



## Iterative Operator-splitting Methods with Embedded Discretization Schemes

Jürgen Geiser

**Abstract.** In this paper we describe a computation of iterative operator-splitting method, which are known as competitive splitting methods, see [10] and [11]. We derived a closed form, based on commutators for the iterative method. The time discretization schemes apply extrapolation schemes and Pade approximations to the exp-functions. Spatial discretization schemes considered Lax-Wendroff methods and are combined with the iterative schemes. The error analysis describe the approximation errors. Numerical examples of ordinary and partial differential equations support the fast computation ideas.

### 1. Introduction

In this paper we concentrate on approximation to the solution of the linear evolution equation

$$\partial_t c = Lc = (A+B)c, \quad c(0) = c(t=0), \quad t \in [0, T], \quad (1)$$

where  $L, A$  and  $B$  are unbounded operators and  $T \in \mathbb{R}^+$ .

We solve our equation with the following numerical scheme:

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

$$\text{and } c_0(t^n) = c^n, \quad c_{-1} = 0.0,$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) + Bc_{i+1}(t), \quad (3)$$

$$\text{with } c_{i+1}(t^n) = c^n,$$

where  $c(t^n)$  is the approximation at  $t = t^n$ .

---

2010 *Mathematics Subject Classification.* 65M15, 65L05, 65M71.

*Key words and phrases.* Iterative operator-splitting method; Extrapolation methods; Lax-Wendroff method; Error analysis; Differential equations.

The numerical method (2)-(3) can be written algebraic in a 2-stage iterative splitting scheme:

$$c_i(t) = \exp(At)c(0) + \int_0^t \exp(As)Bc_{i-1} ds, \quad (4)$$

$$c_{i+1}(t) = \exp(Bt)c(0) + \int_0^t \exp(Bs)Ac_i ds, \quad (5)$$

where  $i = 1, 3, 5, \dots$  and the initial or start solution is given as  $c_0(t) = 0$  or constant. Further we have the conditions, that  $c^n$  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  $n$ .)

Based on our motivation to design effective algorithms for large equation systems. The problem arose in the field of optimizing the computation of the iteration steps of very large systems of differential equations fixed on time-scale and on one discretization method. Here novel ideas in computing *exp*-functions can be used, so we apply the multi-product expansion as an extrapolation method, see [8]. The iterative splitting method can be formulated as a generalization of a waveform relaxation approach, see [5]. So that also numerical schemes of sparse solvers can be taken into account to such a method. For partial differential equations, the spatial discretization methods are very important and we concentrate on efficient embedded discretization schemes, that overcome the problem of order restriction, while using lower order spatial discretization schemes, see [12].

Historically, effective computational methods can derived by considering the local character of each equation part. So in the last years the ideas of splitting into simpler equations are established, see [16], [7] and [14]. We concentrate on choosing extrapolation to obtain higher order schemes without losing efficiency in computing the operators.

The outline of the paper is as follows. The operator-splitting-method is introduced and the error-analysis of the operator-splitting method is presented in Section 2. A closed form is discussed in Section 4, where we discuss an efficient computation of the iterative splitting method with based on extrapolation methods. In Section 6 we present the numerical results for the methods. Finally we discuss future works in the area of iterative methods.

## 2. Error analysis

The following algorithm is based on the iteration with fixed-splitting discretization step-size  $\tau$ , namely, on the time-interval  $[t^n, t^{n+1}]$  we solve

the following sub-problems consecutively for  $i = 0, 2, \dots, 2m$ . (cf. [16]):

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \text{ with } c_i(t^n) = c^n \quad (6)$$

$$\text{and } c_0(t^n) = c^n, \quad c_{-1} = 0.0,$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) + Bc_{i+1}(t), \quad (7)$$

$$\text{with } c_{i+1}(t^n) = c^n,$$

where  $c^n$  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. Two-step iterative schemes for operators

In the following we discuss the consistency of the 2 stage iterative method, taken into account to iterate over both operators.

**Theorem 1.** *Let us consider the abstract Cauchy problem in a Banach space  $\mathbf{X}$*

$$\begin{aligned} \partial_t c(x, t) &= Ac(x, t) + Bc(x, t), \quad 0 < t \leq T \text{ and } x \in \Omega, \\ c(x, 0) &= c_0(x), \quad x \in \Omega, \\ c(x, t) &= c_1(x, t), \quad x \in \partial\Omega \times [0, T], \end{aligned} \quad (8)$$

where  $A, B : \mathbf{X} \rightarrow \mathbf{X}$  are given linear operators which are generators of the analytical semigroups and  $c_0 \in \mathbf{X}$  is a given element. We assume  $\text{dom}(B) \subset \text{dom}(A)$ , so we are restricted to balance the operators. Further, we assume the estimations of an unbounded operator, see [15]:

$$B = A^{1-\alpha}, \quad (9)$$

where  $\alpha \in (0, 1)$  and we assume  $B^{1-\alpha}$  is the infinitesimal generator of an analytical semigroup for all  $\alpha \in (0, 1)$ , see [4].

Further we assume:

$$\|B^\alpha \exp(B\tau_n)\| \leq \kappa \tau_n^{-\alpha}. \quad (10)$$

$$\|B^\alpha \exp((A+B)\tau_n)\| \leq \kappa \tau_n^{-\alpha}, \quad (11)$$

$$\|\exp(A\tau_n)B^{1-\alpha}\| \leq \tilde{\kappa} \tau_n^{p(1-\alpha)}, \quad (12)$$

$$\|A^\beta \exp(A\tau_n)\| \leq \kappa \tau_n^{-\beta}. \quad (13)$$

$$\|A^\beta \exp((A+B)\tau_n)\| \leq \kappa \tau_n^{-\beta}, \quad (14)$$

$$\|\exp(B\tau_n)A^{1-\beta}\| \leq \tilde{\kappa} \tau_n^{q(1-\beta)}, \quad (15)$$

where  $\alpha, \beta, p, q \in (0, 1)$  and  $\tau_n = (t^{n+1} - t^n)$ .

The error of the first time-step is of accuracy  $\mathcal{O}(\tau_n^{i_A \alpha})$ , where  $\tau_n = t^{n+1} - t^n$  and we have equidistant time-steps, with  $n = 1, \dots, N$ . Further  $i_A$  are the iterative steps with operator  $A$ .

Then the iteration process (6)-(7) for  $i = 1, 3, \dots, 2m + 1$  is consistent with the order of the consistency  $\mathcal{O}(\tau_n^{\alpha i_A})$ , where  $0 \leq \alpha < 1$ .

**Proof.** Let us consider the iteration (6)-(7) on the sub-interval  $[t^n, t^{n+1}]$ .

For the first iterations we have:

$$\partial_t c_1(t) = Ac_1(t), \quad t \in (t^n, t^{n+1}], \quad (16)$$

and for the second iteration we have:

$$\partial_t c_2(t) = Ac_1(t) + Bc_2(t), \quad t \in (t^n, t^{n+1}]. \quad (17)$$

In general we have:

For the odd iterations:  $i = 2m + 1$  for  $m = 0, 1, 2, \dots$

$$\partial_t c_i(t) = Ac_i(t) + Bc_{i-1}(t), \quad t \in (t^n, t^{n+1}], \quad (18)$$

where for  $c_0(t) \equiv 0$ .

