Computers and Mathematics with Applications 62 (2011) 1994–2015

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Higher order operator splitting methods via Zassenhaus product formula: Theory and applications

Jürgen Geiser [a,*], Gamze Tanoğlu [b], Nurcan Gücüyenen [b]

[a] Humboldt Univeristät zu Berlin, Berlin, Germany

[b] Izmir Institute of Technology, Gulbahce Campus, Urla, Izmir, Turkey

ARTICLE INFO

Article history:
Received 24 May 2010
Received in revised form 22 June 2011
Accepted 22 June 2011

Keywords:
Operator splitting method
Iterative solver method
Weighting methods
Zassenhaus product
Parabolic differential equations

ABSTRACT

In this paper, we contribute higher order operator splitting methods improved by Zassenhaus product. We apply the contribution to classical and iterative splitting methods. The underlying analysis to obtain higher order operator splitting methods is presented. While applying the methods to partial differential equations, the benefits of balancing time and spatial scales are discussed to accelerate the methods.

The verification of the improved splitting methods are done with numerical examples. An individual handling of each operator with adapted standard higher order time-integrators is discussed. Finally, we conclude the higher order operator splitting methods.

1. Introduction

Our motivation to study the splitting methods are coming from model equations which simulate bio-remediation [1] or radioactive contaminants [2,3]. The efficiency of decoupling different physical processes (e.g., convection, reactions, etc.) helps to accelerate the solver process; see [4].

In this paper, we study the following mathematical equation:

$$\partial_t \, c_i + \nabla \cdot (\mathbf{v} c_i - D \nabla c_i) = f_i(c_1, \dots, c_n), \quad \text{for } i = 1, \dots, n. \tag{1}$$

The unknown $c(x,t) = (c_i(x,t), \dots, c_n(x,t))^t$ is considered in $\Omega \times (0,T) \subset \mathbb{R}^d \times \mathbb{R}$, the space-dimension is given by d. The velocity $\mathbf{v}$ is constant and D is the diffusion–dispersion tensor. The reaction $f_i(c_1, \dots, c_n)$ is a function of all unknowns c_i and couple the equations.

The aim of this paper is to study a novel splitting method which improves operator splitting methods. By weighting methods which embed the so-called Zassenhaus product (see [5]), we improve the initial and starting conditions of the splitting process. To apply the methods, the discretization for the time scales is done by combining explicit and implicit methods. The main advantage is using the standard implicit and explicit Runge–Kutta method and embed this method in an iterative solver.

For the iterative operator splitting methods (see the framework [6]), the delicate problem of low convergence (see [7]) can be improved by starting with sufficient accurate initial conditions. This is satisfied by weighting the method with the help of the Zassenhaus products.

The outline of the paper is as follows. The operator splitting methods are introduced in Section 2. Improvements of standard splitting methods to higher order splitting methods are discussed in Section 3. In Sections 4 and 5, we discuss extension with the Zassenhaus product and the balancing of time and space discretization methods. In Section 6, we present the numerical experiments and the benefits of the higher order splitting methods. Finally, we discuss future works in the area of iterative and non-iterative methods.

* Corresponding author.
E-mail addresses: geiser@mathematik.hu-berlin.de (J. Geiser), gamzetanoglu@iyte.edu.tr (G. Tanoğlu).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.06.043

2. Operator splitting methods

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

$$\frac{\partial c(t)}{\partial t} = Ac(t) + Bc(t), \quad \text{with } t \in [0, T], c(0) = c_0, \tag{2}$$

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 corresponds to physical operators such as convection and diffusion operators. We consider the following operators splitting schemes:

1. Sequential operator splitting: A–B splitting

$$\frac{\partial c^*(t)}{\partial t} = Ac^*(t) \quad \text{with } t \in [t^n, t^{n+1}] \quad \text{and} \quad c^*(t^n) = c^n_{sp} \tag{3}$$

$$\frac{\partial c^{**}(t)}{\partial t} = Bc^{**}(t) \quad \text{with } t \in [t^n, t^{n+1}] \quad \text{and} \quad c^{**}(t^n) = c^*(t^{n+1}), \tag{4}$$

for $n = 0, 1, \dots, N-1$, while $t^0 = 0$ and $t^N = T$, further $c^n_{sp} = c_0$ is given from (2). The approximated split solution at the point $t = t^{n+1}$ is defined as $c^{n+1}_{sp} = c^{**}(t^{n+1})$.

2. Strang–Marchuk operator splitting: A–B–A splitting, see [8,9]

$$\frac{\partial c^*(t)}{\partial t} = Ac^*(t) \quad \text{with } t \in [t^n, t^{n+1/2}] \quad \text{and} \quad c^*(t^n) = c^n_{sp} \tag{5}$$

$$\frac{\partial c^{**}(t)}{\partial t} = Bc^{**}(t) \quad \text{with } t \in [t^n, t^{n+1}] \quad \text{and} \quad c^{**}(t^n) = c^*(t^{n+1/2}), \tag{6}$$

$$\frac{\partial c^{***}(t)}{\partial t} = Ac^*(t) \quad \text{with } t \in [t^{n+1/2}, t^{n+1}] \quad \text{and} \quad c^{***}(t^{n+1/2}) = c^{**}(t^{n+1}), \tag{7}$$

where $t^{n+1/2} = t^n + 0.5\tau_n$, and the approximated split solution at the point $t = t^{n+1}$ is defined as $c^{n+1}_{sp} = c^{***}(t^{n+1})$.

3. Iterative splitting with respect to one operator

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \quad \text{with } c_i(t^n) = c^n, i = 1, 2, \dots, m. \tag{8}$$

4. Iterative splitting with respect to alternating operators

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \quad \text{with } c_i(t^n) = c^n \quad i = 1, 2, \dots, j, \tag{9}$$

$$\frac{\partial c_i(t)}{\partial t} = Ac_{i-1}(t) + Bc_i(t), \quad \text{with } c_{i+1}(t^n) = c^n, \quad i = j+1, j+2, \dots, m. \tag{10}$$

In addition, $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at the time level $t = t^n$. The split approximation at the time level $t = t^{n+1}$ is defined as $c^{n+1} = c_{2m+1}(t^{n+1})$. (Clearly, the function $c_{i+1}(t)$ depends on the interval $[t^n, t^{n+1}]$, too, but, for the sake of simplicity, in our notation we omit the dependence on n.)

3. Higher order operator splitting methods

Often standard splitting methods have the problem to be less effective in the rate of the convergence and CPU times. Here we propose the followings to overcome these difficulties:

- Initialization: improve the starting conditions via Zassenhaus product formula,
- Accelerated the subproblems via *Weighted Polynomials*,
- Extended operator splitting methods via Zassenhaus product formula.

3.1. Classical operator splitting errors

The main problem is the initialization.
Often the $c_0(t) = c(t^n)$ or $c_0(t) = 0$ are to far from the result, see

$$\|c(t) - c_0(t)\| \le err, \tag{11}$$

where err is a given starting error.

By the way the standard initialization errors are local first order, see

$$\|c(t) - c_n\| \leq \|(\exp((A+B)t) - I)c_n\|, \tag{12}$$
$$\|c(t) - c_0(t)\| \leq O(t), \tag{13}$$

and are global of zero order and are to large at all.

Here the ideas of prestepping methods, e.g. A–B splitting or Strang splitting as first or second order exponential splitting schemes can reduce the initial error.

See for the A–B splitting we have a global first order scheme

$$c_0(t) = \exp(At)\exp(Bt)c_n, \tag{14}$$
$$\|c(t) - c_0(t)\| \leq O(t^2), \tag{15}$$

where for the Strang or A–B–A splitting we have a global second order scheme

$$c_0(t) = \exp(A/2t)\exp(Bt)\exp(A/2t)c_n, \tag{16}$$
$$\|c(t) - c_0(t)\| \leq O(t^3). \tag{17}$$

Remark 3.1.1. Here we obtain a starting condition of first or second order and we can improve standard splitting scheme in the initialization process to higher order schemes.

3.2. Higher order A–B splitting by initialization

In this subsection, we improve the order of the A–B splitting via Zassenhaus product formula as follows:

Theorem 3.2.1. *We solve the initial value problem (3) and (4). We assume bounded and constant operators A and B.*

The consistency error of the A –B splitting is $\mathcal{O}(t)$, then we can improve the error of the A –B splitting scheme to $\mathcal{O}(t^p)$, $p > 1$ by improving the starting conditions c_0 as

$$c_0 = (\pi_{j=2}^{p}\exp(C_j t^j))c_0$$

where C_j is called as Zassenhaus exponents given in [10], thus local splitting error of A –B splitting method can be read as follows

$$\begin{aligned}\rho_n &= (\exp(\tau_n(A+B)) - \exp(\tau_n B)\exp(\tau_n A))c_{sp}^n \\ &= C_T\,\tau_n^{p+1} + \mathcal{O}(\tau_n^{p+2})\end{aligned} \tag{18}$$

where C_T is a function of Lie brackets of A and B.

Proof of Theorem 3.2.1. Let us consider the subinterval $[0, t]$, where $\tau = t$, the solution of the subproblem (3) is:

$$c^*(t) = \exp(At)c_0 \tag{19}$$

after improving the initialization we have

$$c^*(t) = \exp(At)(\pi_{j=2}^{p}\exp(C_j t^j))c_0 \tag{20}$$

the solution of the subproblem (4) becomes

$$\begin{aligned}c^{**}(t) &= \exp(Bt)\ \exp(At)(\pi_{j=2}^{p}\exp(C_j t^j))c_0 \\ &= \exp((B+A)t)c_0 + \mathcal{O}(t^{p+1})\end{aligned} \tag{21}$$

with the help of the Zassenhaus product formula. □

Remark 3.2.2. For example, the second order A–B splitting after improving the initialization is

$$\begin{aligned}c^{**}(t) &= \exp(Bt)\ \exp(At)\ \exp\left(-\frac{1}{2}[B,A]t^2\right)c_0 \\ &= \exp((B+A)t)c_0 + \mathcal{O}(t^3)\end{aligned} \tag{22}$$

and the third order A–B splitting, see [11,12], after improving the initialization is

$$\begin{aligned}c^{**}(t) &= \exp(Bt)\ \exp(At)\ \exp\left(-\frac{1}{2}[B,A]t^2\right)\ \exp\left(\left(\frac{1}{6}[B,[B,A]] - \frac{1}{3}[A,[A,B]]\right)t^3\right)c_0 \\ &= \exp((B+A)t)c_0 + \mathcal{O}(t^4).\end{aligned} \tag{23}$$

