

Article

Adaptive Iterative Splitting Methods for Convection-Diffusion-Reaction Equations

Jürgen Geiser ^{1,*} , Jose L. Hueso ² and Eulalia Martínez ²

¹ The Institute of Theoretical Electrical Engineering, Ruhr University of Bochum, Universitätsstrasse 150, D-44801 Bochum, Germany; juergen.geiser@ruhr-uni-bochum.de

² Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain; jlhueso@mat.upv.es (J.L.H.); eumarti@mat.upv.es (E.M.)

* Correspondence: juergen.geiser@ruhr-uni-bochum.de

Received: 8 January 2020; Accepted: 20 February 2020; Published: 25 February 2020



Abstract: This article proposes adaptive iterative splitting methods to solve Multiphysics problems, which are related to convection–diffusion–reaction equations. The splitting techniques are based on iterative splitting approaches with adaptive ideas. Based on shifting the time-steps with additional adaptive time-ranges, we could embedded the adaptive techniques into the splitting approach. The numerical analysis of the adapted iterative splitting schemes is considered and we develop the underlying error estimates for the application of the adaptive schemes. The performance of the method with respect to the accuracy and the acceleration is evaluated in different numerical experiments. We test the benefits of the adaptive splitting approach on highly nonlinear Burgers’ and Maxwell–Stefan diffusion equations.

Keywords: time adaptive integration; adaptive iterative splitting; operator-splitting method; error control; convection–diffusion–reaction equations; iterative solver method; nonlinear equations

MSC: 35K45, 35K90, 47D60, 65M06, 65M55

1. Introduction

In this paper, we propose adaptive splitting schemes to solve nonlinear differential equations. We consider spatially discretized convection–diffusion–reaction equations, which we could apply as semi-discretized nonlinear systems of ordinary differential equations. Based on the nonlinearities, it is important to deal with adaptive schemes, whereas we can control the local errors of the underlying schemes, see [1–4]. In general, the splitting methods have local splitting errors, which can be controlled with the time- or spatial steps of the underlying schemes, see [1,2,5,6].

In this work, we consider time-splitting methods, and here, we distinguish between

- Non-iterative methods, e.g., Lie–Trotter, see [7]; Strang-splitting methods, see [8]; or exponential splitting schemes, see [1,9].
- Iterative methods, e.g., iterative splitting methods, see [3,10], or Waveform-Relaxation methods, see, e.g., [11,12]

For the non-iterative methods, e.g., Lie–Trotter and Strang-splitting schemes, first works exist and discuss the ideas of the adaptive time splitting methods, see [6,8]. Although for the iterative methods only some first ideas exist, see [13], based on dealing with different time-step approaches, our new contributions

are based on the novel strategy of ϵ -shifting, see [14], of the underlying splitting method. We obtain so-called shifted iterative splitting methods, which can be compared with the standard iterative splitting methods, i.e., without the shifting. The local error estimate can be computed with the solution of the shifted and non-shifted iterative splitting approaches. Then, the effective error control is given as a function of the local error estimates and an error-tolerance parameter.

Such a novel adaptive splitting approach is important to reduce the computational time, while we decompose into sub-operators, which can be simulated with faster and more accurate numerical schemes. Further, the novel approach can be optimized with an effective and maximal splitting time step, see also [5,14]. We combine these two ideas of adaptivity and splitting into a novel strategy to control and reduce the local splitting error for the iterative splitting methods. Then, we can obtain more accurate results with a maximal time step and reduce the computational cost.

In this paper, we present the novel adaptive splitting techniques as follows.

- In the first step, we consider the standard splitting techniques with the underlying error analysis, see [3], and
- in the second step, we introduce the adaptive techniques, which is based on the ϵ -shift technique, see [14], such that we can control the local splitting error.

Then, an error-estimate is computed, such that we are allowed to evaluate a maximum splitting time-step with respect to the shifted and non-shifted iterative splitting approach. The analysis is based on the standard iterative splitting approaches, see [13,15], and the additional adaptive techniques, see [5,6]. In the numerical applications, which are based on convection–diffusion–reaction equations, we present the verification and the benefits of the novel adaptive iterative splitting approaches.

The paper is outlined as following. In Section 2, we explain the adaptive splitting approaches. Further, in Section 3, we discuss the error analysis of the adaptive splitting schemes. The applications to different convection–diffusion–reaction equations are done in Section 4. In Section 5, we discuss the theoretical and practical results.

2. Adaptive Splitting Approaches

We take inspiration for our studies, which are presented below, from real-life simulations of nonlinear convection–diffusion–reaction equations with the help of splitting approaches, see [3,16–18].

We deal with convection–diffusion–reaction equations, which can be written as

$$\partial_t u(t) + \partial_x v(u) - \partial_x (D(u) \partial_x u) = f(u(t)), \quad x \in \mathbb{R}^d, t > 0, \quad (1)$$

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}^d, t = 0, \quad (2)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the reaction term; $u : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^n$ is the solution; and $D(u)$ is the diffusion matrix, which is a tensor of order $d \times d \times n$ and the velocity vector, which is of order $d \times n$.

In the present paper, we apply spatial discretization methods, such that we consider the spatial discretized partial differential equations with the included boundary conditions, which are given as ordinary differential equations:

$$\partial_t u(t) = A(u(t)) u(t) + B(u(t)) u(t), \quad t \in (0, T), \quad (3)$$

where $u(0) = u_0$ is the initial condition. $A(u)$ and $B(u)$ are operators, which are spatially discretized. For example, $A(u)$ is the discretized in space convection operator and $B(u)$ is the discretized in space diffusion operators. For convenience, the nonlinear operators are bounded, e.g., bounded matrices.

In our proposed scheme, we embed the idea of the ϵ -shift, which is explained in [14], to the iterative splitting methods, see [3]. Then, we obtain a new so called ϵ -shifted iterative splitting method, which is a new contribution. Such shifted iterative splitting approach are used to design new adaptive time-splitting methods, which can be applied with maximal time-steps and compute the error-controls of the local splitting-errors.

In the following, we discuss the standard and the shifted splitting approaches.

2.1. Standard Splitting Approaches

In this section, we describe the standard splitting methods.

We deal with two splitting schemes:

- Non-iterative splitting scheme (Strang splitting), see [8].
- Iterative splitting scheme (fixpoint scheme), see [3].

2.1.1. Strang-Marchuk Splitting (SMS)

In the SMS method, in the first step, the operator A is solved in the left half of interval $[t^n, t^{n+1}]$; then, in the second step, the operator B is solved in the whole interval $[t^n, t^{n+1}]$; and in the third step, the operator A is solved in the right half of the interval $[t^n, t^{n+1}]$. By the initial conditions, the three subproblems are connected, see

$$\begin{aligned}\frac{d\tilde{u}(t)}{dt} &= A(\tilde{u}(t))\tilde{u}(t), \text{ with } \tilde{u}(t^n) = u(t^n), \text{ and } t \in [t^n, t^n + \tau/2], \\ \frac{d\tilde{\tilde{u}}(t)}{dt} &= B(\tilde{\tilde{u}}(t))\tilde{\tilde{u}}(t), \text{ with } \tilde{\tilde{u}}(t^n) = \tilde{u}(t^n + \tau/2), \text{ and } t \in [t^n, t^{n+1}], \\ \frac{du(t)}{dt} &= A(u(t))u(t), \text{ with } u(t^n + \tau/2) = \tilde{\tilde{u}}(t^{n+1}) \text{ and } t \in [t^n + \tau/2, t^{n+1}],\end{aligned}\quad (4)$$

where $\tau = \Delta(t^n) = t^{n+1} - t^n$ is the time step.

2.1.2. Iterative Splitting Methods

The iterative splitting method defined in [13] and extensively studied for ordinary and partial differential equations in [3,10,15,19,20] are alternative operator splitting methods, which are based on iterative techniques.

We apply two versions of the iterative splitting methods:

- Linear Iterative Splitting (LIS)

The LIS solves in the first equation the linear part of operator A with the given right hand side of operator B . Then, it solves in the second equation the linear part of operator B with the right hand side of operator A , using the solution of the first equation. The two solver steps are iterated m times before we pass to the next interval.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n), \\ \frac{du_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(\tilde{u}_i(t))u_i(t), \text{ with } u_i(t^n) = u(t^n),\end{aligned}\quad (5)$$

where $i = 1, 2, \dots, m$. For the initialisation of the iteration, we start with function $u_0(t)$, which verifies the initial condition $u_0(0) = u_0$. After, we have performed m iterations of the LIS, we apply the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$ for the next time-step, till the final step $n + 1 = N$.

- Quasilinear Iterative Splitting (QIS)

The QIS solves in the first equation the nonlinear part of operator A with the given right hand side of operator B . Then, it solves in the second equation the nonlinear part of operator B with the right hand side of operator A , using the solution of the first equation. The two solver steps are iterated m times before we pass to the next interval.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n), \\ \frac{du_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_i(t))u_i(t), \text{ with } u_i(t^n) = u(t^n),\end{aligned}$$

where $i = 1, 2, \dots, m$. For the initialisation of the iteration, we start with function $u_0(t)$, which verifies the initial condition $u_0(0) = u_0$. After, we have performed m iterations of the QIS, we apply the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$ for the next time-step, till the final step $n + 1 = N$.

2.2. Shifted Splitting Approaches for Error Estimations

In this section, we describe the modified splitting methods to apply error-estimates. We propose shifted splitting as a novel method to design error estimates, see [14].

We deal with two shifted splitting schemes:

- Non-iterative splitting scheme (Strang splitting), see [14].
- Iterative splitting scheme (fixpoint scheme), see [3].