For the even iterations:  $i = 2m$  for  $m = 1, 2, \dots$

$$\partial_t c_i(t) = Ac_{i-1}(t) + Bc_i(t), \quad t \in (t^n, t^{n+1}]. \quad (19)$$

We have the following solutions for the iterative scheme:

The solutions for the first two equations are given by the variation of constants:

$$c_1(t) = \exp(A(t - t^n))c(t^n), \quad t \in (t^n, t^{n+1}], \quad (20)$$

$$c_2(t) = \exp(B(t - t^n))c(t^n) + \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))Ac_1(s)ds, \quad t \in (t^n, t^{n+1}]. \quad (21)$$

For the recursive even and odd iterations we have the solutions:

For the odd iterations:  $i = 2m + 1$  for  $m = 0, 1, 2, \dots$

$$c_i(t) = \exp(A(t - t^n))c(t^n) + \int_{t^n}^t \exp((t - s)A)Bc_{i-1}(s) ds, \quad t \in (t^n, t^{n+1}]. \quad (22)$$

For the even iterations:  $i = 2m$  for  $m = 1, 2, \dots$

$$c_i(t) = \exp(B(t - t^n))c(t^n) + \int_{t^n}^t \exp((t - s)B)Ac_{i-1}(s) ds, \quad t \in (t^n, t^{n+1}]. \quad (23)$$

**The consistency is given as:**

For  $e_1$  we have:

$$c_1(t^{n+1}) = \exp(A\tau_n)c(t^n), \quad (24)$$

$$\begin{aligned}
c(t^{n+1}) &= \exp((A+B)\tau_n)c(t^n) \\
&= \exp(A\tau_n)c(t^n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp((s-t^n)(A+B))c(t^n) ds.
\end{aligned} \tag{25}$$

We obtain:

$$\begin{aligned}
\|e_1\| &= \|c - c_1\| \\
&\leq \|\exp((A+B)\tau_n)c(t^n) - \exp(A\tau_n)c(t^n)\| \\
&\leq \left\| \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp((s-t^n)(A+B))c(t^n) ds \right\| \\
&\leq \left\| \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))A^{1-\alpha} \exp((s-t^n)(A+B))c(t^n) ds \right\| \\
&\leq \int_{t^n}^{t^{n+1}} \|\exp(A(t^{n+1}-s))A^{(1-\alpha)}/2A^{(1-\alpha)/2} \exp((s-t^n)(A+B))\| ds \|c(t^n)\| \\
&\leq \int_{t^n}^{t^{n+1}} \frac{1}{(t^{n+1}-s)^{-(1-\alpha)/2}} \frac{\kappa}{(s-t^n)^{(1-\alpha)/2}} ds \|c(t^n)\| \\
&\quad + \int_{t^n}^{t^{n+1}} Cs(\alpha-1)ds \|c(t^n)\| \\
&\leq C\tau^\alpha \|c(t^n)\|
\end{aligned} \tag{26}$$

where  $\alpha \in (0, 1)$  and  $\tau = (t^{n+1} - t^n)$ .

For  $e_2$  we have:

$$c_2(t^{n+1}) = \exp(B\tau_n)c(t^n) + \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \exp((s-t^n)A)c(t^n) ds, \tag{28}$$

$$\begin{aligned}
c(t^{n+1}) &= \exp(B\tau_n)c(t^n) + \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \exp((s-t^n)A)c(t^n) ds \\
&\quad + \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \\
&\quad \times \int_{t^n}^s \exp(A(s-\rho))B \exp((\rho-t^n)(A+B))c(t^n) d\rho ds.
\end{aligned} \tag{29}$$

We obtain:

$$\|e_2\| \leq \|\exp((A+B)\tau_n)c(t^n) - c_2\| \quad (30)$$

$$= \left\| \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \times \int_{t^n}^s \exp(A(s-\rho))A^{1-\alpha} \exp((\rho-t^n)(A+B))c(t^n) d\rho ds \right\| \quad (31)$$

$$= \int_{t^n}^{t^{n+1}} \|\exp(B(t^{n+1}-s))\| \quad (32)$$

$$\times \int_{t^n}^s \|\exp(A(s-\rho))A^{2-\alpha} \exp((\rho-t^n)(A+B))c(t^n) d\rho\| ds$$

$$= \int_{t^n}^{t^{n+1}} \kappa \int_{t^n}^s (s-\rho)^{\alpha-2} d\rho ds \|c(t^n)\| \quad (33)$$

$$\leq C\tau^\alpha \|c(t^n)\|$$

For odd and even iterations, the recursive proof is given in the following. In the next steps, we shift  $t^n \rightarrow 0$  and  $t^{n+1} \rightarrow \tau_n$  for simpler calculations, see [15]. The initial conditions are given with  $c(0) = c(t^n)$ .

For the odd iterations means the iteration over operator  $A$ :  $i = 2m + 1$ , with  $m = 0, 1, 2, \dots$ , we obtain for  $c_i$  and  $c$ :

$$\begin{aligned} c_i(\tau_n) &= \exp(A\tau_n)c(0) \\ &+ \int_0^{\tau_n} \exp(As)B \exp((\tau_n-s)B)c(0) ds \\ &+ \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n-s_1} \exp(s_2B)A \exp((\tau_n-s_1-s_2)A)c(0) ds_2 ds_1 \\ &+ \dots + \\ &+ \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n-s_1} \exp(s_2A)B \int_0^{\tau_n-s_1-s_2} \exp(s_3A)B \dots \\ &\int_0^{\tau_n-\sum_{j=1}^{i-1} s_j} \exp(As_i)B \exp\left(\left(\tau_n - \sum_{j=1}^{i-1} s_j\right)A\right)c(0) ds_i \dots ds_1, \quad (34) \end{aligned}$$

$$c(\tau_n) = \exp(A\tau_n)c(0) + \int_0^{\tau_n} \exp(As)B \exp((\tau_n-s)B)c(0) ds$$

$$\begin{aligned}
 & + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n-s_1} \exp(s_2B)A \exp((\tau_n - s_1 - s_2)A)c(0) ds_2 ds_1 \\
 & + \dots + \\
 & + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n-s_1} \exp(s_2A)B \int_0^{\tau_n-s_1-s_2} \exp(s_3A)B \dots \\
 & \int_0^{\tau_n-\sum_{j=1}^{i-1} s_j} \exp(As_i)B \exp\left(\left(\tau_n - \sum_{j=1}^{i-1} s_j\right)A\right)c(0) ds_i \dots ds_1 \\
 & + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n-s_1} \exp(s_2A)B \int_0^{\tau_n-s_1-s_2} \exp(s_3A)B \dots \\
 & \int_0^{\tau_n-\sum_{j=1}^i s_j} \exp(As_{i+1})B \exp\left(\left(\tau_n - \sum_{j=1}^i s_j\right)(A+B)\right)c(0) ds_{i+1} \dots ds_1.
 \end{aligned} \tag{35}$$

By shifting  $0 \rightarrow t^n$  and  $\tau_n \rightarrow t^{n+1}$ , we obtain our result:

$$\begin{aligned}
 \|e_i\| & \leq \|\exp((A+B)\tau_n)c(t^n) - c_i\| \\
 & \leq \tilde{C} \tau_n^{i_A} \|c(t^n)\|,
 \end{aligned} \tag{36}$$

where  $\alpha = \min_{j=1}^i \{\alpha_j\}$  and  $0 \leq \alpha_i < 1$  and  $i_A$  is the number of odd iteration steps means over the operator  $A$ .

The same proof idea can be applied to the even iterative scheme.

Where for the even scheme we could not obtain a higher order, see:

$$\begin{aligned}
 c(\tau_n) & = \exp(A\tau_n)c(0) \\
 & + \int_0^{\tau_n} \exp(Bs)A \exp((\tau_n - s)B)c(0) ds \\
 & + \int_0^{\tau_n} \exp(Bs_1)A \int_0^{\tau_n-s_1} \exp(s_2B)A \exp((\tau_n - s_1 - s_2)B)c(0) ds_2 ds_1 \\
 & + \dots + \\
 & + \int_0^{\tau_n} \exp(Bs_1)A \int_0^{\tau_n-s_1} \exp(s_2B)A \int_0^{\tau_n-s_1-s_2} \exp(s_3B)A \dots \\
 & \int_0^{\tau_n-\sum_{j=1}^{i-1} s_j} \exp(Bs_i)A \exp\left(\left(\tau_n - \sum_{j=1}^{i-1} s_j\right)B\right)c(0) ds_i \dots ds_1
 \end{aligned}$$

$$\begin{aligned}
& + \int_0^{\tau_n} \exp(Bs_1)A \int_0^{\tau_n-s_1} \exp(s_2B)A \int_0^{\tau_n-s_1-s_2} \exp(s_3B)A \dots \\
& \int_0^{\tau_n-\sum_{j=1}^i s_j} \exp(Bs_{i+1})A \exp\left(\left(\tau_n - \sum_{j=1}^i s_j\right)(A+B)\right)c(0) ds_{i+1} \dots ds_1
\end{aligned} \tag{37}$$