3.3. Higher order A–B–A splitting by accelerating the subproblems via Weighted Polynomials

In literature, Strang–Marchuk or A–B–A splitting is given as

$$\exp(At/2)\ \exp(Bt)\ \exp(At/2) = \exp((A+B)t) + \mathcal{O}(t^3)$$

since we would like to use the Zassenhaus product formula given as

$$\exp((A+B)t) = \exp(At)\ \exp(Bt)(\pi_{j=2}^{p}\exp(C_j t^j)) + \mathcal{O}(t^{p+1}) \tag{24}$$

in order to obtain higher order A–B–A splitting, we present the idea of the *Weighted Polynomials* in the following theorem:

Theorem 3.3.1. *We solve the initial value problem (5)–(7) on the subinterval* $[0, t]$. *We assume bounded and constant operators A and B.*

The consistency error of the A –B –A splitting is $O(t^2)$, *then we can improve the error of the A –B –A splitting scheme to* $O(t^p), p > 2$ *by applying the following steps:*

- *Step 1: Improve the starting conditions* $c^*(0) = c_0$ *as*

  $$c^*(0) = (\pi_{j=2}^{p}\exp(C_j t^j))c_0$$

  where C_j *is called as Zassenhaus exponents given in* [10],
- *Step 2: Accelerate* $c^{**}(0)$ *as*

  $$c^{**}(0) = (\exp(-A\,t/2))c^*(t/2),$$
- *Step 3: Accelerate* $c^{***}(t/2)$ *as*

  $$c^{***}(t/2) = (\exp(At/2))c^{**}(t),$$

  thus the order of the improved A –B –A splitting method can be read as follows

  $$\exp(\tilde{A}t/2)\ \exp(Bt)\ \exp(\tilde{\tilde{A}}t/2) = \exp((A+B)t) + \mathcal{O}(t^{p+1}).$$

  where $\exp(\tilde{A}t/2) = \exp(At)$ *and* $\tilde{\tilde{A}} = \pi_{j=2}^{p}\exp(C_j(t/2)^j)$, *and* C_j *are the Zassenhaus exponents.*

Proof of Theorem 3.3.1. Let us consider the subinterval $[0, t]$, the solution of subproblem (5) is:

$$c^*(t) = \exp(At)c_0 \tag{25}$$

after improving the initialization we have

$$c^*(t) = \exp(At)(\pi_{j=2}^{p}\exp(C_j t^j))c_0. \tag{26}$$

Next accelerate $c^*(t)$ as

$$c^*(t) = \exp(-At)c^*(t) \tag{27}$$

the solution of subproblem (6) becomes

$$\begin{aligned} c^{**}(t) &= \exp(Bt)c^*(t/2) \\ &= \exp(Bt)\ \exp(-At/2)\exp(At/2)(\pi_{j=2}^{p}\exp(C_j(t/2)^j))c_0 \end{aligned} \tag{28}$$

or

$$c^{**}(t) = \exp(Bt)\ (\pi_{j=2}^{p}\exp(C_j(t/2)^j))c_0 \tag{29}$$

since $[-A/2, A/2] = 0$. Finally, the acceleration of $c^{**}(t)$ is given by the equation

$$c^{**}(t) = \exp(At/2)\exp(Bt)\ (\pi_{j=2}^{p}\exp(C_j(t/2)^j))c_0, \tag{30}$$

then the solution of subproblem (6) becomes

$$c^{***}(t) = \exp(At/2)\exp(At/2)\exp(Bt)\ (\pi_{j=2}^{p}\exp(C_j(t/2)^j))c_0 \tag{31}$$

or

$$c^{***}(t) = \exp(At)\exp(Bt)\ (\pi_{j=2}^{p}\exp(C_j(t/2)^j))c_0 \tag{32}$$

since $[A/2, A/2] = 0$. This can be rewritten as

$$\begin{aligned} c^{***}(t) &= \exp(At)\exp(Bt)\ (\pi_{j=2}^{p}\exp(\tilde{C}_j(t)^j))c_0 \\ &= \exp((A+B)t) + \mathcal{O}(t^{p+1}), \end{aligned} \tag{33}$$

where $\tilde{C}_j = \frac{1}{2^j}C_j$ and C_j are the exponents of the Zassenhaus product formula. □

3.4. Higher order A–B–A splitting based on Zassenhaus product formula

In this subsection, we first derive the Zassenhaus exponents by using the same approach given [10] for the A–B–A splitting. Again, using the formal power series expansion of exponential function, the Zassenhaus product for two non-commutative variables A and B for A–B–A splitting may be written as

$$\begin{aligned} \exp((A+B)t) &= \sum_{k=0}^{\infty} \frac{1}{k!}(A+B)^k t^k = I + (A+B)t + \left(\frac{1}{2}A^2 + \frac{1}{2}AB + \frac{1}{2}BA + \frac{1}{2}B^2\right)t^2 + \cdots \\ &= \left(I + \frac{At}{2} + \frac{A^2t^2}{8} + \cdots\right)\left(I + Bt + \frac{B^2t^2}{2} + \cdots\right)\left(I + \frac{At}{2} + \frac{A^2t^2}{8} + \cdots\right)\prod_{n=3}^{\infty}(e^{D_n t^n}) \\ &= \exp\left(\frac{At}{2}\right)\exp(Bt)\exp\left(\frac{At}{2}\right)\exp(D_3t^3)\exp(D_4t^4)\ldots. \end{aligned} \tag{34}$$

Our aim is to compute the polynomials D_n which are function of commutators [., [[.,]]]. One can find these polynomials by comparison method or Witschel's method. But, we use the ideas and notations which were first presented in [10] as follows:

Let $\tau_1, \ldots \tau_n$ be arbitrary commutative variables and let $J = (J_{ij})$, $K = (K_{ij})$, and $L = (L_{ij})$ be three $(n+1) \times (n+1)$ matrices defined by $J_{ij} = 0$, $K_{ij} = 0$ and $L_{ij} = 0$ for $i > j$ and

$$J_{ij} = \frac{1}{(j-i)!}.\prod_{k=i}^{j-1}(1+\tau_k), \qquad K_{ij} = \frac{(-1)^{(i+j)}}{(j-i)!} \quad \text{and} \quad L_{ij} = \frac{(-1)^{(i+j)}}{(j-i)!}\prod_{k=i}^{j-1}\tau_k \quad \text{for } i \le j.$$

Furthermore, they define the $(n+1) \times (n+1)$ matrices P and Q by

$$P_{ij} = \delta_{i+1,j} \quad \text{and} \quad Q_{ij} = \delta_{i+1,j}\tau_i$$

where $\delta_{i,j}$ is Kronecker delta. The operator U is defined in [10] but $a = At$, $b = Bt$, $c_n = D_n t^n$ We state the following corollary:

Corollary 3.4.1. *The Zassenhaus exponent c_3 defined in Eq. (34) is obtained in terms of the 4×4 matrices L, K, H where $H = \exp(1/2P + Q + 1/2P)$, $K = \exp(-1/2P)$ and $L = \exp(-Q)$ as $c_3 = U(K\ L\ K\ H)_{1,4}$. For $n > 3$, the Zassenhaus exponents c_k is given in terms of the corresponding $(n+1) \times (n+1)$ matrices as*

$$c_k = U(\exp(-C_{k-1})\ldots\exp(-C_3)\ K\ L\ K\ H)_{1,n+1}. \tag{35}$$

Here, $C_m (1 < m < n)$ are the Zassenhaus exponents written in terms of the $(n+1) \times (n+1)$ matrices P and Q, and the index $(1, n+1)$ indicates the upper right element of a matrix.

Proof of Corollary 3.4.1. Each element $n \in \mathbb{N}$ can be written as

$$\exp(P+Q) = \exp(P/2)\exp(Q)\exp(P/2)\prod_{i=3}^{n}(\exp(C_i)). \tag{36}$$

Therefore one obtains

$$\exp(C_n) = \exp(-C_{n-1})\ldots\exp(-C_3)\exp(e^{P/2})\exp(Q)\exp(P/2)\exp(P+Q). \tag{37}$$

The rest of the proof is the same as in [10]. □

Corollary 3.4.2. *The Zassenhaus exponent D_3 given in Eq. (34) can be found as*

$$D_3 = \frac{1}{24}[B, [B, A]] - \frac{1}{12}[A, [A, B]], \tag{38}$$

by comparing the exact solution given in (34) with the expansion up to the order $\mathcal{O}(t^4)$ given Eq. (34). Thus if the weight $w_3 = I + D_3t^3$ is chosen and multiplied by the initial condition, the order of the A –B –A splitting becomes $\mathcal{O}(t^3)$.

Proof of Corollary 3.4.2. The splitting error of Strang splitting or A–B–A splitting is

$$\rho = \exp((A+B)t) - \exp(At/2)\exp(Bt)\exp(At/2) \tag{39}$$

$$= \left(\frac{1}{24}[B, [B, A]] - \frac{1}{12}[A, [A, B]]\right)t^3. \tag{40}$$

The coefficient of t^3 given in the expansion

$$e^{(A+B)t} = e^{\frac{At}{2}} e^{B} e^{\frac{At}{2}} e^{D_3 t^3} + \mathcal{O}(t^4), \tag{41}$$

is

$$D_3 + \frac{(A+B)^3}{3!} - \rho,$$

thus if we choose $D_3 = \rho$, the splitting error becomes $\mathcal{O}(t^3)$. □

3.5. Higher order iterative splitting based on Zassenhaus product formula

Waveform relaxation (one operator):

Theorem 3.5.1. *We solve the initial value problem (8). We assume bounded and constant operators A, B. The initial step is given as* $c_1(t) = \exp(At)c_0$.

Then we can improve the error of the iterative scheme to $\mathcal{O}(t^{i+j})$ *by multiplying a weighted function with the kernel* $\omega_j(t) = \exp(Bt)\Pi_{k=2}^{i} \exp(\hat{c}_k t^k) + O(t^j)$ *to* $c_{i-1}(t)$*, where* $\hat{c}_k$ *are the so called Zassenhaus exponents, see* [10].

Proof of Theorem 3.5.1. The iterative scheme is for the step c_2 as:

$$c_i = \exp(At)c_0 + \int_0^t \exp(A(t-s))B\exp(As)c_0\, ds, \tag{42}$$

where $c_1(t) = \exp(As)c_0$.