2.2.1. Shifted Strang-Marchuk Splitting (SSMS)

In this method, in interval $[t^n, t^{n+1}]$, we first solve for operator A with a half time step minus a small ϵ , means $\tau/2 - \epsilon$, then we solve B with the full time-step τ , and again for A with a half time step plus a small ϵ , means $\tau/2 + \epsilon$. The three subproblems are connected by the initial conditions, according to

$$\begin{aligned}\frac{d\tilde{u}(t)}{dt} &= A(\tilde{u}(t))\tilde{u}(t), \text{ with } \tilde{u}(t^n) = u(t^n) \text{ and } t \in [t^n, t^n + \tau/2 - \epsilon], \text{ time-step } \tau/2 - \epsilon, \\ \frac{d\tilde{\tilde{u}}(t)}{dt} &= B(\tilde{\tilde{u}}(t))\tilde{\tilde{u}}(t), \text{ with } \tilde{\tilde{u}}(t^n) = \tilde{u}(t^n + \tau/2 - \epsilon), \text{ and } t \in [t^n, t^{n+1}], \text{ time-step } \tau, \\ \frac{du(t)}{dt} &= A(u(t))u(t), \text{ with } u(t^n + \tau/2 - \epsilon) = \tilde{\tilde{u}}(t^{n+1}) \text{ and } t \in [t^n + \tau/2 - \epsilon, t^{n+1}], \\ &\text{time-step } \tau/2 + \epsilon.\end{aligned}\tag{6}$$

The ϵ value is a small fraction of τ , for example $\epsilon = 0.005\tau$, so that the shifted interval is close to the original one. The error estimate is given as

$$err = ||u_{Strang}(t^{n+1}) - u_{Strang,\epsilon}(t^{n+1})||.\tag{7}$$

If the error estimate err is higher than a given tolerance η , we redo the computations with a smaller step according to the *refinement* scheme

$$\Delta t_{new} = \nu \Delta t \sqrt{\frac{\eta}{err}},\tag{8}$$

where we apply $\nu > 0$, near 1 as a security factor. Otherwise, if $err \leq \eta$, we accept the obtained value $u_{Strang}(t^{n+1})$, and proceed with the next time interval. In this case, in order to avoid the usage of unnecessary small time steps, we apply a *coarsening* scheme

$$\Delta t_{new} = (1 + \kappa)\Delta t, \quad (9)$$

where κ is a small positive value depending on the tolerance η .

The Algorithm is given in Algorithm 1:

Algorithm 1.

1. Compute the local time-steps with the Strang and shifted Strang method, means $u_{strang}(t^{n+1})$ and $u_{strang,\epsilon}(t^{n+1})$.
2. Compute the error estimation according to (7).
3. If $err > \eta$, reject the time-step and restart the recent time-interval with the $\Delta t_n = \Delta t_{new}$ obtained from (8).

If $err \leq \eta$, then we are in the error tolerance. Proceed with the next time interval with the increased time step $\Delta t_{n+1} = \Delta t_{new}$ given by (9).

Remark 1. The theoretical results of the Algorithm 1, which is based on the ϵ -shifted Strang-splitting method, are given in the literature [4,5]. Here, the authors applied the shifted Strang-splitting method and could apply a local error estimate of the first and second splitting resolution, see [14].

In the Figure 1, we have the graphically introduction of the shifting ideas.

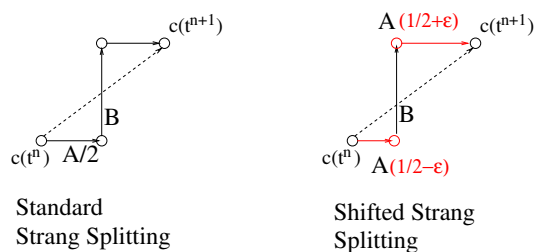


Figure 1. Standard Strang splitting and shifted Strang splitting method.

2.2.2. Shifted Iterative Splitting Methods

In this section, we apply the shifting time-step ideas to the iterative splitting methods. First, we solve for the first iterative step with time-step $\tau - \epsilon$, then we solve for the second iterative step with time-step $\tau + \epsilon$.

We modify the two versions of the iterative splitting methods:

- Shifted Linear Iterative Splitting (SLIS)

The SLIS solves in the first equation the linearized part of operator A with the given right hand side of operator B for a minus shift of ϵ in the time-step. Then, it solves the second equation the linearized part of operator B with a given right hand side of operator A for a plus shift of ϵ in the time-step,

using the solution of the first equation. The two solver steps are iterated m times before we pass to the next interval.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n), \text{ time-step } \tau - \epsilon, \\ \frac{du_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(\tilde{u}_i(t))u_i(t), \text{ with } u_i(t^n) = u(t^n), \text{ time-step } \tau + \epsilon,\end{aligned}\quad (10)$$

where $i = 1, 2, \dots, m$. For the initialization of the iteration, we start with function $u_0(t)$, which verifies the initial condition $u_0(0) = u_0$. After we have performed m iterations of the SLIS, we apply the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$ for the next time-step, until the final step $n + 1 = N$.

We apply the error-estimates as in Algorithm 2 and then we go to the next time-step.

Here, we decided $i = 1$, but the error estimates also work for $i = 1, 2, \dots, m$.

- Shifted Quasilinear Iterative Splitting (SQIS)

The SQIS solves in the first equation the nonlinear part of operator A with the given right hand side of operator B for a minus shift of ϵ in the time-step. Then, it solves the second equation the nonlinear part of operator B with a given right hand side of operator A for a plus shift of ϵ in the time-step, using the solution of the first equation. The two solver steps are iterated m times before we pass to the next interval.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n), \text{ time-step } \tau - \epsilon, \\ \frac{du_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_i(t))u_i(t), \text{ with } u_i(t^n) = u(t^n), \text{ time-step } \tau + \epsilon,\end{aligned}\quad (11)$$

where $i = 1, 2, \dots, m$. For the initialization of the iteration, we start with function $u_0(t)$, which verifies the initial condition $u_0(0) = u_0$. After we have performed m iterations of the SQIS, we apply the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$ for the next time-step, till the final step $n + 1 = N$.

We apply the error-estimates as in Algorithm 2 and then we go to the next time-step.

Here we decided $i = 1$, but the error estimates also works for $i = 1, 2, \dots, m$.

The error estimate is given as

$$err = ||u_i(t^{n+1}) - u_{i,\epsilon}(t^{n+1})|| \leq \eta, \quad (12)$$

whereas η is a given error tolerance, e.g., $\eta = 10^{-5}$.

Further the adaptive time-stepping is

$$\Delta t_{new} = \nu \Delta t \left(\frac{\eta}{err} \right)^{1/(2i)}, \quad (13)$$

where we apply $\nu > 0$, near 1 as a security factor.

The Algorithm is given in Algorithm 2:

Algorithm 2.

1. We compute the local time-steps with the iterative and shifted iterative method, means $u_i(t^{n+1})$ and $u_{i,\epsilon}(t^{n+1})$

2. We compute the error estimation

$$err = \|u_i(t^{n+1}) - u_{i,\epsilon}(t^{n+1})\| \leq \eta, \quad (14)$$

3. If $err \leq \eta$, then we are in the error tolerance and we accept the time-step means $u(t^{n+1}) = u_i(t^{n+1})$ and the next time-step is $\Delta t_{n+1} = \Delta t_{new}$.

Otherwise, we reject the time-step and restarted the recent time-interval with $\Delta t_n = \Delta t_{new}$

In the Figure 2, we have the graphically introduction of the shifting ideas.

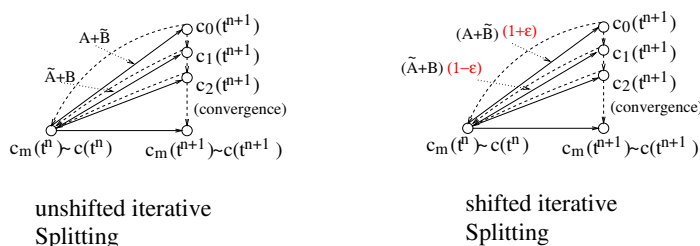


Figure 2. Unshifted iterative splitting and shifted iterative splitting method.

3. Error Analysis

The error analysis of the methods are done in the following.

We deal with the following assumptions to the nonlinear operators:

Assumption 1.

- Estimation of the nonlinear operators:

$$\|A(e_i(t))\| \leq \|\tilde{A}\|, \quad t^n \leq t \leq t^{n+1}, \quad i = 0, 1, \dots, I, \quad (15)$$

$$\|B(e_i(t))\| \leq \|\tilde{B}\|, \quad t^n \leq t \leq t^{n+1}, \quad i = 0, 1, \dots, I, \quad (16)$$

where $\tilde{A}$ and $\tilde{B}$ are bounded operators, such that the Taylor expansion of the operators can be applied.

- For the nonlinear operators A and B , we estimated the linearized parts as bounded operators $\tilde{A}, \tilde{B} : \mathbf{X} \rightarrow \mathbf{X}$, where $\mathbf{X}$ is an appropriate Banach space. Further, we have a Banach-norm for the vector and the matrices, which is given as $\|\cdot\|$.

In Theorem 1, we derive the consistency order of the shifted iterative operator-splitting.

Theorem 1. The operators $A, B \in \mathcal{L}(\mathbf{X})$ are nonlinear bounded operators with the Assumption 1. The abstract Cauchy problem is given as

$$\partial_t c(t) = A(c(t))c(t) + B(c(t))c(t), \quad 0 < t \leq T, \quad (17)$$

$$c(0) = c_0.$$

The abstract Cauchy problem (17) has an existent and unique solution. Then, the shifted iterative splitting method (11) is consistent with the order of the consistency $\mathcal{O}(\tau_n^{2i})$ with $i = 1, \dots, m$.

Proof. We assume $\tilde{A} + \tilde{B} \in \mathcal{L}(\mathbf{X})$ and we assume that the linear operators are generator of a uniformly continuous semigroup, such that we have a unique solution $c(t) = \exp((\tilde{A} + \tilde{B})t)c_0$.

