Hindawi Publishing Corporation
International Journal of Differential Equations
Volume 2013, Article ID 314290, 11 pages
http://dx.doi.org/10.1155/2013/314290

Research Article

Embedded Zassenhaus Expansion to Splitting Schemes: Theory and Multiphysics Applications

Jürgen Geiser

Institute of Physics, Felix-Hausdorff-Street 6, 17489 Greifswald, Germany

Correspondence should be addressed to Jürgen Geiser; geiser@mathematik.hu-berlin.de

Received 25 March 2013; Revised 5 August 2013; Accepted 19 August 2013

Academic Editor: Shuyu Sun

Copyright © 2013 Jürgen Geiser. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present some operator splitting methods improved by the use of the Zassenhaus product and designed for applications to multiphysics problems. We treat iterative splitting methods that can be improved by means of the Zassenhaus product formula, which is a sequential splitting scheme. The main idea for reducing the computation time needed by the iterative scheme is to embed fast and cheap Zassenhaus product schemes, since the computation of the commutators involved is very cheap, since we are dealing with nilpotent matrices. We discuss the coupling ideas of iterative and sequential splitting techniques and their convergence. While the iterative splitting schemes converge slowly in their first iterative steps, we improve the initial convergence rates by embedding the Zassenhaus product formula. The applications are to multiphysics problems in fluid dynamics. We consider phase models in computational fluid dynamics and analyse how to obtain higher order operator splitting methods based on the Zassenhaus product. The computational benefits derive from the use of sparse matrices, which arise from the spatial discretisation of the underlying partial differential equations. Since the Zassenhaus formula requires nearly constant CPU time due to its sparse commutators, we have accelerated the iterative splitting schemes.

1. Introduction

Our motivation to study the operator splitting methods comes from models in fluid dynamics, for example problems in bioremediation [1] or radioactive contaminants [2]. Such multiphysics problems are delicate, and the solver methods can be accelerated by decoupling the different physical behaviours; see [3, 4]. Based on the splitting error of all the splitting schemes, we have to take into account higher order ideas; see [5, 6]. While standard splitting methods deal with lower order convergence, see Lie splitting and Strang splitting [7, 8], we propose a combination of iterative splitting methods with an embedded Zassenhaus product formula. Such a combination allows reducing the splitting error and also reducing the CPU time needed; see [9]. Theoretically, we combine fixed point schemes (iterative splitting methods) with sequential splitting schemes (Zassenhaus products), which are connected with the theory of Lie groups and Lie algebras. Based on that relation, we can construct higher order splitting schemes for an underlying Lie algebra and improve the convergence results with cheap iterative schemes.

Historically, the efficiency of decoupling different physical processes into simpler processes, for example, convection and reaction, has helped to accelerate the solution process and is discussed in [10].

We propose the following ideas.

(i) Iterative splitting schemes are based on fixed point schemes, for example, waveform relaxation, which linearly improve the convergence order. Based on reducing the integral operators to cheaply computable matrices, they can be seen as solver methods; see [9].

(ii) Zassenhaus formula is based on nested commutators, which are the main keys to deriving higher order standard splitting schemes (e.g., Lie-Trotter and Strang splitting). They are simple to compute because of their nilpotent structure; see [11].

In this paper we study the following ordinary differential equations, that can arose of semidiscretized partial differential equations, for example, multiphysics problem, see [2, 3], or theoretical physics; see [12, 13].

We focus our attention on the case of two linear operators (i.e., we consider the Cauchy problem) as follows:

$$\frac{\partial c(t)}{\partial t} = Ac(t) + Bc(t), \quad \text{with } t \in [0,T], \; c(0) = c_0, \tag{1}$$

whereby the initial function c_0 is given, and A and B are assumed to be bounded linear operators in the Banach space $\mathbf{X}$ with $A, B : \mathbf{X} \to \mathbf{X}$. In realistic applications, the operators correspond to discretised matrices, for example, such as convection and diffusion matrices; see [2]. Further, they can be given matrices, for example, such as in perturbation theory [14].

For all such problems, the solution of the Cauchy problem (1) is given as

$$c(t) = \exp(A+B)\, c(0), \quad \text{with } t \in [0,T], \tag{2}$$

where $c(0)$ is the initial condition.

Here one of the main ideas to compute such delicate solutions is to separate or decouple the term $\exp(A+B)$ into simpler and parallel computational expression, for example, $\exp(A)\exp(B)$; see [10, 15]. To reduce the splitting error for such decompositions, iterative splitting method and products of exponential of noncommuting operators, see [16, 17], are important. We concentrate on iterative splitting methods and improve their convergence order with embedded Zassenhaus expansion; see [18].

While iterative splitting methods are cheap to implement and can be computed fast, one of their drawbacks is due to their initial process. Here, Zassenhaus expansion is an attractive tool to overcome such a drawback and improve the initial solutions to higher accuracy. While on the other hand, we deal only with lower nested commutators of the Zassenhaus formula, the drawbacks of stringent regularity conditions on the operators are less; see [19, 20].

The outline of the present paper is as follows. The splitting methods are discussed in Section 2. In Section 3, we present the numerical experiments and the benefits of the higher order splitting methods. Finally, we discuss future work in the area of iterative and noniterative splitting methods.

2. Splitting Methods

We consider the following splitting schemes:

(i) iterative splitting schemes,

(ii) Zassenhaus formula.

Their ideas and their underlying convergence analysis, based on the Cauchy problem (1), are discussed in the next subsections.

2.1. Iterative Operator Splitting. In the following, we develop two ideas of the iterative splitting schemes.

Iterative splitting with respect to one operator (also called a one-sided scheme) as follows:

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \quad \text{with } c_i(t^n) = c^n, \quad i = 1, 2, \ldots, m. \tag{3}$$

Iterative splitting with respect to two operators (also called a two-sided scheme) as follows:

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \quad \text{with } c_i(t^n) = c^n, \tag{4}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) + Bc_{i+1}(t), \quad \text{with } c_{i+1}(t^n) = c^n, \quad i = 1, 3, \ldots, 2m+1. \tag{5}$$

Theorem 1. *Let one considers the abstract Cauchy problem given in* (1). *One obtains for the one-sided iterative operator splitting method* (4) *the following accuracy:*

$$\left\| S_i - \exp((A+B)\tau) \right\| \le C\tau^i, \tag{6}$$

where S_i is the approximate solution for the ith iterative step and C is a constant that can be chosen uniformly on bounded time intervals.