We improve the method as:

$$\tilde{c}_1(t) = \exp(At)\exp(Bt)\Pi_{k=2}^{i} \exp(\hat{c}_k t^k)c_0, \tag{43}$$

where we obtain the exact solution:

$$c_i = \exp(At)c_0 + \int_0^t \exp(A(t-s))B\exp((A+B)s)c_0\, ds, \tag{44}$$

based on comparison with the formulation of the Zassenhaus product formula (see [5])

$$\exp((A+B)t) = \exp(At)\exp(Bt)\Pi_{k=2}^{\infty} \exp(\hat{c}_k t^k), \tag{45}$$

we can derive the weights are given as:

$$w_i(t) = \exp(Bt)\exp(\hat{c_2})\exp(\hat{c_3})\ldots\exp(\hat{c_i}) + \mathcal{O}(t^{i+1}), \tag{46}$$

where $\hat{c}_i, i = 2, \ldots, \infty$ are Zassenhaus exponents as follows:

$$\begin{aligned} \hat{c_2} &= -1/2[A, B], \\ \hat{c_3} &= (-1/3[B, [B, A]] + 1/6[A, [A, B]]), \\ \hat{c_4} &= (-1/24[[[A, B], A], A] - 1/8[[[A, B], A], B] - 1/8[[[A, B], B], B]). \end{aligned} \tag{47}$$

Thus some examples for weights are given as:

$$w_1(t) = I + Bt, \tag{48}$$

$$w_2(t) = I + Bt + B^2t^2/2 - 1/2[A, B]t^2, \tag{49}$$

$$w_3(t) = I + Bt + B^2t^2/2 - 1/2[A, B]t^2 + B^3/3t^3 + (-1/3[B, [B, A]] + 1/6[A, [A, B]])t^3 - 1/2B[A, B]t^3, \tag{50}$$

$$\begin{aligned} w_4(t) = {} & I + Bt + B^2t^2/2 - 1/2[A, B]t^2 + B^3/3!t^3 + (-1/3[B, [B, A]] + 1/6[A, [A, B]])t^3 - 1/2B[A, B]t^4 \\ & + B^4/4!t^4 + (-1/24[[[A, B], A], A] - 1/8[[[A, B], A], B] - 1/8[[[A, B], B], B])t^4 \\ & + (-1/3B[B, [B, A]] + 1/6B[A, [A, B]])t^4 - \left(\frac{B^2}{4}[A, B]\right) t^4 + 1/4[A, B]^2 t^4. \end{aligned} \tag{51}$$

Same can be done for the iterative splitting method. □

3.5.1. Higher order iterative splitting with respect alternating operators based comparison or Witschel's method

Consider Eqs. (9) and (10), the exact solution of the iteration can be found by using variation of constant formula as follows:

$$c_i(t) = \exp(At)c_0 + \exp(At)\int_0^t \exp(-As)Bc_{i-1}\,ds, \tag{52}$$

$$c_{i+1}(t) = \exp(Bt)c_0 + \exp(Bt)\int_0^t \exp(-Bs)Ac_i\,ds. \tag{53}$$

Assume that $c_{i-1} = 0$ then
for $i = 1$,

$$c_1(t) = \exp(At)c_0, \tag{54}$$

for $i = 2$,

$$\begin{aligned} c_2(t) &= \exp(Bt)\left(I + \int_0^t \exp(-Bs)A\exp(As)\,ds\right)c_0 \\ &= \exp(Bt)\left(I + \int_0^t (I - Bs)A(I + As)\ ds\right)c_0 \\ &= \exp(Bt)\left(I + \int_0^t (A + A^2 s) - ABs + \mathcal{O}(s^2)\ ds\right)c_0 \\ &= \exp(Bt)\left(I + At + \frac{A^2t^2}{2} - \frac{ABt^2}{2}\right)c_0 + \mathcal{O}(t^3) \\ &= \left(I + Bt + \frac{B^2t^2}{2}\right)\left(I + At + \frac{A^2t^2}{2} - \frac{ABt^2}{2}\right)c_0 + \mathcal{O}(t^3) \\ &= \left(I + (A+B)t + \frac{B^2t^2}{2} + BAt^2 + \frac{A^2t^2}{2} - \frac{ABt^2}{2}\right)c_0 + \mathcal{O}(t^3) \\ &= \exp(At + Bt) + \mathcal{O}(t^2). \end{aligned} \tag{55}$$

In next theorem, we show how to increase the order of the accuracy with respect to the *Weighted Polynomials*.

Theorem 3.5.2. *There exists a Weighted Polynomial so that the order of the accuracy of iterative splitting with alternating operators can be increased up to* $\mathcal{O}(t^3)$.

Proof of Theorem 3.5.2. We give the proof by construction in the following steps:

- Step 1: Start the initiation as $c_{i-1} = 0$.
- Step 2: Accelerate the c_1 as $c_1 = (I + Wt)c_1$.
- Step 3: Compute c_2 by using Eq. (53) as

$$\begin{aligned} c_2(t) &= \exp(Bt)\left(I + \int_0^t \exp(-Bs)\exp(As)(I + Ws)\,ds\right)c_0 \\ &= \exp(Bt)\left(I + \int_0^t ((I - Bs)A(I + As)(I + Ws) + \mathcal{O}(s^2))\,ds\right)c_0 \\ &= \exp(Bt)\left(I + \int_0^t (A + AWs - BAs + A^2 s + \mathcal{O}(s^2))\,ds\right)c_0 \\ &= \exp(Bt)\left(I + At + \frac{AWt^2}{2} - \frac{BAt^2}{2} + \frac{A^2t^2}{2}\right)c_0 + \mathcal{O}(t^3). \end{aligned} \tag{56}$$

- Step 4: Next expand $\exp(Bt)$ up to $\mathcal{O}(t^3)$,

$$\begin{aligned} c_2(t) &= \left(I + Bt + \frac{B^2t^2}{2}\right)\left(I + At + \frac{AWt^2}{2} - \frac{BAt^2}{2} + \frac{A^2t^2}{2}\right)c_0 + \mathcal{O}(t^3) \\ &= \left(I + (A+B)t + \frac{B^2t^2}{2} + BAt^2 + \frac{AWt^2}{2} + \frac{A^2t^2}{2}\right)c_0 + \mathcal{O}(t^3). \end{aligned} \tag{57}$$

- Step 5: Finally compare this with exact solution up to $\mathcal{O}(t^3)$ to find the commutator and W as follows. The exact solution of the problem is given by,

$$\begin{aligned} c_{exact} &= e^{(A+B)t} \\ &= \left(I + (A+B)t + \left(\frac{A^2}{2} + \frac{AB}{2} + \frac{BA}{2} + \frac{B^2}{2}\right)t^2\right)c_0 + \mathcal{O}(t^3), \end{aligned} \tag{58}$$

and the error can be found by subtracting Eq. (58) from (57) as follows

$$|c_{exact} - c_2| \leq \left(\frac{AB}{2} - \frac{AW}{2}\right)O(t^2) + \mathcal{O}(t^3). \tag{59}$$

From this expression if $W = B$, the order of the accuracy of iterative splitting with respect to alternating operators can be increased up to $O(t^3)$, thus we can find the *Weighted Polynomial* as follows:

$$w_1 = I + Bt. \tag{60}$$

Note that this is the same as the weight found in Eq. (48). We proved that the order of the accuracy of iterative splitting with respect to alternating operators can be increased up to $\mathcal{O}(t^3)$ via *Weighted Polynomial* defined in Eq. (60), Therefore,

$$|c_{exact} - c_{(i=2)}| \leq C\ \mathcal{O}(t^3), \tag{61}$$

where C is the function of Commutators. □

Theorem 3.5.3. *There exists a Weighted Polynomial so that the order of the accuracy of iterative splitting with alternating operators can be increased up to $O(t^4)$ after the second iteration.*

Proof of Theorem 3.5.3. We give the proof by construction in the following steps:
Step 1: Start the initiation as $c_{i-1} = 0$.
Step 2: Accelerate the c_1 as $c_1 = (I + W_1 t + W_2 t^2)c_1$.
Step 3: Compute c_2 by using Eq. (55) as

$$\begin{aligned} c_2(t) &= \exp(Bt)\left(I + \int_0^t \exp(-Bs)A\exp(As)(I + W_1 s + W_2 s^2)\,ds\right)c_0 \\ &= \exp(Bt)\left(I + \int_0^t \left(I - Bs + \frac{B^2s^2}{2}\right)A\left(I + As + \frac{A^2s^2}{2}\right)(I + W_1 s + W_2 s^2) + \mathcal{O}(s^2)\,ds\right)c_0 \\ &= \exp(Bt)\left(I + \left(\int_0^t (A + (A^2 + AW_1 - BA))s\right)ds\right. \\ &\quad \left. + \int_0^t \left(\frac{A^3}{2} + AW_1A - BA^2 + AW_2 - BAW_1 + \frac{B^2A}{2}\right)s^2\,ds\right)c_0 + \mathcal{O}(s^4). \end{aligned} \tag{62}$$

After integrating the expression in Eq. (62) on the right, $c_2(t)$ becomes

$$\begin{aligned} c_2(t) &= \exp(Bt)\left(I + At + (A^2 + AW_1 - BA)\frac{t^2}{2}\right) \\ &\quad + \left(\frac{A^3}{2} + \left(A^2W_1 - BA^2 + AW_2 - BAW_1 + \frac{B^2A}{2}\right)\frac{t^3}{3}\right)c_0 + \mathcal{O}(t^4). \end{aligned} \tag{63}$$

Step 4: Next expand $\exp(Bt)$ up to $O(t^3)$ and insert this into Eq. (62)

$$\begin{aligned} c_2(t) &= \left(I + Bt + \frac{B^2t^2}{2} + \frac{B^3t^3}{3}\right)\left(I + At + (A^2 + AW_1 - BA)\frac{t^2}{2}\right. \\ &\quad \left. + \left(\frac{A^3}{2} + A^2W_1 - BA^2 + AW_2 - BAW_1 + \frac{B^2A}{2}\right)\frac{t^3}{3}\right)c_0 + \mathcal{O}(t^4). \end{aligned} \tag{64}$$

Step 5: Finally by comparing this with exact solution up to $\mathcal{O}(t^3)$, which is given by

$$\begin{aligned} c_{exact} &= e^{(A+B)t} \\ &= \left(I + (A+B)t + \left(\frac{A^2}{2} + \frac{AB}{2} + \frac{BA}{2} + \frac{B^2}{2}\right)t^2\right. \\ &\quad \left. + \left(\frac{A^3}{6} + \frac{A^2B}{6} + \frac{ABA}{6} + \frac{AB^2}{6} + \frac{BA^2}{6} + \frac{BAB}{6} + \frac{B^2A}{6} + \frac{B^3}{6}\right)\right)c_0 + \mathcal{O}(t^4), \end{aligned} \tag{65}$$

then we find a weight function $w_2 = I + W_1 t + W_2 t^2$, where

$$W_1 = B, \qquad W_2 = \frac{B^2 - [A, B]}{2}.$$

Note that this is the same as the weight found in Eq. (49). Finally, by putting these values into the error which can be found by subtracting Eq. (65) from (64), we have

$$|c_{exact} - c_2| \leq D\ \mathcal{O}(t^4), \tag{66}$$

where D is the function of commutators and can be estimated. □

4. Extended splitting method based on Zassenhaus formula

The standard exponential splitting methods are based on the following decomposition idea:

$$\exp((A + B)t) = \Pi_{i=1}^{j} \exp(a_i At) \exp(b_i Bt) + O(t^{j+1}). \tag{67}$$

The extension to the exponential splitting schemes are given as:

$$\exp((A + B)t) = \Pi_{i=1}^{j} \exp(a_i At) \exp(b_i Bt) \Pi_{k=j}^{m} \exp(C_k t^k) + O(t^{m+1}), \tag{68}$$

where C_j is a function of Lie brackets of A and B.

