Computation of Iterative operator-splitting methods

Jürgen Geiser

geiser@mathematik.hu-berlin.de

Abstract. In this paper we describe a computation of iterative operator-splitting method, which are known as competitive splitting methods, see [8] and [9]. We derived a closed form, based on commutators for the iterative method. The schemes apply extrapolation schemes and Pade approximations to the exp-functions. The error analysis describe the approximation errors. Numerical examples of ordinary and partial differential equations support the fast computation ideas.

Keyword Iterative operator-splitting method, pade approximations, extrapolation methods, error analysis, differential equations.

AMS subject classifications. 65M15, 65L05, 65M71.

1 Introduction

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

$$\partial_t \, c = Lc = (A + B)c, \; c(0) = c_0, \tag{1}$$

where L, A and B are unbounded operators.

As numerical method we will apply a 2-stage iterative splitting scheme :

$$c_i(t) = \exp(At)c_0 + \int_0^t \exp(As)Bc_{i-1} \, ds, \tag{2}$$

$$c_{i+1}(t) = \exp(Bt)c_0 + \int_0^t \exp(Bs)Ac_i \, ds, \tag{3}$$

where $i = 1, 3, 5, \ldots$ and $c_0(t) = 0$. 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. 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 [13], [5] and [11]. We concentrate on choosing extrapolation and exponential splitting schemes to obtain higher order schemes without loosing 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 3, where we discuss an efficient computation of the iterative splitting method with based on exponential splitting and extrapolation methods. In Section 4 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 τ, namely, on the time-interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for $i = 0, 2, \ldots 2m$. (cf. [13]):

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \; + \; Bc_{i-1}(t), \text{ with } \; c_i(t^n) = c^n \tag{4}$$
$$\text{and } c_0(t^n) = c^n \; , \; c_{-1} = 0.0,$$
$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) \; + \; Bc_{i+1}(t), \tag{5}$$
$$\text{with } \; c_{i+1}(t^n) = c^n \; ,$$

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.)

2.1 Two unbounded Operators

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

$$\begin{aligned} \partial_t c(x,t) = Ac(x,t) + Bc(x,t), &\quad 0 < t \le T \text{ and } x \in \Omega, \\ c(x,0) = c_0(x), &\; x \in \Omega, \\ c(x,t) = c_1(x,t), &\; x \in \partial\Omega \times [0,T], \end{aligned} \tag{6}$$

where $A, B : D(\mathbf{X}) \to \mathbf{X}$ *are given linear operators which are generators of the* C_0*-semigroup and* $c_0 \in \mathbf{X}$ *is a given element. We assume* A *and* B *have the same domains* $dom(A) = dom(B)$.

Further, we assume the following bounds:

$$||B^\alpha \exp(B\tau_n)|| \leq \kappa\tau_n^{-\alpha}. \tag{7}$$
$$||B^\alpha \exp((A+B)\tau_n)|| \leq \kappa\tau_n^{-\alpha}, \tag{8}$$
$$||\exp(A\tau_n)B^{1-\alpha}|| \leq \tilde{\kappa}\tau_n^{p(1-\alpha)}, \tag{9}$$
$$||A^\beta \exp(A\tau_n)|| \leq \kappa\tau_n^{-\beta}. \tag{10}$$
$$||A^\beta \exp((A+B)\tau_n)|| \leq \kappa\tau_n^{-\beta}, \tag{11}$$
$$||\exp(B\tau_n)A^{1-\beta}|| \leq \tilde{\kappa}\tau_n^{q(1-\beta)}, \tag{12}$$

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 $\mathcal{O}(\tau_n^m)$*, where* $\tau_n = t^{n+1} - t^n$ *and we have equidistant time-steps, with* $n = 1, \ldots, N$*. Then the iteration process (4)–(5) for* $i = 1, 3, \ldots, 2m+1$ *is consistent with the order of the consistency* $\mathcal{O}(\tau_n^{m+\alpha m})$*, where* $0 \leq \alpha < 1$.

Proof. Let us consider the iteration (4)–(5) on the sub-interval $[t^n, t^{n+1}]$.

For the first iterations we have:

$$\partial_t c_1(t) = A c_1(t), \quad t \in (t^n, t^{n+1}], \tag{13}$$

and for the second iteration we have:

$$\partial_t c_2(t) = A c_1(t) + B c_2(t), \quad t \in (t^n, t^{n+1}], \tag{14}$$

In general we have:

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

$$\partial_t c_i(t) = A c_i(t) + B c_{i-1}(t), \quad t \in (t^n, t^{n+1}], \tag{15}$$

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

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

$$\partial_t c_i(t) = A c_{i-1}(t) + B c_i(t), \quad t \in (t^n, t^{n+1}], \tag{16}$$

We have the following solutions for the iterative scheme:

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

$$c_1(t) = \exp(A(t - t^n))c(t^n), \quad t \in (t^n, t^{n+1}], \tag{17}$$

$$\begin{aligned} c_2(t) &= \exp(B(t-t^n))c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))Ac_1(s)ds, \quad t \in (t^n, t^{n+1}]. \end{aligned} \tag{18}$$

For the recursive even and odd iterations we have the solutions: For the odd iterations: $i = 2m+1$ for $m = 0, 1, 2, \ldots$

$$c_i(t) = \exp(A(t-t^n))c(t^n) + \int_{t^n}^t \exp((t-s)A)Bc_{i-1}(s)\; ds, \quad t \in (t^n, t^{n+1}], \tag{19}$$

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

$$c_i(t) = \exp(B(t - t^n))c(t^n) + \int_{t^n}^{t} \exp((t - s)B)Ac_{i-1}(s)\, ds, \quad t \in (t^n, t^{n+1}], \tag{20}$$

The consistency is given as:

For e_1 we have:

$$c_1(t^{n+1}) = \exp(A\tau_n)c(t^n), \tag{21}$$

$$\begin{aligned} c(t^{n+1}) &= \exp((A + B)\tau_n)c(t^n) = \exp(A\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \exp((s - t^n)(A + B))c(t^n)\, ds. \end{aligned} \tag{22}$$

We obtain:

$$\begin{aligned} &||e_1|| = ||c - c_1|| \leq || \exp((A + B)\tau_n)c(t^n) - \exp(A\tau_n)c(t^n)|| \qquad (23) \\ &\leq || \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \exp((s - t^n)(A + B))c(t^n)\, ds|| \\ &\leq || \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B^{1-\alpha} B^{\alpha} \exp((s - t^n)(A + B))c(t^n)\, ds|| \\ &\leq \int_{t^n}^{t^{n+1}} || \exp(A(t^{n+1} - s))B^{1-\alpha}||\ ||B^{\alpha} \exp((s - t^n)(A + B))||\, ds\ ||c(t^n)|| \\ &\leq \int_{t^n}^{t^{n+1}} \frac{1}{(t^{n+1} - s)^{p(1-\alpha)}}\ \frac{\kappa}{(s - t^n)^{\alpha}}\ ds\ ||c(t^n)|| \\ &\leq \int_{t^n}^{t^{n+1/2}} (\frac{\kappa}{(s - t^n)^{\alpha}} + \frac{C}{\tau^{p(1-\alpha)}})ds \\ &+ \int_{t^{n+1/2}}^{t^{n+1}} (\frac{C}{\tau^{\alpha}} + \frac{C}{(t^{n+1} - s)^{p(1-\alpha)}})ds \\ &\leq C(\tau^{1-\alpha} + \tau^{p\alpha} + \tau^{\alpha} + \tau^{p\alpha}) \\ &\leq C\tau^{\min((1-\alpha), p\alpha)}\ ||c(t^n)|| \qquad (24) \end{aligned}$$