Proof. The proof is given for $i = 1, 2, \ldots$ and with the consistency error $e_i(\tau) = c(\tau) - c_i(\tau)$, we have

$$\begin{aligned}
c_i(\tau) = {} & \exp(A\tau)\, c(t^n) \\
& + \int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1} - s\right)\right) B \exp(sA)\, c(t^n)\, ds \\
& + \int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1} - s_1\right)\right) B \\
& \quad \times \int_{t^n}^{t^{n+1}-s_1} \exp\left(\left(t^{n+1} - s_1 - s_2\right) A\right) \\
& \qquad \times B \exp(s_2 A)\, c(t^n)\, ds_2 ds_1 \\
& + \cdots + \int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1} - s_1\right)\right) B \\
& \qquad \cdots \int_{t^n}^{t^{n+1} - \sum_{j=1}^{i-2} s_j} \exp\left(\left(t^{n+1} - \sum_{j=1}^{i-1} s_j\right) A\right) \\
& \qquad \times Bc_{\text{init}}(s_i)\, ds_i ds_{i-1}, \ldots, ds_2 ds_1,
\end{aligned}$$

$$\begin{aligned}
c(\tau) = {} & \exp(A\tau)\, c(t^n) \\
& + \int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1} - s\right)\right) B \exp(sA)\, c(t^n)\, ds \\
& + \int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1} - s_1\right)\right) B \\
& \quad \times \int_{t^n}^{t^{n+1}-s_1} \exp\left(\left(t^{n+1} - s_1 - s_2\right) A\right) \\
& \qquad \times B \exp(s_2 A)\, c(t^n)\, ds_2 ds_1
\end{aligned}$$

$$+\cdots+\int_{t^n}^{t^{n+1}} \exp\left(A\left(t^{n+1}-s_1\right)\right)B \cdots \int_{t^n}^{t^{n+1}-\sum_{j=1}^{i-1}s_j} \exp\left(\left(t^{n+1}-\sum_{j=1}^{i}s_j\right)A\right) \times B\exp\left(\left(s_i\left(A+B\right)\right)c\left(t^n\right)\,ds_i,\ldots,ds_2ds_1. \tag{7}$$

We obtain

$$\begin{aligned}\left\|e_i\right\| &\le \left\|\exp\left(\left(A+B\right)\tau\right)c\left(t^n\right)-S_i\left(\tau\right)c_{\text{init}}\left(\tau\right)\right\| \\ &\le C\tau^i \max_{s_i\in[0,\tau]}\left\|\exp\left(s_i\left(A+B\right)\right)c\left(t^n\right)-c_{\text{init}}\left(s_i\right)\right\| \\ &\le C\tau^i\left\|\exp\left(\tau\left(A+B\right)\right)c\left(t^n\right)-c_{\text{init}}\left(\tau\right)\right\|,\end{aligned} \tag{8}$$

where i is the number of iterative steps.

The same idea can be applied to the even iterative scheme and also for alternating A and B. □

Remark 2. The accuracy of the initialisation $c_{\text{init}}(\tau)$ is important for conserving or improving the underlying iterative splitting scheme.

Here we have the following initialisation schemes:

$$\begin{aligned}c_1\left(\tau\right) &= \exp\left(A\tau\right)c\left(t^n\right) \longrightarrow \left\|e_i\right\| \le C\tau^i c\left(t^n\right), \\ c_1\left(\tau\right) &= \exp\left(A\tau\right)\exp\left(B\tau\right)c\left(t^n\right) \longrightarrow \left\|e_i\right\| \le C\tau^{i+1}c\left(t^n\right).\end{aligned} \tag{9}$$

So the slow convergence in the initial process $c_1(\tau)$ of the iterative splitting schemes can be improved by a cheap computable initial process $c_1(\tau) \approx \exp((A+B)\tau)$. We will next discuss the fast convergent Zassenhaus formula for the initial process.

2.2. Zassenhaus Formula (Sequential Splitting). The Zassenhaus formula is an extension to the exponential splitting schemes and can be given in Theorem 3; see also [20].

Theorem 3. *One assumes $\mathscr{L}(A,B)$ is the free Lie algebra generated by A and B. The kernel function of solution* (1), $\exp((A+B)t)$, *can be uniquely decomposed as*

$$\begin{aligned}\exp\left(\left(A+B\right)t\right) = {} & \exp\left(At\right)\exp\left(Bt\right)\pi_{i=1}^{j}\pi_{k=2}^{m} \\ & \times\exp\left(C_k\left(A,B\right)t^k\right) \\ & +\mathscr{O}\left(t^{m+1}\right),\end{aligned} \tag{10}$$

where $C_j(A,B) \in \mathscr{L}(A,B)$ is a homogeneous Lie polynomial of A and B of degree j.

Proof. The sketch of the proof is given in [20]. The idea is based on the existence of the BCH (Baker-Campbell-Hausdorff) formula, and we obtain

$$\exp\left(-At\right)\exp\left(\left(A+B\right)t\right) = \exp\left(Bt+D\left(t\right)\right), \tag{11}$$

where $D(t)$ is a Lie polynomial of degree >1, and

$$\exp\left(\left(A+B\right)t\right) = \exp\left(At\right)\exp\left(Bt\right)+\mathscr{O}\left(t^2\right). \tag{12}$$

Further we have

$$\exp\left(-Bt\right)\exp\left(Bt+D\left(t\right)\right) = \exp\left(C_2t^2+\widetilde{D}\left(t\right)\right), \tag{13}$$

where $\widetilde{D}(t)$ is a Lie polynomial of degree >2, and

$$\exp\left(\left(A+B\right)t\right) = \exp\left(At\right)\exp\left(Bt\right)\exp\left(C_2t^2\right)+\mathscr{O}\left(t^3\right). \tag{14}$$

We can repeat the recursive process and obtain by complete induction the results. □

The first Zassenhaus polynomials are given as

$$\begin{aligned}C_2\left(A,B\right) &= \frac{1}{2}\left[B,A\right], \\ C_3\left(A,B\right) &= \frac{1}{3}\left[\left[B,A\right],B\right]+\frac{1}{6}\left[\left[B,A\right],A\right], \\ C_4\left(A,B\right) &= \frac{1}{24}\left[\left[\left[B,A\right],A\right],A\right]+\frac{1}{8}\left[\left[\left[B,A\right],B\right],A\right] \\ &\quad+\frac{1}{8}\left[\left[\left[B,A\right],B\right],B\right].\end{aligned} \tag{15}$$

Here we see the benefit of such schemes in the computation of the commutators. If we assume that we have quickly computable commutators, for example, nilpotent matrices or sparse matrices, the computation time needed for the adjacent (or perturbed) operators $\exp(C_kt^k)$ of the scheme is much less than that of the main operators $\exp(At)\exp(Bt)$, and so the scheme is very effective and can be used as an initial solution of the iterative schemes.

Remark 4. We can also generalize the application of the Zassenhaus formula to decomposition methods with existing BCH formulas, for example,

(i) A-B splitting or Lie-Trotter splitting,

(ii) Strang splitting.

Example 5. (1) *A-B or Lie-Trotter Splitting*. For the Lie-Trotter splitting, there exists a BCH formula, and we have derived the polynomials in Theorem 3.

(2) *Strang Splitting*. There exists also a BCH formula, which is given as

$$\exp\left(\frac{At}{2}\right)\exp\left(Bt\right)\exp\left(\frac{At}{2}\right) = \exp\left(tS_1+t^3S_3+t^5S_5+\cdots\right), \tag{16}$$

where the $S_i(A,B) \in \mathscr{L}(A,B)$ is a homogeneous Lie polynomial in A and B of degree i; see [21].

The BCH formula is given as

$$\begin{aligned}\exp\left(\left(\frac{A}{2}+\frac{B}{2}\right)t\right) &= \Pi_{k=2}^{\infty}\exp\left(\widetilde{C}_kt^k\right)\exp\left(\frac{A}{2t}\right)\exp\left(\frac{B}{2t}\right), \\ \exp\left(\left(\frac{B}{2}+\frac{A}{2}\right)t\right) &= \exp\left(\frac{B}{2t}\right)\exp\left(\frac{A}{2t}\right)\Pi_{k=2}^{\infty}\exp\left(C_kt^k\right),\end{aligned} \tag{17}$$

and so there exists a new product based on the Zassenhaus polynomials as follows:

$$\Pi_{k=3}^{\infty} \exp\left(D_k t^k\right) = \Pi_{k=2}^{\infty} \exp\left(\widetilde{C}_k t^k\right) \Pi_{k=2}^{\infty} \exp\left(C_k t^k\right), \quad (18)$$

and we obtain a higher order scheme; see also [22].

