

# ITERATIVE OPERATOR SPLITTING METHODS FOR DIFFERENTIAL EQUATIONS: PROOFTECHNIQUES AND APPLICATIONS

JÜRGEN GEISER\*

**Abstract.** In this paper we describe an iterative operator-splitting method for bounded operators. Our contribution is a novel iterative method that can be applied as a splitting method to ordinary and partial differential equations. A simple relation between the number of iterative steps and order of the splitting scheme makes this an alternative method to a time decomposition method. The iterative splitting scheme can be applied to a physical problem, but the original problem is not divided as in standard splitting schemes. We present error bounds for iterative splitting methods in the presence of bounded operators. We discuss efficient algorithms for computing the integral formulation of the splitting scheme. In experiments, we consider the benefits of the novel splitting method in terms of the number of iterations and time steps. Ordinary differential equations and convection-diffusion-reaction equations are presented in the numerical results.

**Key words.** Iterative operator-splitting method, Error analysis, convection-diffusion equation, heat equation.

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

**1. Introduction.** Iterative operator-splitting methods have been considered in the last few years, see [7], [9] and [16], as efficient solvers for differential equations. Historically, they can be seen as alternating Waveform relaxation methods, while exchanging their operators in the Waveform relaxation scheme. Such schemes are well-known and are considered to be efficient solvers in many fields of application since the 1980s, see [20] and [22]. Here, the iterative scheme is a novel method that generalizes such schemes.

Thus, for iterative splitting methods, we state the following features.

- For non-commuting operators, we may reduce the local splitting error by using more iteration steps to obtain higher-order accuracy.
- We must solve the original problem within a full split step, while keeping all operators in the equations.
- Splitting the original problem into different sub-problems including all operators of the problem is physically the best. We obtain consistent approximations after each inner step, because of the exact or approximate starting conditions for the previous iterative solution.

In our paper, we have taken into account the iterative operator-splitting schemes of a PDE solver and deal with bounded operators. Here, the balance between the operator in an integral formulation is important to bound the scheme. Such a balance, which is not necessary in the bounded case, reduces the order of the scheme. Such deficits can be reduced by using additional iteration steps to balance the spatial and time steps in the discretized form, see [7] and [8].

To discuss the analysis and application to differential equations, we concentrate in this paper on an approximate solution of the linear evolution equation:

$$(1.1) \quad \partial_t u = Lu = (A + B)u, \quad u(0) = u_0,$$

where  $L$ ,  $A$  and  $B$  are bounded operators.

---

\*Department of Mathematics, Humboldt University of Berlin, D-10099 Berlin, Germany (geiser@mathematik.hu-berlin.de).

As the numerical method, we will apply a two-stage iterative splitting scheme:

$$(1.2) \quad u_i(t) = \exp(At)u_0 + \int_0^t \exp(A(t-s))Bu_{i-1} ds,$$

$$(1.3) \quad u_{i+1}(t) = \exp(Bt)u_0 + \int_0^t \exp(B(t-s))Au_i ds,$$

where  $i = 1, 3, 5, \dots$  and  $u_0(t) = 0$ .

The outline of the paper is as follows. Operator-splitting methods are introduced in Section 2 and an error analysis of operator-splitting methods is presented. In Section 3, we discuss an error analysis of iterative methods. In Section 4, we discuss an efficient computation of the iterative splitting method with  $\phi$ -functions. In Section 5 we introduce the application of our methods to existing software tools. Finally, we discuss future work in the area of iterative splitting methods.

**2. Iterative operator splitting.** In this section consistency of the iterative splitting method is proved for unbounded generators of strongly continuous semigroup.

Consider the abstract homogenous Cauchy problem in a Banach space X

$$(2.1) \quad u'(t) = Au(t), \quad t \in [0, T]$$

$$(2.2) \quad u(0) = u_0$$

Let  $\|\cdot\|$  be the norm in X, and let  $\|\cdot\|_{L(X)}$  denote the corresponding induced operator norm. Let A be a densely defined closed linear operator in X for which there exist real constants  $M \geq 1$  and  $\omega$  such that the resolvent set  $\rho(A)$  satisfies  $\rho(A) \supset (\omega, \infty)$ , and we have the resolvent condition

$$(2.3) \quad \|(\lambda I - A)^{-n}\|_{L(X)} \leq \frac{M}{(\lambda - \omega)^n}, \quad \text{for } \lambda > \omega, \quad n = 1, 2, \dots$$

These are necessary and sufficient conditions for A to be the infinitesimal generator of a  $C_0$  semigroup of bounded linear operators, which we denote by  $S(t)$ ,  $t \geq 0$ , satisfying

$$(2.4) \quad \|S(t)\|_{L(X)} \leq Me^{\omega t}$$

see [?]. Then if  $u_0 \in D(A)$ , our cauchy problem has a unique solution on  $[0, T]$  given by

$$(2.5) \quad u(t) = S(t)u_0 .$$

Consider the abstract nonhomogeneous Cauchy problem in a Banach space X

$$(2.6) \quad u'(t) = Au(t) + f, \quad t \in [0, T]$$

$$(2.7) \quad u(0) = u_0$$

and suppose that A generates a  $C_0$ -semigroup  $S(t)$ ,  $t \geq 0$ . Then if  $u_0 \in D(A)$  and  $f \in C'([0, T]; X)$ , our Cauchy problem has a unique solution on  $[0, T]$  given by

$$(2.8) \quad u(t) = S(t)u_0 + \int_0^t S(t-s)f(s)ds$$

see [15]. If we define  $F$  by

$$(2.9) \quad F(t, \tau) = \int_{\tau}^t S(t-s)f(s)ds, \quad t \geq \tau, 0,$$

we can define (2.8) as

$$(2.10) \quad u(t) = S(t)u_0 + F(t, 0).$$

Sometimes it can be useful to apply the iterative operator splitting approach [9, 5, 4], when looking for approximate solutions to (2.1), (2.2). Let  $A_1$  and  $A_2$  be infinitesimal generators of  $C_0$  semigroups  $S_1(t)$ ,  $S_2(t)$ , satisfying

$$(2.11) \quad A_1 + A_2 = A, \quad D(A_1) = D(A_2) = D(A),$$

then we can write (2.5) as

$$(2.12) \quad u(t) = S_2(t)u_0 + F_1(0, t) \quad .$$

where  $F_1(\tau, t) = \int_{\tau}^t S_2(t-s)A_1S(s)u_0ds$ .