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

See assumption to the interval see Figure 1:

For e_2 we have:

$$\begin{aligned} c_2(t^{n+1}) &= \exp(B\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A \exp((s - t^n)A)c(t^n)\, ds, \end{aligned} \tag{25}$$

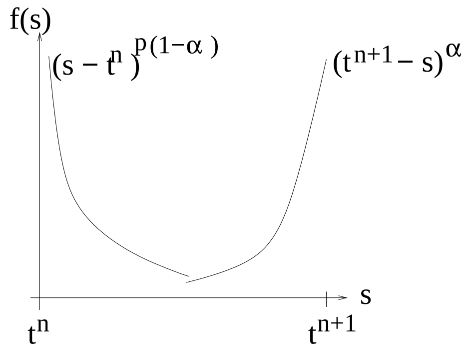

Fig. 1. Function of the estimations.

$$\begin{aligned}
&c(t^{n+1}) = \exp(B\tau_n)c(t^n) \\
&+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A\exp((s-t^n)A)c(t^n)\; ds \\
&+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \qquad (26)\\
&\int_{t^n}^{s} \exp(A(s-\rho))B\exp((\rho-t^n)(A+B))c(t^n)\; d\rho\; ds.
\end{aligned}$$

We obtain:

$$||e_2|| \leq ||\exp((A+B)\tau_n)c(t^n) - c_2|| \qquad (27)$$

$$= ||\int_{t^n}^{t^{n+1}} \exp(B(t^{n+1}-s))A \qquad (28)$$

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

$$= \int_{t^n}^{t^{n+1}} ||\exp(B(t^{n+1}-s))A^{1-\alpha}|| \qquad (29)$$

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

$$= \int_{t^n}^{t^{n+1/2}} \left(\frac{\kappa_1}{(t^{n+1}-s)^{p_1(1-\alpha_1)}} + \frac{\kappa_2}{(t^{n+1}-s)^{p_2(1-\alpha_2)}} + \frac{C_1}{\tau^{1-\alpha_2}}\right) ds \qquad (30)$$

$$+ \int_{t^{n+1/2}}^{t^{n+1}} \left(\frac{C_2}{\tau^{1-\alpha_1}} + \frac{C_1}{\tau^{(1-\alpha_2)}} + \frac{\kappa_2}{(s-t^n)^{\alpha_2}}\right) ds \qquad (31)$$

$$\leq C\tau^{\min((1-\alpha_1),p_1\alpha_1,(1-\alpha_2),p_2\alpha_2)}\; ||c(t^n)||$$

For odd and even iterations, the recursive proof is given in the following. In the next steps, we shift $t^n \to 0$ and $t^{n+1} \to \tau_n$ for simpler calculations, see [12]. The initial conditions are given with $c(0) = c(t^n)$.

For the odd iterations: $i = 2m+1$, with $m = 0, 1, 2, \ldots$, we obtain for c_i and c:

$$\begin{aligned}
& c_i(\tau_n) = \exp(A\tau_n)c(0) \\
& + \int_0^{\tau_n} \exp(As)B\exp((\tau_n - s)B)c(0)\ ds \\
& + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2 B)A\exp((\tau_n - s_1 - s_2)A)c(0)\ ds_2\ ds_1 \\
& + \ldots + \\
& + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2 A)B \int_0^{\tau_n - s_1 - s_2} \exp(s_3 A)B \ldots \\
& \int_0^{\tau_n - \sum_{j=1}^{i-1} s_j} \exp(As_i)B\exp((\tau_n - \sum_{j=1}^{i-1} s_j)A)c(0)\ ds_i \ldots ds_1,
\end{aligned} \tag{32}$$

$$\begin{aligned}
& c(\tau_n) = \exp(A\tau_n)c(0) \\
& + \int_0^{\tau_n} \exp(As)B\exp((\tau_n - s)B)c(0)\ ds \\
& + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2 B)A\exp((\tau_n - s_1 - s_2)A)c(0)\ ds_2\ ds_1 \\
& + \ldots + \\
& + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2 A)B \int_0^{\tau_n - s_1 - s_2} \exp(s_3 A)B \ldots \\
& \int_0^{\tau_n - \sum_{j=1}^{i-1} s_j} \exp(As_i)B\exp((\tau_n - \sum_{j=1}^{i-1} s_j)A)c(0)\ ds_i \ldots ds_1 \\
& + \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2 A)B \int_0^{\tau_n - s_1 - s_2} \exp(s_3 A)B \ldots \\
& \int_0^{\tau_n - \sum_{j=1}^{i} s_j} \exp(As_{i+1})B\exp((\tau_n - \sum_{j=1}^{i} s_j)(A+B))c(0)\ ds_{i+1} \ldots ds_1.
\end{aligned} \tag{33}$$

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

$$\begin{aligned}
& ||e_i|| \leq ||\exp((A+B)\tau_n)c(t^n) - c_i|| \\
& \leq \tilde{C}\tau_n^{\min_{j=1}^{i}(1-\alpha_i, p_i\alpha_i)} ||c(t^n)||,
\end{aligned} \tag{34}$$

where $\alpha = \min_{j=1}^{i}\{\alpha_i\}$ and $0 \leq \alpha_i < 1$, $0 < p_i < 1$.

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

Remark 1. An application is given to $A = \nabla D_1 \nabla$, $B = \nabla D_2 \nabla$, where D_1, D_2 are diffusion coefficients and the convergence order is given as

$$||e_1|| = \tilde{C}\tau_n^{\min(1-\alpha_1, p_1\alpha_1)}||c(t^n)|| + \mathcal{O}(\tau_n^{1+\alpha_1}) \quad (35)$$

and hence

$$||e_2|| = \tilde{\tilde{C}}||e_0||\tau_n^{\min(1-\alpha_1,p_1\alpha_1)+\min(1-\beta_1,q_1\beta_1)}$$
$$+\mathcal{O}(\tau_n^{1+\min(1-\alpha_1,p_1\alpha_1)+\min(1-\beta_1,q_1\beta_1)}), \quad (36)$$

where $0 \leq \alpha_1, \alpha_2 < 1$.

Remark 2. If we assume the consistency of $\mathcal{O}(\tau_n^m)$ for the initial value $e_1(t^n)$ and $e_2(t^n)$, we can redo the proof and obtain at least a global error of the splitting methods of $\mathcal{O}(\tau_n^{m-1})$.

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

3 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 resubstitute the integral with closed functions. Such benefits accelerate the computation and made the ideas to parallelize, see [2] and [6].

Recursion We study the stability of the linear system (4) and (5), 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:

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

where $K_A, K_B \in \mathbb{R}^+$ are given as the growth estimation of the exponential functions, see [4].

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 (4) and (5), we consider a well-conditioned system of eigenvectors whereby we can consider the eigenvalues λ_1 of A and λ_2 of B instead of the operators A, B themselves.

We assume that all initial values $c_i(t^n) = c_{approx}(t^n)$ with $i = 0, 1, 2, \ldots$, are as $||c_{approx}(t_n) - c_n|| \leq O(\tau^m)$ where m is the order, see [4].