Remark 6. The application of the Zassenhaus formula to partial differential equations, for example, second-order parabolic evolution equations, needs additional stringent regularity conditions on the operators. Due to the nested commutators, which are needed to apply the Zassenhaus formula, one needs at least higher regularity conditions.

We need some functional analytical results on the domains of the operators. The domains of the operators have to satisfy the following compatibility conditions for the first Zassenhaus exponents.

(i) Condition for the first term C_2 of the Zassenhaus formula

$$\mathscr{D}\left(L^2\right) \subset \mathscr{D}\left(A^2\right) \cap \mathscr{D}(AB) \cap \mathscr{D}(BA) \cap \mathscr{D}\left(B^2\right), \quad (19)$$

where $L = A+B$, and $\mathscr{D}$ is the domain of the operators.

(ii) Condition for the second term C_3 of the Zassenhaus formula

$$\begin{aligned} \mathscr{D}\left(L^3\right) \subset \mathscr{D}\left(A^3\right) \cap \mathscr{D}\left(A^2 B\right) \\ \cap \mathscr{D}(ABA) \cap \mathscr{D}\left(AB^2\right) \cap \mathscr{D}\left(B^3\right), \end{aligned} \quad (20)$$

where $L = A+B$, and $\mathscr{D}$ is the domain of the operators.

Example 7. If we assume to apply the Zassenhaus formula to a second-order diffusion equation and we assume to split each spatial direction, see the functional analytical framework [23], where the domains are given as

$$\begin{aligned} \mathscr{D}(L) &= H^2(\Omega) \cap H_0^1(\Omega), \\ \mathscr{D}(A) &= \left\{c \in L^2(\Omega); \partial_{xx} c, \partial_x c \in L^2\Omega, \right. \\ &\qquad \left. u(0, y) = u(1, y) = 0, \forall y \in (0,1)\right\}, \\ \mathscr{D}(B) &= \left\{c \in L^2(\Omega); \partial_{yy} c, \partial_y c \in L^2\Omega, \right. \\ &\qquad \left. u(x, 0) = u(x, 1) = 0, \forall x \in (0,1)\right\}, \end{aligned} \quad (21)$$

where $\Omega = (0,1)^2$.

Then we need the following regularity result:

(i) condition for the first term C_2 of the Zassenhaus formula

$$\mathscr{D}\left(L^2\right) \subset H^4(\Omega), \quad (22)$$

(ii) condition for the second term C_3 of the Zassenhaus formula

$$\mathscr{D}\left(L^3\right) \subset H^6(\Omega). \quad (23)$$

In the example, we need a very high regular domain to apply a first or second Zassenhaus exponent. This is one of the limitations and drawbacks of applying the Zassenhaus formula. In our applications, we restrict on applying the first or at least the second Zassenhaus exponent. We apply the third and fourth Zassenhaus exponents to the benchmark problems given as matrix equations without such restrictions to the domains; see [20].

2.3. Embedding the Zassenhaus Expansion into an Iterative Splitting Scheme. We now discuss the embedding of the Zassenhaus formula into the iterative operator splitting schemes.

Definition 8. We apply the Lie-Trotter splitting with the embedded Zassenhaus formula, which is given as

$$\begin{aligned} E_{\text{Zassen,Comp},1}(t) &= \exp(At)\exp(Bt), \\ E_{\text{Zassen,Comp},j}(t) &= \exp\left(C_j t^j\right), \quad \text{for } j \in 2, \ldots, i, \end{aligned} \quad (24)$$

where the sequential Zassenhaus operators are defined as

$$E_{\text{Zassen},i}(t) = \Pi_{j=1}^{i} E_{\text{Zassen,Comp},j}(t), \quad (25)$$

and we obtain an accuracy of $\mathcal{O}(t^i)$, and i is the number of Zassenhaus components.

In the following theorem, we present the improvement of the iterative splitting scheme with the Zassenhaus expansion.

Theorem 9. *One solves the initial value problem* (1). *We assume bounded and constant operators A, B.*

The initialisation process is done with the Zassenhaus formula

$$c_i(t) = E_{Zassen,i}(t)\, c_0, \quad (26)$$

where $E_{Zassen,i}(t)$ is given in (25).

Furthermore, the improved solutions are embedded in the iterative splitting scheme (4) *and one has after j iterative steps the following result:*

$$c_{i+j}(t) = E_{iter,j}(t)\, E_{Zassen,i}(t)\, c_0. \quad (27)$$

This has improved the error of the iterative scheme to $\mathcal{O}(t^{i+j})$.

Proof. The solution of the iterative splitting scheme (4) is

$$c_{i+j}(t) = E_{\text{iter},i}(t)\, c_{\text{init},j}, \quad (28)$$

where $S_i(t) = E_{\text{iter},i}(t)$, given in Theorem 1.

The initialisation is given by the Zassenhaus formula

$$c_{\text{init},j}(t) = E_{\text{Zassen},j}(t)\, c_0. \quad (29)$$

Combining both splitting schemes, we have a local error of

$$\begin{aligned} e_{i+j}(t) &= \left\| c(t) - c_{i+j}(t) \right\| \\ &= \left\| c(t) - E_{\text{iter},i}(t)\, E_{\text{Zassen},j}(t)\, c_0 \right\| \\ &\le \mathcal{O}\left(t^{i+j}\right) c_0. \end{aligned} \quad (30)$$

□

Remark 10. We obtain an acceleration of the iterative scheme by applying cheap higher order Zassenhaus formula. Such a benefit allows reducing to 2-3 iterative steps, while the initial solution is of a higher order. Taking into account the sparsity or nilpotency of the commutators, we now have a fast iterative splitting scheme.

Remark 11. For our numerical experiment, we apply the Zassenhaus formulaat the beginning of each integration step, as the so-called preprocessor. Therefore, we obtain more accurate starting conditions for the integration steps of the iterative splitting scheme. Based on the regularity conditions of the nested commutator, we apply only 1-2 Zassenhaus exponents.

In the following, we explain the final algorithm of the embedded Zassenhaus formula to the iterative splitting scheme.

Algorithm 12. We compute N time steps, where each time step Δt_n is given as $\Delta t_n = t^{n+1} - t^n$ with $n = 0, \ldots, N-1$. Further $c(t^0)$ is the initial condition of the Cauchy problem (1), and we compute $c(t^1), \ldots, c(t^N)$ successively. We have the following two steps; first we apply the Zassenhaus formula as initialization of $c_0(t^{n+1})$ for the iterative scheme; second we apply the iterative splitting scheme to the initial value problem (1) and compute the final solution $c(t^{n+1})$ as follows.

(1) We start with $j = 2$ and compute the Zassenhaus formula.

(2) We compute the matrix norm for the Zassenhauss exponents C_j.

If $\mathrm{err}_{\mathrm{Zass}_j} \leq \mathrm{err}_1$, then we start with the iterative splitting scheme with $c_0(t) = c_{\mathrm{Zass}_j}$, $i = 1$ and go to (3), else $j = j + 1$ and we go to (1).