and we have

$$\|e_{i_B}\| \leq \|\exp((A+B)\tau_n)c(t^n) - c_{i_B}\| \tag{38}$$

$$\begin{aligned}
& = \left\| \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s_1))A \dots \right. \\
& \quad \left. \int_{t^n}^{s_{i_B-1}} \exp(B(s_{i_B-1} - s_{i_B}))A \exp((s_{i_B} - t^n)(A+B))c(t^n) ds_{i_B} ds_1 \right\|
\end{aligned} \tag{39}$$

$$= \int_{t^n}^{t^{n+1}} \kappa \int_{t^n}^s \kappa \|A^2 \exp((\rho - t^n)(A+B))c(t^n) d\rho\| ds \tag{40}$$

$$= \int_{t^n}^{t^{n+1}} \kappa \dots \int_{t^n}^{s_{i_B}} (s - \rho)^{-i_B} ds_{i_B} \dots ds_1 \|c(t^n)\| \tag{41}$$

$$\leq C \tau^0 \|c(t^n)\|$$

So we could not improve the order in the weaker iterations, so we need at least some strong iterative steps.  $\square$

**Remark 1.** An example can be assumed with  $A = \nabla D_1 \nabla$  and  $B = -\mathbf{v} \cdot \nabla + \nabla D_2 \nabla$ , where  $D_1 \gg D_2 \in \mathbb{R}^+$ , with  $B = A^{1-\alpha}$ ,  $0 \leq \alpha < 1$ . We apply two iterative steps with operator  $A$  and have the following local errors:

$$\|e_1\| = \tilde{C} \tau_n^\alpha \tag{42}$$

and hence

$$\|e_2\| = \tilde{\tilde{C}} \tau_n^{2\alpha}, \tag{43}$$

where  $0 \leq \alpha < 1$  and  $\tilde{C}, \tilde{\tilde{C}}$  are constants independent of  $\tau_n$ .

### 3.1. Convergence results

To derive convergent results, we apply stability and consistency done in the previous sections.

The convergence result for the assumption  $B = A^{1-\alpha}$  and apply the iteration to operator  $A$  (one-side iteration).

We obtain the following results:

For  $i_A = 1$ :

$$\begin{aligned} & \|S_1^m - S_{\text{num},1}^m\| \\ &= \|(\exp((A+B)\tau_n)^m - \exp(A\tau_n)^m)c(t^n)\| \end{aligned} \quad (44)$$

$$\begin{aligned} &\leq \left\| \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^m - \exp(A\tau_n)^m \right\| c(t^n) \\ &\leq \left\| \left( \exp(A\tau_n)^m + \binom{m}{1} \exp(A\tau_n)^{m-1} \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right. \right. \\ &\quad \left. \left. + \dots + \left( \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^m \right) - \exp(A\tau_n)^m \right\| c(t^n) \end{aligned} \quad (45)$$

$$\leq mC\tau_n^\alpha \|c(t^n)\|, \quad (46)$$

where  $m$  are number of the local steps,  $C$  is a constant independent of  $\tau_n$ .

For  $i_A = 2$ :

$$\begin{aligned} & \|S_2^m - S_{\text{num},2}^m\| \\ &= \left\| \left( \exp((A+B)\tau_n)^m \right. \right. \\ &\quad \left. \left. - \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^m \right) c(t^n) \right\| \end{aligned} \quad (47)$$

$$\begin{aligned} &\leq \left\| \left( \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right. \right. \right. \\ &\quad \left. \left. + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \int_{t^n}^s \exp(A(t^{n+1}-\sigma))B \exp((A+B)\sigma)d\sigma ds \right)^m \right. \\ &\quad \left. - \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^m \right\| c(t^n) \end{aligned} \quad (48)$$

$$\begin{aligned} &\leq \left\| \left( \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^m \right. \right. \\ &\quad \left. \left. + \binom{m}{1} \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp(As)ds \right)^{m-1} \right. \right. \end{aligned}$$

$$\begin{aligned}
& \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \int_{t^n}^s \exp(A(t^{n+1} - \sigma))B \exp((A+B)\sigma) d\sigma ds \Big) \\
& + \dots + \left( \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \int_{t^n}^s \exp(A(t^{n+1} - \sigma))B \exp((A+B)\sigma) d\sigma ds \right)^m \\
& - \left( \exp(A\tau_n) + \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \exp(As) ds \right)^m \Big) c(t^n) \Big\| \quad (49)
\end{aligned}$$

$$\leq mC\tau_n^{2\alpha} \|c(t^n)\|, \quad (50)$$

where  $m$  are number of the local steps,  $C$  is a constant independent of  $\tau_n$ .

The same recursive argument can be done for arbitrary  $i_A \in \mathbb{N}^+$  and we obtain

$$\|S_{i_A}^m - S_{\text{num}, i_A}^m\| \leq mC\tau_n^{i_A\alpha} \|c(t^n)\|, \quad (51)$$

where  $m$  are number of the local steps,  $C$  is a constant independent of  $\tau_n$  and  $i_A$  are the number of iterative steps.

**Remark 2.** Local we have an error  $O(t^{i_A\alpha})$   
global we have an error  $O(t^{i_A\alpha-1})$ .

This means we need sufficient iterative steps  $i_A$ . At least for  $\alpha \in (0, 1)$  and an assumed convergence ( $t^c$ ) with order  $c > 0$ , we need

$$i_A \geq \frac{c-1}{\alpha} \quad (52)$$

iterative steps.

In numerical experiments we obtain much more better results and achieve second or third order methods with two and three iterative steps.

In the next section we describe the computation of the integral formulation with exp-functions.

#### 4. Computation of the iterative splitting schemes: Closed formulation

In the last years, the computational effort to compute integral with exp-function has increased, we present a closed form, and re-substitute the integral with closed functions. Such benefits accelerate the computation and made the ideas to parallelize, see [2] and [8].

*Recursion.* We study the stability of the linear system (6) and (7), based on different closed formulations.

We consider the suitable vector norm  $\|\cdot\|$  on  $\mathbb{R}^M$ , together with its induced operator norm. The matrix exponential of  $Z \in \mathbb{R}^{M \times M}$  is denoted by  $\exp(Z)$ . We assume that:

$$\|\exp(\tau_n A)c^n\| \leq K_A \|c^n\| \quad \text{and} \quad \|\exp(\tau B)c^n\| \leq K_B \|c^n\| \quad \text{for all } \tau_n > 0,$$

where  $K_A, K_B \in \mathbb{R}^+$  are given as the growth estimation of the exponential functions, see [5].

It can be shown that the system (1) implies  $\|\exp(\tau_n (A + B))c^n\| \leq \tilde{K}\|c^n\|$  and is itself stable.

For more transparency of the splitting scheme (6) and (7), we consider a well-conditioned system of eigenvectors whereby we can consider the eigenvalues  $\lambda_1$  of  $A$  and  $\lambda_2$  of  $B$  instead of the operators  $A, B$  themselves.

We assume that all initial values  $c_i(t^n) = c_{\text{approx}}(t^n)$  with  $i = 0, 1, 2, \dots$ , are as  $\|c_{\text{approx}}(t_n) - c_n\| \leq O(\tau^m)$  where  $m$  is the order, see [5].

Further we assume  $\lambda_1 \neq \lambda_2$ , otherwise we do not consider the iterative splitting method, while the time-scales are equal, see [7].

*A( $\alpha$ )-stability.* We define  $z_k = \tau\lambda_k$ ,  $k = 1, 2$ . We start with  $c_0(t) = u^n$  and we obtain:

$$c_{2m}(t^{n+1}) = S_m(z_1, z_2) c_n, \quad (53)$$

where  $S_m$  is the stability function of the scheme with  $m$ -iterations.