CASE I: Initial guess  $u_0(t) \equiv 0$

Let  $0 < h \leq T$  and  $t_n = nh$ ,  $n = 0, 1, 2, \dots, [T/h]$ , then we have the following iterative splitting algorithm for  $i = 1, 3, \dots, 2m+1$

$$(2.13) \quad u'_i(t) = A_1u_i(t) + A_2u_{i-1}(t), \quad t^n < t < t^{n+1},$$

$$(2.14) \quad u_i(t^n) = u^n,$$

$$(2.15) \quad u'_{i+1}(t) = A_1u_i(t) + A_2u_{i+1}(t), \quad t^n < t < t^{n+1},$$

$$(2.16) \quad u_{i+1}(t^n) = u^n,$$

where  $u_1, u_2 \in C'([0, T]; X)$ ,  $u^n$  is the known split approximation at time level  $t = t^n$  and  $u_0 \equiv 0$  is our initial guess. The split approximation at the time-level  $t = t^{n+1}$  is defined as  $u^{n+1} = u_{2m+2}(t^n)$ , and we choose to approximate  $u(t^n)$  by  $u_{2m+2}(t^n)$ , see [9, 5, 4].

When  $i = 1$ , we have

$$(2.17) \quad u'_1(t) = A_1u_1(t), \quad u_1(0) = u_0$$

$$(2.18) \quad u'_2(t) = A_1u_1(t) + A_2u_2(t), \quad u_2(0) = u_0$$

for  $[0, t]$  interval and we choose to approximate  $u(t)$  by  $u_2(t)$ . The Eqn. (2.17) has a unique solution on  $[0, t]$  interval given by

$$(2.19) \quad u_1(t) = S_1(t)u_0$$

$$(2.20)$$

and Eqn.(2.18) has the exact solution from separation of variables such that

$$(2.21) \quad u_2(t) = S_2(t)u_0 + \int_0^t S_2(t-s)A_1S_1(s)u_0ds$$

where  $u_2(t)$  is the approximate solution of our Cauchy problem. If we define

$$(2.22) \quad F_2(t, \tau) = \int_{\tau}^t S_2(t-s)A_1S_1(s)u_0ds, \quad t \geq \tau \geq 0$$

and if we denote  $u_2(t)$  by  $U^n$ , a single step of iterative splitting method can be expressed in the form

$$(2.23) \quad U^{n+1} = S_2(h)U^n + F_2(t^{n+1}, t^n)$$

and we can formulate the expression in terms of  $U^0$  as

$$(2.24) \quad U^{n+1} = (S_2(h))^{n+1}U^0 + \sum_{j=0}^n (S_2(h))^j F_2(t^{n-j+1}, t^{n-j}).$$

CASE II: Initial guess  $u_0(t) \equiv u_0$

Let  $0 < h \leq T$  and  $t_n = nh$ ,  $n = 0, 1, 2, \dots, [T/h]$ , then we have the following iterative splitting algorithm for  $i = 1, 3, \dots, 2m+1$

$$(2.25) \quad u'_i(t) = A_1 u_i(t) + A_2 u_{i-1}(t), \quad t^n < t < t^{n+1},$$

$$(2.26) \quad u_i(t^n) = u^n,$$

$$(2.27) \quad u'_{i+1}(t) = A_1 u_i(t) + A_2 u_{i+1}(t), \quad t^n < t < t^{n+1},$$

$$(2.28) \quad u_{i+1}(t^n) = u^n,$$

where  $u_1, u_2 \in C'([0, T]; X)$ ,  $u^n$  is the known split approximation at time level  $t = t^n$  and  $u_0 \equiv 0$  is our initial guess. The split approximation at the time-level  $t = t^{n+1}$  is defined as  $u^{n+1} = u_{2m+2}(t^n)$ , and we choose to approximate  $u(t^n)$  by  $u_{2m+2}(t^n)$ , see [9, 5, 4].

When  $i = 1$ , we have

$$(2.29) \quad u'_1(t) = A_1 u_1(t) + A_2 u_0(t), \quad u_1(0) = u_0$$

$$(2.30) \quad u'_2(t) = A_1 u_1(t) + A_2 u_2(t), \quad u_2(0) = u_0$$

for  $[0, t]$  interval and we choose to approximate  $u(t)$  by  $u_2(t)$ . The Eqn. (2.29) has a unique solution on  $[0, t]$  interval given by

$$(2.31) \quad u_1(t) = S_1(t)u_0 + \int_0^t S_1(t-s)A_2 u_0 ds$$

$$(2.32)$$

and Eqn.(2.30) has the exact solution from separation of variables such that

$$(2.33) \quad \begin{aligned} u_2(t) &= S_2(t)u_0 + \int_0^t S_2(t-s)A_1 S_1(s)u_0 ds \\ &+ \int_0^t S_2(t-s)A_1 \left( \int_0^s S_1(s-\sigma)A_2 u_0 d\sigma \right) ds \end{aligned}$$

where  $u_2(t)$  is the approximate solution of our Cauchy problem. If we define

$$(2.34) \quad F_2(t, \tau) = \int_\tau^t S_2(t-s)A_1 S_1(s)u_0 ds, \quad t \geq \tau \geq 0$$

$$(2.35) \quad F_3(t, \tau) = \int_\tau^t S_2(t-s)A_1 \left( \int_0^s S_1(s-\sigma)A_2 u_0 d\sigma \right) ds, \quad t \geq \tau \geq 0$$

and if we denote  $u_2(t)$  by  $U^n$ , a single step of iterative splitting method can be expressed in the form

$$(2.36) \quad U^{n+1} = S_2(h)U^n + F_2(t^{n+1}, t^n) + F_3(t^{n+1}, t^n)$$

and we can formulate the expression in terms of  $U^0$  as

$$(2.37) \quad \begin{aligned} U^{n+1} &= (S_2(h))^{n+1}U^0 \\ &+ \sum_{j=0}^n (S_2(h))^j (F_2(t^{n-j+1}, t^{n-j}) + F_3(t^{n-j+1}, t^{n-j})). \end{aligned}$$

**3. Consistency of iterative operator splitting method.** Let's recall the following definitions for consistency of iterative splitting methods from [1]

When CASE I

DEFINITION 3.1. Define  $T_h : \mathbf{X} \times [0, T - h] \rightarrow \mathbf{X}$  by

$$(3.1) \quad T_h(u_0, t) = S(h)u(t) - [S_2(h)u(t) + F_2(t + h, t)],$$

where  $u(t)$  is given in (2.5), and for each  $u_0, t$ ,  $T_h(u_0, t)$  is called the local truncation error of the iterative splitting method.