(3) If the error of the iterative scheme $\mathrm{err}_{\mathrm{iter}} \leq \mathrm{err}_2$, then we are finished; else $i = i+1$, and we compute the next iterative step, till $\mathrm{err}_{\mathrm{iter}} \leq \mathrm{err}_2$.

3. Numerical Examples

In this section, we discuss examples of the use of embedding Zassenhaus product methods in iterative splitting schemes. In the first example, we demonstrate somewhat artificially how the proposed Zassenhaus splitting method avoids the splitting error up to a certain order. The next example shows the benefit of the combined splitting schemes for applications to multiphysics problems.

In the following, we consider a benchmark example to validate the theoretical results and real life problems to apply our schemes.

We applied the following splitting schemes, given in Table 1.

Table 1: Applied splitting schemes.

Abbreviation in the examples	Splitting methods
A-B	*A-B* splitting
Strang	Strang splitting
c_1	Iterative splitting with one iterative step
c_2	Iterative splitting with one iterative step and one Zassenhaus step as prestep
c_3	Iterative splitting with one iterative step and two Zassenhaus steps as presteps
c_4	Iterative splitting with two iterative steps and two Zassenhaus steps as presteps
c_5	Iterative splitting with three iterative steps and two Zassenhaus steps as presteps
c_6	Iterative splitting with four iterative steps and two Zassenhaus steps as presteps

3.1. Benchmark Problem: Matrix Examples. In the following, the idea of the matrix example is to present the decomposition into two simpler matrices. In context of the multiphysics problems, even such simple problems can be seen as test examples. We assume that each simple matrix is presenting a basic single physics problems, for example, reaction process between two species, while the combination of the two matrices results in a more complicate physics problem, for example, coupled reaction process between two species; see [2].

We consider the matrix equation

$$u'(t) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} u, \quad u(0) = u_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad t \in [0, 1], \tag{31}$$

the exact solution of which is

$$u(t) = \frac{2\left(e^{3t} - e^{-2t}\right)}{5}. \tag{32}$$

We split the matrix as

$$\begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix} = A + B, \tag{33}$$

where $[A, B] \neq 0$. For this simple example, we did not obtain nilpotent matrices for the commutators.

We apply 10–100 time steps for the computations. The final integration time of each single time step was less than <1 [sec], such that we need at least for such benchmark problems about 1–10 [sec]. Figure 1 presents the numerical errors, that is, the difference between the exact and the numerical solution.

Figure 2 presents the CPU times needed by the standard and the iterative splitting schemes. The application of the Zassenhaus formula did not increase, while the next Zassenhaus coefficients are at least nearly nilpotent matrices, and we can neglect the computational time. Moreover different time-steps did not influence the computations of the Zassenhaus exponents, which is a preprocess. First we compute and save all the Zassenhaus exponents, and later we multiply with the necessary time steps.

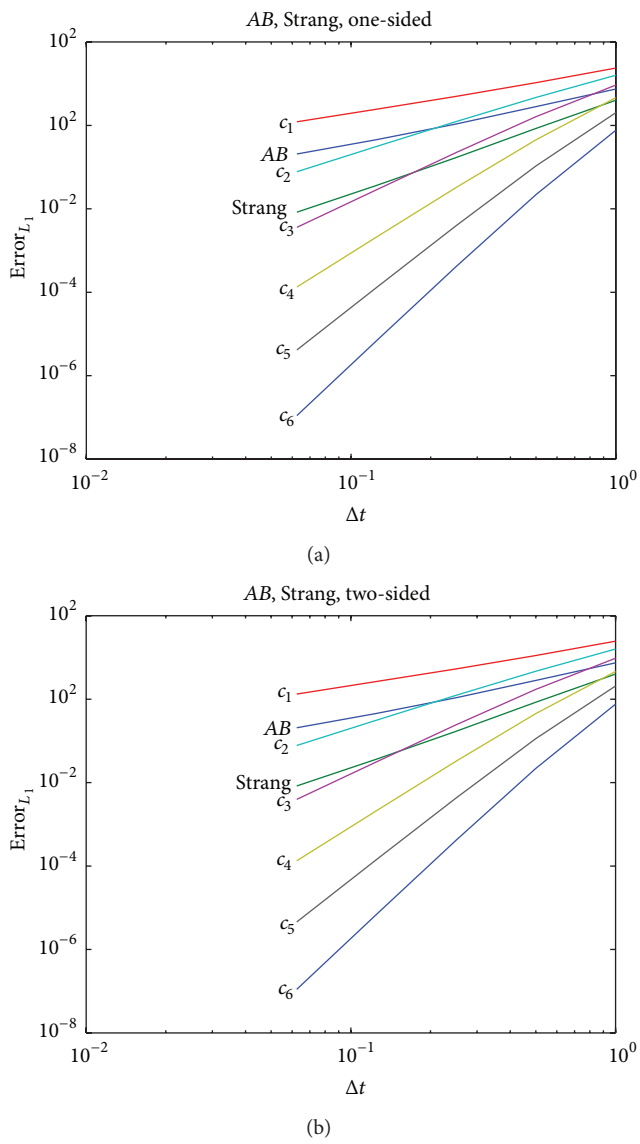

FIGURE 1: Numerical errors of the standard splitting scheme and the iterative schemes with the Zassenhaus formula, where we have at least $1, \ldots, 4$ iterative steps.

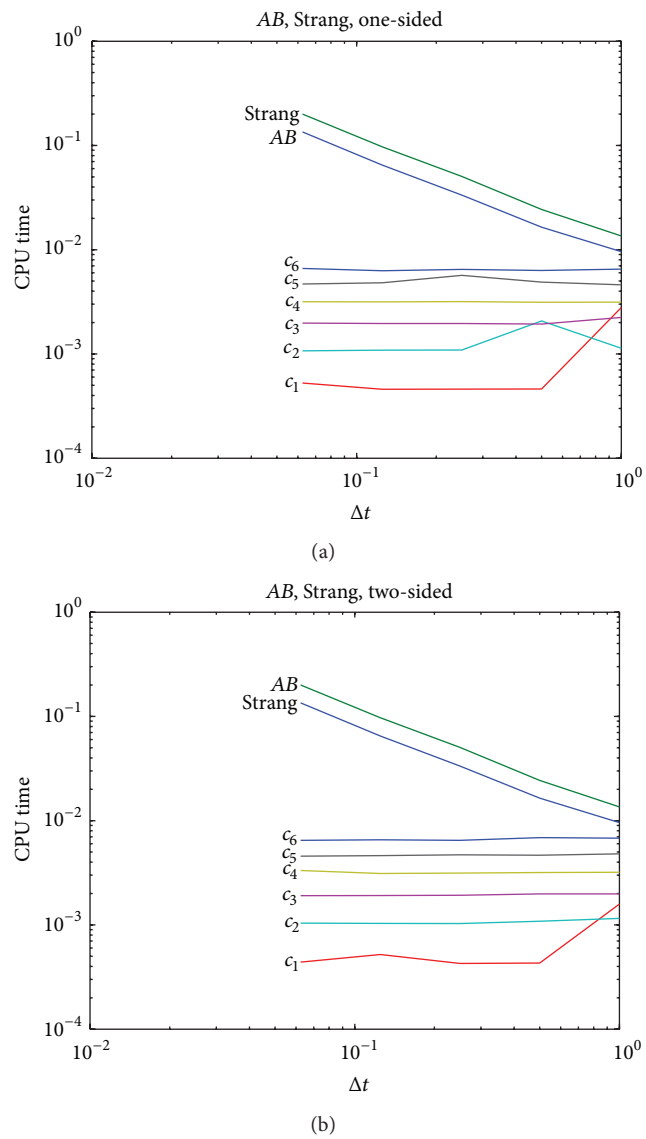

FIGURE 2: CPU times of the standard splitting scheme and the iterative schemes with the Zassenhaus formula, where we have at least $1, \ldots, 4$ iterative steps.