Let us consider the  $A(\alpha)$ -stability given by the following eigenvalues in a wedge:

$$\mathcal{W} = \{\zeta \in \mathbb{C} : |\arg(\zeta)| \leq \alpha\}.$$

For the A-stability we have  $|S_m(z_1, z_2)| \leq 1$  whenever  $z_1, z_2 \in \mathcal{W}_{\pi/2}$ .

The stability of the splitting schemes are given in the following theorems with respect to  $A$  and  $A(\alpha)$ -stability.

#### 4.1. Computation of the iterative splitting methods: Closed formulation with integral computations

A further computation of the iterative schemes are given by the variation of constants, see for exponential splitting schemes [15].

To obtain analytical solutions of the differential equations:

$$\partial_t c_{2,\text{iter}} = Ac_{2,\text{iter}} + Bc_{1,\text{iter}} \quad (54)$$

$$\partial_t c_{3,\text{iter}} = Ac_{3,\text{iter}} + Bc_{2,\text{iter}} \quad (55)$$

$$\vdots$$

$$\partial_t c_{i+1} = Ac_{i+1} + Bc_i \quad (56)$$

where  $c(t^n)$  is the initial condition and  $A, B$  are bounded operators, the initialization is with  $c_{1,\text{iter}}(t) = \exp(Bt)\exp(At)c(t_n)$  is a first order splitting scheme.

The application of the variation of constants is given as:

$$c_{2,\text{iter}}(t) = \exp(At)c(t_n) + \int_{t_n}^t \exp(A(t-s))Bc_{1,\text{iter}}(s) ds, \quad (57)$$

$$c_{3,iter}(t) = \exp(Bt)c(t_n) + \int_{t_n}^t \exp(B(t-s))Ac_{2,iter}(s) ds. \quad (58)$$

We apply the numerical integration of the integral with Trapezoidal rule for the first integral and Simpson's rule for the second integral and obtain:

$$c_{1,iter}(s) = \exp(B(s-t^n)) \exp(A(s-t^n))c(t^n) \quad (59)$$

$$c_{2,iter}(s) = \exp(A(s-t^n))c(t^n) + \frac{(s-t^n)}{2}(\exp(A(s-t^n))Bc_{1,iter}(t^n) + Bc_{1,iter}(s)), \quad (60)$$

$$c_{3,iter}(s) = \exp(B(s-t^n))c(t^n) + \frac{(s-t^n)}{6} \times (\exp(B(s-t^n))Ac_{2,iter}(t^n) + 4 \exp(B(s-t^n)/2)Ac_{2,iter}(t^n + (s-t^n)/2) + Ac_{2,iter}(s)), \quad (61)$$

$$c_{4,iter}(s) = \exp(A(s-t^n))c(t^n) + \frac{(s-t^n)}{8} \times (\exp(A(s-t^n))Bc_{3,iter}(t^n) + 3 \exp(A2/3(s-t^n))Bc_{3,iter}(t^n + 2/3(s-t^n)) + 3 \exp(A1/3(s-t^n))Bc_{3,iter}(t^n + 1/3(s-t^n)) + Bc_{3,iter}(s)). \quad (62)$$

where we compute  $c_{1,iter}(t^{n+1}), c_{2,iter}(t^{n+1}), \dots$ , and  $s \in [t^n, t^{n+1}]$ ,  $\tau = t^{n+1} - t^n$ .

The fourth order method can also be computed with Bode's or Romberg's rules:

$$c_{4,iter}(s) = \exp(A(s-t^n))c(t^n) + \frac{(s-t^n)}{90} \times (7 \exp(A(s-t^n))Bc_{3,iter}(t^n) + 32 \exp(A3/4(s-t^n))Bc_{3,iter}(t^n + 1/4(s-t^n)) + 12 \exp(A1/2(s-t^n))Bc_{3,iter}(t^n + 1/2(s-t^n)) + 32 \exp(A1/4(s-t^n))Bc_{3,iter}(t^n + 3/4(s-t^n)) + 7Bc_{3,iter}(s)). \quad (63)$$

### Example.

$$c_{3,iter}(t^{n+1}) = \exp(B\tau)c(t^n) + \frac{\tau}{6}(\exp(B\tau)Ac_{2,iter}(t^n) + 4 \exp(B\tau/2)Ac_{2,iter}(t^n + \tau/2) + Ac_{2,iter}(t^{n+1})), \quad (64)$$

where we have to compute the subinterval results:

$$c_{2,iter}(t^n + \tau/2) = \exp(A\tau/2)c(t^n) + \frac{\tau}{4}(\exp(A\tau/2)Bc_{1,iter}(t^n) + Bc_{1,iter}(t^n + \tau/2)), \quad (65)$$

$$c_{1,iter}(t^n + \tau/2) = \exp(B\tau/2) \exp(A\tau/2)c(t^n). \quad (66)$$

We we have to compute  $t \in [0, T]$ , with  $t_0, t_1, \dots, t_N$  and  $N$  number of time steps, where the time steps are equidistant of  $\tau = t_j - t_{j-1}$ ,  $j = 1, \dots, N$ .

We have to compute successively  $c(t^1), c(t^2), \dots, c(t^N)$ , where the highest iterative solutions, are the initialization to the next interval, i.e.  $c_{i,\text{iter}}(t^1) \approx c(t^1), \dots, c_{i,\text{iter}}(t^{N-1}) \approx c(t^{N-1})$ .

The generalization is given with Romberg's extrapolation scheme is given in the following algorithm.

**Algorithm 1.** We apply the iterative scheme by Romberg's extrapolation method. We divide into time intervals  $[t_0, t_1], [t_1, t_2], \dots, [t_{N-1}, t_N]$ , and each subinterval  $t_n, t_{n+1}$  is solved with the iterative splitting scheme with Romberg's extrapolation method.

- (i) We start with  $n = 0$  and the initial condition  $c(0)$ , and starting solution  $c_{1,\text{iter}}(t) = \exp(A(t - t_n)) \exp(B(t - t_n))$
- (ii) We compute the time interval  $t_n, t_{n+1}$  and the solution  $c(t_{n+1})$  is obtained by:
  - (a) We start with  $i = 2$

$$c_{i,\text{iter}}(t) = \exp(A(t - t_n))c(t_n) + \int_{t_n}^t f_{A,i-1}(s)ds, \quad (67)$$

$$c_{i+1,\text{iter}}(t) = \exp(B(t - t_n))c(t_n) + \int_{t_n}^t f_{B,i}(s)ds, \quad (68)$$

where

$$f_{A,i-1}(s) = \exp(A(t - s))Bc_{i-1,\text{iter}}(s)$$

and

$$f_{B,i}(s) = \exp(B(t - s))Ac_{i,\text{iter}}(s).$$

We compute the integrals of the functions  $f_{A,i-1}, f_{B,i}$  by:

$$R(0,0) = \frac{1}{2}(t_{n+1} - t_n)(f(t_n) + f(t_{n+1})), \quad (69)$$

$$R(j,0) = \frac{1}{2}R(j-1,0) + h_j \sum_{k=1}^{2^{j-1}} f(t_1 + (2k-1)h_j), \quad (70)$$

$$R(j,j) = R(j,j-1) + \frac{1}{4^j - 1}(R(j,j-1) - R(j-1,j-1)), \quad (71)$$

$$R(j,j) = \frac{1}{4^j - 1}(4^j R(j,j-1) - R(j-1,j-1)), \quad (72)$$

where  $j \geq 1, h_j = \frac{t_{n+1} - t_n}{2^j}$  and  $f = f_{A,i-1}$  or  $f = f_{B,i}$ .

- (b) we increase  $i = i + 1$ , till  $i = I$  and we go to (iii)
- (iii) The result is given as  $c(t_{n+1}) = c_i(t_{n+1})$ , we increase  $n = n + 1$  and goto (ii), if  $n = N$  we are finished.

**Remark 3.** The same recurrent argument can be applied to the next iterative scheme. A higher numerical integration method is necessary. Here we have only to apply matrix multiplications and can skip the time-consuming integral computations. Only two evaluations for the exponential function for  $A$  and  $B$  are necessary. The main disadvantage of computing the iterative scheme exactly are the time-consuming inverse matrices. These can be skipped with numerical methods.