DEFINITION 3.2. The iterative splitting method is said to be consistent on  $[0, T]$  if

$$(3.2) \quad \lim_{h \rightarrow 0} \sup_{0 \leq t_n \leq T-h} \frac{\|T_h(u_0, t_n)\|}{h} = 0$$

whenever  $u_0 \in \mathcal{B}$ ,  $\mathcal{B}$  being some dense subspace of  $\mathbf{X}$ .

DEFINITION 3.3. If in the consistency relation (3.2) we have

$$(3.3) \quad \sup_{0 \leq t_n \leq T-h} h^{-1} \|T_h(u_0, t_n)\| = O(h^p), \quad p > 0,$$

the method is said to be (consistent) of order  $p$ . All we have to do is to show that the local truncation error

$$(3.4) \quad S(t)u(t) - S_2(h)u(t) - F_2(t + h, t)$$

which appears inside the norm in (3.3) is  $O(h^2)$  uniformly in  $t$ .

THEOREM 3.4. For any  $C_0$ -semigroups  $\{S(t)\}_{t \geq 0}$  of bounded linear operators with corresponding infinitesimal generator  $A$ , we have the Taylor series expansion

$$(3.5) \quad S(t)x = \sum_{j=0}^{n-1} \frac{t^j}{j!} A^j x + \frac{1}{(n-1)!} \int_0^t (t-s)^{n-1} S(s) A^n x ds, \quad \text{for all } x \in D(A^n)$$

see[12], Section 11.8. Particularly, for  $n = 3, 2$  and  $1$  we get the relations,

$$(3.6) \quad S(h)x = x + hAx + \frac{h^2}{2} A^2 x + \frac{1}{2} \int_0^h (h-s)^2 S(s) A^3 x ds,$$

$$(3.7) \quad S(h)x = x + hAx + \int_0^h (h-s) S(s) A^2 x ds,$$

$$(3.8) \quad S(h)x = x + \int_0^h S(s)Ax ds.$$

In the following lemma, we denote the bounded operator definition with respect to their underlying operator norms.

LEMMA 3.5. *Let  $A$  (resp.  $B$ ) be a closed linear operator from  $D(A) \subset X$  (resp.  $D(B) \subset X$ ) into  $X$ . If  $D(A) \subset D(B)$ , then there exists a constant  $C$  such that*

$$(3.9) \quad \|Bx\| \leq C(\|Ax\| + \|x\|) \quad \text{for all } x \in D(A).$$

Proof is given in [23], Chapter II.6, Theorem 2.

For iterative splitting method we have the truncation error (3.4) as

$$(3.10) \quad S_2(h)u(t) + F_1(t+h, t) - S_2(h)u(t) - F_2(t+h, t)$$

or

$$(3.11) \quad F_1(t+h, t) - F_2(t+h, t).$$

It can be written as

$$(3.12) \quad \begin{aligned} & F_1(t+h, t) - F_2(t+h, t) \\ &= \int_0^h S_2(h-s)A_1S(s)u_0 ds - \int_0^h S_2(h-s)A_1S_1(s)u_0 ds \\ &= \int_0^h S_2(h-s)A_1(S(s) - S_1(s))u_0 ds. \end{aligned}$$

We start with the first iterative step and obtain the following proposition.

PROPOSITION 3.6. *Let  $A$  (resp.  $A_1, A_2$ ) be an infinitesimal generator of a  $C_0$  semigroup  $S(s)$  (resp.  $S_1(s), S_2(s)$ ),  $s \geq 0$ . Let (2.11) be satisfied, and let  $T > 0$ . Then we have the approximation properties*

$$(3.13) \quad \|S(s)x - S_1(s)x\| \leq sC(T)(\|Ax\| + \|x\|), \quad 0 \leq s \leq T,$$

whenever  $x \in D(A)$ , where  $C(T)$  is constant independent of  $s$ .

*Proof.* For  $x \in D(A)$ , since  $A_1 + A_2 = A$ ,  $D(A_1) = D(A_2) = D(A)$ , we have

$$(3.14) \quad S(s)x - S_1(s)x = x + \int_0^s S(\tau)Ax d\tau - x - \int_0^s S_1(\tau)A_1x d\tau$$

$$(3.15) \quad = \int_0^s S(\tau)Ax d\tau - \int_0^s S_1(\tau)A_1x d\tau.$$

□

Since  $S(s)$  and  $S_1(s)$  are infinitesimal generators of  $C_0$  semigroups of bounded linear operators for  $A$  and  $A_1$ , closed linear operators, satisfying

$$(3.16) \quad \|S(s)\|_{L(X)} \leq Me^{\omega s} \quad \text{and} \quad \|S_1(s)\|_{L(X)} \leq M_1e^{\omega_1 s},$$

and also from Lemma 3.5, we have

$$(3.17) \quad \| S(s)x - S_1(s)x \| \leq sC(T)(\| Ax \| + \| x \|),$$

where  $C(T)$  is independent of  $s$ .

For additional iterative steps we define the Proposition 3.7.

**PROPOSITION 3.7.** *Let  $A, A_1$  (resp.  $A_2$ ) be infinitesimal generators of  $C_0$  semi-groups  $S(s), S_1(s)$  (resp.  $S_2(s)$ ),  $s \geq 0$ . Let (2.11) be satisfied, and let  $T > 0$ . Then we have the approximation properties*

$$(3.18) \quad \| A_1(S(s)x - S_1(s)x) \| \leq sC'(T)(\| A^2x \| + \| Ax \|), \quad 0 \leq s \leq T,$$

whenever  $x \in D(A)$ , where  $C'(T)$  is constant independent of  $s$ .