Theorem 4.0.4. *The initial value problem (8) is solved by classical exponential splitting schemes. We assume bounded and constant operators A, B.*

Then we can find extensions based on the Zassenhaus formula given as

$$\exp((A + B)t) = \Pi_{i=1}^{j} \exp(a_i At) \exp(b_i Bt) \Pi_{k=j}^{m} \exp(C_k t^k) + O(t^{m+1}), \tag{69}$$

where C_j is a function of Lie brackets of A and B.

Proof of Theorem 4.0.4. (1) Lie–Trotter splitting:

For the Lie–Trotter splitting there exists coefficients with respect to the extension:

$$\exp((A + B)t) = \exp(At) \exp(Bt) \Pi_{k=2}^{\infty} \exp(C_k t^k), \tag{70}$$

where the coefficients C_k are given in [10].

Based on an existing BCH formula of the Lie–Trotter splitting one can apply the Zassenhaus formula.

(2) Strang splitting:

A existing BCH formula is given as:

$$\exp(At/2) \exp(Bt) \exp(At/2) = \exp(tS_1 + t^3 S_3 + t^5 S_5 + \cdots), \tag{71}$$

where the coefficients S_i are given as in [13].

There exists an Zassenhaus formula based on the BCH formula.

By the expression:

$$\exp((A/2 + B/2)t) = \Pi_{k=2}^{\infty} \exp(\tilde{C}_k t^k) \exp(A/2t) \exp(B/2t), \tag{72}$$

and

$$\exp((B/2 + A/2)t) = \exp(B/2t) \exp(A/2t) \Pi_{k=2}^{\infty} \exp(C_k t^k), \tag{73}$$

then there exists a new product:

$$\Pi_{k=3}^{\infty} \exp(D_k t^k) = \Pi_{k=2}^{\infty} \exp(\tilde{C}_k t^k) \Pi_{k=2}^{\infty} \exp(C_k t^k), \tag{74}$$

with one order higher, see also [14].

(3) General exponential splitting:

Same can be done with the general exponential splitting schemes. □

5. Balancing of time and spatial discretization

Splitting methods are important for partial differential equations, because of reducing computational time to solve the equations and accelerating the solver process, see [15].

Here additional balancing is taken into account, because of the spatial step.

The following theorem, addresses the delicate situation of time and spatial steps and the fact of reducing the theoretical promised order of the scheme:

Theorem 5.0.5. *We solve the initial value problem by applying iterative operator splitting scheme (9) and (10). We assume bounded and constant operators A, B. While iterating i-time with A and j-time with B the theoretical order is given as $O(t^{i+j})$ The initial step is given as $c_1(t) = \exp(At)\exp(Bt)c_0$.*

Then we reduce order of the iterative scheme to $O(t^i)$, while norm of B is larger or equal than $O(\frac{1}{t})$ same is also with the operator A.

So the balancing below the so called CFL condition is important to preserve the order of the splitting method.

Proof of Theorem 5.0.5. The theoretical order of the iterative splitting scheme is given as:

$\|c_{i+j} - c\| \leq \|A\|\,\|B\|,\ t^{i+j} + O(t^{i+j+1})$ where $\|A\| = \rho(A)$ is the spectral or the maximum eigenvalue of operator A and $\|B\| = \rho(B)$ is the spectral or the maximum eigenvalue of operator B.

Based on the spatial discretization we have the following eigenvalues:

$\rho(A) = \frac{a_1}{\Delta x^p}$, $\rho(B) = \frac{a_2}{\Delta x^q}$ where we have a p-th order spatial discretization of A and a q-th order spatial discretization of B, a_1, a_2 are the diagonal entries of the finite difference stencil, see [16].

If we assume to have a CFL condition ≥ 1 for the operator B we obtain:

$$\frac{a_1}{\Delta x^p} t \geq 1, \tag{75}$$

and therefore:

$$\|A_2\|^j t^j = O(1). \tag{76}$$

We lost the order for operator B and reduce to the order of the operator A.

Same can be done for operator A.

Therefore we have a necessary restriction to preserve the order of the splitting method given as:

$$O(1) \geq \rho(A) \geq O\left(\frac{1}{t}\right).$$

We preserve the order:

$$\|B\|^j t^j = O(t^j). \quad \square \tag{77}$$

Remark 5.0.6. By using implicit method for the discretization scheme, we did not couple the time scale and the spatial scale by a CFL condition and are so fare independent of the reduction but taken into account less accurate results.

6. Numerical examples

We consider the following test problems in order to verify our theoretical findings in the previous sections.

We discuss the application of the Zassenhaus product to iterative methods (e.g. iterative operator splitting methods) and non-iterative methods (e.g. Lie–Trotter, Strang splitting).

6.1. First test-example: eigenvalue problem

We first deal with the following ordinary differential equation

$$\frac{\partial c(t)}{\partial t} = \lambda c(t), \quad \text{with } t \in [0, T], c(0) = 1, \tag{78}$$

and we assume $\lambda = 1$.

We divide our ODE's in sub-equations after applying the one operator splitting method as following

$$\frac{\partial c_i(t)}{\partial t} = -\lambda_1 c_i(t) + \lambda_2 c_{i-1}(t), \quad \text{with } c_i(t^n) = c^n,\ i = 1, 2, \ldots, m, \tag{79}$$

where $\lambda_1 = -1$ and $\lambda_2 = 2$, initial condition is $c(0) = 1$. The exact solution of the problem is $c_{exact} = \exp(x)$. We applied the midpoint rule to find the approximate solution. Since there is no splitting error we have following proposition:

Proposition 6.1.1. *The order of the accuracy of the iterative splitting (8) is two after applying the midpoint to each sub-equations.*

Proof of Proposition 6.1.1. We obtain following finite difference approximation after discretization Eq. (8) by midpoint method on $[0, \tau]$,

$$c_i(\tau) = \chi_1 c^0 + \chi_2 \frac{\tau}{2}\lambda_2(c_{i-1}(0) + c_{i-1}(\tau))(\tau) \quad i = 1, 2, \dots, m, \tag{80}$$

where χ_1 is defined as follows if $|\frac{\lambda_1}{2}\tau| < 1$,

$$\chi_1 = \frac{1 - \frac{\lambda_1}{2}\tau}{1 + \frac{\lambda_1}{2}\tau} \tag{81}$$

$$= 1 - (\lambda_1\tau) + \frac{\lambda_1^2\tau^2}{2} + O(\tau^3) \tag{82}$$

$$= e^{-\lambda_1\tau} + O(\tau^3), \tag{83}$$

and Pade Approximation of the $e^{-\lambda_1\tau}$ up to the order $O(\tau^3)$ and χ_2 is defined as follows if $|\frac{\lambda_1}{2}\tau| < 1$

$$\chi_2 = \frac{\lambda_2\frac{\tau}{2}}{1 + \frac{\lambda_1\tau}{2}} \tag{84}$$

$$= \frac{\lambda_2\tau}{2} - \frac{\lambda_1\lambda_2\tau^2}{4} + O(\tau^3), \tag{85}$$

assume that $c_{i-1} = 0$, by inserting this into Eq. (80), we have for $i = 1$,

$$c_1(\tau) = \chi_1 c^0, \tag{86}$$

for $i = 2$,

$$c_2(\tau) = (\chi_1 + \chi_2(1 + \chi_1))c^0. \tag{87}$$

We can easily see that this approximation does not give the exact solution up to the second order, then we need to compute next iteration as follows,

$$c_3(\tau) = (\chi_1 + \chi_2(1 + (\chi_1 + \chi_2(1 + \chi_1))))c^0 \tag{88}$$

$$= \chi_1(1 + {\chi_1}^{-1}\chi_2(1 + (\chi_1 + \chi_2(1 + \chi_1))))c^0, \tag{89}$$

after expanding the terms up to third order we may see the following result

$$c_3(\tau) = e^{(-\lambda_1+\lambda_2)\tau} + O(\tau^3). \quad \square \tag{90}$$

In the first experiment, we exhibit the solution of the eigenvalue problem by using the weight in Eq. (48) as $w = 1+\tau\lambda_2$, since $W = B = \lambda_2$. Fig. 1 shows that the same order of accuracy can be reached by using the less iteration via *Weighted Polynomial*.

Remark 6.1.2. By applying to standard eigenvalue problems, we concentrate on the accuracy of each method without considering the splitting error. Here the weighted method can reach the same results as an iterative splitting method with one iteration step more. So weighted splitting helps to reduce computational time.

6.2. Second test-example: matrix problem

To see the error of the commutation we contribute a non-commutative example.

We deal with the following problem:

$$\frac{\partial u(t)}{\partial t} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} u, \tag{91}$$

with the initial conditions $u_0 = (1, -1)$ on the interval $[0, T]$.

The analytical solution is given by :

$$u(t) = \begin{pmatrix} e^{-t} \\ e^{t} \end{pmatrix}, \tag{92}$$

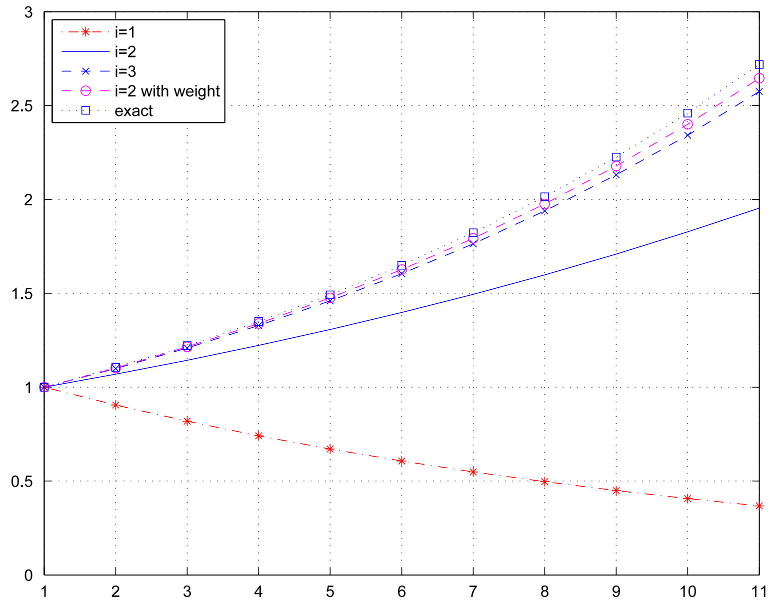

Fig. 1. Comparison of the solutions of eigenvalue problem obtained by midpoint method for different number of iterations and iteration with weight for $\Delta t = 0.01$.