Further we assume $\lambda_1 \neq \lambda_2$, otherwise we do not consider the iterative splitting method, while the time-scales are equal, see [5].

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

$$c_{2m}(t^{n+1}) = S_m(z_1, z_2)\; c_n \;, \tag{37}$$

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.

3.1 Exponential Splitting schemes

Here we present a closed form for the iterative splitting method based on exponential splitting schemes.

For $i = 1$, we have

$$c_1(t^{n+1}) = \exp(A\tau)\exp(B\tau)c(t^n), \tag{38}$$

where $\tau = t^{n+1} - t^n$ and we have a first order method, also known as AB splitting methods, see [4].

For $i = 2$, we have

1.) Parallel splitting method:

$$c_2(t^{n+1}) = \frac{1}{2}(\exp(A\tau)\exp(B\tau) + \exp(B\tau)\exp(A\tau))c(t^n), \tag{39}$$

where $\tau = t^{n+1} - t^n$ and we have a second order method, also known as parallel AB splitting method, see [4].

2.) Strang Splitting method:

$$c_2(t^{n+1}) = (\exp(A\frac{\tau}{2})\exp(B\tau)\exp(A\frac{\tau}{2}))c(t^n) \tag{40}$$

where $\tau = t^{n+1} - t^n$ and we have a second order method, also known as Strang splitting method, see [16]

3.) Third order method: For $i = 3$, we have (new method):

$$\begin{aligned} c_3(t^{n+1}) = (&\frac{4}{6}\exp(\frac{1}{2}A\tau)\exp(B\tau)\exp(\frac{1}{2}A\tau) \\ &+\frac{4}{6}\exp(\frac{1}{2}B\tau)\exp(A\tau)\exp(\frac{1}{2}B\tau) \\ &-\frac{1}{6}\exp(A\tau)\exp(B\tau) - \frac{1}{6}\exp(B\tau)\exp(A\tau))c(t^n) \end{aligned} \tag{41}$$

where $\tau = t^{n+1} - t^n$ and we can reduce the operators with assumptions to the commutators, e.g. $[A,[A,B]] = [B,[A,A]]$.

Higher orders are at least the derivation of the remaining form of all the commutations.

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

Theorem 2. *We have the following stability for the exponential splitting schemes:*

For the stability function S_i of the exponential splitting schemes $i = 1,2,3$ we have the following A-stability

$$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_i(z_1, z_2)| \leq 1 \ , \ \forall \ \alpha \in [0, \pi/2], \textit{and}\, i = 1,2,3, \tag{42}$$

with $\omega \in [0,1]$, the initialization is given as $c_{-1} = 0$ and the initial conditions are $c_i(t^n) = c_n$.

Proof. We consider the two possibilities, $z_1 \to -\infty$ and $z_2 \to -\infty$, for the schemes.

We obtain for $i = 1$:

$$S_1(z_1, -\infty) = \lim_{z_2 \to -\infty} \exp(z_1) \exp(z_2) = 0 \ , S_1(-\infty, z_2) = 0 \ , \tag{43}$$

and additionally for both variables:

$$S_1(-\infty, -\infty) = 0 \ , \tag{44}$$

Further, we obtain for $i = 2$:

1.) Parallel splitting

$$S_2(z_1, -\infty) = 0 \ , S_2(-\infty, z_2) = 0 \ , \tag{45}$$

and additionally for both variables:

$$S_2(-\infty, -\infty) = 0 \ , \tag{46}$$

and also for 2.) Strang-Splitting we also obtain the stability in both variables.

The same argumentation can be done for $i = 3$:

$$\begin{aligned} S_3(z_1, -\infty) = & \lim_{z_2 \to -\infty} (\frac{4}{6} \exp(\frac{1}{2} z_1) \exp(z_2) \exp(\frac{1}{2} z_1) \\ & + \frac{4}{6} \exp(\frac{1}{2} z_2) \exp(z_1) \exp(\frac{1}{2} z_2) \\ & - \frac{1}{6} \exp(z_1) \exp(z_2) - \frac{1}{6} \exp(z_2) \exp(z_1)) \end{aligned} \tag{47}$$

and additionally for both variables:

$$S_3(-\infty, -\infty) = 0 \ , \tag{48}$$

All methods are full A-stable.

3.2 Computation of higher order iterative splitting methods with extrapolation methods

We apply standard second order and fourth order methods of exponential splitting schemes and extrapolate the methods. The methods are improvement of the initialization process of iterative splitting schemes.

Second Order

For a kernel of second order splitting methods, e.g.

$$\mathcal{T}_2(\tau) = \frac{1}{2}\left(\exp(A\tau)\exp(B\tau) + \exp(B\tau)\exp(A\tau)\right), \tag{49}$$

We have the multi-product expansion as:

$$\mathcal{T}_4(\tau) = -\frac{1}{3}\mathcal{T}_2(\tau) + \frac{4}{3}\mathcal{T}_2^2\left(\frac{\tau}{2}\right) \tag{50}$$

$$\mathcal{T}_6(\tau) = \frac{1}{24}\mathcal{T}_2(\tau) - \frac{16}{15}\mathcal{T}_2^2\left(\frac{\tau}{2}\right) + \frac{81}{40}\mathcal{T}_2^3\left(\frac{\tau}{3}\right) \tag{51}$$

$$\mathcal{T}_8(\tau) = -\frac{1}{360}\mathcal{T}_2(\tau) + \frac{16}{45}\mathcal{T}_2^2\left(\frac{\tau}{2}\right) - \frac{729}{280}\mathcal{T}_2^3\left(\frac{\tau}{3}\right) + \frac{1024}{315}\mathcal{T}_2^4\left(\frac{\tau}{4}\right) \tag{52}$$

$$\begin{aligned}\mathcal{T}_{10}(\tau) = &\frac{1}{8640}\mathcal{T}_2(\tau) - \frac{64}{945}\mathcal{T}_2^2\left(\frac{\tau}{2}\right) + \frac{6561}{4480}\mathcal{T}_2^3\left(\frac{\tau}{3}\right) \\ &- \frac{16384}{2835}\mathcal{T}_2^4\left(\frac{\tau}{4}\right) + \frac{390625}{72576}\mathcal{T}_2^5\left(\frac{\tau}{5}\right) ..\end{aligned} \tag{53}$$

Fourth Order

For a kernel of fourth order splitting methods, e.g. Chin see [1]

$$\begin{aligned}&\mathcal{T}_4(\tau) = \exp(\frac{\tau}{2}(1-\frac{1}{\sqrt{3}})A)\exp(\frac{\tau}{2}B)\exp(\frac{\tau}{\sqrt{3}}\tilde{A})\exp(\frac{\tau}{2}B)\exp(\frac{\tau}{2}(1-\frac{1}{\sqrt{3}})A), \\ &\tilde{A} = A + \frac{\tau^2}{24}(2\sqrt{3}-3)[B,[B,A]],\end{aligned} \tag{54}$$

Here we can construct extrapolations with the kernels: $\mathcal{T}_4, \mathcal{T}_6, \mathcal{T}_8$ etc., i.e. $m = 0, 1, 2, \ldots$.