*Proof.* For  $x \in D(A)$ , since  $A_1 + A_2 = A$ ,  $D(A_1) = D(A_2) = D(A)$ , we have

$$(3.19) \quad A_1(S(s)x - S_1(s)x) = A_1x + \int_0^s S(\tau)A_1Ax d\tau - A_1x - \int_0^s S_1(\tau)A_1^2x d\tau$$

$$(3.20) \quad = \int_0^s S(\tau)A_1Ax d\tau - \int_0^s S_1(\tau)A_1^2x d\tau.$$

Since  $S(s)$  and  $S_1(s)$  are infinitesimal generators of  $C_0$  semigroups of bounded linear operators for  $A$  and  $A_1$ , closed linear operators, satisfying

$$(3.21) \quad \| S(s) \|_{L(X)} \leq Me^{\omega s} \quad \text{and} \quad \| S_1(t) \|_{L(X)} \leq M_1e^{\omega_1 s},$$

we have

$$(3.22) \quad \| A_1(S(s)x - S_1(s)x) \| \leq sC'(T)(\| A_1Ax \| + \| A_1x \|),$$

where  $C'(T)$  is independent of  $s$ . From Lemma 3.5, we know that we can bound  $\| A_1Ax \|$  in terms of  $\| A^2x \| + \| Ax \|$ , similarly  $\| A_1^2x \|$  in terms of  $\| A^2x \| + \| Ax \|$ . It follows that we have the estimate (3.18), possibly with a modified constant  $C'(T)$ .

We analyse Eq. (3.12).

$$(3.23) \quad \begin{aligned} & \left\| \int_0^h S_2(h-s)A_1(S(s) - S_1(s))u_0 ds \right\| \\ & \leq \int_0^h \| S_2(h-s)A_1(S(s) - S_1(s))u_0 ds \| \\ & \leq \int_0^h \| S_2(h-s) \| \| A_1(S(s) - S_1(s)) \| \| u_0 ds \\ & \leq M_2e^{\omega_2 h} \int_0^h \| A_1(S(s) - S_1(s)) \| \| u_0 ds \\ & \leq M_2e^{\omega_2 h} u_0 \int_0^h s(\| A^2x \| + \| Ax \|) ds \\ & \leq h^2 C''(T)(\| A^2x \| + \| Ax \|) ds. \end{aligned}$$

□

When CASE II

All we have to do is to show that the local truncation error

$$(3.24) \quad S(t)u(t) - S_2(h)u(t) - F_2(t+h, t) - F_3(t+h, t),$$

which appears inside the norm in (3.3) is  $O(h^3)$  uniformly in  $t$ .

For iterative splitting method we have the truncation error (2.36) as

$$(3.25) \quad S_2(h)u(t) + F_1(t+h, t) - S_2(h)u(t) - F_2(t+h, t) - F_3(t+h, t),$$

or

$$(3.26) \quad F_1(t+h, t) - F_2(t+h, t) - F_3(t+h, t).$$

It can be written as

$$(3.27) \quad \begin{aligned} & F_1(t+h, t) - F_2(t+h, t) - F_3(t+h, t) \\ &= \int_0^h S_2(h-s)A_1S(s)u_0ds - \int_0^h S_2(h-s)A_1S_1(s)u_0ds \\ & \quad - \int_0^h S_2(h-s)A_1\left(\int_0^s S_1(s-\sigma)A_2u_0d\sigma\right)ds \end{aligned}$$

$$(3.28) \quad = \int_0^h S_2(h-s)A_1\left(S(s) - S_1(s) - \int_0^s S_1(s-\sigma)A_2d\sigma\right)u_0ds$$

In the next proposition, we derive the estimation of the second iterative step.

**PROPOSITION 3.8.** *Let  $A$  (resp.  $A_1, A_2$ ) be an infinitesimal generator of a  $C_0$  semigroup  $S(s)$  (resp.  $S_1(s), S_2(s)$ ),  $s \geq 0$ . Let (2.11) be satisfied, and let  $T > 0$ . Then we have the approximation properties*

$$(3.29) \quad \begin{aligned} & \|S(s)x - S_1(s)x - \int_0^s S_1(s-\sigma)A_2xd\sigma\| \\ & \leq s^2C(T)(\|A^2x\| + \|Ax\| + \|x\|), \quad 0 \leq s \leq T, \end{aligned}$$

whenever  $x \in D(A)$ , where  $C(T)$  is constant independent of  $s$ .

*Proof.* For  $x \in D(A)$ , since  $A_1 + A_2 = A$ ,  $D(A_1) = D(A_2) = D(A)$ , we have

$$(3.30) \quad \begin{aligned} & S(s)x - S_1(s)x - \int_0^s S_1(s-\sigma)A_2xd\sigma \\ &= x + sAx + \int_0^s (s-\sigma)S(\sigma)A^2xd\sigma \\ & \quad - (x + sA_1x + \int_0^s (s-\sigma)S_1(\sigma)A_1^2xd\sigma) \\ & \quad - \left(\int_0^s (A_2x + \int_0^\sigma S_1(\sigma)A_1A_2xd\sigma)\right)ds \\ &= \int_0^s (s-\sigma)S(\sigma)A^2xd\sigma \\ & \quad - \int_0^s (s-\sigma)S_1(\sigma)A_1^2xd\sigma - \int_0^s \int_0^\sigma S_1(\sigma)A_1A_2xd\sigma ds. \end{aligned}$$



Since  $S(s)$  and  $S_1(s)$  are infinitesimal generators of  $C_0$  semigroups of bounded linear operators for  $A$  and  $A_1$ , closed linear operators, satisfying

$$(3.31) \quad \| S(s) \|_{L(X)} \leq M e^{\omega s} \quad \text{and} \quad \| S_1(t) \|_{L(X)} \leq M_1 e^{\omega_1 t},$$

we have

$$(3.32) \quad \begin{aligned} & \| S(s)x - S_1(s)x - \int_0^s S_1(s-\sigma)A_2x d\sigma \| \\ & \leq s^2 C''(T) (\| A^2x \| + \| A_1^2x \| + \| A_1A_2x \|), \end{aligned}$$

where  $C''(T)$  is independent of  $s$ . From Lemma 3.5, we know that we can bound  $\| A^2x \|$  in terms of  $\| Ax \| + \| x \|$ , similarly  $\| A_1^2x \|$  in terms of  $\| A_1x \| + \| x \|$  and  $\| A_1A_2x \|$  in terms of  $\| Ax \| + \| x \|$ . It follows that we have the estimate (3.29), possibly with a modified constant  $C''(T)$ .

