport: Results and Optimal Parameters. *Symmetry* **2022**, *14*, 2616. [CrossRef]
17. Kumar, S.; Ramcharan, B.; Yadav, V.S.; Singh, A.; Rajpoot, M.K. Exploring Acoustic and Sound Wave Propagation Simulations: A Novel Time Advancement Method for Homogeneous and Heterogeneous Media. *Adv. Appl. Math. Mech.* **2024**, *17*, 315–349. [CrossRef]
18. 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]
19. Ketcheson, D.I. Highly Efficient Strong Stability-Preserving Runge–Kutta Methods with Low-Storage Implementations. *SIAM J. Sci. Comput.* **2008**, *30*, 2113–2136. [CrossRef]
20. Qin, X.; Jiang, Z.; Yan, C. Strong Stability Preserving Two-Derivative Two-Step Runge-Kutta Methods. *Mathematics* **2024**, *12*, 2465. [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. Bansal, S.; Natesan, S. A novel higher-order efficient computational method for pricing European and Asian options. *Numer. Algorithms* **2024**. [CrossRef]
23. Rieth, Á.; Kovács, R.; Fülöp, T. Implicit numerical schemes for generalized heat conduction equations. *Int. J. Heat Mass Transf.* **2018**, *126*, 1177–1182. [CrossRef]
24. 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]
25. 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]
26. 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]
27. Manaa, S.; Sabawi, M. Numerical Solution and Stability Analysis of Huxley Equation. *AL-Rafidain J. Comput. Sci. Math.* **2005**, *2*, 85–97. [CrossRef]
28. Macías-Díaz, J.E. On an exact numerical simulation of solitary-wave solutions of the Burgers–Huxley equation through Cardano's method. *BIT Numer. Math.* **2014**, *54*, 763–776. [CrossRef]
29. 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]

30. Appadu, A.R.; Inan, B.; Tijani, Y.O. Comparative study of some numerical methods for the Burgers-Huxley equation. *Symmetry* **2019**, *11*, 1333. [CrossRef]
31. Songolo, M.E. 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.
32. 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]
33. Liang, Z.; Yan, Y.; Cai, G. A Dufort-Frankel Difference Scheme for Two-Dimensional Sine-Gordon Equation. *Discret. Dyn. Nat. Soc.* **2014**, *2014*, 784387. [CrossRef]
34. Gasparin, S.; Berger, J.; Dutykh, D.; Mendes, N. Stable explicit schemes for simulation of nonlinear moisture transfer in porous materials. *J. Build. Perform. Simul.* **2018**, *11*, 129–144. [CrossRef]
35. Harley, C. Hopscotch method: The numerical solution of the Frank-Kamenetskii partial differential equation. *Appl. Math. Comput.* **2010**, *217*, 4065–4075. [CrossRef]
36. Nagy, Á.; Saleh, M.; Omle, I.; Kareem, H.; Kovács, E. New stable, explicit, shifted-hopscotch algorithms for the heat equation. *Math. Comput. Appl.* **2021**, *26*, 61. [CrossRef]
37. 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]
38. Hirsch, C. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*; Wiley: Hoboken, NJ, USA, 1988.
39. 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]
40. 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]
41. Askar, A.H.; Omle, I.; Kovács, E.; Majár, J. Testing Some Different Implementations of Heat Convection and Radiation in the Leapfrog-Hopscotch Algorithm. *Algorithms* **2022**, *15*, 400. [CrossRef]
42. Chapra, S.C.; Canale, R.P. *Numerical Methods for Engineers*, 7th ed.; McGraw-Hill Science/Engineering/Math: New York, NY, USA, 2015.
43. Linear Multistep Method. *Wikipedia*. 23 September 2024. Available online: https://en.wikipedia.org/w/index.php?title=Linear_multistep_method&oldid=1247156850 (accessed on 1 January 2025).

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.