The closed form of the coefficients for the extrapolation is given as with closed form solutions

$$c_i = \frac{k_i^{2m}}{\sum_{j=1}^{n+1} k_j^2} \prod_{j=1(\neq i)}^{n+1} \frac{k_i^2}{k_i^2 - k_j^2} \tag{55}$$

with $\{k_1, k_2, k_3, \ldots k_n\} = \{1, 2, 3, \ldots, n\}$ and error coefficient,

$$e_{2m+2n+1} = (-1)^{n-1} \frac{k_i^{2m}}{\sum_{j=1}^{n+1} k_j^2} \prod_{i=1}^{n} \frac{1}{k_i^2}. \tag{56}$$

Generalization to the Extrapolation schemes of different Orders

The closed form of the coefficients for the extrapolation is given as with closed form solutions

$$c_i = \frac{k_i^{am}}{\sum_{j=1}^{n+1} k_j^a} \prod_{j=1(\neq i)}^{n+1} \frac{k_i^a}{k_i^a - k_j^a}, \tag{57}$$

with $\{k_1, k_2, k_3, \ldots k_n\} = \{1, 2, 3, \ldots, n\}$ and $a = 2, 3, 5, \ldots$ (prime numbers) and error coefficient,

$$e_{am+an+1} = (-1)^{n-1} \frac{k_i^{am}}{\sum_{j=1}^{n+1} k_j^a} \prod_{i=1}^{n} \frac{1}{k_i^a}, \tag{58}$$

The derivation of the scheme is given in [6].

Examples

We have the multi-product expansion for $m = 2$ as:

$$\mathcal{T}_{4+2n}(\tau) = \sum_{i=1}^{n+1} c_i \mathcal{T}_4^{k_i}(\frac{\tau}{k_i}) \tag{59}$$

Here the first terms :

$$\mathcal{T}_6(\tau) = -\frac{1}{15}\mathcal{T}_4(\tau) + \frac{16}{15}\mathcal{T}_4^2\left(\frac{\tau}{2}\right) \tag{60}$$

$$\mathcal{T}_8(\tau) = \frac{1}{336}\mathcal{T}_4(\tau) - \frac{64}{210}\mathcal{T}_4^2\left(\frac{\tau}{2}\right) + \frac{729}{560}\mathcal{T}_4^3\left(\frac{\tau}{3}\right) \tag{61}$$

We have the multi-product expansion for $m = 3$ as:

$$\mathcal{T}_{3+3n}(\tau) = \sum_{i=1}^{n+1} c_i \mathcal{T}_3^{k_i}(\frac{\tau}{k_i}) \tag{62}$$

Here the first terms :

$$\mathcal{T}_6(\tau) = -\frac{1}{7}\mathcal{T}_3(\tau) + \frac{8}{7}\mathcal{T}_3^2\left(\frac{\tau}{2}\right) \tag{63}$$

$$\mathcal{T}_9(\tau) = \frac{1}{182}\mathcal{T}_3(\tau) - \frac{64}{133}\mathcal{T}_3^2\left(\frac{\tau}{2}\right) + \frac{729}{494}\mathcal{T}_3^3\left(\frac{\tau}{3}\right) \tag{64}$$

The stability of the methods are given in the following theorem 3

Theorem 3. *We have the following stability for the extrapolation splitting schemes:*

For the stability function $S_{\mathcal{T}_2}$ of the extrapolation splitting schemes is given as

$$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_{\mathcal{T}_2}(z_1, z_2)| \leq 1 \ , \ \forall \ \alpha \in [0, \pi/2], \tag{65}$$

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

Further all related extrapolation schemes with the stability function $S_{\mathcal{T}_2}$ are stable.

Proof. We proof the stability of $\mathcal{T}_2$.

Based on the stability of $S_2 = S_{\mathcal{T}_2}$ of Theorem 2 we have a stable scheme.

For the extrapolation schemes we have the stability function:

$$S_{\mathcal{T}_{2n+2}}(z_1, z_2) = \sum_{i=1}^{n+1} c_i S_{\mathcal{T}_2}(z_1/k_i, z_2/k_i)^{k_i}. \quad (66)$$

Based on the stability of $S_{\mathcal{T}_2}$ and also $S_{\mathcal{T}_2}^{k_i}$.

We have a stable extrapolation scheme.

3.3 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 [12].

To obtain analytical solutions of the differential equations:

$$\partial_t c_{2,iter} = A c_{2,iter} + B c_{1,iter} \quad (67)$$

$$\partial_t c_{3,iter} = A c_{2,iter} + B c_{3,iter} \quad (68)$$

$$\vdots$$

$$\partial_t c_{i+1} = A c_{i+1} + B c_i \quad (69)$$

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

The application of the variation of constants is given as:

$$c_{2,iter}(t) = \exp(At)c(t_n) + \int_{t_n}^{t} \exp(A(t-s))B c_{1,iter}(s)\, ds, \quad (70)$$

$$c_{3,iter}(t) = \exp(Bt)c(t_n) + \int_{t_n}^{t} \exp(B(t-s))A c_{2,iter}(s)\, ds, \quad (71)$$

$$(72)$$

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:

$$c_{1,iter}(s) = \exp(B(s-t^n))exp(A(s-t^n))c(t^n) \quad (73)$$
$$c_{2,iter}(s) = \exp(A(s-t^n))c(t^n)$$
$$+\frac{(s-t^n)}{2}\left(exp(A(s-t^n))Bc_{1,iter}(t^n) + Bc_{1,iter}(s)\right), \quad (74)$$
$$c_{3,iter}(s) = \exp(B(s-t^n))c(t^n)$$
$$+\frac{(s-t^n)}{6}(exp(B(s-t^n))Ac_{2,iter}(t^n) + 4\exp(B(s-t^n)/2)Ac_{2,iter}(t^n+(s-t^n)/2)$$
$$+Ac_{2,iter}(s)), \quad (75)$$
$$c_{4,iter}(s) = \exp(A(s-t^n))c(t^n)$$
$$+\frac{(s-t^n)}{8}(exp(A(s-t^n))Bc_{3,iter}(t^n) + 3\exp(A2/3(s-t^n))Bc_{3,iter}(t^n+2/3(s-t^n))$$
$$+3exp(A1/3(s-t^n))Bc_{3,iter}(t^n+1/3(s-t^n)) + Bc_{3,iter}(s)). \quad (76)$$

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

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

$$c_{4,iter}(s) = \exp(A(s-t^n))c(t^n)$$
$$+\frac{(s-t^n)}{90}(7exp(A(s-t^n))Bc_{3,iter}(t^n) + 32\exp(A3/4(s-t^n))Bc_{3,iter}(t^n+1/4(s-t^n))$$
$$+12\exp(A1/2(s-t^n))Bc_{3,iter}(t^n+1/2(s-t^n))$$
$$+32exp(A1/4(s-t^n))Bc_{3,iter}(t^n+3/4(s-t^n)) + 7Bc_{3,iter}(s)). \quad (77)$$

Example:

$$c_{3,iter}(t^{n+1}) = \exp(B\tau)c(t^n)$$
$$+\frac{\tau}{6}(exp(B\tau)Ac_{2,iter}(t^n) + 4\exp(B\tau/2)Ac_{2,iter}(t^n+\tau/2)$$
$$+Ac_{2,iter}(t^{n+1})), \quad (78)$$

where we have to compute the subinterval results:

$$c_{2,iter}(t^n+\tau/2) = \exp(A\tau/2)c(t^n)$$
$$+\frac{\tau}{4}\left(exp(A\tau/2)Bc_{1,iter}(t^n) + Bc_{1,iter}(t^n+\tau/2)\right), \quad (79)$$
$$c_{1,iter}(t^n+\tau/2) = \exp(B\tau/2)exp(A\tau/2)c(t^n). \quad (80)$$