Remark 13. We see that the errors decrease significantly with increasing order of approximation for the initialisation process from the Zassenhaus splitting (we apply 1–3 Zassenhaus coefficients). Here we apply commutators, which did not reduce their matrix rank; therefore we have the same computation time as for the pure iterative schemes. Overall, we have their benefits in the application of the Zassenhaus product schemes to the standard Lie-Trotter or Strang-Marchuk splitting. Similar results are given with more iterative steps $i = 3, \ldots, 6$, as expected.

3.2. Real Life Problems: Multiphase Examples. In the next examples, we simulate coupled transport and reaction equations with different phases, so-called multiphase problems; see [9]. The equations are coupled with the reaction terms and are presented as follows:

$$\begin{aligned}
\partial_t R_i u_i + \nabla \cdot \mathbf{v} u_i &= -\lambda_i R_i u_i + \lambda_{i-1} R_{i-1} u_{i-1} \\
&\quad + \beta(-u_i + g_i), \quad \text{in } \Omega \times (0, T), \\
u_{i,0}(x) &= u_i(x, 0), \quad \text{on } \Omega, \\
\partial_t R_i g_i &= -\lambda_i R_i g_i + \lambda_{i-1} R_{i-1} g_{i-1} \\
&\quad + \beta(-g_i + u_i), \quad \text{in } \Omega \times (0, T), \\
g_{i,0}(x) &= g_i(x, 0), \quad \text{on } \Omega, \\
&\quad i = 1, \ldots, m,
\end{aligned} \tag{34}$$

where m is the number of equations and i is the index of each component. The unknown mobile concentrations $u_i = u_i(x,t)$ are taken in $\Omega \times (0,T) \subset \mathbb{R}^n \times \mathbb{R}^+$, where n is the spatial dimension. The unknown immobile concentrations $g_i = g_i(x,t)$ are taken in $\Omega \times (0,T) \subset \mathbb{R}^n \times \mathbb{R}^+$, where n is the spatial dimension. The retardation factors R_i are constant, and $R_i \geq 0$. The kinetic part is given by the factors λ_i. They are constant and $\lambda_i \geq 0$. For the initialisation of the kinetic part, we set $\lambda_0 = 0$. The kinetic part is linear and irreversible, so the successors have only one predecessor. The initial conditions are given for each component i as constants or linear impulses. For the boundary conditions, we have trivial inflow and outflow conditions with $u_i = 0$ at the inflow boundary. The transport part is given by the velocity $\mathbf{v} \in \mathbb{R}^n$ and is piecewise constant; see [2]. The exchange between the mobile and immobile part is given by β.

In the following, we discuss the one-phase and two-phase examples.

3.2.1. One-Phase Example. The next example is a simplified real life problem with a multiphase transport-reaction equation. We treat the mobile and immobile pores in a porous media. Such simulations are made about waste scenarios; see [2, 24].

We concentrate on the computational benefits of a fast computation of the mixed iterative scheme with the Zassenhaus formula.

The one phase equation is

$$\begin{gathered}
\partial_t c_1 + \nabla \cdot \mathbf{F}_1 c_1 = -\lambda_1 c_1, \quad \text{in } \Omega \times [0,t], \\
\partial_t c_2 + \nabla \cdot \mathbf{F}_2 c_2 = \lambda_1 c_1 - \lambda_2 c_2, \quad \text{in } \Omega \times [0,t], \\
\mathbf{F}_i = \mathbf{v}_i - D_i \nabla, \quad i = 1,2, \\
c_1(\mathbf{x},t) = c_{1,0}(\mathbf{x}), \qquad c_2(\mathbf{x},t) = c_{2,0}(\mathbf{x}), \quad \text{on } \Omega, \\
c_1(\mathbf{x},t) = c_{1,1}(\mathbf{x},t), \qquad c_2(\mathbf{x},t) = c_{2,1}(\mathbf{x},t), \\
\text{on } \partial\Omega \times [0,t].
\end{gathered} \tag{35}$$

Here the parameters take the values $\nu_1 = 0.1$, $\nu_2 = 0.05$, $D_1 = 0.01$, $D_2 = 0.005$, and $\lambda_1 = \lambda_2 = 0.1$.

We now turn to the finite difference schemes and semidiscretise the equation, with its convection and diffusion operators as follows:

$$\partial_t \mathbf{c} = (A_1 + A_2)\mathbf{c}. \tag{36}$$

We obtain the two matrices and decouple the diffusion and convection parts as follows:

$$\begin{gathered}
A_1 = \begin{pmatrix} A_{\text{diff}} & 0 \\ 0 & A_{\text{diff}} \end{pmatrix} \in \mathbb{R}^{2I \times 2I}, \\
A_2 = \begin{pmatrix} A_{\text{Conv}} & 0 \\ 0 & A_{\text{Conv}} \end{pmatrix} + \begin{pmatrix} -\Lambda_1 & 0 \\ \Lambda_1 & -\Lambda_2 \end{pmatrix} \in \mathbb{R}^{2I \times 2I}.
\end{gathered} \tag{37}$$

We apply the splitting method to the operators A_1 and A_2 given in Section 2.1.

The submatrices are

$$\begin{gathered}
A_{\text{diff},i} = \frac{D_i}{\Delta x^2} R_1 = \frac{D_i}{\Delta x^2} \cdot \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{I \times I}, \\
A_{\text{conv},i} = -\frac{v_i}{\Delta x} R_2 = -\frac{v_i}{\Delta x} \cdot \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{I \times I},
\end{gathered} \tag{38}$$

where I is the number of spatial points and Δx is the spatial step size. Consider

$$\begin{gathered}
\Lambda_1 = \begin{pmatrix} \lambda_1 & 0 & & & \\ 0 & \lambda_1 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & \lambda_1 & 0 \\ & & & 0 & \lambda_1 \end{pmatrix} \in \mathbb{R}^{I \times I}, \\
\Lambda_2 = \begin{pmatrix} \lambda_2 & 0 & & & \\ 0 & \lambda_2 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & \lambda_2 & 0 \\ & & & 0 & \lambda_2 \end{pmatrix} \in \mathbb{R}^{I \times I}.
\end{gathered} \tag{39}$$

We have the commutator

$$[A_1, A_2] = \begin{pmatrix} 0 & 0 \\ \dfrac{D_2 - D_1}{\Delta x^2} R_1 \Lambda_1 & 0 \end{pmatrix} \in \mathbb{R}^{2I \times 2I}, \tag{40}$$

where we assume that $[R_1, R_2] \approx 0$. This is a nilpotent commutator, and the computation time needed for the commutators of such operators is less than that for the full operators. Furthermore, we assume that we have a bounded spatial step size, which is given by $\Delta x = 0.1 \gg 0$, so that we scale the singular entry $(D_2 - D_1)/\Delta x^2 \approx 1$ and evade a blowup in such matrices.