We have the following assumptions for the stability formulations:

$$\lim_{z_1 \rightarrow -\infty} \tau^{-1} z_1 \exp(z_1) = 0. \quad (73)$$

The stability of the methods are given in the following Theorem 2

**Theorem 2.** *We have the following stability for the integral formulated iterative schemes:*

*For the stability function  $S_{i,\text{iter}}$  of iterative splitting schemes we have*

$$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_{i,\text{iter}}(z_1, z_2)| \leq 1, \quad \text{for all } \alpha \in [0, \pi/2], \quad (74)$$

*with  $\omega \in [0, 1]$  and the initial conditions are  $c(t^n) = c_n$  and  $i$  is the iteration index.*

**Proof.** We proof the stability of  $S_{1,\text{iter}}$ .

For the extrapolation schemes we have the stability function:

$$S_{1,\text{iter}}(z_1, z_2) = \exp(z_2) \exp(z_1). \quad (75)$$

For both possibilities  $z_1 \rightarrow -\infty$  and  $z_2 \rightarrow -\infty$  we have  $S_{1,\text{iter}} \rightarrow 0$ .

For the higher iteration steps we taken into account the assumptions (59)-(60).

Based on this assumptions, we write for  $i = 2$

$$S_{2,\text{iter}}(z_1, z_2) = \exp(z_1) + \frac{1}{2}(\exp(z_1)z_2 + z_2 \exp(z_2) \exp(z_1)). \quad (76)$$

For both possibilities  $z_1, z_2 \rightarrow -\infty$  we have  $S_{2,\text{iter}} \rightarrow 0$ .

Same proof idea is used for the higher iterative steps.  $\square$

## 5. Spatial discretization schemes

To apply our method to partial differential equations, we have to consider higher order spatial discretization schemes.

In the following we discuss an efficient scheme based on Lax-Wendroff.

### 5.1. The Lax-Wendroff scheme in one dimension

The 2nd order Lax-Wendroff scheme for the 1D advection diffusion equation is:

$$\begin{aligned} u_j(t^{n+1}) = & u_j(t^n) + \frac{v}{2}(u_{j+1}(t^n) - u_{j-1}(t^n)) \\ & - \frac{zv^2}{2}(u_{j+1}(t^n) - 2u_j(t^n) + u_{j-1}(t^n)) \end{aligned} \quad (77)$$

where  $\nu = \frac{\nu \Delta t}{\Delta x}$  is the Courant-Friedrich-Levy number and

$$z(D, \nu, \Delta t) = \frac{2D}{\nu^2 \Delta t} + 1. \quad (78)$$

Since the Lax-Wendroff scheme is made for hyperbolic PDEs only we have to examine the stability closely and carry out a von-Neumann stability analysis. We encounter the following stability condition for the Fourier coefficients of  $u_j(t^n)$

$$\left| \frac{C_k^{n+1}}{C_k^n} \right|^2 = (1 + z\nu^2 \cos(\theta_k) - z\nu^2)^2 + \nu^2 \sin^2(\theta_k) \leq 1 \quad (79)$$

$$\Rightarrow \cos(\theta_k) \leq \frac{\sqrt{(1 - z\nu^2)^2 + (z^2\nu^2 - 1)(2z - z^2 - 1)} + z\nu^2 - 1}{z^2\nu^2 - 1} \quad (80)$$

Obviously this statement is always true for right-hand sides  $\geq 1$  so it is sufficient to estimate the solutions of

$$\Rightarrow z^2\nu^2 - z\nu^2 \leq \sqrt{(1 - z\nu^2)^2 + (z^2\nu^2 - 1)(2z - z^2 - 1)} \quad (81)$$

In Figure 5.1 we see the pairs  $(\nu, z)$  for which the statement holds. Of course only the region  $\{(\nu, z), \nu \geq 0, z \geq 1\}$  is of practical interest. The figure suggests that for all  $\nu \geq 0$  there are  $z \geq 1$  leading to a stable scheme. For the small  $D(z, \nu, \Delta t)$  for which this is true we may say that the problem is convection dominated.

## 5.2. Generalization to two dimensions

The advection-diffusion equation in two dimensions:

$$\partial_t u = -\mathbf{v} \nabla u + D \Delta u = -v_x \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + D \frac{\partial^2 u}{\partial y^2}, \quad (82)$$

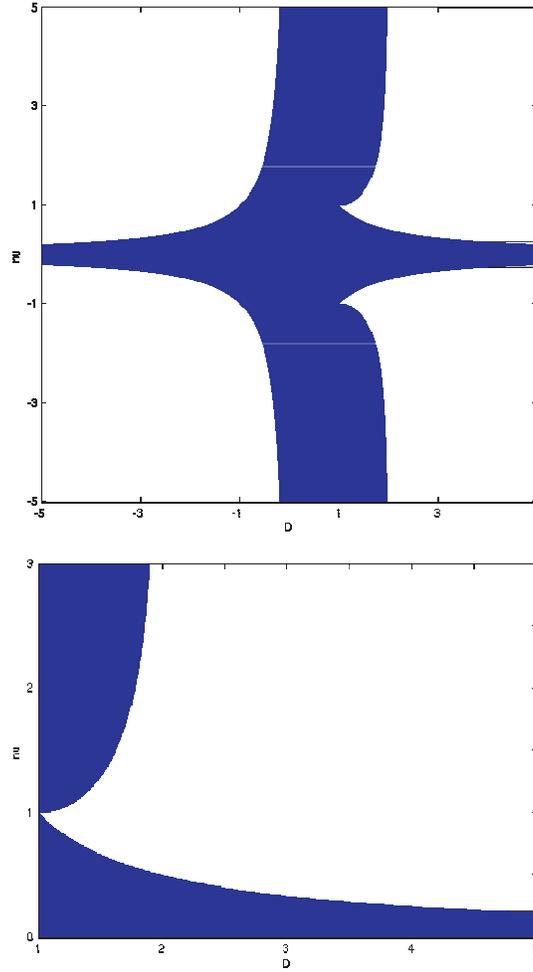
$$u(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}).$$

Application of the Lax-Wendroff scheme yields the second order accurate formula:

$$\begin{aligned} u(t^{n+1}) \approx & u(t^n) + \Delta t (-v_x \partial_x u(t^n) - v_y \partial_y u(t^n)) + \Delta t \left( D + \frac{v_x^2 \Delta t}{2} \right) \partial_{xx} u(t^n) \\ & + \Delta t \left( D + \frac{v_y^2 \Delta t}{2} \right) \partial_{yy} u(t^n) + \Delta t^2 v_x v_y \partial_{xy} u(t^n) \end{aligned} \quad (83)$$

We use central differences to achieve a second order scheme. This is similar to the 1D scheme (77) except for the new cross term incorporating  $\partial_{xy} u$ :

$$\begin{aligned} u_{ij}(t^{n+1}) = & u_{ij}(t^n) - \frac{v_x}{2} (u_{i+1j}(t^n) - u_{i-1j}(t^n)) - \frac{v_y}{2} (u_{ij+1}(t^n) - u_{ij-1}(t^n)) \\ & + \frac{z_x v_x^2}{2} (u_{i+1j}(t^n) - 2u_{ij}(t^n) + u_{i-1j}(t^n)) \end{aligned}$$



**Figure 5.1.**  $(v, z)$  inside the blue area satisfy the stability condition. The region of interest is also shown in detail.

$$\begin{aligned}
 & + \frac{z_y v_y^2}{2} (u_{ij+1}(t^n) - 2u_{ij}(t^n) + u_{ij-1}(t^n)) \\
 & + v_x v_y (u_{ij}(t) + u_{i-1j-1}(t^n) - u_{i-1j}(t^n) - u_{ij-1}(t^n)), \quad (84)
 \end{aligned}$$

introducing the Courant-Friedrich-Levy numbers and the constants  $z_x, z_y$

$$v_x = \frac{v_x \Delta t}{\Delta x} \quad v_y = \frac{v_y \Delta t}{\Delta y} \quad (85)$$

$$z_x = \left( \frac{2D}{\Delta t v_x^2} + 1 \right) \quad z_y = \left( \frac{2D}{\Delta t v_y^2} + 1 \right). \quad (86)$$