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

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

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

Algorithm 31 *We apply the iterative scheme by Romberg's extrapolation method. We divide into time intervals* $[t_0, t_1], [t_1, t_2], \ldots, [t_{N-1}, t_N]$*, and each subinterval* t_n, t_{n+1} *is solved with the iterative splitting scheme with Romberg's extrapolation method.*

1.) We start with $n = 0$ *and the initial condition* $c(0)$*.*
and starting solution $c_{1,iter}(t) = \exp(A9t - t_n)) \exp(B(t - t_n))$

2.) We compute the time interval t_n, t_{n+1} *and the solution* $c(t_{n+1})$ *is obtained by :*

a.) We start with $i = 2$

$$c_{i,iter}(t) = \exp(A(t - t_n))c(t_n) + \int_{t_n}^{t} f_{A,i-1}(s)ds, \tag{81}$$

$$c_{i+1,iter}(t) = \exp(B(t - t_n))c(t_n) + \int_{t_n}^{t} f_{B,i}(s)ds, \tag{82}$$

where $f_{A,i-1}(s) = \exp(A(t-s))Bc_{i-1,iter}(s)$ *and* $f_{B,i}(s) = \exp(B(t-s))Ac_{i,iter}(s)$

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

$$R(0,0) = \frac{1}{2}(t_{n+1} - t_n)(f(t_n) + f(t_{n+1})) \tag{83}$$

$$R(j,0) = \frac{1}{2}R(j-1,0) + h_j \sum_{k=1}^{2^{j-1}} f(t_1 + (2k-1)h_j) \tag{84}$$

$$R(j,j) = R(j,j-1) + \frac{1}{4^j - 1}(R(j,j-1) - R(j-1,j-1)) \tag{85}$$

$$R(j,j) = \frac{1}{4^j - 1}(4^j R(j,j-1) - R(j-1,j-1)) \tag{86}$$

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 3.)*

3.) The result is given as $c(t_{n+1}) = c_I(t_{n+1})$*, we increase* $n = n+1$ *and goto 2.), 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:

$$\lim_{z_1 \to -\infty} \tau^{-1} z_1 \exp(z_1) = 0 \tag{87}$$

The stability of the methods are given in the following theorem 4

Theorem 4. *We have the following stability for the integral formulated iterative schemes:*

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

$$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_{i,iter}(z_1, z_2)| \leq 1 \ , \ \forall \, \alpha \in [0, \pi/2], \tag{88}$$

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 $S_{1,iter}$.

For the extrapolation schemes we have the stability function:

$$S_{1,iter}(z_1, z_2) = \exp(z_2)\exp(z_1). \tag{89}$$

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

For the higher iteration steps we taken into account the assumptions (7)-(12).

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

$$S_{2,iter}(z_1, z_2) = \exp(z_1) + \frac{1}{2}(\exp(z_1)z_2 + z_2\exp(z_2)\exp(z_1)), \tag{90}$$

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

Same proof idea is used for the higher iterative steps.

4 Numerical experiments

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

4.1 First Experiment

We deal in the first with an ODE and separate the complex operator in two simpler operators.

We deal with the following equation :

$$\partial_t u_1 = -\lambda_1 u_1 + \lambda_2 u_2 \ , \tag{91}$$

$$\partial_t u_2 = \lambda_1 u_1 - \lambda_2 u_2 \ , \tag{92}$$

$$u_1(0) = u_{10} \ , \ u_2(0) = u_{20} \text{ (initial conditions)} \ , \tag{93}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are the decay factors and $u_{10}, u_{20} \in \mathbb{R}^+$. We have the time-interval $t \in [0, T]$.

We rewrite the equation (91) in operator notation, we concentrate us to the following equations :

$$\partial_t u = A(t)u + B(t)u \ , \tag{94}$$

$$\tag{95}$$

where $u_1(0) = u_{10} = 1.0$, $u_2(0) = u_{20} = 1.0$ are the initial conditions, where the operators are

$$A = \begin{pmatrix} -\lambda_1 & \lambda_2 \\ 0 & 0 \end{pmatrix} , \; B = \begin{pmatrix} 0 & 0 \\ \lambda_1 & -\lambda_2 \end{pmatrix} . \tag{96}$$

The concrete parameters for the experiments are given as:
$\lambda_1 = 0.05$, $\lambda_2 = 0.01$, $T = 1.0$, $u_0 = (1,1)^t$.
The L_1-error is computed as:

$$err_{num,L_1} = \sum_{k=1}^{N} |u_{exact}(t_k) - u_{num}(t_k)| \tag{97}$$

where $t_k = k\Delta t$, where $t_0, t_1, \ldots$ and $\Delta t = 0.1$.

The L_{max}-error is computed as:

$$err_{num,max} = \max_{k=1}^{N} |u_{exact}(t_k) - u_{num}(t_k)| \tag{98}$$

where $t_k = k\Delta t$, where $t_0, t_1, \ldots$ and $\Delta t = 0.1$.

In the first steps we apply the AB, Stang and 3rd order method and compared with the unsplitted solutions. The numerical results for the exponential splitting methods are given in Figure 2.

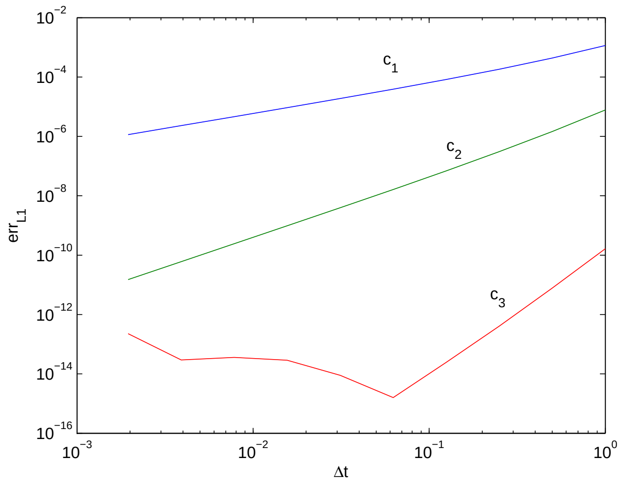

Fig. 2. Numerical errors of the methods 1.)-4.), x-axis: time, y-axis: max-error.

In a next series we apply the extrapolation schemes to our exponential splitting kernels of second and fourth order.

The numerical results of a second order kernel are given in Figure 3.

The numerical results of a fourth order kernel are given in Figure 4.

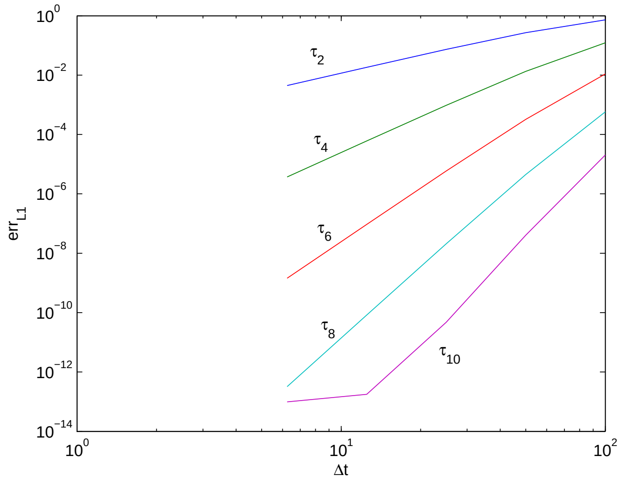

Fig. 3. Numerical errors of the methods of a second order kernel, x-axis: time, y-axis: max-error.