We have obtained the optimal results of a basically constant CPU time for decreasing time steps, which increase in the standard scheme.

We apply 10–100 timesteps for the computations with 10–100 spatial points. The final integration time of each single time step was about 1–10 [sec], such that we need at least for such benchmark problems about 100–1000 [sec].

Figure 3 presents the numerical error, that is, the difference between the exact and the numerical solution.

Figure 4 presents the CPU time of the standard and the iterative splitting schemes. Based on the preprocess of computing the Zassenhaus exponents at the beginning and store the matrices, we could save additional computational time with larger time steps.

Remark 14. For the iterative schemes with embedded Zassenhaus products, we can reach improved results more quickly. The benefits come from the constant CPU time needed for

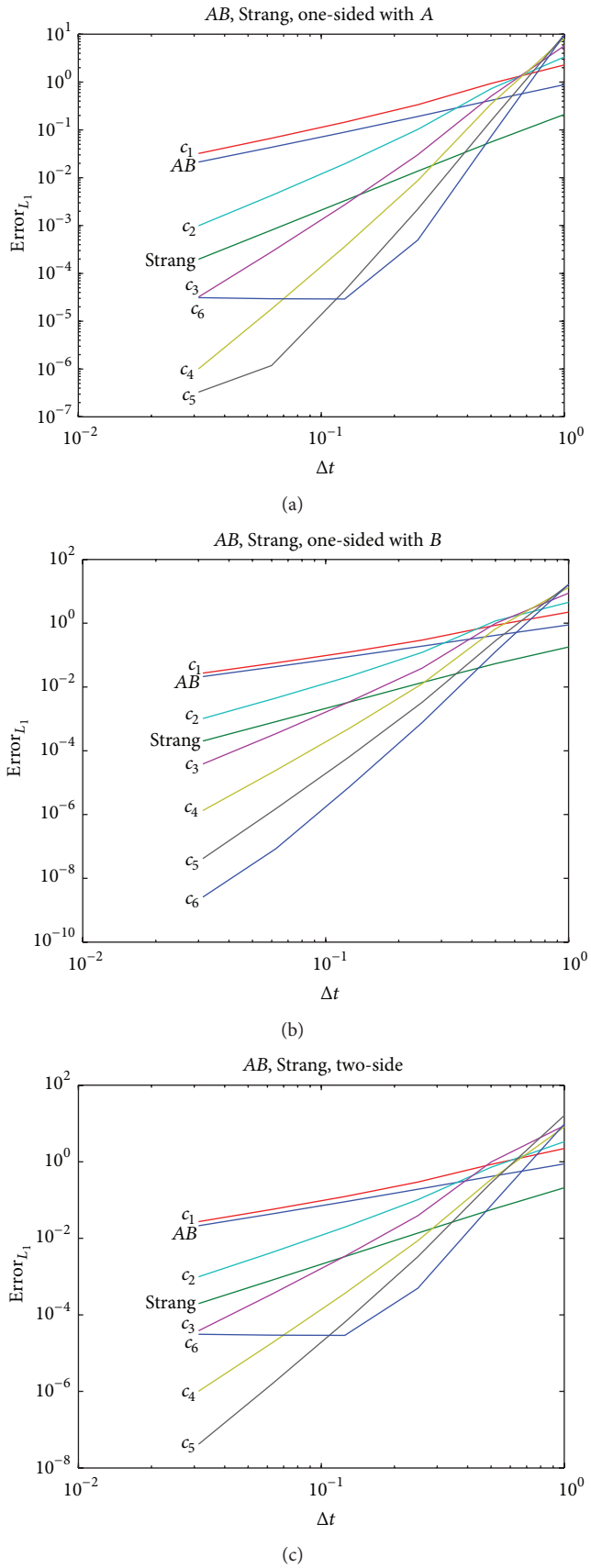

FIGURE 3: Numerical errors of the standard splitting scheme and the iterative schemes with the Zassenhaus formula, where we have at least $1, \ldots, 4$ iterative steps.

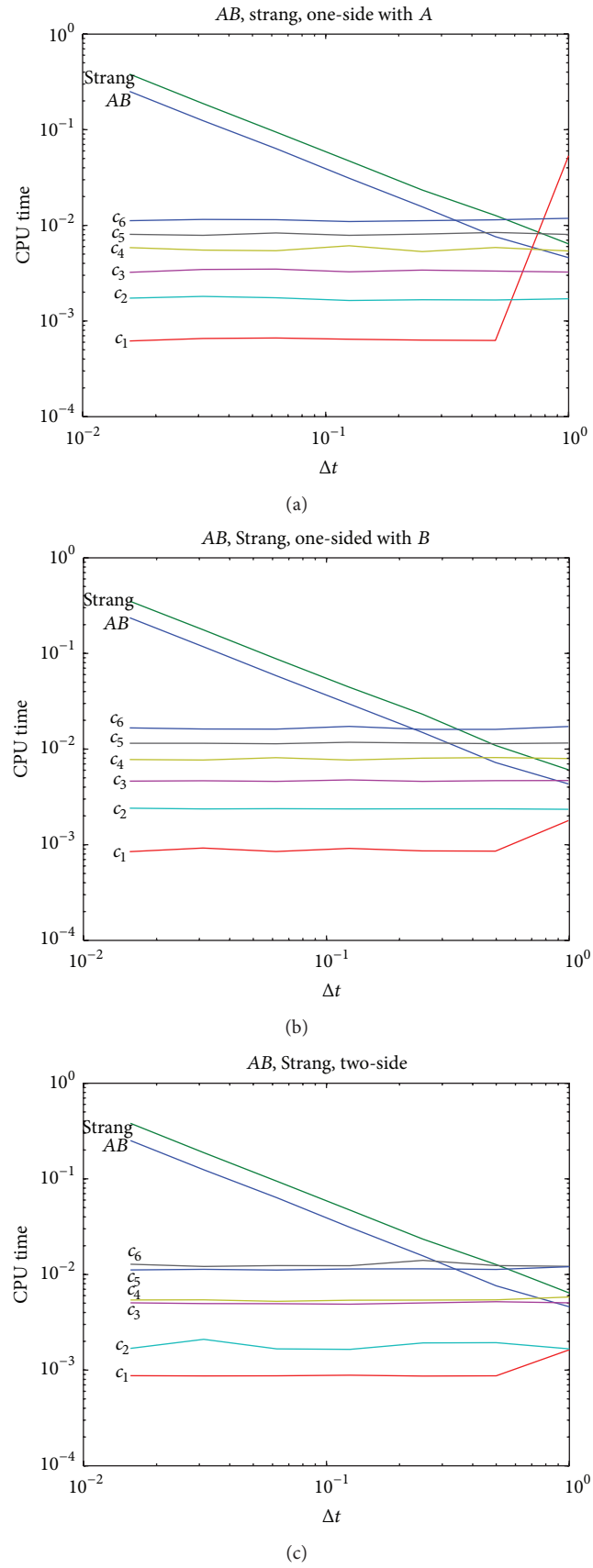

FIGURE 4: CPU time of the standard splitting scheme and the iterative schemes with the Zassenhaus formula, where we have at least $1, \ldots, 4$ iterative steps.

calculating the nilpotent commutators in the Zassenhaus formula. So it makes sense to have 1–3 Zassenhaus steps before starting the iterative scheme. At that point, only 2-3 iterative steps are needed to obtain more accurate results than those from the expensive standard schemes. Here, too, we obtain the best results with the one-sided iterative schemes.