In the following, we consider the local time-interval $[t^n, t^{n+1}]$.

$e_i(t) = c(t) - \tilde{c}_i(t)$ and $e_{i+1}(t) = c(t) - c_i(t)$ are the local error functions.

The error functions for the shifted time-intervals are computed as

$$\begin{aligned}\partial_t e_i(t) &= \tilde{A}e_i(t) + \tilde{B}e_{i-1}(t), \quad t \in (t^n, t^{n+1} - \epsilon], \\ e_i(t^n) &= 0,\end{aligned}\tag{18}$$

and

$$\begin{aligned}\partial_t e_{i+1}(t) &= \tilde{A}e_i(t) + \tilde{B}e_{i+1}(t), \quad t \in (t^n, t^{n+1} + \epsilon], \\ e_{i+1}(t^n) &= 0,\end{aligned}\tag{19}$$

for $i = 1, 3, 5, \dots$, with $e_1(0) = 0$ and $e_0(t) = c(t)$.

Based on the Assumptions 1, we can assume that the linearized operators $\tilde{A}$ and $\tilde{B}$ are generators of the one-parameter C_0 semigroup, which are given as $(\exp(\tilde{A}(t)))_{t \geq 0}$ and $(\exp(\tilde{B}(t)))_{t \geq 0}$.

In the following, we can write the abstract Cauchy problem with homogeneous initial conditions as

$$\begin{aligned}e_i(t) &= \int_{t^n}^t \exp(\tilde{A}(t-s)) \tilde{B}e_{i-1}(s) ds, \quad t \in [t^n, t^{n+1} - \epsilon], \\ e_{i+1}(t) &= \int_{t^n}^t \exp(\tilde{B}(t-s)) \tilde{A}e_{i+1}(s) ds, \quad t \in [t^n, t^{n+1} + \epsilon],\end{aligned}\tag{20}$$

We apply the norms for the vectors and matrices and we can estimate

$$\begin{aligned}\|e_i(t)\| &\leq \|\tilde{B}\| \|e_{i-1}\| \int_{t^n}^t \|\exp(\tilde{A}(t-s))\| ds, \quad t \in [t^n, t^{n+1} - \epsilon], \\ \|e_{i+1}(t)\| &\leq \|\tilde{A}\| \|e_i\| \int_{t^n}^t \|\exp(\tilde{B}(t-s))\| ds, \quad t \in [t^n, t^{n+1} + \epsilon],\end{aligned}\tag{21}$$

We assume, that $(\tilde{A}(t))_{t \geq 0}$ and $(\tilde{B}(t))_{t \geq 0}$ are generators of semigroups and we apply the so-called *growth estimation*. Then, we can estimate

$$\begin{aligned}\|\exp(\tilde{A}t)\| &\leq K \exp(\omega t); \quad t \geq 0, \\ \|\exp(\tilde{A}t)\| &\leq \tilde{K} \exp(\tilde{\omega} t); \quad t \geq 0,\end{aligned}\tag{22}$$

where the estimations are held for some numbers $K \geq 0$ and $\omega, \tilde{\omega} \in \mathbb{R}$.

In the following, we distinguish between the following two operator-types.

- We assume $(\tilde{A}(t))_{t \geq 0}$ and $(\tilde{B}(t))_{t \geq 0}$ are bounded operators, which generates stable semigroups, meaning $\omega, \tilde{\omega} \leq 0$, see [13,21], or
- We assume $(\tilde{A}(t))_{t \geq 0}$ and $(\tilde{B}(t))_{t \geq 0}$ are operators with exponential growth, which generates stable semigroups, means $\omega, \tilde{\omega} > 0$, see [13,21].

Then, we have the following two estimates of the two groups of operators:

- Bounded operators.
They are estimated as

$$\begin{aligned}\|\exp(\tilde{A}t)\| &\leq K, \quad t \geq 0, \\ \|\exp(\tilde{B}t)\| &\leq \tilde{K}, \quad t \geq 0,\end{aligned}\tag{23}$$

and we apply the estimations to (21) and obtain the relation

$$\begin{aligned}\|e_i\|(t) &\leq K \|\tilde{B}\| (\tau_n - \epsilon) \|e_{i-1}\|, \quad t \in [t^n, t^{n+1} - \epsilon], \\ \|e_{i+1}\|(t) &\leq \tilde{K} \|\tilde{A}\| (\tau_n + \epsilon) \|e_i\|, \quad t \in [t^n, t^{n+1} - \epsilon].\end{aligned}\tag{24}$$

- Operators with exponential growth.

Here, we assume that $(\exp(\tilde{A}t))_{t \geq 0}$, $(\exp(\tilde{B}t))_{t \geq 0}$ are exponentially growing with some $\omega > 0$, $\tilde{\omega} > 0$. Therefore, we can estimate

$$\int_{t^n}^t \|\exp(\tilde{A}(t-s))\| ds \leq K_\omega(t), \quad t \in [t^n, t^{n+1} - \epsilon], \quad (25)$$

$$\int_{t^n}^t \|\exp(\tilde{B}(t-s))\| ds \leq \tilde{K}_\omega(t), \quad t \in [t^n, t^{n+1} + \epsilon], \quad (26)$$

where

$$K_\omega(t) = \frac{K}{\omega} (\exp(\omega(t-t^n)) - 1), \quad t \in [t^n, t^{n+1} - \epsilon], \quad (27)$$

$$\tilde{K}_\omega(t) = \frac{\tilde{K}}{\tilde{\omega}} (\exp(\omega(t-t^n)) - 1), \quad t \in [t^n, t^{n+1} + \epsilon]. \quad (28)$$

Further, we apply

$$K_\omega(t) \leq \frac{K}{\omega} (\exp(\omega\tau_n) - 1) = K\tau_n + \mathcal{O}(\tau_n^2), \quad (29)$$

$$\tilde{K}_\omega(t) \leq \frac{\tilde{K}}{\tilde{\omega}} (\exp(\tilde{\omega}\tau_n) - 1) = \tilde{K}\tau_n + \mathcal{O}(\tau_n^2). \quad (30)$$

The estimations (24) and (30) result in

$$\|e_i\| \leq (K\|\tilde{B}\|(\tau_n - \epsilon)\|e_{i-1}\| + \mathcal{O}((\tau_n - \epsilon)^2), \quad (31)$$

$$\|e_{i+1}\| \leq (\tilde{K}\|\tilde{A}\|(\tau_n + \epsilon)\|e_i\| + \mathcal{O}((\tau_n + \epsilon)^2), \quad (32)$$

and we can apply recursively the error-estimation of Equations (31) and (32) and we obtain,

$$\|e_{i+1}\| \leq K\|\tilde{B}\|\|\tilde{A}\|\|\tilde{B}\|\|e_{i-1}\|(\tau_n^2 - 2\epsilon\tau_n) + \mathcal{O}(\tau_n^3) + \mathcal{O}(\epsilon\tau_n^2). \quad (33)$$

Then, we recursively applied the Equation (33) and obtain the proved statement. $\square$

Remark 2. Based on the derivation of the error of the shifted iterative method, we obtain the error

$$err = \|u_i(t^{n+1}) - u_{i,\epsilon}(t^{n+1})\| \leq \eta, \quad (34)$$

whereas $err = C(\Delta t^{2i-1})$ and also $\eta = C(\Delta t_{new}^{2i-1})$, and we obtain

$$\Delta t_{new} = \nu \Delta t \left(\frac{\eta}{err} \right)^{2i-1}. \quad (35)$$

Remark 3. In realistic applications, an optimal relation between the time-step τ_n and the number of iterative steps $2i - 1$, see Equation (35), is necessary. In practical experiments, we saw that $\nu > 0$, but near to 1 and $i \approx 3, 4, 5$ iterations are sufficient, such that we obtain an optimal new time-step Δt_{new} . To improve the criterion for stopping the iterative processes, we can additionally define an error bound. For example $|c_i - c_{i-1}| \leq err$ with $err = 10^{-4}$ can be used to restrict us to an appropriate low number of iterative steps.

The order of accuracy can be improved by the choice of the initial iteration function, e.g., additional pre-steps with standard splitting approaches, see [13].

Based on our assumption about the initial solutions, we initialize with exact solutions or we apply higher order interpolated split solutions. This assumption allow to derive a theory for the exactness of the iterative methods, see also [13].

4. Numerical Results

In this section, we present the numerical results based on our novel iterative splitting methods for nonlinear ordinary and partial differential equation. We verified our theoretical results of the error-estimates and applied the shifted iterative splitting methods as a new-solver class.

4.1. First Numerical Example: Bernoulli Equation

In the first example, we apply a nonlinear differential equation, which is given as the Bernoulli equation, see

$$\frac{\partial u(t)}{\partial t} = (\lambda_1 + \lambda_3)u(t) + (\lambda_2 + \lambda_4)(u(t))^p, \quad t \in [0, T], \quad \text{with } u(0) = 1. \quad (36)$$

For the Bernoulli equation, we can derive analytical solutions as reference solutions, see [15,22]. The analytical solutions are given as

$$u(t) = \exp((\lambda_1 + \lambda_3)t) \left[-\frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3} \exp((\lambda_1 + \lambda_3)(p-1)t) + c \right]^{1/(1-p)}.$$

Using $u(0) = 1$ we find that $c = 1 + \frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3}$, so

$$u(t) = \exp((\lambda_1 + \lambda_3)t) \left\{ 1 + \frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3} [1 - \exp((\lambda_1 + \lambda_3)(p-1)t)] \right\}^{1/(1-p)}. \quad (37)$$

For the applications, we apply the following parameters, $p = 2$, $\lambda_1 = -1$, $\lambda_2 = -0.5$, $\lambda_3 = -100$, $\lambda_4 = -20$, $T = 0.2$, and $u(0) = 1$.