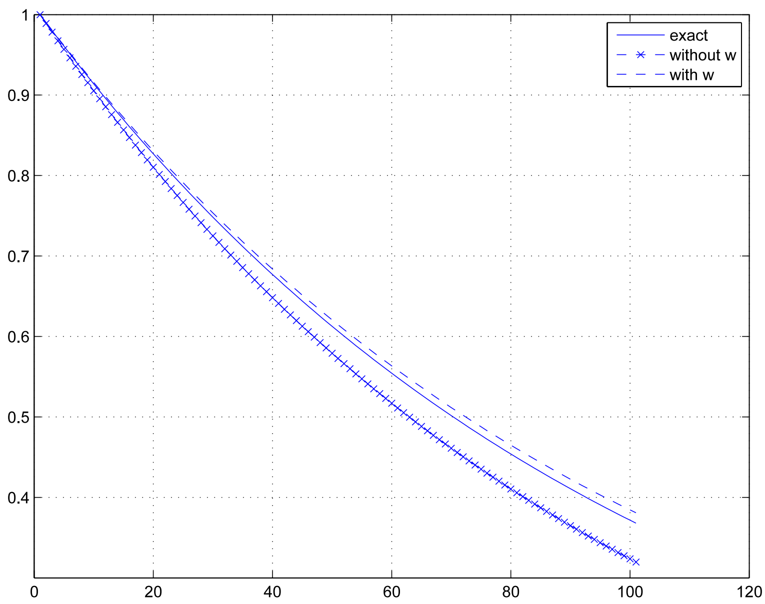

Fig. 2. Comparison of the solutions of matrix problem by one-operator splitting solved by the third order Runge–Kutta method with weight or without weight for $\Delta t = 0.01$.

We split our linear operators into two operators by setting:

$$\frac{\partial u(t)}{\partial t} = \begin{pmatrix} 2 & -1 \\ -1 & 0 \end{pmatrix} u + \begin{pmatrix} -2 & 2 \\ 2 & 0 \end{pmatrix} u, \tag{93}$$

Not that the matrices do not commute. For integration constants we use a step size of $\Delta t = 10^{-2}$. We apply the third order Runge–Kutta method to our iterative scheme with respect to the one operator. We compare the first component of the solution obtained from weighted and without weighted iterative scheme with exact solution in Fig. 2.

In Fig. 3, we show the rate of convergence on $[0, \triangle t]$ obtained from weighted and without weighted iterative scheme.

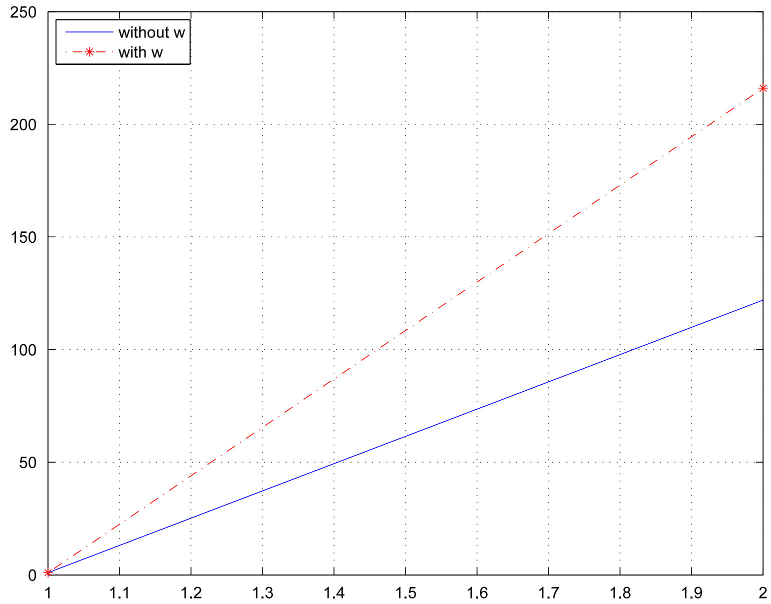

Fig. 3. Rate of the convergency of the matrix problem solved one-operator splitting solved by the third order Runge–Kutta method for $\Delta t = 0.02$.

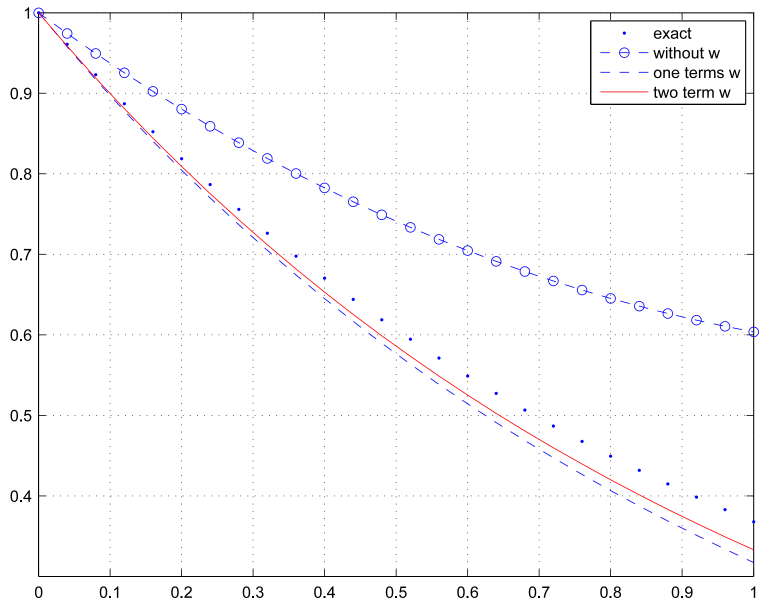

Fig. 4. Comparison of the solutions of matrix problem obtained by iterative splitting method solved the third order Runge–Kutta method with weight, without weight, one term weight, two terms weight for $\Delta t = 0.04$.

In Figs. 4 and 5, we compare the different weight polynomials, one term weight we mean $w_1 = I + Bt$, two term weight we mean

$$w_2 = I + Bt + (B^2 - [A, B])\frac{t^2}{2}$$

for $\Delta t = 0.04$ and $\Delta t = 0.02$, respectively for alternating operator splitting.

Next, we apply fourth order Runge–Kutta method with Lie–Trotter splitting to the same problem and compare the solutions without weight, one term weight and two term weight. Results are given in Figs. 6–11 and Table 1.

In Table 2, we used the weight obtained in Corollary 3.4.2 for Strang splitting solved by fourth order Runge–Kutta method:

Remark 6.2.1. We obtained the same benefits in the non-commuting case of a differential equation. The splitting error is also reduced by applying a weighted method and we save computational in having the same accuracy.

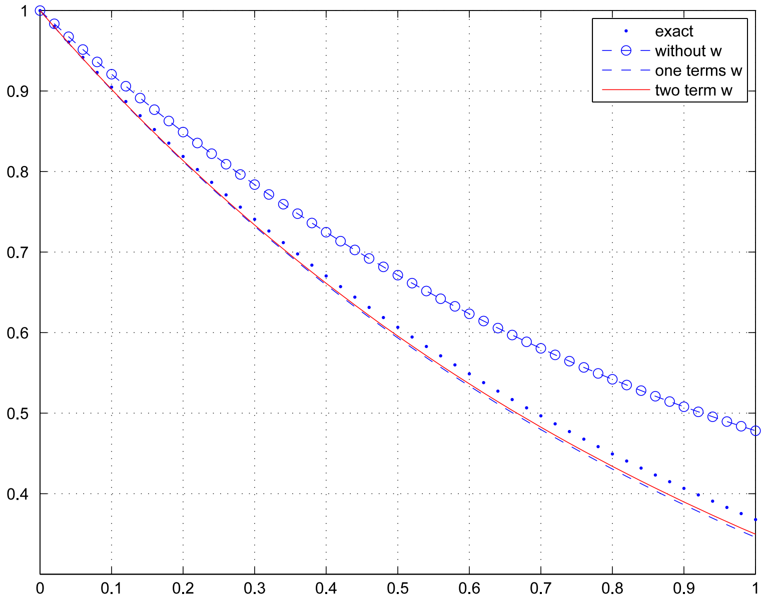

Fig. 5. Comparison of the solutions of matrix problem obtained by iterative splitting method and the third order Runge–Kutta method with weight, without weight, one term weight, two terms weight for $\Delta t = 0.02$.

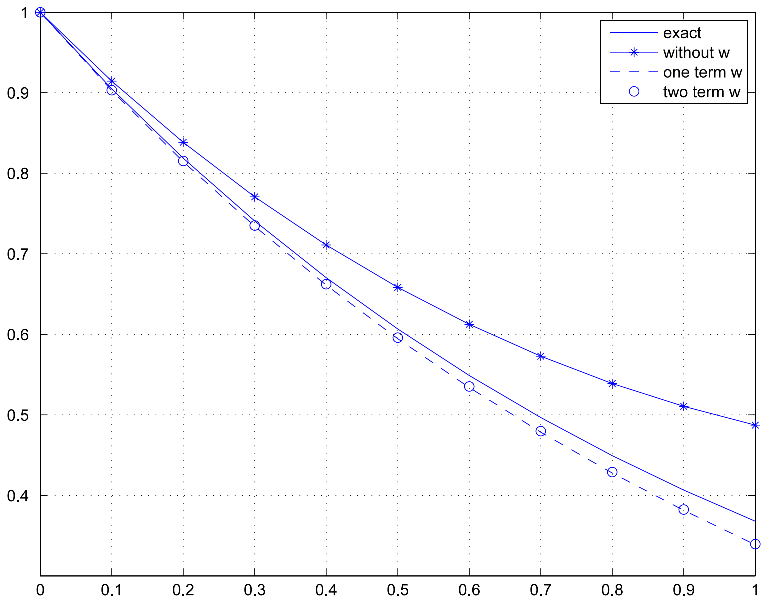

Fig. 6. Comparison of the solutions of matrix problem obtained by Lie–Trotter splitting solved by fourth order Runge–Kutta method without weight, one term weight, two term weight for $\Delta t = 0.01$.