□

REMARK 3.1. *The generalisation to  $i$ -th iterative steps is given as:*

$$(3.33) \quad \| S(s)x - S_i(s)x d\sigma \| \leq s^i \tilde{C}(T) \left( \sum_{j=0}^i \| A^j x \| \right)$$

where  $\tilde{C}$  is a constant independent of the method and  $S^i$  the  $i$ th iterative operator.

The proof can be done recursively with the help of Propositions 3.7 and 3.8.

**4. Computation of iterative splitting method.** Based on the integral formulation of iterative splitting schemes, it is important to formulate efficient algorithms to solve these schemes. One efficient method is to compute integral formulations of exp-functions with so called  $\phi$ -functions, which reduces the integration to a product of exp-functions, see [11]. Such algorithms can be expressed in a discrete formulation by exponential Runge-Kutta methods, see [?].

With regard to computations of the matrix exponential, an overview is given in [18].

For linear operators  $A, B : \mathcal{D}(X) \subset X \rightarrow X$  generating a  $C_0$  semigroup and a scalar  $t \in \mathbb{R}^+$ , we define the operators  $a = tA$  and  $b = tB$  and the bounded operators  $\phi_{0,A} = \exp(a)$ ,  $\phi_{0,B} = \exp(b)$  and:

$$(4.1) \quad \phi_{k,A} = \int_0^1 \exp((1-s)\tau A) \frac{s^{k-1}}{(k-1)!} ds,$$

$$(4.2) \quad \phi_{k,B} = \int_0^1 \exp((1-s)\tau B) \frac{s^{k-1}}{(k-1)!} ds,$$

for  $k \geq 1$ .

From this definition it is straightforward to prove the recurrence relation:

$$(4.3) \quad \phi_{k,A} = \frac{1}{k!} I + \tau A \phi_{k+1},$$

$$(4.4) \quad \phi_{k,B} = \frac{1}{k!} I + \tau B \phi_{k+1}.$$

We apply equations (4.3) and (4.4) to our iterative schemes (2.13)-(2.15) and obtain:

$$(4.5) \quad c_1(\tau) = \exp(A\tau)c(t_n) = \phi_{0,A}c(t_n),$$

$$(4.6) \quad c_2(\tau) = \phi_{0,A}c(t_n) + \sum_{k=1}^{\infty} B^k A \phi_{k,A},$$

where we assume that  $B$  is bounded and  $\exp B = \sum_{k=0}^{\infty} \frac{1}{k!} B^k$ .

For a bounded operator  $B$  we can apply the convolution of integrals exactly using the Laplace transformation or using numerical integration rules.

**4.1. Exact Computation of the Integrals.** The algorithm of the iterative splitting method (2.13)-(2.15) can be written in the form:

$$(4.7) \quad \partial_t c_1 = A c_1$$

$$(4.8) \quad \partial_t c_2 = A c_1 + B c_2$$

$$\vdots$$

$$(4.9) \quad \partial_t c_{i+1} = A c_{i+1} + B c_{i+1},$$

where  $c(t_n)$  is the initial condition and  $A, B$  are bounded operators.

We use the Laplace transform to solve the ordinary differential equations (4.7)-(4.9), see also [2].

The transformations for this case are given in [2]. We need to define the transformed function  $\hat{u} = \hat{u}(s, t)$ :

$$(4.10) \quad \hat{u}_i(s, t) := \int_0^{\infty} u_i(x, t) e^{-sx} dx.$$

We obtain the following analytical solution of the first iterative steps with the re-transformation:

$$(4.11) \quad c_1 = \exp(At)c(t_n),$$

$$(4.12) \quad c_2 = A(B - A)^{-1} \exp(At)c(t_n) + A(A - B)^{-1} \exp(Bt)c(t_n).$$

The solutions of the next steps can be done recursively.

The Laplace transform is given as:

$$(4.13) \quad \tilde{c}_1 = (Is + A)^{-1} c_{01}$$

$$(4.14) \quad \tilde{c}_2 = (Is + B)^{-1} c_{02} + (Is + B)^{-1} A \tilde{c}_1$$

$$(4.15) \quad \tilde{c}_3 = (Is + A)^{-1} c_{03} + (Is + A)^{-1} A \tilde{c}_2$$

$$\dots$$

Here, we assume that we fulfill the following commutator:

$$(4.16) \quad (A - B)^{-1} A = A(A - B)^{-1},$$

and that we can apply the decomposition of partial fractions:

$$(4.17) \quad (Is + A)^{-1} A (Is + B)^{-1} = (Is + A)^{-1} (B - A)^{-1} A + A(B - A)^{-1} (Is + B)^{-1}.$$

Here, we can derive our solutions:

$$(4.18) \quad \begin{aligned} c_2 &= \exp(Bt)c(t_n) \\ &+ A(B - A)^{-1} \exp(At)c(t_n) + A(A - B)^{-1} \exp(Bt)c(t_n). \end{aligned}$$

We have the following recurrent argument for the Laplace transform:  
for odd iterations:  $i = 2m + 1$ ,  
for  $m = 0, 1, 2, \dots$

$$(4.19) \quad \tilde{c}_i = (Is - A)^{-1} c_n + (Is - A)^{-1} B \tilde{c}_{i-1},$$

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

$$(4.20) \quad \tilde{c}_i(t) = (Is - B)^{-1} A \tilde{c}_{i-1} + (Is - B)^{-1} c_n,$$

We develop the next iterative solution  $c_3$  as follows:

$$(4.21) \quad \begin{aligned} c_3 &= \exp(At)c(t_n) \\ &+ BA(B - A)^{-1} t \exp(At)c(t_n) \\ &+ BA(A - B)^{-1} (B - A)^{-1} \exp(At)c(t_n) \\ &+ BA(A - B)^{-1} (A - B)^{-1} \exp(Bt)c(t_n). \end{aligned}$$

We apply the iterative steps recursively and obtain for the odd iterative scheme the following recurrent argument:

$$(4.22) \quad \begin{aligned} c_i &= \exp(At)c(t_n) \\ &+ BA(B - A)^{-1} t \exp(At)c(t_n) \\ &+ \dots \\ &+ BA \dots BA(B - A)^{-1} \dots (B - A)^{-1} t^{i-2} \exp(At)c(t_n) \\ &+ BA \dots BA(B - A)^{-1} \dots (B - A)^{-1} (A - B)^{-1} (B - A)^{-1} \exp(At)c(t_n) \\ &+ \dots + BA \dots BA(B - A)^{-1} \dots (B - A)^{-1} (A - B)^{-1} \exp(Bt)c(t_n). \end{aligned}$$

**REMARK 4.1.** *The same recurrent argument can be applied to the even iterative scheme. Here, we only have to apply matrix multiplications and can skip the time-consuming integral computations. Only two evaluations of 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.*

**4.2. Numerical Computation of the Integrals.** Here, our main contributions are to skip the integral formulation of the exponential functions and to apply only matrix multiplications of the given exponential functions. Such operators can be computed at the beginning of the evaluation.