This finite difference scheme is now brought into conservation form\*:

$$u_{ij}(t^{n+1}) = u_{ij} - v_x(F_{i+1j} - F_{ij}) - v_y(G_{ij+1} - G_{ij}), \quad (87)$$

$$F_{ij} = u_{i-1j} + \frac{1}{2}\phi(\theta_{ij}^x)(1 - z_x v_1)(u_{ij} - u_{i-1j}) - \frac{1}{2}v_y(u_{i-1j} - u_{i-1j-1}), \quad (88)$$

$$G_{ij} = u_{ij-1} + \frac{1}{2}\phi(\theta_{ij}^y)(1 - z_y v_2)(u_{ij} - u_{ij-1}) - \frac{1}{2}v_x(u_{ij-1} - u_{i-1j-1}). \quad (89)$$

At this point we introduced the flux limiter  $\phi(\theta_i)$  which is used to handle steep gradients of  $u(x, y, t)$  where the Lax-Wendroff scheme adds spurious oscillations (see also [6]).  $\theta_i = \frac{u_{i-1} - u_{i-2}}{u_i - u_{i-1}}$  is a measure for the slope of  $u$  and is estimated in the  $x$  and in the  $y$  direction respectively. For our purposes we chose the van Leer limiter:

$$\phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|} \quad (90)$$

### 5.3. Dimensional splitting

During the numerical experiments we will compare the discussed Lax-Wendroff scheme to the dimensional splitting

$$\frac{\partial u}{\partial t} = A_x u + A_y u, \quad (91)$$

$$A_x = -v_x \frac{\partial u}{\partial x} + D \frac{\partial^2 u}{\partial x^2}, \quad (92)$$

$$A_y = -v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial y^2} \quad (93)$$

which is also translated into a second order finite difference scheme, filtering out the numerical viscosity  $D_{\text{num}} = \frac{v_x \Delta x}{2}$ . In this case we use an implicit BDF2 method to achieve the second order in time yielding:

$$\frac{3}{2}u_{ij}(t^{n+1}) = 2u_{ij}(t^n) - \frac{1}{2}u_{ij}(t^{n-1}) + L_x[u_{ij}(t^{n+1})] + L_y[u_{ij}(t^{n+1})] \quad (94)$$

with the spatial discretization

$$\begin{aligned} L_x[u_{ij}(t^n)] &= -v_x(u_{ij}(t^n) - u_{i-1j}(t^n)) \\ &+ \frac{v_x}{2} \left( \frac{D}{D_{\text{num}}^x} - 1 \right) (u_{i+1j}(t^n) - 2u_{ij}(t^n) + u_{i-1j}(t^n)). \end{aligned} \quad (95)$$

---

\*We suppressed the dependencies  $u_{ij}(t^n)$  on the right hand side

#### 5.4. Advection-diffusion splitting

Another way of splitting the advection-diffusion equation is the following:

$$\begin{aligned}\frac{\partial u}{\partial t} &= Au + Bu & (96) \\ A &= -\mathbf{v}\nabla \\ B &= D\Delta\end{aligned}$$

This splitting will also be used during the experiments. The implementation of the operators is very similar to the above finite differencing schemes. However numerical aspects demand the use of an explicit method namely the Adams-Bashforth method. For further discussions on the practicability of finite differencing schemes in conjunction with the iterative splitting see [10].

### 6. Numerical experiments

In the following we present numerical experiments with the closed computable iterative splitting methods and their benefits to standard schemes.

#### 6.1. First experiment

We deal with the 2-dimensional advection-diffusion equation and periodic boundary conditions

$$\partial_t u = -\mathbf{v}\nabla u + D\Delta u = -v_x \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + D \frac{\partial^2 u}{\partial y^2},$$

$$u(\mathbf{x}, t_0) = u_0(\mathbf{x}),$$

with the parameters

$$v_x = v_y = 1$$

$$D = 0.01$$

$$t_0 = 0.25.$$

The given advection-diffusion problem has an analytical solution

$$u_a(\mathbf{x}, t) = \frac{1}{t} \exp\left(\frac{-(\mathbf{x} - \mathbf{v}t)^2}{4Dt}\right)$$

which we will use as a convenient initial function:

$$u(\mathbf{x}, t_0) = u_a(\mathbf{x}, t_0).$$

We apply dimensional splitting to our problem

$$\frac{\partial u}{\partial t} = A_x u + A_y u$$

where

$$A_x = -v_x \frac{\partial u}{\partial x} + D \frac{\partial^2 u}{\partial x^2}.$$

We use a 1st order upwind scheme for  $\frac{\partial}{\partial x}$  and a 2nd order central difference scheme for  $\frac{\partial^2}{\partial x^2}$ . By introducing the artificial diffusion constant  $D_x = D - \frac{v_x \Delta x}{2}$  we achieve a 2nd order finite difference scheme

$$L_x u(x) = -v_x \frac{u(x) - u(x - \Delta x)}{\Delta x} + D_x \frac{u(x + \Delta x) + u(x) + u(x - \Delta x)}{\Delta x^2}.$$

because the new diffusion constant eliminates the first order error (i.e. the numerical viscosity) of the Taylor expansion of the upwind scheme.  $L_y u$  is derived in the same way.

We apply a BDF5 method to gain 5th order accuracy in time:<sup>†</sup>

$$L_t u(t) = \frac{1}{\Delta t} \left( \frac{137}{60} u(t + \Delta t) - 5u(t) + 5u(t - \Delta t) - \frac{10}{3} u(t - 2\Delta t) + \frac{5}{4} u(t - 3\Delta t) - \frac{1}{5} u(t - 4\Delta t) \right). \quad (97)$$

Our aim is to compare the iterative splitting method with AB-splitting. Since  $[A_x, A_y] = 0$  there is no splitting-error for the AB-splitting and therefore we cannot expect to achieve better results with the iterative splitting in terms of general numerical accuracy. Instead we will show that the iterative splitting out competes AB-splitting regarding the computational effort and round-off-errors. But first there are some remarks which have to be made concerning the special behavior of both methods when combined with high-order Runge-Kutta and BDF methods.

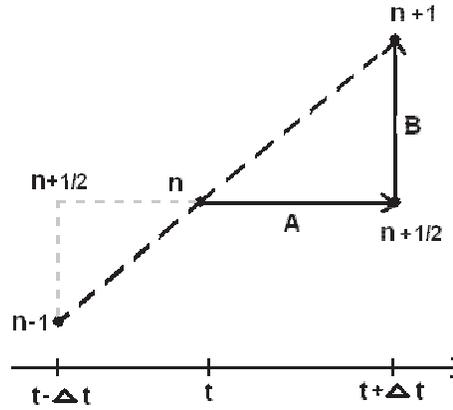


Figure 6.2. Principle of the AB-Splitting.

<sup>†</sup>Please note that the dependencies of  $u(x, t)$  are suppressed for the sake of simplicity.

*Splitting and schemes of high order in time.*

**Concerning AB-Splitting.** The principle of AB-splitting is well known and simple. The equation  $\frac{du}{dt} = Au + Bu$  is broken up into

$$\begin{aligned}\frac{du^{n+1/2}}{dt} &= Au^{n+1/2} \\ \frac{du^{n+1}}{dt} &= Bu^{n+1}\end{aligned}$$

which are connected via  $u^{n+1}(t) = u^{n+1/2}(t + \Delta t)$ . This is pointed out in figure (6.2). AB-splitting works very well for any given one-step method like the Crank-Nicholson-Scheme. Not taking into account the splitting-error (which is an error in time) it is also compatible with high order schemes such as explicit/implicit Runge-Kutta-schemes.