Table 1
Comparison of errors for matrix problem solved by Lie–Trotter splitting and fourth order Runge–Kutta method for $\Delta t = 0.01$.

		err_{L_∞}	err_{L_1}
Lie–Trotter splitting	Without w	0.1194	0.0060
	With one w	0.0292	0.0014
	With two w	0.0284	0.0013

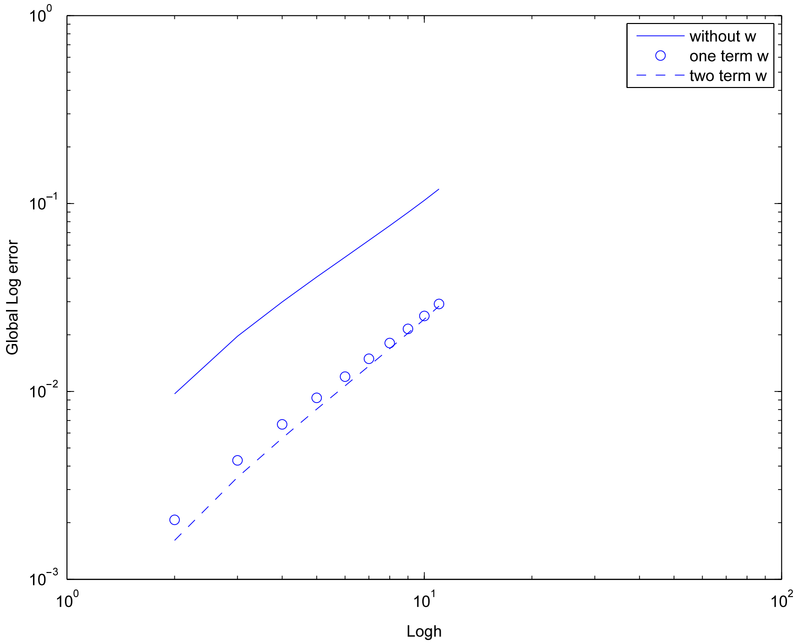

Fig. 7. Comparison of the errors for matrix problem obtained by Lie–Trotter splitting solved by fourth order Runge–Kutta Method without weight, one term weight, two term weight for $\Delta t = 0.01$.

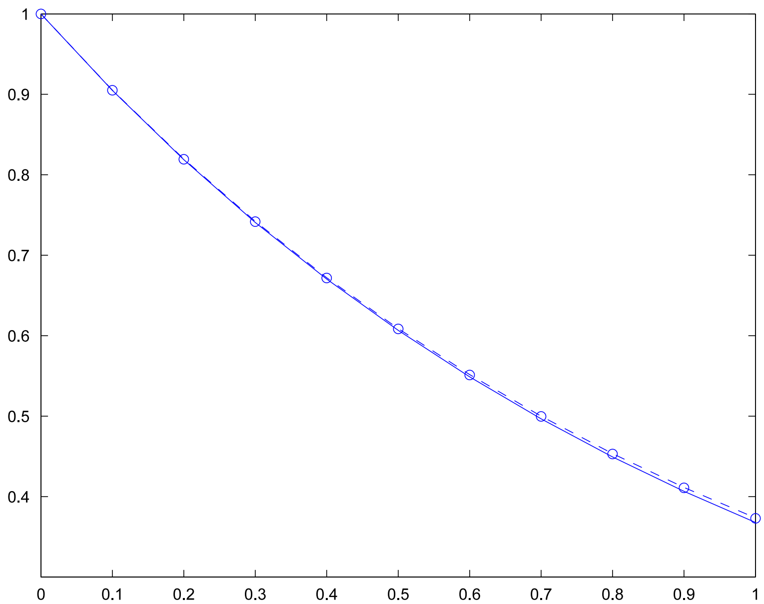

Fig. 8. Comparison of the solutions obtained by Strang splitting solved by fourth order Runge–Kutta method without weight, one term weight for $\Delta t = 0.01$.

Table 2
Comparison of errors for matrix problem with Strang splitting and fourth order Runge–Kutta method for $\Delta t = 0.01$.

		err_{L_∞}	err_{L_1}
Strang splitting	Without w	0.0055	2.7104e−004
	With one w	0.0051	2.3562e−004

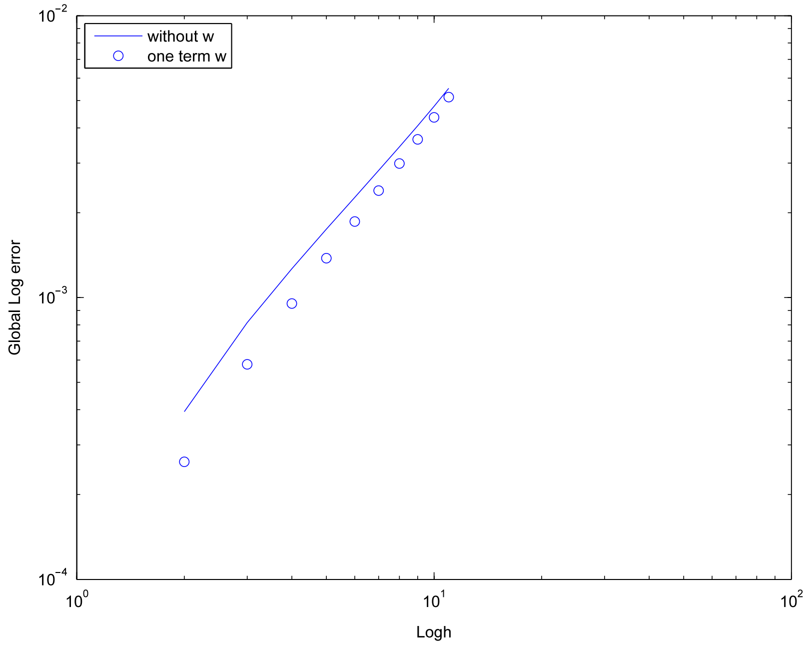

Fig. 9. Comparison of the errors for Matrix problem obtained by Strang splitting solved by fourth order Runge–Kutta Method without weight, one term weight for $\Delta t = 0.01$.

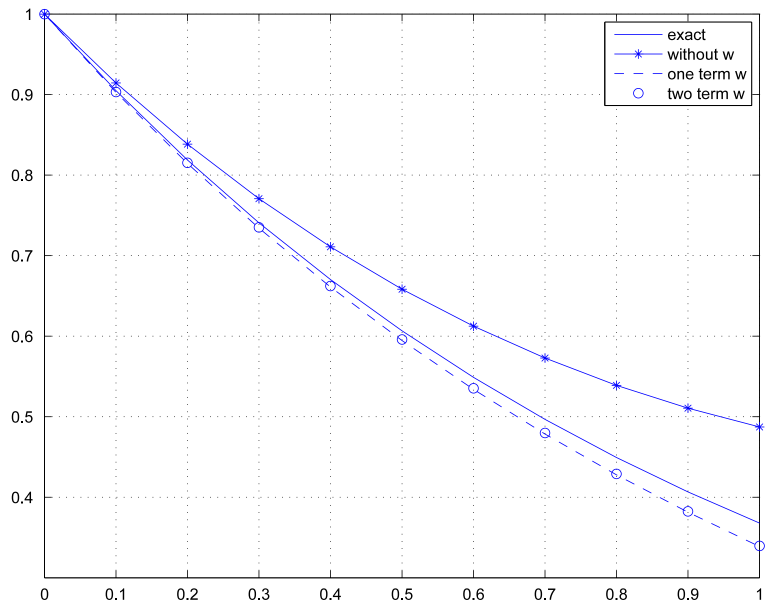

Fig. 10. Comparison of the solutions of matrix problem with Strang splitting solved by fourth order Runge–Kutta with weight, without weight, one term weight for $\Delta t = 0.1$.

6.3. Third test-example: parabolic equation

For more realistic models, we apply our theoretical results to partial differential equations.
We consider a parabolic equation in the following test problem as a next example:

$$u_t = Du_{xx}, \tag{94}$$

where $(x, t) \in [0, 1] \times [0, 1]$, $D = \frac{0.5 * \Delta x^2}{\sqrt[4]{0.5}}$ and Δx is the spatial grid size.

We have the exact solution $u(x, t) = \sin(\pi x)e^{-D\pi^2 t}$ and initial conditions are taken from exact solution, boundary conditions are Dirichlet boundary condition.

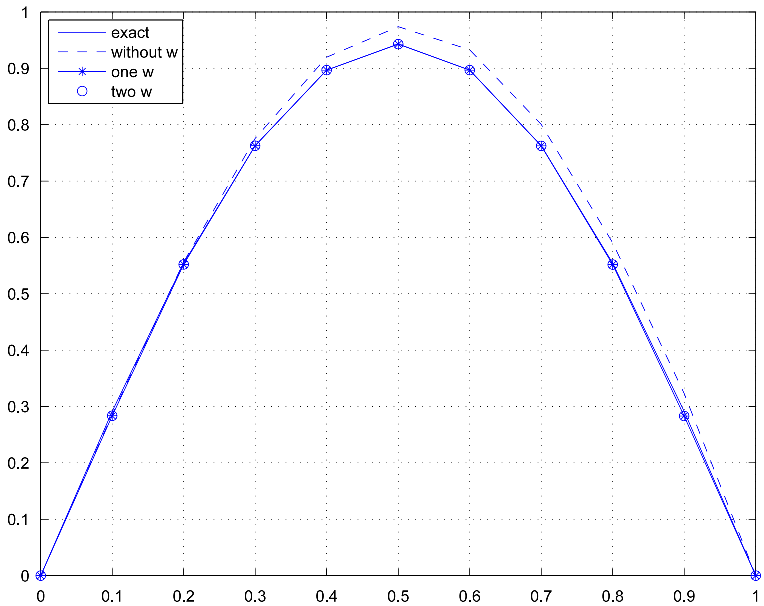

Fig. 11. Comparison of the solutions of parabolic problem obtained by Lie–Trotter splitting solved by implicit Euler method without weight and midpoint rule with one term weight for $\Delta x = 0.1$ and $\Delta t = 0.1$.