We apply the following operators for the splitting.

- operator A: $A = (\lambda_1 + \lambda_3)$,
- operator B: $B(u) = (\lambda_2 + \lambda_4)(u(t))^{p-1}$.

We apply backward Euler method to approximate the derivative in each subinterval $[t^n, t^{n+1}]$, $n = 0, 1, \dots, N$, and solve the resulting equation by using the fixed point method and Newton's method with tolerance 10^{-12} allowing a maximum of three iterations. The accuracy of the methods is assessed by comparing the numerical result u_{num} with the analytical solution u given by (37). We compute the maximum and mean difference at the nodes t_n , according to

$$e_{\max} = \max_n |u_{num}(t_n) - u(t_n)|,$$

and

$$e_{\text{mean}} = \frac{1}{N} \sum_n |u_{num}(t_n) - u(t_n)|.$$

For the Shifted Strang–Marchuk splitting, we analyze the accuracy (with respect to the analytic solution) and the cost of the algorithm for different tolerances η and coarsening factors $1 + \kappa$, where we have set $\kappa = 4\eta^{\frac{1}{4}}$. Taking larger values of κ reduces the number of time intervals, but increases the number

of tentative steps, where the Δt_n must be reduced in order to satisfy the error tolerance criterion. The value of κ has been experimentally chosen in order to minimize the total number of steps.

In each splitting step, the differential equation is approximated by back-Euler's method and solved using the fixed point (BEFP) or Newton's (BEN) methods. The cost of the algorithm and the final accuracy depend on the error tolerance η and the coarsening factor $1 + \kappa$. We compare the shifted ABA-operator splitting method with the shifted variants of the iterative splitting methods above considered.

Table 1 shows that the accuracy is roughly proportional to the square root of the tolerance η , and the number of functional evaluations is inversely proportional to the same quantity. Newton's method requires less iterations to fulfill the tolerance; thus, if the number of time steps is similar to that of the fixed point method, it needs less functional evaluations. Nevertheless, Newton's method also evaluates the derivative, which reduces its advantage over the fixed point method.

Table 1. Shifted Strang–Marchuk splitting method for Bernoulli's equation.

Solver	Tolerance η	Coarsening $1 + \kappa$	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-6}$	1.1265	47	16	63	1091	$6.1055 \cdot 10^{-3}$	$2.5916 \cdot 10^{-3}$
	$1.0 \cdot 10^{-8}$	1.0400	344	80	424	7530	$7.3088 \cdot 10^{-4}$	$3.2822 \cdot 10^{-4}$
	$1.0 \cdot 10^{-10}$	1.0126	3048	303	3351	59938	$9.4175 \cdot 10^{-5}$	$3.5575 \cdot 10^{-5}$
	$1.0 \cdot 10^{-12}$	1.0040	29706	1055	30761	550518	$1.2504 \cdot 10^{-5}$	$3.6420 \cdot 10^{-6}$
BEN	$1.0 \cdot 10^{-6}$	1.1265	47	16	63	838	$6.1709 \cdot 10^{-3}$	$2.6494 \cdot 10^{-3}$
	$1.0 \cdot 10^{-8}$	1.0400	341	79	420	5742	$7.6231 \cdot 10^{-4}$	$3.2987 \cdot 10^{-4}$
	$1.0 \cdot 10^{-10}$	1.0126	3047	303	3350	40100	$9.4168 \cdot 10^{-5}$	$3.5588 \cdot 10^{-5}$
	$1.0 \cdot 10^{-12}$	1.0040	29706	1055	30761	368648	$1.2526 \cdot 10^{-5}$	$3.6422 \cdot 10^{-6}$

For the Shifted Linear Iterative Splitting method, we take $\kappa = 4\sqrt{\eta}$. We obtain similar accuracies to Strang–Marchuk's algorithm working now with higher error tolerances η , as it is shown in Table 2. The accuracy is of the same order as η , whereas the computational cost is slightly higher than the cost of Strang–Marchuk's algorithm.

Table 2. Shifted linear iterative splitting method for Bernoulli's equation.

Solver	Tolerance η	Coarsening $1 + \kappa$	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-4}$	1.0400	109	7	116	1358	$7.0454 \cdot 10^{-3}$	$3.0242 \cdot 10^{-3}$
	$1.0 \cdot 10^{-5}$	1.0126	692	39	731	8656	$8.4372 \cdot 10^{-4}$	$4.1488 \cdot 10^{-4}$
	$1.0 \cdot 10^{-6}$	1.0040	5752	164	5916	70530	$9.1785 \cdot 10^{-5}$	$4.7166 \cdot 10^{-5}$
	$1.0 \cdot 10^{-7}$	1.0013	54105	585	54690	654195	$9.5495 \cdot 10^{-6}$	$4.9235 \cdot 10^{-6}$
BEN	$1.0 \cdot 10^{-4}$	1.0400	108	7	115	912	$7.1531 \cdot 10^{-3}$	$3.0507 \cdot 10^{-3}$
	$1.0 \cdot 10^{-5}$	1.0126	690	39	729	5824	$8.5046 \cdot 10^{-4}$	$4.1542 \cdot 10^{-4}$
	$1.0 \cdot 10^{-6}$	1.0040	5748	164	5912	47288	$9.2284 \cdot 10^{-5}$	$4.7179 \cdot 10^{-5}$
	$1.0 \cdot 10^{-7}$	1.0013	54170	586	54756	438040	$9.3885 \cdot 10^{-6}$	$4.9213 \cdot 10^{-6}$

The results for the Shifted Quasilinear Iterative Splitting, see Table 3, are quite alike to the ones of the linear splitting. Increasing the number of iterations, $iter = 2, 3, \dots$, results in a linear increment of the cost without any accuracy improvement.

Table 3. Shifted quasilinear iterative splitting method for Bernoulli's equation.

Solver	Tolerance η	Coarsening $1 + \kappa$	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEF	$1.0 \cdot 10^{-4}$	1.0400	108	7	115	1350	$7.1191 \cdot 10^{-3}$	$3.0513 \cdot 10^{-3}$
	$1.0 \cdot 10^{-5}$	1.0126	691	39	730	8652	$8.4635 \cdot 10^{-4}$	$4.1525 \cdot 10^{-4}$
	$1.0 \cdot 10^{-6}$	1.0040	5750	164	5914	70534	$9.1939 \cdot 10^{-5}$	$4.7178 \cdot 10^{-5}$
	$1.0 \cdot 10^{-7}$	1.0013	54179	586	54765	655207	$9.4160 \cdot 10^{-6}$	$4.9210 \cdot 10^{-6}$
BEN	$1.0 \cdot 10^{-4}$	1.0400	107	7	114	1108	$7.2075 \cdot 10^{-3}$	$3.0781 \cdot 10^{-3}$
	$1.0 \cdot 10^{-5}$	1.0126	689	39	728	7146	$8.5547 \cdot 10^{-4}$	$4.1556 \cdot 10^{-4}$
	$1.0 \cdot 10^{-6}$	1.0040	5747	164	5911	58464	$9.2240 \cdot 10^{-5}$	$4.7190 \cdot 10^{-5}$
	$1.0 \cdot 10^{-7}$	1.0013	54168	586	54754	438030	$9.3864 \cdot 10^{-6}$	$4.9214 \cdot 10^{-6}$

4.2. Second Numerical Example: Mixed Convection–Diffusion and Burgers Equation

In the second numerical example, we apply coupled partial differential equation (PDE). We apply a coupling of a convection–diffusion equation with a Burgers' equation in 2D, which is called mixed convection–diffusion and Burgers equation (MCDB), and given as

$$\begin{aligned}
 \partial_t u &= -\frac{1}{2}u(\partial_x u + \partial_y u) - \frac{1}{2}(\partial_x u + \partial_y u) \\
 &\quad + \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t), \quad (x, y, t) \in \Omega \times [0, T], \\
 u(x, y, 0) &= u_{\text{ana}}(x, y, 0), \quad (x, y) \in \Omega, \\
 u(x, y, t) &= u_{\text{ana}}(x, y, t), \quad (x, y, t) \in \partial\Omega \times [0, T],
 \end{aligned} \tag{38}$$

where the domains are given as $\Omega = [0, 1] \times [0, 1]$ and $T = 1.25$. The viscosity is μ .

For such an mixed PDE, we can derive an analytical solution, which is

$$u_{\text{ana}}(x, y, t) = \left(1 + \exp\left(\frac{x + y - t}{2\mu}\right)\right)^{-1} + \exp\left(\frac{x + y - t}{2\mu}\right),$$

where we can derive the right hand side $f(x, y, t)$.

By considering the following operators

$$\begin{aligned}
 A(u)v &= -\frac{1}{2}u(\partial_x v + \partial_y v) + \frac{1}{2}\mu(\partial_{xx} v + \partial_{yy} v), \\
 Bv &= -\frac{1}{2}(\partial_x v + \partial_y v) + \frac{1}{2}\mu(\partial_{xx} v + \partial_{yy} v) + f(x, y, t).
 \end{aligned}$$

The MCDB Equation (38) is splitted into the Burgers' term, A and the convection–diffusion term, B and we obtain the operators:

$$\partial_t u = A(u)u + Bu.$$

We deal with different viscosities: low viscosity $\mu = 0.5$, high viscosity, $\mu = 5$. The spatial domain is discretized taking a rectangular mesh with $n_x = n_y = 16$ intervals and applying standard second order divided difference approximations. The resulting differential system is solved by the same methods as in the previous example. The coarsening strategy applied here when $err < \eta$ is

$$\Delta t_{\text{new}} = \min(1 + \eta^2 / err, 2)\Delta t,$$

where $err = \|u_{i,j}\|$ is the vector norm of the computed values in the nodes (x_i, y_j) at each time step.