Things look different if one tries to use a multi-step method like the implicit BDF or the explicit Adams method with AB-splitting, these cannot be properly applied as is shown by the following example:

Choose for instance a BDF2 method which, in case of  $du/dt = f(u)$ , has the scheme

$$\frac{3}{2}u(t + \Delta t) - 2u(t) + \frac{1}{2}u(t - \Delta t) = \Delta t f(u(t + \Delta t)).$$

So the first step of the AB-splitting looks like:

$$\frac{3}{2}u^{n+1/2}(t + \Delta t) - 2u^{n+1/2}(t) + \frac{1}{2}u^{n+1/2}(t - \Delta t) = \Delta t Au(t + \Delta t).$$

Clearly  $u^{n+1/2}(t) = u^n(t)$  but what is  $u^{n+1/2}(t - \Delta t)$ ? This is also shown in figure (6.2) and it is obvious that we won't have knowledge about  $u^{n+1/2}(t - \Delta t)$  unless we compute it separately which means additional computational effort. This overhead even increases dramatically when we move to a multi-step method of higher order.

The mentioned problems with the AB-splitting will not occur with a higher order Runge-Kutta method since only knowledge of  $u^n(t)$  is needed.

**Remarks about the iterative splitting.** The BDF methods apply very well to the iterative splitting. Let us recall at this point that this method, although being a real splitting scheme, always remains a combination of the operators  $A$  and  $B$  so no steps have to be done into one direction only<sup>‡</sup>.

In particular we do a subdivision of our given time-discretization  $t_j = t_0 + j\Delta t$  into  $I$  parts. So we have subintervals  $t_{j,i} = t_j + i\Delta t/I$ ,  $0 \leq i \leq I$  on which we solve

---

<sup>‡</sup>As we will see there is an exception to this.

the following equations iteratively:

$$\frac{du^{i/I}}{dt} = Au^{i/I} + Bu^{(i-1)/I}, \tag{98}$$

$$\frac{du^{(i+1)/I}}{dt} = Au^{i/I} + Bu^{(i+1)/I} \tag{99}$$

$u^{-1/I}$  is either 0 or a reasonable approximation<sup>§</sup> while  $u^0 = u(t_j)$  and  $u^1 = u(t_j + \Delta t)$ . The crucial point here is that we only know our approximations at given times which don't happen to be the times at which a Runge-Kutta method needs to know them. Therefore, in case of a RK method, the values of the approximations have to be interpolated with at least the accuracy one wishes to attain with the splitting and this means a lot of additional computational effort. We may summarize our results now in Table 6.1 that shows which methods are practicable for each kind of splitting scheme.<sup>¶</sup>

**Table 6.1.** Practicability of single- and multi-step methods (s.s.m: single-step methods, m.s.m. multi-step methods).

	low order s.s.m.	high order s.s.m.	m.s.m.
AB-splitting	X	X	-
Iterative splitting	X	-	X

*Numerical results.* After resolving the technical aspects of this issue we can now proceed to the actual computations. The question which arises is which of the splitting methods has the least computational effort since we can expect them to solve the problem with more or less the same accuracy if we use practicable methods with equal order because  $[A_x, B_x] = 0$ . We tested the dimensional splitting of the 2d-advection-diffusion equation with the AB-splitting combined with a 5th order RK method after Dormand and Prince and with the iterative splitting in conjunction with a BDF5 scheme. We used  $40 \times 40$ - and  $80 \times 80$ -grids and completed  $n_t$  time-steps with each of which subdivided into 10 smaller steps until we reached time  $t_{\text{end}} = 0.6$  which is sufficient to see the main effects. The iterative splitting was done with 2 iterations which was already enough to attain the desired order. In Tables 7.2 and 6.3 the errors at time  $t_{\text{end}}$  and the computation times are shown.

<sup>§</sup>In fact the order of the approximation is not of much importance if we fulfill a sufficient number of iterations. In case of  $u^{-1/I} = 0$  we have the exception that a step in A-direction is done while B is left out. The error of this step vanishes after a few but mostly only one iteration

<sup>¶</sup>In favor of the iterative splitting scheme take also into the account that AB-splitting may be used along with the mentioned high order methods but cannot maintain the order if  $[A, B] \neq 0$  while the iterative splitting re-establishes the maximum order of the scheme when a sufficient number of iterations is done.

**Table 6.2.** Errors and computation times of AB-splitting and iterative splitting for a  $40 \times 40$ -grid.

Number of steps	Error AB	Error It. spl.	AB computation time	It. spl. computation time
5	0.1133	0.1154	0.203 s	0.141 s
10	0.1114	0.1081	0.500 s	0.312 s
30	0.1074	0.1072	1.391 s	0.907 s
50	0.1075	0.1074	2.719 s	1.594 s

**Table 6.3.** Errors and computation times of AB-splitting and iterative splitting for a  $80 \times 80$ -grid.

Number of steps	Error AB	Error It. spl.	AB computation time	It. spl. computation time
5	0.0288	0.0621	0.812 s	0.500 s
10	0.0276	0.0285	2.031 s	1.266 s
30	0.0268	0.0267	6.109 s	4.000 s
50	0.0265	0.0265	12.703 s	7.688 s

As we can see, the error of the iterative splitting reaches the AB-splitting error after a certain number of time-steps and stays below it for all additional steps we accomplish. Of course the error cannot sink under a certain amount which is governed by the spatial discretization. It is to be noticed that while the computation time used for the iterative splitting is always about 20%-40% less than that of the AB-splitting<sup>||</sup> the accuracy is, with a sufficient number of time-steps, slightly better than that of the AB-splitting. This is due to the roundoff error which is higher for the Runge-Kutta method because of the greater amount of basic operations needed to compute the RK steps.

A future task will be to introduce non-commuting operators in order to show the superiority of the iterative splitting over the AB-splitting when the order in time is reduced due to the splitting error.

## 7. Second experiment

In the second experiment we consider the linear and nonlinear advection-diffusion equation in two dimensions.

**Two dimensional advection-diffusion equation.** We deal with the two dimensional advection-diffusion equation with the parameters

$$v_x = v_y = 1$$

<sup>||</sup>The code for both methods is kept in the simplest possible form.

$$D = 0.01$$

$$t_0 = 0.25.$$

A convenient initial function is the exact solution

$$u_a(\mathbf{x}, t) = \frac{1}{t} \exp\left(\frac{-(\mathbf{x} - \mathbf{v}t)^2}{4Dt}\right)$$

at initial time  $t_0$ .

We used the Lax-Wendroff scheme from equation (84) with and without the flux limiter. The results are shown in Table 7.1.

**Table 7.1.** Errors for different diffusion constants on a  $80 \times 80$ -grid.

$D$	Err. Ordinary LW	Err. Flux Limiter
0.003	0.0230	0.0238
0.005	0.0603	0.1268
0.0066	0.0659	0.1543
0.01	unstable	0.2186
0.02	unstable	0.7331
0.04	unstable	1.3732
> 0.04	unstable	unstable

The values 0.0066 and 0.04 are rough estimates for the maximum diffusion for which the Lax-Wendroff scheme is stable, depending on the use of the flux limiter. We see that the flux limiter enhances the stability of the scheme while it leads to larger errors in regions where the ordinary scheme is stable.

**Dimensional splitting of a two-dimensional advection-diffusion equation.** Here the dimensional splitting from (94) is used in conjunction with the iterative splitting scheme. Results are shown in Table 7.2.

**Table 7.2.** Errors for different diffusion constants on a  $80 \times 80$ -grid.

Substeps	Iterations	$D = 0.005$	$D = 0.01$
5	1	0.1760	0.0816
	2	0.0773	0.0275
	4	0.0774	0.0275
10	1	0.1733	0.0803
	2	0.0749	0.0267
	4	0.0750	0.0267

**One and two step iterative schemes for advection-diffusion equations.** We use the splitting (96) and apply the one and two step iterative splitting

$$\frac{\partial u_i(t)}{\partial t} = Au_i + Bu_{i-1} \quad (100)$$

and

$$\frac{\partial u_i(t)}{\partial t} = Au_i + Bu_{i-1} \quad (101)$$

$$\frac{\partial u_{i+1}(t)}{\partial t} = Au_i + Bu_{i+1} \quad (102)$$

respectively.

Results are shown in Tables 7.3 and 7.4. We see in case of the one step method that the choice of  $A$  and  $B$  makes a difference for 1 Iteration. This is due to the dominance of the advection part. In general there is no improvement with higher iteration numbers.