Table 3
Comparison of errors in parabolic problem measured by L_∞ norm and L_1 norm after applying Lie–Trotter splitting solved by implicit Euler method without weight and midpoint rule with one term weight for $\Delta x = 0.1$ and $\Delta t = 0.1$.

		err_{L_∞}	err_{L_1}
Lie–Trotter splitting	Without w	0.0376	0.0021
	With w_1	0.0082	2.1848e−004

Table 4
Comparison of errors in parabolic problem measured by L_∞ norm and L_1 norm after applying Strang splitting solved by midpoint rule for $\Delta x = 0.1$ and $\Delta t = 0.1$.

		err_{L_∞}	err_{L_1}
Strang splitting	Without w	0.0100	4.1548e−004
	With w_1	0.0011	8.8875e−005

We shall imply the fourth order difference approximation for u_{xx} as

$$u_{xx} \cong \frac{1}{\Delta x^2}[-1/12 \quad 4/3 \quad -5/2 \quad 4/3 \quad -1/12].$$

Therefore we obtain the first order differential equations given by

$$\frac{du}{dt} = Au, \tag{95}$$

where A is the global matrix coefficients given by the following stencil

$$A = \frac{1}{\Delta x^2}[-1/12 \quad 4/3 \quad -5/2 \quad 4/3 \quad -1/12] = A_1 + A_2, \tag{96}$$

and $A_1 = (A_l + D)$, $A_2 = A_u$ where A_l is lower triangular matrix, D is Diagonal matrix, A_u is upper triangular matrix.

Errors computed by L_∞ norm and L_1 norm are presented in Tables 3 and 4.

It is easily seen that we get the same result by using Lie–Trotter with one weight, with Strang splitting without weight.

Remark 6.3.1. To concentrate on the splitting error, we have balanced higher order time discretization with higher order spatial discretization using higher order stencils. In all experiments, we have presented the improvement of the weighted methods. In detail the benefits of the initialization with one and two terms can be obtained.

Table 5
Comparison of errors of hyperbolic problem with iterative splitting solved by midpoint method for $\Delta x = 0.05$ and $\Delta t = 0.01$.

		err_{L_∞}	err_{L_1}	CPU times
Iterative method	Without w	0.1119	0.1756	0.073314
	With one w	0.0769	0.1112	0.079872
	With two w	0.0766	0.1107	0.080240

Table 6
Comparison of errors of hyperbolic problem with iterative splitting solved by midpoint method for $\Delta x = 0.2$ and $\Delta t = 0.1$.

		err_{L_∞}	err_{L_1}	CPU times
Iterative method	Without w	0.3721	0.3016	0.037469
	With one w	0.0693	0.0619	0.037681
	With two w	0.0612	0.0403	0.038231

Table 7
Comparison of errors of hyperbolic problem with iterative splitting solved by fourth order Runge–Kutta method without weight, with one term weight and with two term weight for $\Delta x = 0.2$ and $\Delta t = 0.1$.

		err_{L_∞}	err_{L_1}	CPU times
Iterative method	Without w	0.2197	0.1831	0.029454
	With one w	0.0858	0.0804	0.030352
	With two w	0.0631	0.0625	0.030736

Table 8
Comparison of errors of hyperbolic problem with iterative splitting solved by fourth order Runge–Kutta method without weight, with one term weight and with two term weight for $\Delta x = 0.1$ and $\Delta t = 0.02$.

		err_{L_∞}	err_{L_1}	CPU times
Iterative method	Without w	0.1561	0.2587	0.038844
	With one w	0.0906	0.1408	0.040069
	With two w	0.0900	0.1392	0.040208

6.4. Fourth test-example: hyperbolic equation

We consider the following test problem:

$$u_t + au_x - bu = 0, \tag{97}$$

where $(x, t) \in [0, 1] \times [0, 1]$ with exact solution $u(x, t) = e^x e^{(b-a)t}$ where $a = 1, b = 1$ and initial conditions, boundary conditions are taken from exact solution.

We use the second order expansion for u_x:

$$u_x \cong \frac{1}{\Delta x}[-3/2 \quad 2 \quad -1/2].$$

Comparison of errors with iterative splitting and midpoint rule for solutions without weight, with one term weight and with two term weight are given in Table 5:

Comparison of errors with iterative splitting and Midpoint for solutions without weight, with one term weight and with two term weight are given in Table 6:

Comparison of errors of hyperbolic problem with iterative splitting solved by fourth order Runge–Kutta method without weight, with one term weight and with two term weight are given in Tables 7 and 8:

Remark 6.4.1. For hyperbolic problem, the iterative operator splitting scheme is considered and with improved by the weighted method. While the numerical error is reduced twice and more, the amount of computational time is nearly the same. Here we have the benefit of the weighted method, that accelerates the initialization process at the beginning of the method.

6.5. Fifth test-example: system of parabolic equations (decoupled problem)

We consider the following test problem as an example:

$$R_1 u_{1,t} + v_1 u_{1,x} - D_{11} u_{1,xx} = -\lambda_1 u_1, \tag{98}$$

$$R_2 u_{2,t} + v_2 u_{2,x} - D_{22} u_{2,xx} = -\lambda_2 u_2, \tag{99}$$

where $u(x, t_0) = u_{exact}(x, t_0)$, $u(0, t) = u_{exact}(0, t)$, $u(L, t) = u_{exact}(L, t)$, $R_1, R_2 = 1$, $v_1 = 0.001$, $v_2 = 0.002$, $D_{11} = 0.0001$, $D_{22} = 0.0004$ and $\lambda_1 = 10^{-5}$, $\lambda_2 = 2 \times 10^{-5}$ and exact solution of the problem is given by

$$u_{exact}(x, t) = \frac{1}{2\sqrt{D\pi t}} e^{\frac{-(x-vt)^2}{4Dt}} e^{-\lambda t}. \tag{100}$$

The equation in a system notation is given as:

$$\mathbf{R}u_t + \mathbf{v}u_x - Du_{xx} = \Lambda u, \tag{101}$$

where $u = (u_1, u_2)^t$, $\mathbf{R} = (R_1, R_2)^t$, $\mathbf{v} = (v_1, v_2)^t$, $D = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix}$ and $\Lambda = \begin{pmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{pmatrix}$.

Further we have only diagonal entries in the convection part:
$V = -\begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix}$.

We apply the following cases for the different operator splitting.

(1) Commuting case:

$$\mathbf{R}u_t = Au + Bu, \tag{102}$$

with the operators $A = V\frac{\partial}{\partial_x} + \Lambda$, $B = D\frac{\partial}{\partial_{xx}}$.

(2) Non-commuting case (a):

$$\mathbf{R}u_t = Au + Bu, \tag{103}$$

with the operators

$$A = \begin{pmatrix} -v_1\frac{\partial}{\partial_x} - \lambda_1 & 0 \\ 0 & -v_2\frac{\partial}{\partial_x} - \lambda_2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} D_{11}\frac{\partial^2}{\partial_x^2} & 0 \\ 0 & D_{22}\frac{\partial^2}{\partial_x^2} \end{pmatrix}.$$

(3) Non-commuting case (b):

$$\mathbf{R}u_t = Au + Bu, \tag{104}$$

with the operators

$$A = \begin{pmatrix} -v_1\frac{\partial}{\partial_x} - \lambda_1 & 0 \\ 0 & -v_2\frac{\partial}{\partial_x} - \lambda_2 \end{pmatrix}, \qquad B = \begin{pmatrix} D_{11}\frac{\partial^2}{\partial_x^2} & 0 \\ 0 & D_{22}\frac{\partial^2}{\partial_x^2} \end{pmatrix}.$$

The solution of system of parabolic equations with iterative splitting solved by fourth order midpoint method with one weight and without weight is given in Fig. 12:

Remark 6.5.1. For systems of differential equations we have dealt with commutative and non-commutative cases. In both cases we have obtained the benefit of the weighted methods, while reducing the splitting error and reserve the computational time.

6.6. *Sixth example: system of parabolic equations (coupled problem)*

We consider the following test problem as an example:

$$R_1 u_{1,t} + v_1 u_{1,x} - D_{11}u_{1,xx} - D_{12}u_{2,xx} = -\lambda_1 u_1 + \lambda_2 u_2, \tag{105}$$

$$R_2 u_{2,t} + v_2 u_{2,x} - D_{21}u_{1,xx} - D_{22}u_{2,xx} = \lambda_1 u_1 - \lambda_2 u_2, \tag{106}$$

where $(x, t) \in [0, 10] \times [1900, 2000]$.

For simplification and derivation of the analytical solution we choose:

$$R_1 = R_2 = 1, \qquad v_1 = v_2 \quad \text{and} \quad D_{11} = D_{22} \quad \text{with } D_{12} = D_{21} = 0$$

and we have:

$$R_1, R_2 = 1, \qquad v_1 = v_2 = 0.001, \qquad D_{11} = D_{22} = 0.0001, \qquad D_{12} = D_{21} = 0 \quad \text{and}$$
$$\lambda_1 = 10^{-5}, \qquad \lambda_2 = 2 \times 10^{-5}.$$

The equation in a system notation is given as:

$$\mathbf{R}u_t + \mathbf{v}u_x - Du_{xx} = \Lambda u, \tag{107}$$

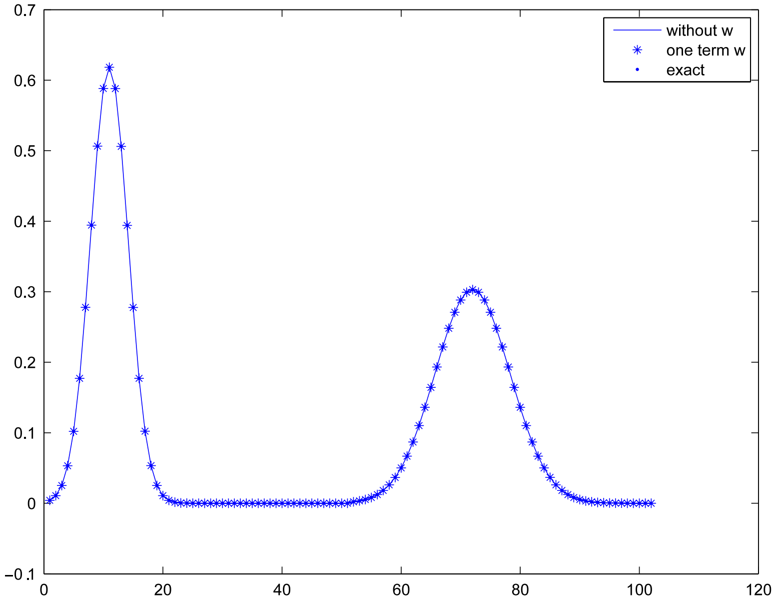

Fig. 12. Comparison of solutions of system of parabolic equations with iterative splitting solved by fourth order midpoint method with one weight and without weight for $\Delta x = 0.2$ and $\Delta t = 1$.

where

$$u = (u_1, u_2)^t, \qquad \mathbf{R} = (R_1, R_2)^t, \qquad \mathbf{v} = (v_1, v_2)^t, \qquad D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} \text{ and } \Lambda = \begin{pmatrix} -\lambda_1 & \lambda_2 \\ \lambda_1 & -\lambda_2 \end{pmatrix}.$$

Further we have only diagonal entires in the convection part:

$$V = -\begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix}.$$

The analytical solution is given, see [17]:

$$u_t + \mathbf{v} u_x - D u_{xx} = \Lambda u. \tag{108}$$

The reaction matrix can be diagonalized given as:

$$\Lambda = S \Lambda_{diag} S^{-1}, \tag{109}$$

and we have a decoupled system given as:

$$c_t + \mathbf{v} c_x - D c_{xx} = \Lambda_{diag} c, \tag{110}$$

where $c = S^{-1} u$.