For the shifted Strang–Marchuk splitting method we take $\epsilon = 0.05$ and different values of η .

Table 4 shows the results of solving the equation with low viscosity $\mu = 0.5$ using different tolerances. The solutions of the differential equations are approximated by using back-Euler fixed point method (BEFP) or back-Euler–Newton’s method (BEN). Both methods perform similarly in cost and accuracy in this case.

The corresponding results for solving the equation with high viscosity, $\mu = 5$, are shown in Table 5. BEFP requires much more time steps than BEN, but reaches more accuracy.

Table 4. Solution of mixed convection–diffusion and Burgers (MCDB) equation for $\mu = 0.5$ using the shifted Strang–Marchuk splitting method.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	336	211	547	8752	$1.8537 \cdot 10^{-2}$	$1.5565 \cdot 10^{-3}$
	$1.0 \cdot 10^{-2}$	493	38	531	8496	$7.7173 \cdot 10^{-3}$	$1.0212 \cdot 10^{-3}$
	$1.0 \cdot 10^{-3}$	1581	5	1586	25376	$2.8237 \cdot 10^{-3}$	$4.6368 \cdot 10^{-4}$
	$1.0 \cdot 10^{-4}$	6356	2	6358	101728	$9.6438 \cdot 10^{-4}$	$2.6878 \cdot 10^{-4}$
BEN	$1.0 \cdot 10^{-1}$	65	28	93	1302	$4.6585 \cdot 10^{-2}$	$6.1648 \cdot 10^{-3}$
	$1.0 \cdot 10^{-2}$	381	18	399	5586	$9.8225 \cdot 10^{-3}$	$1.2524 \cdot 10^{-3}$
	$1.0 \cdot 10^{-3}$	1566	5	1571	21994	$2.8012 \cdot 10^{-3}$	$4.6639 \cdot 10^{-4}$
	$1.0 \cdot 10^{-4}$	6353	2	6355	88970	$9.6547 \cdot 10^{-4}$	$2.6882 \cdot 10^{-4}$

Table 5. Solution of MCDB equation for $\mu = 5$ using the shifted Strang–Marchuk splitting method.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	4899	1077	5976	95616	$7.9463 \cdot 10^{-3}$	$1.0352 \cdot 10^{-4}$
	$1.0 \cdot 10^{-2}$	3375	977	4352	69632	$6.2542 \cdot 10^{-4}$	$1.4242 \cdot 10^{-5}$
	$1.0 \cdot 10^{-3}$	2980	159	3139	50224	$1.7561 \cdot 10^{-5}$	$8.0790 \cdot 10^{-6}$
	$1.0 \cdot 10^{-4}$	3382	4	3386	54176	$1.2602 \cdot 10^{-5}$	$6.7706 \cdot 10^{-6}$
BEN	$1.0 \cdot 10^{-1}$	11	0	11	154	$1.6393 \cdot 10^{-3}$	$4.0286 \cdot 10^{-4}$
	$1.0 \cdot 10^{-2}$	43	1	44	616	$9.7024 \cdot 10^{-4}$	$3.0109 \cdot 10^{-4}$
	$1.0 \cdot 10^{-3}$	339	0	339	4746	$1.4041 \cdot 10^{-4}$	$6.0483 \cdot 10^{-5}$
	$1.0 \cdot 10^{-4}$	1118	1	1119	15662	$2.3620 \cdot 10^{-5}$	$1.9871 \cdot 10^{-5}$

For the linear and quasilinear shifted iterative splitting methods we take $\epsilon = 0.5$ and the same coarsening strategy. Tables 6–9 show the cost and the accuracy in the low and high viscosity cases for the shifted linear and quasilinear iterative splitting methods using the back-Euler fixed point method and back-Euler Newton’s method as solvers.

The shifted linear and quasilinear iterative splitting methods give similar results in all the considered cases. The behavior of the back-Euler fixed point method is worse in the low viscosity case than in the high viscosity case, as in the shifted Strang–Marchuk splitting method.

Table 6. Results of the shifted linear iterative splitting for the MCDB equation with $\mu = 0.5$.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	646	303	949	11388	$5.0943 \cdot 10^{-3}$	$9.7351 \cdot 10^{-4}$
	$5.0 \cdot 10^{-2}$	642	238	880	10560	$2.7441 \cdot 10^{-3}$	$3.9256 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	788	172	960	11520	$1.6284 \cdot 10^{-3}$	$1.5681 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	1343	152	1495	17940	$5.9673 \cdot 10^{-4}$	$1.5654 \cdot 10^{-4}$
BEN	$1.0 \cdot 10^{-1}$	222	164	386	3860	$1.4076 \cdot 10^{-2}$	$6.1344 \cdot 10^{-4}$
	$5.0 \cdot 10^{-2}$	388	157	545	5450	$4.7347 \cdot 10^{-3}$	$2.0174 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	706	152	858	8580	$1.5222 \cdot 10^{-3}$	$1.3188 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	1337	150	1487	14870	$2.6003 \cdot 10^{-3}$	$1.5871 \cdot 10^{-4}$

Table 7. Results of the shifted linear iterative splitting for the MCDB equation with $\mu = 5$.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	6530	2962	9492	113904	$1.9099 \cdot 10^{-2}$	$1.0672 \cdot 10^{-3}$
	$5.0 \cdot 10^{-2}$	6458	2022	8480	101760	$1.9099 \cdot 10^{-2}$	$5.1457 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	6440	1250	7690	92280	$1.1951 \cdot 10^{-3}$	$2.2108 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	6407	807	7214	86568	$5.1211 \cdot 10^{-4}$	$8.9769 \cdot 10^{-5}$
BEN	$1.0 \cdot 10^{-1}$	37	21	58	580	$5.3262 \cdot 10^{-2}$	$6.0472 \cdot 10^{-3}$
	$5.0 \cdot 10^{-2}$	63	20	83	830	$2.6259 \cdot 10^{-2}$	$2.5666 \cdot 10^{-3}$
	$2.5 \cdot 10^{-2}$	109	18	127	1270	$9.7793 \cdot 10^{-3}$	$9.2603 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	188	15	203	2030	$3.4315 \cdot 10^{-3}$	$3.1950 \cdot 10^{-4}$

Table 8. Results of the shifted quasi linear iterative splitting for the MCDB equation with $\mu = 0.5$.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	647	321	968	11616	$5.2855 \cdot 10^{-3}$	$9.7675 \cdot 10^{-4}$
	$5.0 \cdot 10^{-2}$	643	238	881	10572	$2.4622 \cdot 10^{-3}$	$3.9273 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	788	173	961	11532	$1.7081 \cdot 10^{-3}$	$1.5881 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	6419	809	7228	86736	$7.2082 \cdot 10^{-4}$	$8.9011 \cdot 10^{-5}$
BEN	$1.0 \cdot 10^{-1}$	221	163	384	3840	$1.3667 \cdot 10^{-2}$	$5.5944 \cdot 10^{-4}$
	$5.0 \cdot 10^{-2}$	388	157	545	5450	$4.6486 \cdot 10^{-3}$	$2.0269 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	706	152	858	8580	$2.2658 \cdot 10^{-3}$	$1.3493 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	1343	152	1495	17940	$6.2250 \cdot 10^{-4}$	$1.5775 \cdot 10^{-4}$

Table 9. Results of the shifted quasi linear iterative splitting for the MCDB equation with $\mu = 5$.

Solver	Tolerance η	Time Steps	Tentative Steps	Total Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
BEFP	$1.0 \cdot 10^{-1}$	6525	2974	9499	113988	$1.9100 \cdot 10^{-2}$	$1.0647 \cdot 10^{-3}$
	$5.0 \cdot 10^{-2}$	6461	2024	8485	101820	$1.9100 \cdot 10^{-2}$	$5.1301 \cdot 10^{-4}$
	$2.5 \cdot 10^{-2}$	6445	1235	7680	92160	$1.1921 \cdot 10^{-3}$	$2.1909 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	6419	809	7228	86736	$7.2082 \cdot 10^{-4}$	$8.9011 \cdot 10^{-5}$
BEN	$1.0 \cdot 10^{-1}$	37	21	58	580	$5.3071 \cdot 10^{-2}$	$6.0407 \cdot 10^{-3}$
	$5.0 \cdot 10^{-2}$	63	20	83	830	$2.6196 \cdot 10^{-2}$	$2.5650 \cdot 10^{-3}$
	$2.5 \cdot 10^{-2}$	109	18	127	1270	$9.7690 \cdot 10^{-3}$	$9.2559 \cdot 10^{-4}$
	$1.25 \cdot 10^{-2}$	188	15	203	2030	$3.4299 \cdot 10^{-3}$	$3.1941 \cdot 10^{-4}$

4.3. Third Numerical Example: Convection-Diffusion-Reaction Equation

In the third numerical example, we deal with a PDE, which is a convection–diffusion–reaction equation in 3D (CDR), see the example in [23]:

$$\begin{aligned}\partial_t u &= -\mathbf{v} \cdot \nabla u + \nabla D \nabla u - ku, \quad (x, y, z, t) \in \Omega \times [t_0, T], \\ u(x, y, z, t_0) &= u_0(x, y, z), \quad (x, y, z) \in \Omega, \\ u(x, y, z, t) &= 0 \in \partial\Omega \times [t_0, T],\end{aligned}\quad (39)$$

where we have $\mathbf{v} = (v_x, v_y, v_z)^t$, $D \in \mathbb{R}^3 \times \mathbb{R}^3$ a diffusion matrix, $\mathbf{u} \in$ is the velocity field, k is a reaction parameter, and $\Omega = [0, 4]^3 \times [t_0, T]$, $T = 10.0$.