**Evaluation by Trapezoidal rule (two iterative steps).**

We have to evaluate:

$$(4.23) \quad c_2(t) = \exp(Bt)c(t_n) + \int_{t_n}^{t_{n+1}} \exp(B(t_{n+1} - s)) A c_1(s) ds, \quad t \in (t_n, t_{n+1}],$$

where  $c_1(t) = \exp(At) \exp(Bt)c(t_n)$ .

We apply the Trapezoidal rule and obtain:

$$(4.24) \quad c_2(t) = \exp(Bt)c(t_n) + \frac{1}{2} \Delta t (B \exp(At) \exp(Bt) + \exp(At)B),$$

where  $c_1(t) = \exp(At) \exp(Bt)c(t_n)$  and  $\Delta t = t - t_n$ .

**Evaluation by Simpson rule (three iterative steps).**

We have to evaluate:

$$(4.25) \quad c_3(t) = \exp(At)c(t_n) + \int_{t_n}^{t_{n+1}} \exp(A(t_{n+1} - s))Bc_2(s)ds, \quad t \in (t_n, t_{n+1}],$$

where  $c_1(t) = \exp(A\frac{t}{2}) \exp(Bt) \exp(A\frac{t}{2})c(t_n)$ .

We apply the Simpson rule and obtain:

$$(4.26) \quad c_3(t) = \exp(At)c(t_n) + \frac{1}{6}\Delta t \left( B \exp(A\frac{t}{2}) \exp(Bt) \exp(A\frac{t}{2}) \right. \\ \left. + 4 \exp(A\frac{t}{2}) B \exp(A\frac{t}{4}) \exp(B\frac{t}{2}) \exp(A\frac{t}{4}) + \exp(At)B \right),$$

where  $c_1(t) = \exp(At) \exp(Bt)c(t_n)$  and  $\Delta t = t - t_n$ .

REMARK 4.2. *The same result can also be derived by applying BDF3 (Backward Differentiation Formula of Third Order).*

**Evaluation by Bode rule (four iterative steps).**

We have to evaluate:

$$(4.27) \quad c_4(t) = \exp(Bt)c(t_n) + \int_{t_n}^{t_{n+1}} \exp(B(t_{n+1} - s))Ac_3(s)ds, \quad t \in (t_n, t_{n+1}],$$

where  $c_3(t)$  has to be evaluated with a third-order method.

We apply the Bode rule and obtain:

$$(4.28) \quad c_4(t) = \exp(At)c(t_n) + \frac{1}{90}\Delta t \left( 7Ac_3(0) + 32 \exp(B\frac{t}{4})Ac_3(\frac{t}{4}) \right. \\ \left. + 12 \exp(B\frac{t}{2})Ac_3(\frac{t}{2}) + 32 \exp(B\frac{3t}{4})Ac_3(\frac{3t}{4}) + 7 \exp(Bt)Ac_3(t) \right),$$

where  $c_3(t)$  is evaluated by the Simpson rule or another third order method. We have  $\Delta t = t - t_n$ .

REMARK 4.3. *The same result can also be derived by applying the fourth order Gauss-Runge-Kutta method.*

In the next section, we describe the numerical results of our methods.

**5. Numerical Examples.** In the examples, we compare the iterative splitting method to standard splitting methods.

In the first and second examples, we apply Benchmark problems with differential equations to test the benefits of our iterative schemes with respect to computational time.

In the third example, we compare the standard ADI (Alternating Direction Implicit) method with the iterative operator-splitting method with embedded SBDF (Stiff Backward Differential Formula) methods as time-discretisation.

In the fourth example, we compare the standard A-B splitting method with the iterative operator-splitting method.

We also compare CPU times for each method to reveal the benefit of the iterative schemes.

**5.1. First example: Systems of ODEs.** In the first example, we deal with a large system of ODEs to verify the benefit to differential equations. Here, the computational cost increases and the balance between iterative steps and time-step size is considered.

We deal with a system of  $10 \times 10$  ODEs:

$$(5.1) \quad \partial_t u_1 = -\lambda_{1,1}u_1 + \lambda_{2,1}u_2 + \dots + \lambda_{10,1}u_{10},$$

$$(5.2) \quad \partial_t u_2 = \lambda_{1,2}u_1 - \lambda_{2,2}(t)u_2 + \dots + \lambda_{10,2}u_{10},$$

$$(5.3) \quad \vdots$$

$$(5.4) \quad \partial_t u_{10} = \lambda_{1,10}u_1 + \lambda_{2,10}(t)u_2 + \dots - \lambda_{10,10}u_{10},$$

$$(5.5) \quad u_1(0) = u_{1,0}, \dots, u_{10}(0) = u_{10,0} \text{ (initial conditions)},$$

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

We rewrite the equation (5.1) in operator notation and we concentrate on the following equations :

$$(5.6) \quad \partial_t u = A(t)u + B(t)u,$$

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

$$(5.7) \quad A = \begin{pmatrix} -\lambda_{1,1}(t) & \dots & \lambda_{10,1}(t) \\ \lambda_{1,5}(t) & \dots & \lambda_{10,5}(t) \\ 0 & \dots & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & \\ \lambda_{1,6}(t) & \dots & \lambda_{10,6}(t) \\ \lambda_{1,10}(t) & \dots & -\lambda_{10,10}(t) \end{pmatrix}.$$

$$(5.8) \quad \lambda_{1,1} = 0.09, \lambda_{2,1} = 0.01, \dots, \lambda_{10,1} = 0.01$$

$$\vdots$$

$$(5.9) \quad \lambda_{1,10} = 0.01, \dots, \lambda_{9,10} = 0.01, \dots, \lambda_{10,10} = 0.09.$$

Further, we also consider  $\tilde{A} = A^t$  and  $\tilde{B} = B^t$  as a combination of operators  $A$  and  $B$ . Such combinations have no influence on the splitting scheme, see also [8], and we consider the balance between time and number of iterations.