Remark 4. The numerical results with extrapolation methods show higher order results. Here we can apply low order kernels, e.g. 2nd or 3rd order and improve their results by choosing an cheap extrapolation schemes, based on Multi-product expansion.

4.2 Second Experiment

We deal in the first with an ODE and separate the complex operator in two simpler operators.

We deal with the 10×10 ODE system:

$$\partial_t u_1 = -\lambda_{1,1} u_1 + \lambda_{2,1} u_2 + \dots + \lambda_{10,1} u_{10}, \tag{99}$$

$$\partial_t u_2 = \lambda_{1,2} u_1 - \lambda_{2,2}(t) u_2 + \dots + \lambda_{10,2} u_{10} \ , \tag{100}$$

$$\vdots \tag{101}$$

$$\partial_t u_{10} = \lambda_{1,10} u_1 + \lambda_{2,10}(t) u_2 + \dots - \lambda_{10,10} u_{10} \ , \tag{102}$$

$$u_1(0) = u_{1,0}, \dots, \ u_{10}(0) = u_{10,0} \text{ (initial conditions)} \ , \tag{103}$$

where $\lambda_1(t) \in \mathbb{R}^+$ and $\lambda_2(t) \in \mathbb{R}^+$ are the decay factors and $u_{1,0}, \dots, u_{10,0} \in \mathbb{R}^+$. We have the time-interval $t \in [0, T]$.

We rewrite the equation (99) in operator notation, we concentrate us to the following equations :

$$\partial_t u = A(t)u + B(t)u \ , \tag{104}$$

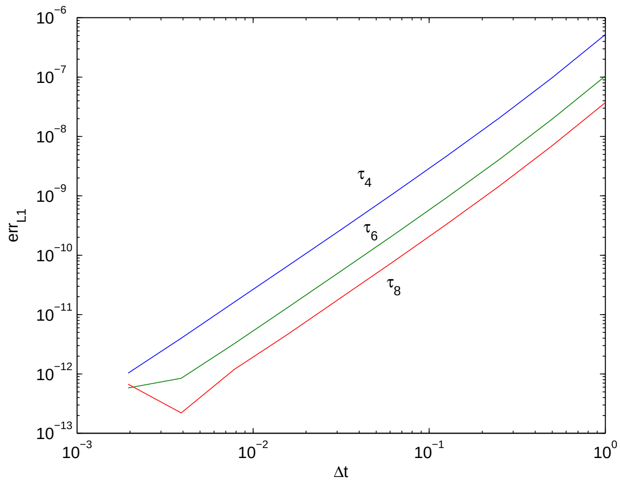

Fig. 4. Numerical errors of a fourth order kernel, x-axis: time, y-axis: max-error.

where $u_1(0) = u_{10} = 1.0$, $u_2(0) = u_{20} = 1.0$ are the initial conditions, where the operators are

$$A = \begin{pmatrix} -\lambda_{1,1}(t) \dots \lambda_{10,1}(t) \\ \lambda_{1,5}(t) \ \dots \lambda_{10,5}(t) \\ 0 \quad \dots \quad 0 \end{pmatrix} , \ B = \begin{pmatrix} 0 \quad 0 \\ \lambda_{1,6}(t) \ \dots \ \lambda_{10,6}(t) \\ \lambda_{1,10}(t) \dots -\lambda_{10,10}(t) \end{pmatrix} . \tag{105}$$

$\lambda_{1,1} = 0.09$, $\lambda_{2,1} = 0.01, \dots, \lambda_{10,1} = 0.01$

$\vdots$

$\lambda_{1,10} = 0.01, \dots \lambda_{9,10} = 0.01, \dots, \lambda_{10,10} = 0.09$

Further $\tilde{A} = A^t$ and $\tilde{B} = B^t$ are at least combination of operators A and B that influence the schemes.

The benefits of the splitting schemes are presented in Figure 5.

If we neglect the benefit of the ordered operators, the splitting schemes loose their accuracy see Figure 6.

The computational benefit of higher order schemes are given in the following Figure 7.

Remark 5. For larger systems of differential equations, we obtain the same higher order results as for lower systems. For more accuracy also the computational time for at least a 10th order extrapolation scheme is less expensive. At least a balance between the order and the computational time is important, while lower order schemes save computational time with moderate accuracy, e.g. 10^{-12}, higher order schemes have their computational benefits above an accuracy of 10^{-15}.

4.3 Third Experiment

We deal with a second order partial differential equation given as:

$$\begin{aligned} \partial_t u &= D\partial_{xx} u \\ u(x,0) &= \sin(\pi x) \\ u &= 0 \text{ on } \partial\Omega \end{aligned}$$

with exact solution

$$u_{\text{exact}}(x,t) = \sin(\pi x)\exp(-D\pi^2 t) \ .$$

We choose $D = 0.0025$, $t \in [0,1]$ and $x \in [0,1]$.
For the spatial discretization we use an upwind finite difference discretization:

$$\partial^- \partial^+ u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \ .$$

and we set the space step size to $\Delta x = \frac{1}{100}$.
Our operator is then given as

$$A = \frac{D}{\Delta x^2} \cdot \begin{pmatrix} 0 & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 0 \end{pmatrix} \tag{106}$$

We split the space-interval into two intervals by splitting the Matrix A into two Matrixes:

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} := A \ .$$

We now solve the problem

$$\partial_t u = A_1 u + A_2 u.$$

We use the Iterative Operator Splitting with Pade approximant and different discretization steps of Δx.

Based on the CFL condition of discretized scheme with the iterative splitting scheme we have:

$$\frac{2D}{\Delta x^2} \, \tau^i \leq err \leq 1 \tag{107}$$

$$2D \, N^2 \, \tau^i \leq err \leq 1 \tag{108}$$

where $\Delta x = \frac{1}{N}$ is the spatial step and N are the number of spatial points, τ is the time step and i the order of the iterative scheme. Further D is the diffusion parameter.

We obtain for the restriction of the time-step :

$$\frac{2D}{\Delta x^2}\,\tau^i \leq err \leq 1 \tag{109}$$

$$\tau \leq \Big(\frac{err}{2D\,N^2})^{\frac{1}{i}}\Big)\,. \tag{110}$$

The numerical results of the higher order schemes are given in Figure 8.

Remark 6. For partial differential equations, additional balancing problems between time and spatial scales are involved. We obtain the same higher order results as for ODE systems, when we consider the CFL conditions and control the spatial scale with the time scale. For more accuracy we have taken into account at least a 4th order scheme, which benefits for small time steps. Therefore also fine spatial grids are necessary. Here parallel computations of the Pade approximants are necessary to achieve a computational benefit with higher order schemes.

4.4 Forth Experiment

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

$$\begin{aligned} \partial_t u &= -\mathbf{v}\nabla u + D\Delta u, \\ &= -v_x \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} + D\frac{\partial^2 u}{\partial x^2} + D\frac{\partial^2 u}{\partial y^2}, \\ u(\mathbf{x}, \mathbf{t_0}) &= u_0(\mathbf{x}), \end{aligned}$$

with the parameters

$$\begin{aligned} v_x &= v_y = 1 \\ D &= 0.01 \\ t_0 &= 0.25. \end{aligned}$$

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

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

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:[1]

$$\begin{aligned} L_t u(t) \ = & \frac{1}{\Delta t} \left(\frac{137}{60} u(t + \Delta t) - 5u(t) + 5u(t - \Delta t) \right. \\ & \left. - \frac{10}{3} u(t - 2\Delta t) + \frac{5}{4} u(t - 3\Delta t) - \frac{1}{5} u(t - 4\Delta t) \right). \end{aligned} \tag{111}$$

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.