To find the analytical solution, we compute the Λ_{diag} by $det(\Lambda - aI) = 0$ where $a = (a_1, a_2)$, hence eigenvalues are $a_1 = 0$ and $a_2 = -(\lambda_1 + \lambda_2)$ and corresponding eigenvectors are $(\lambda_2, \lambda_1)^\top$ and $(\lambda_1, -\lambda_1)^\top$. Therefore Λ_{diag} and S are given as follows

$$\Lambda_{diag} = \begin{pmatrix} 0 & 0 \\ 0 & -(\lambda_1 + \lambda_2) \end{pmatrix},$$

and

$$S = \begin{pmatrix} \lambda_2 & \lambda_1 \\ \lambda_1 & -\lambda_1 \end{pmatrix}.$$

Then we have decoupled equations:

$$\partial_t c_1 + \mathbf{v} \partial_x c_1 - D_{11} \partial_{xx} c_1 = 0, \tag{111}$$

$$\partial_t c_2 + \mathbf{v} \partial_x c_2 - D_{22} \partial_{xx} c_2 = -(\lambda_1 + \lambda_2) c_2, \tag{112}$$

where $v_1 = v_2 = 0.001$, $D_{11} = D_{22} = 0.0001$ and $\lambda_1 = 10^{-5}$, $\lambda_2 = 2 \times 10^{-5}$.

Table 9
Comparison of errors of coupled problem with iterative splitting solved by fourth order Runge–Kutta method without weight, with one term weight for $\Delta x = 0.2$ and $\Delta t = 1$.

		err_{L_∞}	err_{L_1}
Iterative method	Without w	4.0013e−008	1.2834e−006
	With one w	4.0008e−008	1.2819e−006

Finally, the exact solution can be found as follows in terms of the transformed variables:

$$c_{1,exact}(x,t) = \frac{1}{2\sqrt{D_{11}\pi t}} e^{\frac{-(x-v_1 t)^2}{4D_{11}t}}, \tag{113}$$

$$c_{2,exact}(x,t) = \frac{1}{2\sqrt{D_{22}\pi t}} e^{\frac{-(x-v_2 t)^2}{4D_{22}t}} e^{-(\lambda_1+\lambda_2)t}, \tag{114}$$

where $c = S^{-1}u$ and exact solutions becomes:

$$u_{1,exact}(x,t) = \lambda_2 c_{1,exact}(x,t) + \lambda_1 c_{2,exact}(x,t), \tag{115}$$

$$u_{2,exact}(x,t) = \lambda_1 c_{1,exact}(x,t) - \lambda_1 c_{2,exact}(x,t), \tag{116}$$

where $u = Sc$.

For computations, we split the operator as follow:

(1) Commuting case:

$$\mathbf{R}u_t = Au + Bu, \tag{117}$$

with the operators $A = V\frac{\partial}{\partial_x} + \Lambda$ and $B = D\frac{\partial}{\partial_{xx}}$.

(2) Non-commuting case (a):

$$\mathbf{R}u_t = Au + Bu, \tag{118}$$

with the operators

$$A = \begin{pmatrix} -v_1\frac{\partial}{\partial_x} - \lambda_1 & 0 \\ \lambda_1 & -v_2\frac{\partial}{\partial_x} - \lambda_2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} D_{11}\frac{\partial^2}{\partial_x^2} & D_{12}\frac{\partial^2}{\partial_x^2} + \lambda_2 \\ D_{21}\frac{\partial^2}{\partial_x^2} & D_{22}\frac{\partial^2}{\partial_x^2} \end{pmatrix}.$$

(3) Non-commuting case (b):

$$\mathbf{R}u_t = Au + Bu \tag{119}$$

with the operators

$$A = \begin{pmatrix} -v_1\frac{\partial}{\partial_x} - \lambda_1 & D_{12}\frac{\partial^2}{\partial_x^2} \\ \lambda_1 & -v_2\frac{\partial}{\partial_x} - \lambda_2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} D_{11}\frac{\partial^2}{\partial_x^2} & \lambda_2 \\ D_{21}\frac{\partial^2}{\partial_x^2} & D_{22}\frac{\partial^2}{\partial_x^2} \end{pmatrix}.$$

For the decoupled equations with the unknown $c1 = (c_{11}, \dots, c_{1n}), c2 = (c_{21}, \dots, c_{2n})$ where $c = (c1, c2)$ we can apply the scalar analytical solutions, see [17,18].

For each case, comparison of errors of coupled problem with iterative splitting solved by fourth order Runge–Kutta method without weight, with one term weight are given in Table 9:

Fig. 13 shows the solutions of coupled problem obtained by iterative splitting and fourth order Runge–Kutta method.

Remark 6.6.1. For systems of differential equations we have dealt with commutative and non-commutative cases. In both cases we have obtained the benefit of the weighted methods, while reducing the splitting error and reserve the computational time.

7. Conclusion

In the paper we presented the benefits of improving standard splitting methods with weighting schemes. The ideas are based on the Zassenhaus product to improve the initialization process of the splitting method. Here an acceleration of the well-known Lie–Trotter and Strang splitting methods can be done and also iterative splitting schemes. By adding additional terms to the starting conditions of the splitting methods, we obtained that all weighted methods achieved more accurate results. The applications in parabolic equations shows the verification of the theoretical results. In future, we present a framework for iterative and non-iterative operator splitting methods.

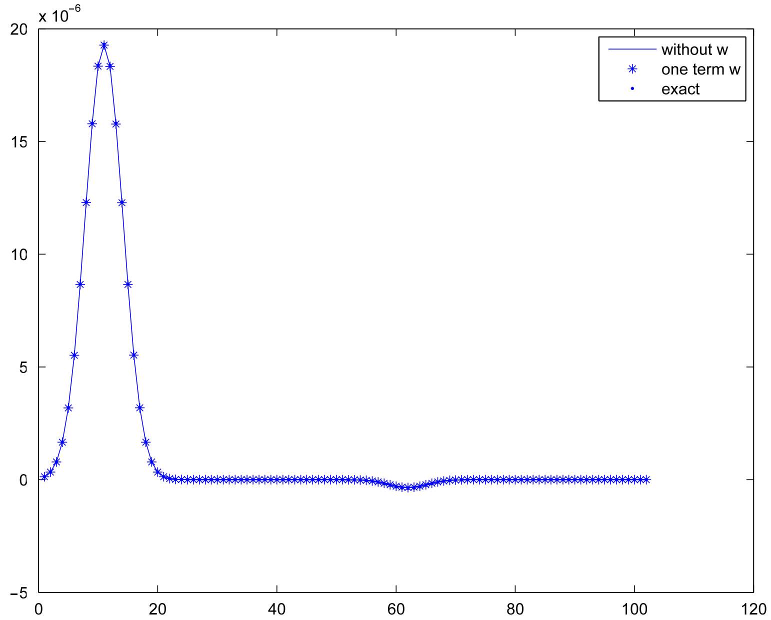

Fig. 13. Comparison of the solutions of coupled problem with iterative splitting solved by fourth order Runge–Kutta with weight, without weight, one term weight for $\Delta x = 0.2$ and $\Delta t = 1$.

References

[1] R.E. Ewing, Up-scaling of biological processes and multiphase flow in porous media, in: IIMA Volumes in Mathematics and its Applications, vol. 295, Springer-Verlag, 2002, pp. 195–215.
[2] J. Geiser, Numerical simulation of a model for transport and reaction of radionuclides, in: Proceedings of the Large Scale Scientific Computations of Engineering and Environmental Problems, Sozopol, Bulgaria, 2001.
[3] P. Frolkovič, J. Geiser, Numerical simulation of radionuclides transport in double porosity media with sorption, in: Proceedings of Algorithmy 2000, Conference of Scientific Computing, 2000, pp. 28–36.
[4] R.I. McLachlan, G. Reinoult, W. Quispel, Splitting methods, Acta Numerica (2002) 341–434.
[5] J.A. Oteo, The Baker–Campbell–Hausdorff formula and nested commutator identities, Journal of Mathematical Physics 32 (1991) 419.
[6] I. Farago, J. Geiser, Iterative operator-splitting methods for linear problems, Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany, June 2005.
[7] S. Vandewalle, Parallel multigrid waveform relaxation for parabolic problems, Teubner Skripten zur Numerik, B.G. Teubner Stuttgart, 1993.
[8] G. Strang, On the construction and comparision of difference schemes, SIAM Journal on Numerical Analysis 5 (1968) 506–517.
[9] G.I Marchuk, Some applicatons of splitting-up methods to the solution of problems in mathematical physics, Aplikace Matematiky 1 (1968) 103–132.
[10] D. Scholz, M. Weyrauch, A note on the Zassenhaus product formula, Journal of Mathematical Physics 47 (2006) 033505.
[11] S. Descombes, Convergence of a splitting method of high order for reaction–diffusion systems, Mathematics of Computations 70 (2001) 1481–1501.
[12] J. Geiser, G. Tanoglu, Operator-splitting methods via Zassenhaus product formula, Applied Mathematics and Computation 217 (2011) 4557–4575.
[13] E. Hairer, C. Lubich, G. Wanner, Geometric numerical integration, in: Springer Series in Computational Mathematics, vol. 31, Springer Verlag, Berlin, New York, Heidelberg, 2002.
[14] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A 150 (5–7) (1990).
[15] J. Geiser, in: Magoules, Lai (Eds.), Decomposition Methods for Differential Equations: Theory and Applications, Chapman & Hall/CRC Press, Taylor and Francis Group, 2009.
[16] B. Gustafsson, High Order Difference Methods for Time dependent PDE, in: Springer Series in Computational Mathematics, vol. 38, Springer Verlag, Berlin, New York, Heidelberg, 2007.
[17] X. Lu, Y. Sun, J.N. Petersen, Analytical solutions of TCE transport with convergent reactions, Transport in Porous Media 51 (2003) 211–225.
[18] J. Geiser, Gekoppelte Diskretisierungsverfahren für Systeme von Konvektions-Dispersions-Diffusions-Reaktionsgleichungen, Doktor-Arbeit, Universität Heidelberg, 2003.