We compare with standard schemes, such as A-B splitting or Strang splitting, see table 5.1. The algorithms are implemented in Maple 7.0 and tested on a Linux PC with a 2.0 GHz Athlon processor.

In various tests, we consider the optimal balance between time and number of iterations, which is presented in Figure 5.1.

**REMARK 5.1.** *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 3rd order iterative scheme is less expensive. At least a balance between the order and computational time is important, while lower order schemes save computational time with moderate accuracy, e.g.  $10^{-12}$ , higher order schemes have computational benefits above an accuracy of  $10^{-15}$ .*

**5.2. Second example: Diffusion Equation.** In the second example, we deal with a partial differential equation and again verify the theoretical results. The equation is given by:

$$(5.10) \quad \frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}, \text{ in } \Omega \times [0, T],$$



We split the space interval into two intervals by splitting Matrix  $A$  into two Matrices:

$$(5.15) \quad \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} := A.$$

We now solve the problem

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

We use Iterative Operator-Splitting with a Pade approximation and different discretization steps  $\Delta x$ .

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

$$(5.16) \quad \frac{2D}{\Delta x^2} \tau^i \leq err \leq 1,$$

$$(5.17) \quad 2D N^2 \tau^i \leq err \leq 1,$$

where  $\Delta x = \frac{1}{N}$  is the spatial step and  $N$  are the number of spatial points,  $\tau$  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:

$$(5.18) \quad \frac{2D}{\Delta x^2} \tau^i \leq err \leq 1$$

$$(5.19) \quad \tau \leq \left( \frac{err}{2D N^2} \right)^{\frac{1}{i}}.$$

We must consider a balance between the spatial step size, time step size and number of iterative steps.

Here, we could not obtain more accurate results, while using additional iterative steps gave improved results. The CFL condition is important, while balancing time and space. Additionally we see an improvement with more iterative steps, if we consider sufficiently small time steps.

We compare with standard schemes, such as A-B splitting or Strang splitting, see table 5.2:

Method n	Number of iterations	CPU Time in s
iter.	2	0.0815
iter.	3	0.1408
iter.	4	0.2065
iter.	5	0.2970
iter.	6	0.4020
AB	-	0.1399
Strang	-	0.1875

TABLE 5.2

*Numerical results for the second example compared with standard AB and Strang Splitting methods.*

The numerical results of the iterative schemes are given in Figure 5.2.

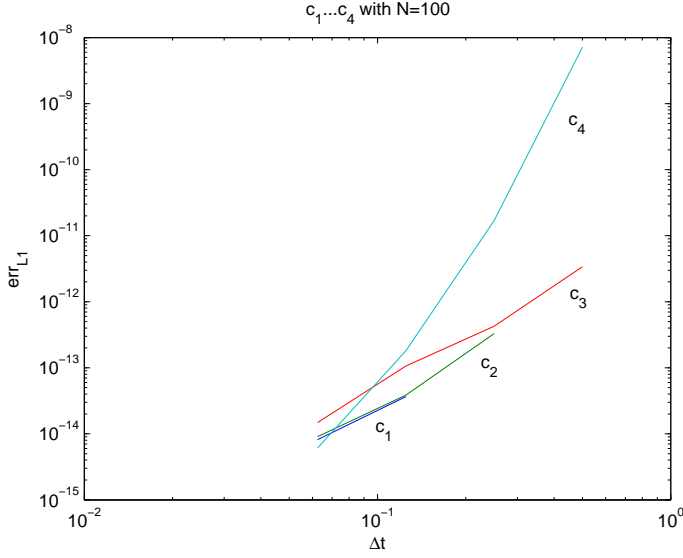


FIG. 5.2. Numerical results for exponential splitting schemes of diffusion equation with  $N = 100$  spatial points.

REMARK 5.2. For partial differential equations, additional balancing problems between time and spatial scales are involved and we must deal with the CFL condition. We obtain the same higher order results as for ODE systems, by considering optimal CFL conditions and controlling the spatial scale with the time scale. For more accuracy, we have taken into account at least four iterative steps, which benefits small time steps and we obtain higher accuracy. Therefore, fine spatial grids are necessary to see the benefits of iterative splitting schemes.

**5.3. Third example: Heat Equation.** In this section, we apply the proposed methods to the following two-dimensional problems. We consider the following heat conduction equation as a test problem with initial and boundary conditions:

$$(5.20) \quad \frac{\partial u}{\partial t} = D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2},$$

$$(5.21) \quad u(x, y, 0) = u_{analy}(x, y, 0), \quad \text{on } \Omega,$$

$$(5.22) \quad \frac{\partial u(x, y, t)}{\partial n} = 0, \quad \text{on } \Omega \times (0, T),$$

where  $u(x, y)$  is a scalar function,  $\omega = [0, 1] \times [0, 1]$ . The analytical solution of the problem is given by

$$(5.23) \quad u_{analy}(x, y, t) = \exp(-(D_x + D_y)\pi^2 t) \cos(\pi x) \cos(\pi y).$$

For approximation error, we choose  $L_\infty$  and  $L_1$  which are given by

$$err_{L_\infty} := \max(\max(|u(x_i, y_j, t^n) - u_{analy}(x_i, y_j, t^n)|))$$

$$err_{L_1} := \sum_{i,j=1}^m \Delta x \Delta y |u(x_i, y_j, t^n) - u_{analy}(x_i, y_j, t^n)|$$



TABLE 5.3

Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  when  $D_x = D_y = 1$  and  $dt = 0.0005$ .