Remark 15. Based on our restriction to D_1, D_2, and Δx with $(D_2 - D_1)/\Delta x^2 \approx 1$, we can control the delicate blowup of the computed commutators. By the way, if we neglect the restriction and Δx with $(D_2 - D_1)/\Delta x^2 \to \infty$, the commutator blows up and we lose the benefits of such an algorithm. This is a limitation of applying such an embedded Zassenhaus formula.

3.2.2. Two Phase Example. The next example is a more delicate real life problem involving a multiphase transport-reaction equation. We treat mobile and immobile pores in a porous media. Such simulations are made for waste scenarios; see [2].

We concentrate on the computational benefits of a fast computation of the commutators and their use for the initialisation of the iterative splitting scheme.

The equation is

$$\begin{gathered}
\partial_t c_1 + \nabla \cdot \mathbf{F} c_1 = g\left(-c_1 + c_{1,im}\right) - \lambda_1 c_1, \quad \text{in } \Omega \times [0,t], \\
\partial_t c_2 + \nabla \cdot \mathbf{F} c_2 = g\left(-c_2 + c_{2,im}\right) + \lambda_1 c_1 - \lambda_2 c_2, \quad \text{in } \Omega \times [0,t], \\
\mathbf{F} = \mathbf{v} - D\nabla, \\
\partial_t c_{1,im} = g\left(c_1 - c_{1,im}\right) - \lambda_1 c_{1,im}, \quad \text{in } \Omega \times [0,t], \\
\partial_t c_{2,im} = g\left(c_2 - c_{2,im}\right) + \lambda_1 c_{1,im} - \lambda_2 c_{2,im}, \quad \text{in } \Omega \times [0,t], \\
c_1(\mathbf{x},t) = c_{1,0}(\mathbf{x}), \qquad c_2(\mathbf{x},t) = c_{2,0}(\mathbf{x}), \quad \text{on } \Omega, \\
c_1(\mathbf{x},t) = c_{1,1}(\mathbf{x},t), \qquad c_2(\mathbf{x},t) = c_{2,1}(\mathbf{x},t), \\
\text{on } \partial\Omega \times [0,t], \\
c_{1,im}(\mathbf{x},t) = 0, \qquad c_{2,im}(\mathbf{x},t) = 0, \quad \text{on } \Omega, \\
c_{1,im}(\mathbf{x},t) = 0, \qquad c_{2,im}(\mathbf{x},t) = 0, \quad \text{on } \partial\Omega \times [0,t].
\end{gathered} \tag{41}$$

Here the parameters are $\nu = 0.1$, $D = 0.01$, $\lambda_1 = \lambda_2 = 0.1$, and $g = 0.01$.

We next treat the semidiscretised equation given by the matrices

$$\partial_t \mathbf{C} = \begin{pmatrix} A - \Lambda_1 - G & 0 & G & 0 \\ \Lambda_1 & A - \Lambda_2 - G & 0 & G \\ G & 0 & -\Lambda_1 - G & 0 \\ 0 & G & \Lambda_1 & -\Lambda_2 - G \end{pmatrix} \mathbf{C}, \tag{42}$$

where $\mathbf{C} = (\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_{1}}_{im}, \mathbf{c_{2}}_{im})^T$, while $\mathbf{c_1} = (c_{1,1}, \ldots, c_{1,I})$ is the solution of the first species in the mobile phase in each spatial discretisation point ($i = 1, \ldots, I$), and the same for the other solution vectors.

We have the following two operators for the splitting method:

$$\begin{aligned}
A = {} & \frac{D}{\Delta x^2} \cdot \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}, \\
& + \frac{\nu}{\Delta x} \cdot \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{I \times I},
\end{aligned} \tag{43}$$

where I is the number of spatial points.

Consider

$$\begin{aligned}
\Lambda_1 &= \begin{pmatrix} \lambda_1 & 0 & & & \\ 0 & \lambda_1 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & \lambda_1 & 0 \\ & & & 0 & \lambda_1 \end{pmatrix}, \\
\Lambda_2 &= \begin{pmatrix} \lambda_2 & 0 & & & \\ 0 & \lambda_2 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & \lambda_2 & 0 \\ & & & 0 & \lambda_2 \end{pmatrix} \in \mathbb{R}^{I \times I}, \\
G &= \begin{pmatrix} g & 0 & & & \\ 0 & g & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & g & 0 \\ & & & 0 & g \end{pmatrix} \in \mathbb{R}^{I \times I}.
\end{aligned} \tag{44}$$

We decouple this into the following matrices:

$$\begin{aligned}
\widetilde{A} &= \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{4I \times 4I}, \\
B_1 &= \begin{pmatrix} -\Lambda_1 & 0 & 0 & 0 \\ \Lambda_1 & -\Lambda_2 & 0 & 0 \\ 0 & 0 & -\Lambda_1 & 0 \\ 0 & 0 & \Lambda_1 & -\Lambda_2 \end{pmatrix}, \\
B_2 &= \begin{pmatrix} -G & 0 & G & 0 \\ 0 & -G & 0 & G \\ G & 0 & -G & 0 \\ 0 & G & 0 & -G \end{pmatrix} \in \mathbb{R}^{4I \times 4I}.
\end{aligned} \tag{45}$$

For the operators $\widetilde{A}$ and $\widetilde{B} = B_1 + B_2$ and B we apply the iterative splitting method given in Section 2.1.

Based on the decomposition, $\widetilde{A}$ is block diagonal and $\widetilde{B}$ is tridiagonal.

The commutator is

$$\left[\widetilde{A}, \widetilde{B}\right] = \begin{pmatrix} 0 & 0 & AG & 0 \\ 0 & 0 & 0 & AG \\ -AG & 0 & 0 & 0 \\ 0 & -AG & 0 & 0 \end{pmatrix} \in \mathbb{R}^{4I \times 4I}, \tag{46}$$