We can have a special analytical solution for an instantaneous point source, which is given as:

$$u_{\text{ana}}(x, y, z, t) = \frac{M}{4\pi t \sqrt{D_{11}D_{22}D_{33}t}} \exp\left(-\frac{((x-x_1)-v_x t)^2}{4D_{11}t} - \frac{(y-y_1)^2}{4D_{22}t} - \frac{(z-z_1)^2}{4D_{22}t}\right).$$

We have the following parameters.

- instantaneous point source: $(x_1, y_1, z_1) = (1, 1, 1)$, $M = 1.0$,
- initial start at $t_0 = 1$, where we initialise the equation with $\mathbf{u}_0(x, y, z) = u_{\text{ana}}(x, y, z, t_0)$,
- the diffusion parameters are given as $D_{11} = 0.01$, $D_{22} = 0.02$, $D_{33} = 0.03$ all other parameters are 0,
- the velocity is given as $(v_x, v_y, v_z) = (0.1, 0, 0)$,
- the reaction parameter is given as $k = 0.1$.

By considering the following operators, we decouple into the fast velocity–reaction part and the slow diffusion parts:

$$\begin{aligned}A &= \nabla D \nabla, \\ B &= -\mathbf{v} \cdot \nabla - k,\end{aligned}$$

we split (50) in fast and slow parts

$$\partial_t u = Au + Bu.$$

The equation is spatially discretized taking a number, n_x , n_y , n_z , of equal subintervals in each direction in Ω , and approximating the spatial derivatives by standard second order divided differences, resulting in a linear differential system.

We first check that the discretization error decreases with the size of the spatial subintervals by solving the differential system using Heun's method and Strang–Marchuk method with different number of spatial subintervals. The numerical results u_{num} in the node points (x_i, y_j, z_k) are compared with the analytical solution u_{ana} in the same points at the final time $T = 10$, computing the maximum and the mean absolute differences as before. Table 10 shows that there is no significant difference between both methods.

Table 10. Differences between the analytical solution of convection–diffusion–reaction (CDR) equation and the numerical solutions obtained by direct (Heun) integration and by Strang–Marchuk method with different number of spatial subintervals.

Spatial Subintervals	Time Steps	Heun		Strang–Marchuk	
		e_{max}	e_{mean}	e_{max}	e_{mean}
8	16	0.342564	0.019429	0.338664	0.019334
12	24	0.108504	0.006381	0.104289	0.006366
16	32	0.065986	0.002939	0.063730	0.002901
20	40	0.044801	0.001792	0.043494	0.001774

Now we fix the number of spatial subintervals $n_x = n_y = n_z = 16$, and analyze the performance of the adaptive methods for the CDR example. To estimate the convergence of the methods, we compare their results with the approximation obtained by integrating the differential equation by Heun's method using the same time steps. Table 11 shows the results of the shifted Strang–Marchuk splitting and the shifted linear iterative splitting for different tolerances, η . Lower tolerances produce lower maximum and mean errors but require more time steps. The relationship between the number of time steps and the mean error is depicted in Figure 3.

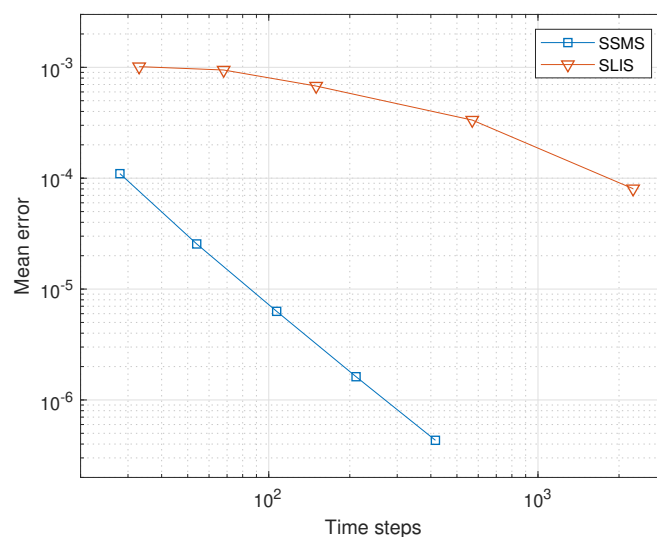


Figure 3. Mean error e_{mean} of the shifted Strang–Marchuk splitting (SSMS) and the shifted linear iterative splitting (SLIS) for the CDR equation for different tolerances, η .

Remark 4. In the Figure 3, we see the differences in the convergence behaviour between the shifted Strang–Marchuk splitting (SSML) and the shifted linear iterative splitting (SLIS) method. We see in the figure, that the SSML method has only a linear convergence order ≤ 1 , the SLIS method has higher order of convergence, here we have at least an

order ≥ 2 . Therefore the adaptive iterative scheme is much more effective and accurate than the noniterative splitting scheme. The result verified the proposition, that the iterative splitting scheme is a higher order scheme, see [3,15], and that we also conserve the higher order approach in the adaptive version.

Table 11. Results of the shifted Strang–Marchuk’s splitting, (SSMS), and the shifted linear iterative splitting, SLIS, for the CDR equation for different tolerances.

Shifted Method	Tolerance η	Time Steps	Tentative Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
SSMS	$1.0 \cdot 10^{-4}$	28	4	372	$4.1517 \cdot 10^{-3}$	$1.0990 \cdot 10^{-4}$
	$2.5 \cdot 10^{-5}$	54	11	768	$9.1570 \cdot 10^{-4}$	$2.5527 \cdot 10^{-5}$
	$6.25 \cdot 10^{-6}$	107	24	1560	$2.2147 \cdot 10^{-4}$	$6.2970 \cdot 10^{-6}$
	$1.5625 \cdot 10^{-6}$	211	52	3144	$5.6628 \cdot 10^{-5}$	$1.6200 \cdot 10^{-6}$
	$3.90625e-7$	416	103	6216	$1.5064 \cdot 10^{-5}$	$4.3245e-7$
SLIS	$6.4 \cdot 10^{-2}$	33	3	280	$1.1334 \cdot 10^{-2}$	$1.0145 \cdot 10^{-3}$
	$1.6 \cdot 10^{-2}$	68	4	568	$1.2867 \cdot 10^{-2}$	$9.4605 \cdot 10^{-4}$
	$4.0 \cdot 10^{-3}$	150	17	1328	$9.4607 \cdot 10^{-3}$	$6.7794 \cdot 10^{-4}$
	$1.0 \cdot 10^{-3}$	570	130	5592	$6.1880 \cdot 10^{-3}$	$3.3441 \cdot 10^{-4}$
	$2.5 \cdot 10^{-4}$	2256	577	22656	$1.4651 \cdot 10^{-3}$	$8.0392 \cdot 10^{-5}$

4.4. Fourth Numerical Example: Nonlinear Diffusion Equation

Our fourth numerical example is a partial differential equation which is nonlinear diffusion equation in 2D, see the example in [24].

The multicomponent diffusion equation is based on the idea of a Maxwell–Stefan diffusion equation, which is highly nonlinear, see [20,24]:

$$\begin{aligned}\partial_t \mathbf{u} &= \nabla A(\mathbf{u}) \nabla \mathbf{u}, \quad (x, y, t) \in \Omega \times [t_0, T], \\ \mathbf{u}(x, y, t_0) &= \mathbf{u}_0(x, y), \quad (x, z) \in \Omega, \\ \mathbf{u}(x, y, t) &= 0 \in \partial\Omega \times [t_0, T],\end{aligned}\tag{40}$$

where we have $A \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ a nonlinear diffusion matrix and $\Omega = [0, 1]^3 \times [t_0, T]$, $t_0 = 0$, $T = 1.0$.

An application of such a nonlinear diffusion (NLD) is given by

$$\partial_t u_1 = D_{12} \nabla \cdot \nabla u_1, \quad (x, y, t) \in \Omega \times [t_0, T],\tag{41}$$

$$\partial_t u_2 = \nabla \cdot \left(\left(\frac{1}{D_{23}} + \beta u_1 \right)^{-1} (\nabla u_2 + \beta D_{12} u_2 \nabla u_1) \right), \quad (x, y, t) \in \Omega \times [t_0, T],\tag{42}$$

$$\mathbf{u}(x, y, t) = 0 \in \partial\Omega \times [t_0, T],$$

where we have $\alpha = \left(\frac{1}{D_{12}} - \frac{1}{D_{13}} \right)$ and $\beta = \left(\frac{1}{D_{12}} - \frac{1}{D_{23}} \right)$.

Further, we apply with the following parameters in the NLD Equations (41) and (42).

The parameters and the initial and boundary conditions are given as:

- Uphill example, which is known as semi-degenerated Duncan and Toor experiment, see [25]: $D_{12} = D_{13} = 0.833$ and $D_{23} = 0.168$, where we have $\alpha = 0$.

- Asymptotic example, which is known as asymptotic Duncan and Toor experiment, see [25]: $D_{12} = 0.0833$, $D_{13} = 0.680$ and $D_{23} = 0.168$, where we have $\alpha \neq 0$.
- We apply $J = 140$, where J is the number of spatial grid points.
- Based on the explicit discretization method, we have to fulfill the time-step-restriction, which is given as CFL-condition:

$$\Delta t \leq \frac{(\Delta x)^2}{2 \max\{D_{12}, D_{13}, D_{23}\}}.$$
- The computational domains are given with: $\Omega = [0, 1]$ is the spatial domain and $[0, T] = [0, 1]$ is the time domain.
- The initial conditions are as follows.