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

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

[1] Please note that the dependencies of $u(\mathbf{x}, t)$ are suppressed for the sake of simplicity.

which are connected via $u^{n+1}(t) = u^{n+1/2}(t + \Delta t)$. This is pointed out in figure (9). 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:

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

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 (9) and it is obvious that we wont 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 [2].
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 the following equations iteratively:

$$\frac{du^{i/I}}{dt} = Au^{i/I} + Bu^{(i-1)/I} \tag{112}$$

$$\frac{du^{(i+1)/I}}{dt} = Au^{i/I} + Bu^{(i+1)/I} \tag{113}$$

(114)

[2] As we will see there is an exception to this.

$u^{-1/I}$ is either 0 or a reasonable approximation[3] 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 4.4 that shows which methods are practicable for each kind of splitting scheme.[4]

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

Table 1. Practicability of single- and multi-step methods (s.s.m: single-step methods, m.s.m. multi-step methods).

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 $40x40$- and $80x80$-grids and completed n_t time-steps with each of which subdivided into 10 smaller steps until we reached time $t_{end} = 0.6$ 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 2 and 3 the errors at time t_{end} and the computation times are shown.

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

[3] 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

[4] 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.

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 2. Errors and computation times of AB-splitting and iterative splitting for a $40x40$-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

Table 3. Errors and computation times of AB-splitting and iterative splitting for a $80x80$-grid.

computation time used for the iterative splitting is always about 20%-40% less than that of the AB-splitting[5] 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.

5 Conclusions and Discussions

We have presented an iterative operator-splitting method computed with exponential splitting and 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 exponential operators efficient. 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 nonlinear differential equations.

6 Appendix

Pade approximation

[5] The code for both methods is kept in the simplest possible form.

Remark 7. To apply the exponential functions $\exp(At)$, we apply the Pade approximations, that can be computed with a general scheme.

Here the idea of the Gauss continued fractions are considered, see [18]. The Pade approximants can be formulated in such a framework.

We define a first approximate ${}_1F_1(1;b;z)$ is given as

$$ {}_1F_1(1;b;z) = \cfrac{1}{1 + \cfrac{-\,z}{b + \cfrac{z}{(b+1) + \cfrac{-\,bz}{(b+2) + \cfrac{2z}{(b+3) + \cfrac{-\,(b+1)z}{(b+4) + \ddots}}}}}}, \tag{115} $$

and the application to $\exp(z) = {}_1F_11(1;1;z)$

$$ e^z = \cfrac{1}{1 + \cfrac{-\,z}{1 + \cfrac{z}{2 + \cfrac{-\,z}{3 + \cfrac{2z}{4 + \cfrac{-\,2z}{5 + \ddots}}}}}}. \tag{116} $$

Then the Pade approximant is given as

$$ R_{m,n} = \frac{{}_1F_1(-m;-m-n;z)}{{}_1F_1(-n;-m-n;-z)}, \tag{117} $$

where the standard notation for this series is given as

$$ {}_pF_q(a_1,\ldots,a_p;b_1,\ldots,b_q;z), \tag{118} $$

although variations are sometimes used see [17].

Using the rising factorial or Pochhammer symbol:

$$ (a)_n = a(a+1)(a+2)...(a+n-1),\ (a)_0 = 1, \tag{119} $$

this can be written

$$ {}_pF_q(a_1,\ldots,a_p;b_1,\ldots,b_q;z) = \sum_{n=0}^{\infty} \frac{(a_1)_n \ldots (a_p)_n}{(b_1)_n \ldots (b_q)_n} \frac{z^n}{n!}. \tag{120} $$

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. I. Farago, 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.
5. 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.
6. 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.
7. 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.
8. J. Geiser and Chr. Kravvaritis. *A Domain Decomposition method based on iterative Operator Splitting method.* Applied Numerical Mathematics, 59, 608-623, 2009.
9. 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.
10. M. Hochbruck and A. Ostermann. *Explicit Exponential Runge-Kutta Methods for Semilinear Parabolic Problems.* SIAM Journal on Numerical Analyis, Vol. 43, Iss. 3, 1069-1090, 2005.
11. E. Hansen and A. Ostermann. *Exponential splitting for unbounded operators.* Mathematics of Computation, accepted, 2008.
12. T. Jahnke and C. Lubich. *Error bounds for exponential operator splittings.* BIT Numerical Mathematics, 40:4, 735-745, 2000.
13. 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.
14. C.T. Kelly. *Iterative Methods for Linear and Nonlinear Equations.* Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 1995.
15. G.I Marchuk. *Some applicatons of splitting-up methods to the solution of problems in mathematical physics.* Aplikace Matematiky, 1, 103-132, 1968.
16. G. Strang. *On the construction and comparision of difference schemes.* SIAM J. Numer. Anal., 5, 506-517, 1968.
17. E.W Weisstein. *CRC Concise Encyclopedia of Mathematics.* CRC Press, Boca Raton, Florida, USA, 1998.
18. H.S. Wall. *Analytic Theory of Continued Fractions.* Chelsea Publishing Company, 335-361, 1973.