**Table 7.3.** Errors for different diffusion constants on a  $40 \times 40$ -grid with the one step iterative splitting with  $A = -\mathbf{v}\nabla$  and  $B = D\Delta$ .

Substeps	Iterations	$D = 0.1$	$D = 0.2$
5	1	0.0362	0.0128
	2	0.0374	0.0134
	4	0.0374	0.0134
10	1	0.0361	0.0132
	2	0.0375	0.0134
	4	0.0375	0.0134

**Table 7.4.** Errors for different diffusion constants on a  $40 \times 40$ -grid with the one step iterative splitting with  $A = D\Delta$  and  $B = -\mathbf{v}\nabla$ .

Substeps	Iterations	$D = 0.1$	$D = 0.2$
5	1	0.1357	0.0631
	2	0.0375	0.0134
	4	0.0374	0.0134
10	1	0.1557	0.0712
	2	0.0376	0.0135
	4	0.0375	0.0134

**One and two step iterative schemes for 2D Burgers equation.** In our last experiment we apply the iterative splitting to the 2D Burgers equation:

$$\partial_t u + \frac{1}{2} \partial_x u^2 + \frac{1}{2} \partial_y u^2 - D \partial_{xx} u - D \partial_{yy} u = 0. \quad (103)$$

**Table 7.5.** Errors for different diffusion constants on a  $40 \times 40$ -grid with the two step iterative splitting.

Substeps	Iterations	$D = 0.1$	$D = 0.02$
5	1	0.0246	0.0049
	2	0.0374	0.0134
	4	0.0374	0.0134
10	1	0.0128	0.0050
	2	0.0134	0.0134
	4	0.0134	0.0134

In this case the operators are defined as follows:

$$A = -u_{i-1} \partial_x + D \partial_{xx}, \tag{104}$$

$$B = -u_{i-1} \partial_y + D \partial_{yy} \tag{105}$$

$u_{i-1}$  denotes the  $(i - 1)$ th iteration. This takes care of the coupling between the iteration steps. We use a finite differencing scheme of third order in space (central differences) and time (explicit Adams-Bashforth) and compare the results with a scheme of only second order. We also compare the iterative scheme with a standard Strang-Marchuk splitting of second order. Numerical results are shown in Table 7.6.

**Table 7.6.** Burgers equation on a  $40 \times 40$ -grid with the two step iterative splitting and schemes of second and third order ( $t_0 = 0.25, T = 30$ ).

Substeps	Iterations	2nd order		3rd order		Strang-Splitting	
		$D = 2$	$D = 4$	$D = 2$	$D = 4$	$D = 2$	$D = 4$
5	1	2.4188e-002	6.3023e-003	2.6636e-002	8.0688e-003	4.0475e-002	3.3462e-002
	2	6.3253e-003	1.2224e-003	7.3006e-004	4.6906e-005	-	-
	3	6.3257e-003	1.2224e-003	7.2989e-004	4.6906e-005	-	-
10	1	2.6467e-002	7.1241e-003	2.9618e-002	9.3597e-003	2.7678e-002	1.7381e-002
	2	6.3420e-003	1.2264e-003	7.3038e-004	6.1562e-005	-	-
	3	6.3428e-003	1.2264e-003	7.2997e-004	4.6923e-005	-	-

**Remark 4.** In the second experiment, we can improve iterative splitting methods with embedded spatial discretization methods. Higher order schemes as Lax-Wendroff schemes are embedded as dimensional splitting schemes to our iterative splitting method. Also nonlinear equations can be embedded. To compute the iterative scheme, we have only to deal with spatial discretization schemes on each dimensions. At least less iterative steps are needed to achieve higher order results.

### 8. Conclusions and Discussions

We have presented an iterative operator-splitting method computed with extrapolation schemes. We have analyzed the splitting error for the operators.

Under weak assumptions we could proof the higher order error bounds. Closed formulations allow to compute the delicate iterative scheme efficient. We embedded spatial discretization schemes to the iterative splitting method and achieved higher order results. Numerical examples confirm the applications to differential equations and achieve the theoretical results. In the future we will focus us on the development of improved operator-splitting methods with respect to their application in timedependent and nonlinear differential equations.

## References

- [1] S.A. Chin and P. Anisimov, Gradient symplectic algorithms for solving the radial Schrödinger equation, *J. Chem. Phys.* **124**, 054106, 2006.
- [2] S.A. Chin, Multi-product splitting and Runge-Kutta-Nyström integrators, ArXiv:0809.0914; submitted to *Applied Numerical Mathematics*, 2009.
- [3] B. Davis. *Integral Transform and Their Applications. Applied Mathematical Sciences*, **25**, Springer Verlag, New York — Heidelberg — Berlin, 1978.
- [4] K.-J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, Springer, New York, 2000.
- [5] I. Farago and J. Geiser, Iterative operator-splitting methods for linear problems, *Preprint No. 1043* of the Weierstass Institute for Applied Analysis and Stochastics, (2005), 1–18; *International Journal of Computational Science and Engineering*, accepted September 2007.
- [6] R. Fazio and A. Jannelli, Second order positive schemes by means of flux limiters for the advection equation, *IANG International Journal of Applied Mathematics* **39**(1), 2009.
- [7] J. Geiser, Higher order splitting methods for differential equations: Theory and applications of a fourth order method. *Numerical Mathematics: Theory, Methods and Applications*, Global Science Press, Hong Kong, China, accepted, April 2008.
- [8] J. Geiser and S. Chin, Multi-product expansion, Suzuki's method and the Magnus integrator for solving time-dependent problems, *Preprint 2009-4*, Humboldt University of Berlin, Department of Mathematics, Germany, 2009.
- [9] J. Geiser and L. Noack, Iterative operator-splitting methods for nonlinear differential equations and applications of deposition processes, *Preprint 2008-4*, Humboldt University of Berlin, Department of Mathematics, Germany, 2008.
- [10] J. Geiser and Chr. Kravvaritis, A domain decomposition method based on iterative operator splitting method, *Applied Numerical Mathematics* **59**, 608–623, 2009.
- [11] J. Geiser and Chr. Kravvaritis, Overlapping operator splitting methods and applications in stiff differential equations, Special issue: *Novel Difference and Hyprod Methods for Differential and Integro-Differential Equations and Applications*, (Guest editors: Qin Sheng and Johnny Henderson), *Neural, Parallel, and Scientific Computations (NPSC)* **16**, 189–200, 2008.
- [12] J. Geiser, Weighted iterative operator-splitting methods: stability-theory, *Proceedings of the 6th International Conference, NMA 2006*, Borovets, Bulgaria, August 20-24, 2006, published in: *Lecture Notes in Computer Science* **4310**, 40–47, 2007.
- [13] M. Hochbruck and A. Ostermann, Explicit exponential Runge-Kutta methods for semilinear parabolic problems, *SIAM Journal on Numerical Analysis* **43**(3), 1069–1090, 2005.

- [14] E. Hansen and A. Ostermann, Exponential splitting for unbounded operators, *Mathematics of Computation*, accepted, 2008.
- [15] T. Jahnke and C. Lubich, Error bounds for exponential operator splittings, *BIT Numerical Mathematics* **40**(4), 735–745, 2000.
- [16] J. Kanney, C. Miller and C. Kelley, Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems, *Advances in Water Resources* **26**, 247–261, 2003.
- [17] C.T. Kelly, *Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics*, SIAM, Philadelphia, USA, 1995.
- [18] G.I. Marchuk, Some applications of splitting-up methods to the solution of problems in mathematical physics, *Aplikace Matematiky* **1**, 103–132, 1968.
- [19] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **5**, 506–517, 1968.
- [20] E.W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, CRC Press, Boca Raton, Florida, USA, 1998.
- [21] H.S. Wall, *Analytic Theory of Continued Fractions*, Chelsea Publishing Company, 335–361, 1973.

Jürgen Geiser, *Ernst-Moritz-Arndt University of Greifswald, Institute of Physics, Felix-Hausdorff-Str. 6, D-17489 Greifswald, Germany.*  
E-mail: juergen.geiser@uni-greifswald.de

Received January 21, 2012

Accepted March 2, 2012