1. Uphill example

$$u_1^{in}(x) = \begin{cases} 0.8 & \text{if } 0 \leq x < 0.25, \\ 1.6(0.75 - x) & \text{if } 0.25 \leq x < 0.75, \\ 0.0 & \text{if } 0.75 \leq x \leq 1.0, \end{cases} \quad (43)$$

$$u_2^{in}(x) = 0.2, \text{ for all } x \in \Omega = [0, 1]. \quad (44)$$

2. Asymptotic example

$$u_1^{in}(x) = \begin{cases} 0.8 & \text{if } 0 \leq x \leq 0.5, \\ 0.0 & \text{else,} \end{cases} \quad (45)$$

$$u_2^{in}(x) = 0.2, \text{ for all } x \in \Omega = [0, 1]. \quad (46)$$

- For the boundary conditions, we apply no-flux type conditions:

$$u_1 = u_2 = 0, \text{ on } \partial\Omega \times [0, 1]. \quad (47)$$

We apply the following splitting of the operators with the one-dimensional spatial derivations:

$$\partial_t \mathbf{u} = (A(\mathbf{u}) + B(\mathbf{u}))\mathbf{u}, \quad (48)$$

where we have the operators in the following decomposition of the u_1 and u_2 parts with $\xi \in [0, 1]$:

$$A(\mathbf{u}) = \begin{pmatrix} \xi D_{12} \frac{\partial^2}{\partial x^2} & 0 \\ \xi \frac{\partial}{\partial x} \left(\left(\frac{1}{D_{23}} + \beta u_1 \right)^{-1} \beta D_{12} u_2 \frac{\partial}{\partial x} \right) & (1 - \xi) \frac{\partial}{\partial x} \left(\left(\frac{1}{D_{23}} + \beta u_1 \right)^{-1} \frac{\partial}{\partial x} \right) \end{pmatrix} \quad (49)$$

$$B(\mathbf{u}) = \begin{pmatrix} (1 - \xi) D_{12} \frac{\partial^2}{\partial x^2} & 0 \\ (1 - \xi) \frac{\partial}{\partial x} \left(\left(\frac{1}{D_{23}} + \beta u_1 \right)^{-1} \beta D_{12} u_2 \frac{\partial}{\partial x} \right) & \xi \frac{\partial}{\partial x} \left(\left(\frac{1}{D_{23}} + \beta u_1 \right)^{-1} \frac{\partial}{\partial x} \right) \end{pmatrix} \quad (50)$$

where we have $\xi = 0.5$ a symmetric decomposition.

We first check that the non adaptive methods require a very small time step to converge and estimate its convergence by comparing the results doubling successively the number of time steps. The results are shown in Table 12 for the direct integration and for the unshifted Strang–Marchuk method. The errors are computed measuring the difference between the result obtained with a given number of time steps and the result with twice that number at every shared temporal and spatial node. The error estimates for the

Strang–Marchuk splitting in the case of 40,000 time steps are not available because the method diverges with 20,000 time steps.

Figure 4 illustrates the uphill phenomenon, where the solutions u_1 and u_2 increase before reaching the stationary state.

The adaptive methods result in an important reduction of the number of time steps obtaining similar error estimations. Tables 13 and 14 show the results for the uphill case and for the asymptotic case of the nonlinear diffusion equation, respectively. Here, the errors are computed by comparing the solution of the shifted methods with the ones obtained by direct integration, using the same time steps as the adaptive method.

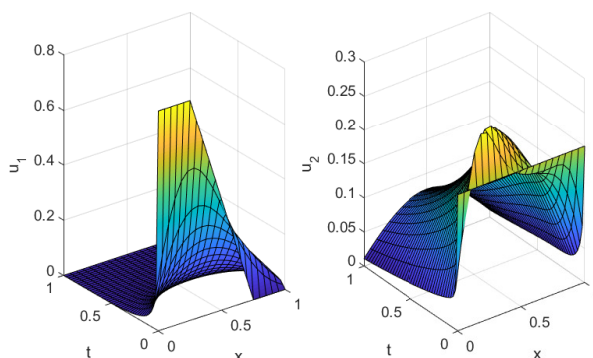


Figure 4. Evolution of the magnitudes u_1 and u_2 in the uphill example.

Table 12. Cost and error estimates of the Heun integration (HI) and the Strang–Marchuk (SM) splitting for the nonlinear diffusion equation.

Solver	Time Steps	Functional Evaluations	Uphill Example		Asymptotic Behavior	
			Max Error e_{max}	Mean Error e_{mean}	Max Error e_{max}	Mean Error e_{mean}
HI	120000	240000	$1.2050 \cdot 10^{-2}$	$7.0883 \cdot 10^{-7}$	$1.0045 \cdot 10^{-1}$	$5.7176e-7$
	240000	480000	$2.1710 \cdot 10^{-3}$	$1.1072 \cdot 10^{-7}$	$1.8856 \cdot 10^{-2}$	$1.0060 \cdot 10^{-7}$
	480000	960000	$4.5606 \cdot 10^{-4}$	$1.8113 \cdot 10^{-8}$	$4.1844 \cdot 10^{-3}$	$2.0626 \cdot 10^{-8}$
SM	40000	240000	$1.9853 \cdot 10^{-2}$	$7.3680 \cdot 10^{-7}$	—	—
	80000	480000	$4.3103 \cdot 10^{-3}$	$1.7807e-7$	$2.2560 \cdot 10^{-1}$	$7.4157 \cdot 10^{-7}$
	160000	960000	$8.2574 \cdot 10^{-4}$	$4.2266 \cdot 10^{-8}$	$3.3259 \cdot 10^{-2}$	$2.4397 \cdot 10^{-7}$

The shifted Strang–Marchuk method behaves better for $\epsilon = 0.03$, whereas the shifted linear and quasilinear splitting methods work well for $\epsilon = 0.03$, except for in Table 15, where the behavior of the considered splitting methods is studied for different splitting weights ζ .

Figure 5 depicts the regions in the space-time plane where the uphill phenomenon takes place, that is where N_2 and $\partial_x u_2$ have the same sign. The equation is solved by the shifted linear iterative splitting with $\eta = 1.0 \cdot 10^{-5}$.

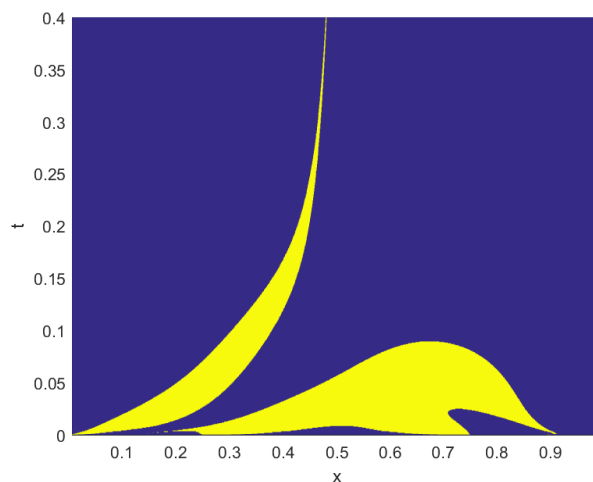


Figure 5. Regions in the space-time domain where N_2 and $\partial_x u_2$ have the same sign.

Table 13. Results of the shifted Strang–Marchuk’s splitting (SSMS) the shifted linear iterative splitting (SLIS), and the shifted quasilinear iterative splitting (SQIS), for the uphill case of the NLD equation.

Shifted Method	Parameter η	Time Steps	Tentative Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
SSMS $\epsilon = 0.03$	$1.0 \cdot 10^{-2}$	22711	477	278256	$4.4092 \cdot 10^{-1}$	$5.4509 \cdot 10^{-2}$
	$1.0 \cdot 10^{-3}$	12113	1297	160920	$8.4409 \cdot 10^{-2}$	$4.1336 \cdot 10^{-3}$
	$1.0 \cdot 10^{-4}$	12105	1290	160740	$4.5592 \cdot 10^{-3}$	$1.6081 \cdot 10^{-4}$
	$1.0 \cdot 10^{-5}$	12109	1314	161076	$6.9255 \cdot 10^{-4}$	$1.4140 \cdot 10^{-5}$
SLIS $\epsilon = 0.01$	$1.0 \cdot 10^{-2}$	19788	3177	183720	$9.3208 \cdot 10^{-2}$	$1.6424 \cdot 10^{-2}$
	$1.0 \cdot 10^{-3}$	19787	3133	183360	$7.2316 \cdot 10^{-3}$	$9.9959 \cdot 10^{-4}$
	$1.0 \cdot 10^{-4}$	19878	3158	184288	$1.1644 \cdot 10^{-3}$	$9.4958 \cdot 10^{-5}$
	$1.0 \cdot 10^{-5}$	25126	4880	240048	$1.1369 \cdot 10^{-4}$	$8.2151 \cdot 10^{-6}$
SQIS $\epsilon = 0.01$	$1.0 \cdot 10^{-2}$	20021	3119	185120	$8.3612 \cdot 10^{-1}$	$1.5261 \cdot 10^{-1}$
	$1.0 \cdot 10^{-3}$	19784	3194	183824	$7.0992 \cdot 10^{-3}$	$1.5841 \cdot 10^{-3}$
	$1.0 \cdot 10^{-4}$	19879	3145	184192	$1.1644 \cdot 10^{-3}$	$9.5553 \cdot 10^{-5}$
	$1.0 \cdot 10^{-5}$	25125	4895	240160	$1.1370 \cdot 10^{-4}$	$8.1793 \cdot 10^{-6}$

Table 14. Results of the shifted Strang–Marchuk’s splitting (SSMS), the shifted linear iterative splitting (SLIS), and the shifted quasilinear iterative splitting (SQIS), for the uphill case of the NLD equation with $\eta = 1 \cdot 10^{-5}$ and different values of the weight parameter ξ .