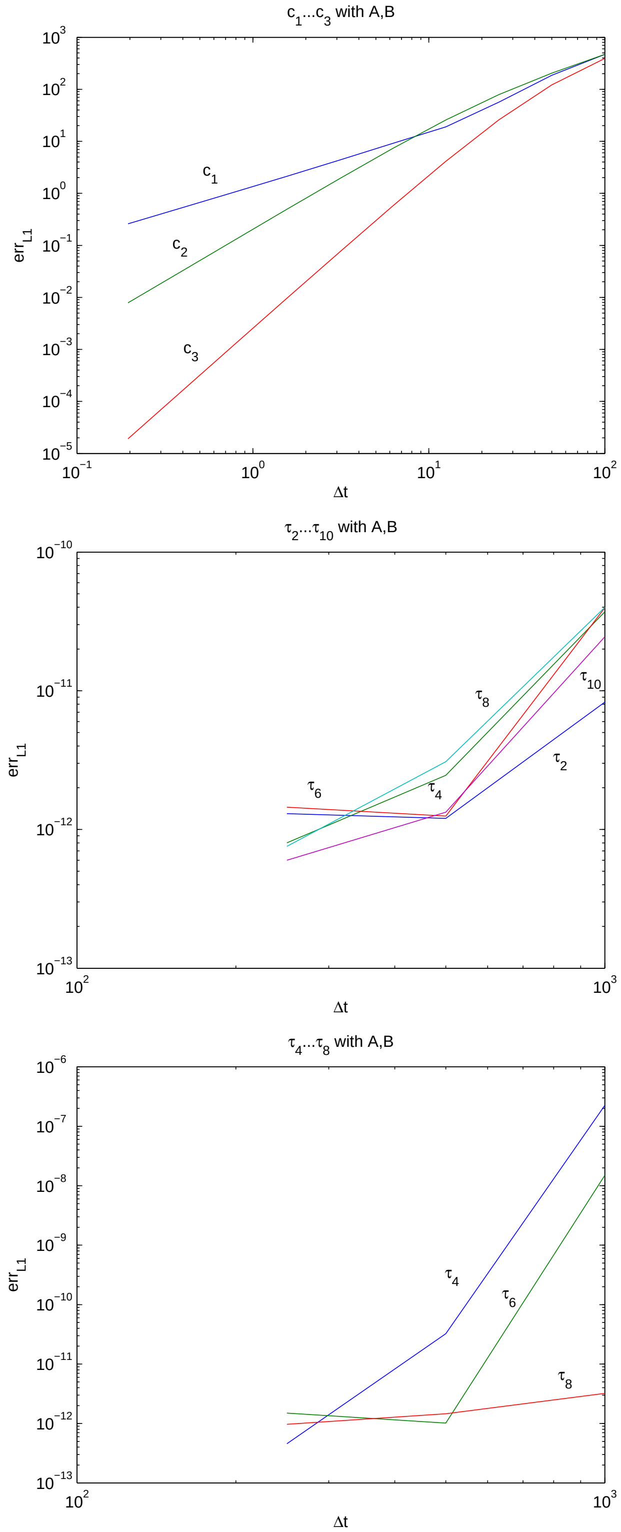

Fig. 5. Numerical errors of the splitting schemes, x-axis: time, y-axis: L_1-error (upper picture: Results of the exponential splitting schemes, middle picture: Results of the extrapolation schemes with 2nd order kernel, lower picture: Results of the extrapolation schemes with 4th order kernel).

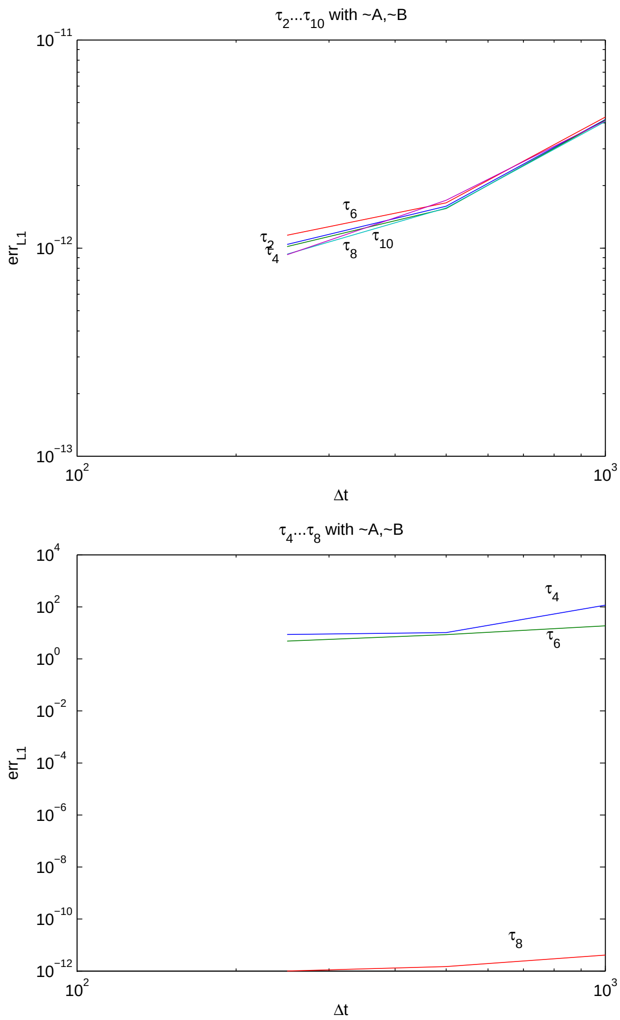

Fig. 6. Numerical errors of the splitting schemes with bad ordered operators, x-axis: time, y-axis: L_1-error (upper picture: Results of the extrapolation schemes with 2nd order kernel, lower picture: Results of the extrapolation schemes with 4th order kernel).

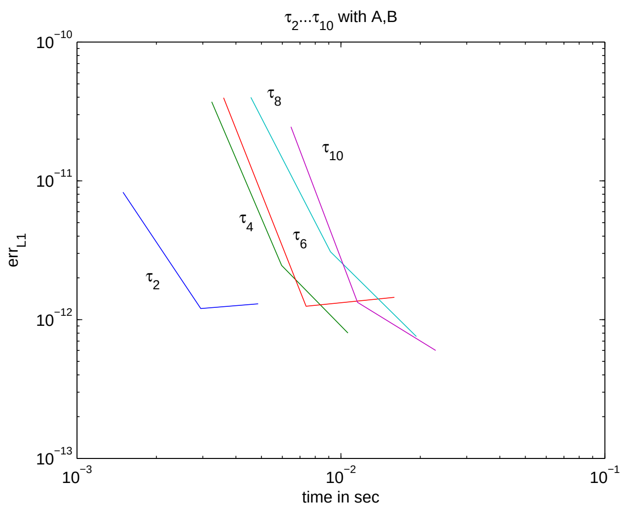

Fig. 7. Numerical errors of the splitting schemes, x-axis: computational time in [sec], y-axis: L_1-error (Results of the extrapolation schemes with 2nd order kernel).

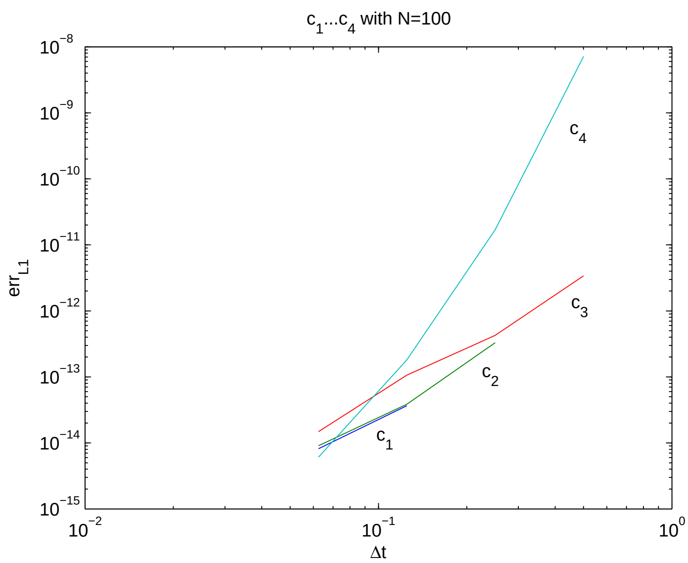

Fig. 8. Numerical result for the exponential splitting schemes of the diffusion equation with $N = 100$ spatial points.

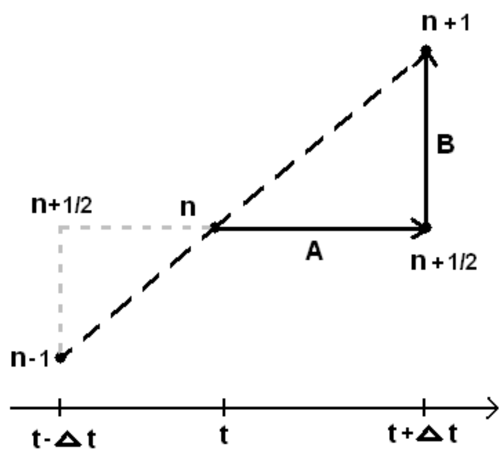

Fig. 9. Principle of the AB-Splitting.