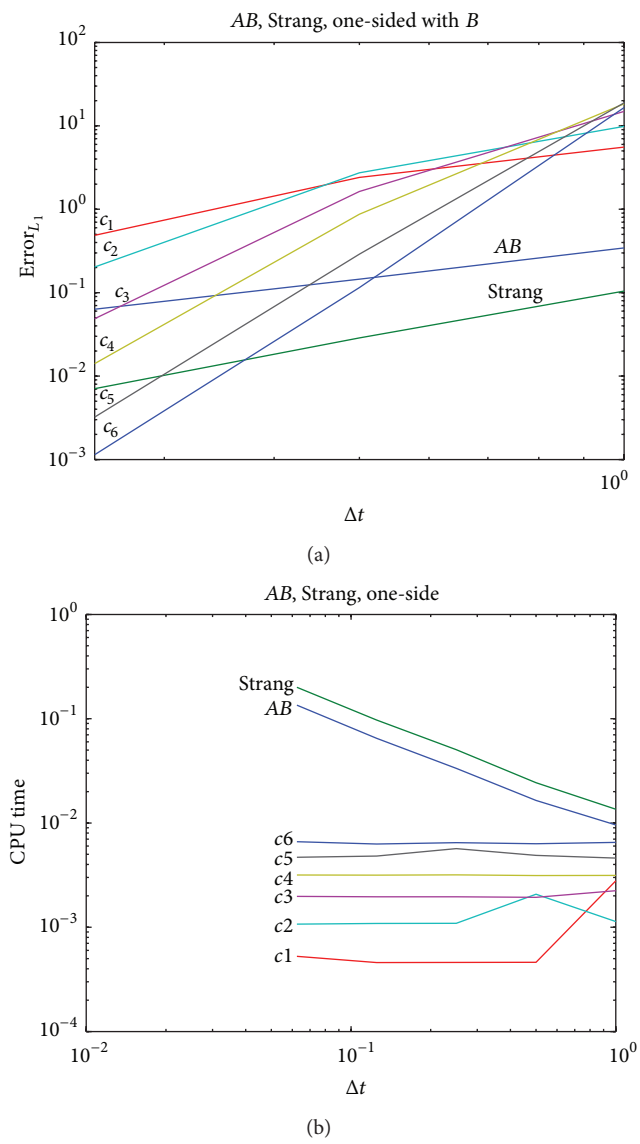

Figure 5: Numerical errors and CPU time of the standard splitting scheme and the iterative schemes with Zassenhaus formula, where we have at least $1, \ldots, 4$ iterative steps.

where we have a sparser operator than the main operator $\widetilde{B}$, and therefore we save computation time in computing such operators. Further we assume that we have a bounded spatial step size, which is given by $\Delta x = 0.1 \gg 0$, so that we scale the singular entry $D/\Delta x^2 \approx 1$ and $v/\Delta x \approx 1$.

Figure 5 presents the numerical error, that is, the difference between the exact and the numerical solution and the CPU time required. The optimal results are from the one-sided iterative schemes on the operator $\widetilde{B}$.

Remark 16. We obtain the same results as in the one-phase example, because of the basically constant CPU time for decreasing time steps of the iterative scheme; we obtain optimal results when applying 1–3 Zassenhaus steps. Therefore, with the embedded Zassenhaus formula, we can reach improved results more quickly than with the standard schemes. With 2-3 more iterative steps, we obtain more accurate results. With the one-sided iterative schemes, we reach the best convergence results when using the more delicate operator $\widetilde{B}$.

4. Conclusions and Discussion

In this paper, we have presented a novel splitting scheme combining the ideas of iterative and sequential schemes based on the Zassenhaus formula. Here the idea is to decouple the expensive iterative computation of the schemes based only on the matrix exponential, obtaining simpler embedded Zassenhaus schemes based on nilpotent commutators. Such combinations have the benefit of reducing the computation time because the commutators can be computed very cheaply. On the other hand, simple linear iterative steps can be done very cheaply if they are initialised with a sufficiently accurate starting solution, and this accelerates the solvers. The error analysis showed that these are stable methods for higher order schemes. In the applications, we were able to demonstrate the speedup from the Zassenhaus enhanced method and were able to apply it to multiphysics applications. In the future, we will concentrate on linear and nonlinear matrix-dependent schemes, that switch between noniterative and iterative schemes based on Zassenhaus products.

References

[1] R. E. Ewing, "Up-scaling of biological processes and multiphase ow in porous media," in *IIMA Volumes in Mathematics and Its Applications*, vol. 295, pp. 195–215, Springer, New York, NY, USA, 2002.

[2] J. Geiser, *Discretization and Simulation of Systems for Convection-Diffusion-Dispersion Reactions with Applications in Groundwater Contamination*, Groundwater Modelling, Management and Contamination, Nova Science; Monograph, New York, NY, USA, 2008.

[3] N. Antonic, C. J. van Duijn, W. Jäger, and A. Mikelic, *Multiscale Problems in Science and Technology: Challenges to Mathematical Analysis and Perspectives*, Springer, Berlin, Germany, 2002.

[4] E. Weinan, *Principle of Multiscale Modelling*, Cambridge University Press, Cambridge, UK, 2011.

[5] S. Blanes and F. Casas, "On the necessity of negative coefficients for operator splitting schemes of order higher than two," *Applied Numerical Mathematics*, vol. 54, no. 1, pp. 23–37, 2005.

[6] E. Hansen and A. Ostermann, "High order splitting methods for analytic semigroups exist," *BIT. Numerical Mathematics*, vol. 49, no. 3, pp. 527–542, 2009.

[7] G. Strang, "On the construction and comparison of difference schemes," *SIAM Journal on Numerical Analysis*, vol. 5, pp. 506–517, 1968.

[8] G. I. Marchuk, "Some applicatons of splitting-up methods to the solution of problems in mathematical physics," *Aplikace Matematiky*, vol. 1, pp. 103–132, 1968.

[9] J. Geiser, "Computing exponential for iterative splitting methods: algorithms and applications," *Journal of Applied Mathematics*, vol. 2011, Article ID 193781, 27 pages, 2011.

[10] R. I. McLachlan and G. R. W. Quispel, "Splitting methods," *Acta Numerica*, vol. 11, pp. 341–434, 2002.

[11] J. Geiser, G. Tanoğlu, and N. Gücüyenen, "Higher order operator splitting methods via Zassenhaus product formula: theory and applications," *Computers & Mathematics with Applications*, vol. 62, no. 4, pp. 1994–2015, 2011.

[12] G. H. Weiss and A. A. Maradudin, "The Baker-Hausdorff formula and a problem in crystal physics," *Journal of Mathematical Physics*, vol. 3, pp. 771–777, 1962.

[13] R. M. Wilcox, "Exponential operators and parameter differentiation in quantum physics," *Journal of Mathematical Physics*, vol. 8, pp. 962–982, 1967.

[14] J. Geiser, "An iterative splitting approach for linear integro-differential equations," *Applied Mathematics Letters*, vol. 26, no. 11, pp. 1048–1052, 2013.

[15] F. Casas and A. Murua, "An efficient algorithm for computing the Baker-Campbell-Hausdorff series and some of its applications," *Journal of Mathematical Physics*, vol. 50, no. 3, Article ID 033513, 2009.

[16] J. Geiser, *Iterative Splitting Methods for Differential Equations*, Chapman & Hall/CRC Numerical Analysis and Scientific Computing, CRC Press, Boca Raton, Fla, USA, 2011.

[17] S. Vandewalle, *Parallel Multigrid Waveform Relaxation for Parabolic Problems*, Teubner, Stuttgart, Germany, 1993.

[18] J. Geiser and G. Tanoglu, "Operator-splitting methods via the Zassenhaus product formula," *Applied Mathematics and Computation*, vol. 217, no. 9, pp. 4557–4575, 2011.

[19] F. Bayen, "On the convergence of the Zassenhaus formula," *Letters in Mathematical Physics*, vol. 3, no. 3, pp. 161–167, 1979.

[20] F. Casas, A. Murua, and M. Nadinic, "Efficient computation of the Zassenhaus formula," *Computer Physics Communications*, vol. 183, no. 11, pp. 2386–2391, 2012.

[21] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2002.

[22] H. Yoshida, "Construction of higher order symplectic integrators," *Physics Letters A*, vol. 150, no. 5–7, pp. 262–268, 1990.

[23] E. Faou, A. Ostermann, and K. Schratz, "Analysis of exponential splitting methods for inhomogeneous parabolic equations," http://arxiv.org/abs/1212.5827.

[24] J. Geiser, "Discretization methods with embedded analytical solutions for convection-diffusion dispersion-reaction equations and applications," *Journal of Engineering Mathematics*, vol. 57, no. 1, pp. 79–98, 2007.