Shifted Method	Parameter ξ	Time Steps	Tentative Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
$\epsilon = 0.03$	0	40160	4775	539,220	$2.2711 \cdot 10^{-1}$	$4.0969 \cdot 10^{-2}$
	0.25	21036	3991	300,324	$6.1904 \cdot 10^{-2}$	$1.2819 \cdot 10^{-2}$
	0.5	12105	1290	160740	$4.5592 \cdot 10^{-3}$	$1.6081 \cdot 10^{-4}$
	0.75	12094	527	151452	$3.8271 \cdot 10^{-3}$	$1.9800 \cdot 10^{-5}$
	1	16435	1581	216192	$3.9292 \cdot 10^{-3}$	$1.4966 \cdot 10^{-4}$
$\epsilon = 0.01$	0	39313	7195	372064	$9.0557 \cdot 10^{-4}$	$1.5864 \cdot 10^{-4}$
	0.25	29515	5859	282992	$9.5854 \cdot 10^{-4}$	$2.1259 \cdot 10^{-4}$
	0.5	19878	3158	184288	$1.1644 \cdot 10^{-3}$	$9.4958 \cdot 10^{-5}$
	0.75	22010	2272	194256	$1.8200 \cdot 10^{-3}$	$2.9126 \cdot 10^{-5}$
	1	30867	2064	263448	$2.5255 \cdot 10^{-3}$	$2.4335 \cdot 10^{-5}$
$\epsilon = 0.01$	0	39313	7233	372368	$8.2133 \cdot 10^{-4}$	$1.6467 \cdot 10^{-4}$
	0.25	29515	5933	283584	$1.0667 \cdot 10^{-3}$	$2.2256 \cdot 10^{-4}$
	0.5	19879	3145	184192	$1.1644 \cdot 10^{-3}$	$9.5553 \cdot 10^{-5}$
	0.75	22077	2384	195688	$1.8200 \cdot 10^{-3}$	$3.0349 \cdot 10^{-5}$
	1	30906	2084	263920	$2.5255 \cdot 10^{-3}$	$2.5535 \cdot 10^{-5}$

Table 15. Results of the shifted Strang–Marchuk’s splitting (SSMS), the shifted linear iterative splitting (SLIS), and the shifted quasilinear iterative splitting (SQIS) for the asymptotic case of the NLD equation.

Shifted Method	Parameter η	Time Steps	Tentative Steps	Functional Evaluations	Max Error e_{max}	Mean Error e_{mean}
$\epsilon = 0.03$	$1.0 \cdot 10^{-2}$	31442	238	380160	$9.1699 \cdot 10^{-1}$	$1.2015 \cdot 10^{-2}$
	$1.0 \cdot 10^{-3}$	12810	546	160272	$5.1000 \cdot 10^{-1}$	$4.5127 \cdot 10^{-3}$
	$1.0 \cdot 10^{-4}$	10223	511	128808	$7.6494 \cdot 10^{-2}$	$1.6126 \cdot 10^{-4}$
	$1.0 \cdot 10^{-5}$	10508	854	136344	$7.2481 \cdot 10^{-3}$	$8.4180 \cdot 10^{-6}$
$\epsilon = 0.01$	$1.0 \cdot 10^{-2}$	15087	1748	134680	$1.5719 \cdot 10^{-1}$	$2.7535 \cdot 10^{-2}$
	$1.0 \cdot 10^{-3}$	14736	1702	131504	$1.2306 \cdot 10^{-2}$	$5.0463 \cdot 10^{-4}$
	$1.0 \cdot 10^{-4}$	15047	1815	134896	$2.2724 \cdot 10^{-3}$	$3.9500 \cdot 10^{-5}$
	$1.0 \cdot 10^{-5}$	24371	5071	235536	$2.2461 \cdot 10^{-4}$	$8.5634 \cdot 10^{-6}$
$\epsilon = 0.01$	$1.0 \cdot 10^{-2}$	16502	1509	144088	$2.7868 \cdot 10^{-1}$	$7.1696 \cdot 10^{-2}$
	$1.0 \cdot 10^{-3}$	14709	1683	131136	$1.2001 \cdot 10^{-2}$	$2.1926 \cdot 10^{-3}$
	$1.0 \cdot 10^{-4}$	15049	1819	134944	$1.9668 \cdot 10^{-3}$	$4.4380 \cdot 10^{-5}$
	$1.0 \cdot 10^{-5}$	24381	5042	235384	$1.9277 \cdot 10^{-4}$	$6.6005 \cdot 10^{-6}$

5. Conclusions and Discussion

We present a novel adaptive iterative splitting approach for partial differential equations of the type convection–diffusion–reaction equation. The numerical analysis shows the convergence of the schemes, while we could apply a shift in time of the methods. In the numerical experiments, we apply different state-of-the-art nonlinear convection–diffusion equations, where we receive benefits in the computational time and also in the accuracy of the methods. The adaptive splitting schemes allow to control the errors of the scheme and reduce the computational time, while we could apply smaller and larger time-steps.

Author Contributions: The theory, the formal analysis and the methodology presented in this paper was developed by J.G. The software development and the numerical validation of the methods was done by J.H. and E.M. The paper was written by J.G., J.H. and E.M. and was corrected and edited by J.G., J.H. and E.M.. The writing–review was done by J.G.. The supervision and project administration was done by J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by German Academic Exchange Service grant number 91588469.

Acknowledgments: We acknowledge support by the DFG Open Access Publication Funds of the Ruhr-Universität of Bochum, Germany and by Ministerio de Economía y Competitividad, Spain, under grant PGC2018-095896-B-C21-C22.

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

References

1. Auzinger, W.; Herfort, W. Local error structures and order conditions in terms of Lie elements for exponential splitting schemes. *Opusc. Math.* **2014**, *34*, 243–255.
2. Auzinger, W.; Koch, O.; Quell, M. Adaptive high-order splitting methods for systems of nonlinear evolution equations with periodic boundary conditions. *Numer. Algor.* **2017**, *75*, 261–283.
3. Geiser, J. *Iterative Splitting Methods for Differential Equations*; Numerical Analysis and Scientific Computing Series; Taylor & Francis Group: Boca Raton, FL, USA; London, UK; New York, NY, USA, 2011.
4. Descombes, S.; Massot, M. Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: Singular perturbation and order reduction. *Numer. Math.* **2004**, *97*, 667–698.
5. Descombes, S.; Dumont, T.; Louvet, V.; Massot, M. On the local and global errors of splitting approximations of reaction-diffusion equations with high spatial gradients. *Int. J. Comput. Math.* **2007**, *84*, 749–765.
6. McLachlan, R.I.; Quispel, G.R.W. Splitting methods. *Acta Numer.* **2002**, *11*, 341–434.
7. Trotter, H.F. On the product of semi-groups of operators. *Proc. Am. Math. Soc.* **1959**, *10*, 545–551.
8. Strang, G. On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **1968**, *5*, 506–517.
9. Jahnke, T.; Lubich, C. Error bounds for exponential operator splittings. *BIT Numer. Math.* **2000**, *40*, 735–745.
10. Geiser, J. Iterative Semi-implicit Splitting Methods for Stochastic Chemical Kinetics. In *Finite Difference Methods: Theory and Applications*; Dimov, I.; Faragó, I.; Vulkov, L., Eds.; Springer International Publishing: Cham, Germany, 2019; pp. 35–47.
11. Nevanlinna, O. Remarks on Picard-Lindelöf Iteration, Part I. *BIT* **1989**, *29*, 328–346.
12. Vandewalle, S. *Parallel Multigrid Waveform Relaxation for Parabolic Problems*; Teubner Skripten zur Numerik, B.G. Teubner Stuttgart: Stuttgart, Germany, 1993.
13. Farago, I.; Geiser, J. Iterative Operator-Splitting Methods for Linear Problems. *Int. J. Comput. Sci. Eng.* **2007**, *3*, 255–263.
14. Descombes, S.; Duarte, M.; Dumont, T.; Louvet, V.; Massot, M. Adaptive time splitting method for multi-scale evolutionary partial differential equations. *Confluentes Math.* **2011**, *3*, 413–443.
15. Geiser, J. Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations. *J. Comput. Appl. Math.* **2008**, *217*, 227–242.
16. Dimov, I.; Farago, I.; Havasi, A.; Zlatev, Z. Different splitting techniques with application to air pollution models.. *International Journal of Environment and Pollution* **2008**, *32*, 174–199.
17. Karlsen, K.H.; Lie, K.-A.; Natvig, J.R.; Nordhaug, H.F.; Dahle, H.K. Operator splitting methods for systems of convection–diffusion equations: Nonlinear error mechanisms and correction strategies. *J. Comput. Phys.* **2001**, *173*, 636–663.
18. Vabishchevich, P. *Additive Operator-Difference Schemes: Splitting Schemes*; De Gruyter: Berlin, Germany, 2014.
19. Geiser, J. Iterative operator-splitting methods for nonlinear differential equations and applications. *Numer. Methods Partial Differ. Equ.* **2011**, *27*, 1026–1054.
20. Geiser, J. Iterative solvers for the Maxwell–Stefan diffusion equations: Methods and applications in plasma and particle transport. *Cogent Math.* **2015**, *2*, 1092913.

21. Engel, K.-J.; Nagel, R. *One-Parameter Semigroups for Linear Evolution Equations*; Springer: New York, NY, USA, 2000.
22. Geiser, J.; Hueso, J.L.; Martinez, E. New versions of iterative splitting methods for the momentum equation. *J. Comput. Appl. Math.* **2017**, *309*, 359–370.
23. Socolofsky, S.A.; Jirka, G.H. Environmental fluid mechanics. Part I: Mass transfer and diffusion. In *Engineering-Lectures*, 2nd ed.; University of Karlsruhe, Institute of Hydromechanics: Karlsruhe, Germany, 2004.
24. Boudin, L.; Grec, B.; Salvarani, F. A mathematical and numerical analysis of the Maxwell–Stefan diffusion equations. *Discrete Contin. Dyn. Syst. Ser. B* **2012**, *17*, 1427–1440.
25. Duncan, J.B.; Toor, H.L. An experimental study of three component gas diffusion. *AIChE J.* **1962**, *8*, 38–41.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).