	$\Delta x = \Delta y$	$err_{L_\infty}$	$err_{L_1}$	CPU times
ADI	1/2	2.8374e-004	2.8374e-004	0.787350
	1/4	3.3299e-005	2.4260e-005	2.074755
	1/16	1.6631e-006	8.0813e-007	22.227760
SBDF2	1/2	2.811e-004	2.811e-004	0.167132
	1/4	3.2519e-005	2.3692e-005	0.339014
	1/16	1.1500e-006	5.5882e-007	2.907924
SBDF3	1/2	2.7841e-004	2.7841e-004	0.312774
	1/4	3.1721e-005	2.3111e-005	0.460088
	1/16	6.2388e-007	3.0315e-007	4.217704
SBDF4	1/2	2.7578e-004	2.7578e-004	0.400968
	1/4	3.0943e-005	2.2544e-005	0.718028
	1/16	1.1155e-007	5.4203e-008	5.550207

$$err_{L_1} := |u(t^n) - u_{analy}(t^n)|$$

Our numerical results obtained by a standard ADI method and by an iterative operator-splitting method with  $k = 1, 2, 3$ , where the underlying time-discretization method is a SBDFk,  $k=1,2,3$ , where  $k$  stands for the order of the SBDF (Stiff Backward Differential Formula) method, are presented in Tables (5.3), (5.4) and (5.5) for various diffusion coefficients.

First, we fix the diffusion coefficients at  $D_x = D_y = 1$  with time step  $dt = 0.0005$ . Comparison of  $L_\infty$ ,  $L_1$  at  $T=0.5$  and CPU time are presented in Table (5.3) for various spatial step sizes.

In the second experiment, the diffusion coefficients are fixed at  $D_x = D_y = 0.001$  for the same time step. Comparison of errors  $L_\infty$ ,  $L_1$  at  $T=0.5$  and CPU time are presented in Table 2 for various spatial steps,  $\Delta x$  and  $\Delta y$ .

In the third experiment, the diffusion coefficients are fixed at  $D_x = D_y = 0.00001$  for the same time step. Comparison of errors  $L_\infty$ ,  $L_1$  at  $T=0.5$  and CPU time are presented in Table (5.5) for various spatial steps,  $\Delta x$  and  $\Delta y$ .

As a second example, we deal with the following time-dependent partial differential equation:

$$(5.24) \quad \partial_t u(x, y, t) = \epsilon^2 u_{xx} + u_{yy} - (1 + \epsilon^2 + 4\epsilon^2 y^2 + 2\epsilon) e^{-t} e^{x+\epsilon y^2}$$

$$(5.25) \quad u(x, y, 0) = e^{x+\epsilon y^2} \text{ in } \Omega = [-1, 1] \times [-1, 1]$$

$$(5.26) \quad u(x, y, t) = e^{-t} e^{x+\epsilon y^2} \text{ on } \partial\Omega$$

with exact solution

$$(5.27) \quad u(x, y, t) = e^{-t} e^{x+\epsilon y^2}$$

The operators are split with respect to the  $\epsilon$  scale as follows:

$$Au = \begin{cases} \epsilon^2 u_{xx} - (1 + \epsilon^2 + 4\epsilon^2 y^2 + 2\epsilon) e^{-t} e^{x+\epsilon y^2} & \text{for } (x, y) \in \Omega \end{cases}$$

TABLE 5.4

Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $D_x = D_y = 0.001$  and  $dt = 0.0005$ .

	$\Delta x = \Delta y$	$err_{L_\infty}$	$err_{L_1}$	CPU times
ADI	1/2	0.0019	0.0019	0.786549
	1/4	4.9226e-004	3.5864e-004	2.090480
	1/16	3.1357e-005	1.5237e-005	22.219374
SBDF2	1/2	0.0018	0.0018	0.167021
	1/4	4.8298e-004	3.5187e-004	0.341781
	1/16	2.1616e-005	1.0503e-005	2.868618
SBDF3	1/2	0.0018	0.0018	0.215563
	1/4	4.7369e-004	3.4511e-004	0.461214
	1/16	1.1874e-005	5.7699e-006	4.236695
SBDF4	1/2	0.0018	0.0018	0.274806
	1/4	4.6441e-004	3.3835e-004	0.717014
	1/16	2.1330e-006	1.0365e-006	5.517444

TABLE 5.5

Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $D_x = D_y = 0.00001$  and  $dt = 0.0005$ .

	$\Delta x = \Delta y$	$err_{L_\infty}$	$err_{L_1}$	CPU times
ADI	1/2	1.8694e-005	1.8694e-005	0.783630
	1/4	4.9697e-006	3.6207e-006	2.096761
	1/16	3.1665e-007	1.5386e-007	22.184733
SBDF2	1/2	1.8614e-005	1.8614e-005	0.167349
	1/4	4.8760e-006	3.5524e-006	0.342751
	1/16	2.1828e-007	1.0606e-007	2.864787
SBDF3	1/2	1.8534e-005	1.8534e-005	0.216137
	1/4	4.7823e-006	3.4842e-006	0.465666
	1/16	1.1991e-007	5.8265e-008	4.256818
SBDF4	1/2	1.8454e-005	1.8454e-005	0.399424
	1/4	4.688e-006	3.4159e-006	0.714709
	1/16	2.1539e-008	1.0466e-008	5.501323

and

$$Bu = \{ u_{yy} \quad \text{for } (x, y) \in \Omega$$

Comparison of errors for  $\Delta x = 2/5$  with respect to various  $\Delta t$  and  $\epsilon$  values with SBDF3 are given in table (5.3):

In table (5.8), we compare the numerical solution of our second model problem implemented with the Iterative Operator-Splitting method solved by the trapezoidal rule, SBDF2 and SBDF3.

The exact solution, approximate solution obtained by iterative splitting and SBDF3 and error are shown in Figure (5.6).

REMARK 5.3. *In the numerical example, we apply standard splitting methods and the iterative splitting methods. The benefit of the iterative splitting methods are more accurate solutions in shorter CPU time. The application of SBDF methods as standard*

TABLE 5.6

Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  when  $D_x = 1, D_y = 0.001$  and  $dt = 0.0005$ .

	$\Delta x = \Delta y$	$err_{L_\infty}$	$err_{L_1}$	CPU times
ADI	1/2	0.0111	0.0111	0.787006
	1/4	0.0020	0.0015	2.029179
	1/16	1.1426e-004	5.5520e-005	21.959890
SBDF2	1/2	0.0109	0.0109	0.210848
	1/4	0.0019	0.0014	0.385742
	1/16	2.5995e-005	1.2631e-005	2.913781
SBDF3	1/2	0.0108	0.0108	0.316777
	1/4	0.0018	0.0013	0.454392
	1/16	4.4834e-005	2.1785e-005	4.227773
SBDF4	1/2	0.0106	0.0106	0.395751
	1/4	0.0017	0.0013	0.709488
	1/16	1.1445e-004	5.5613e-005	5.562917

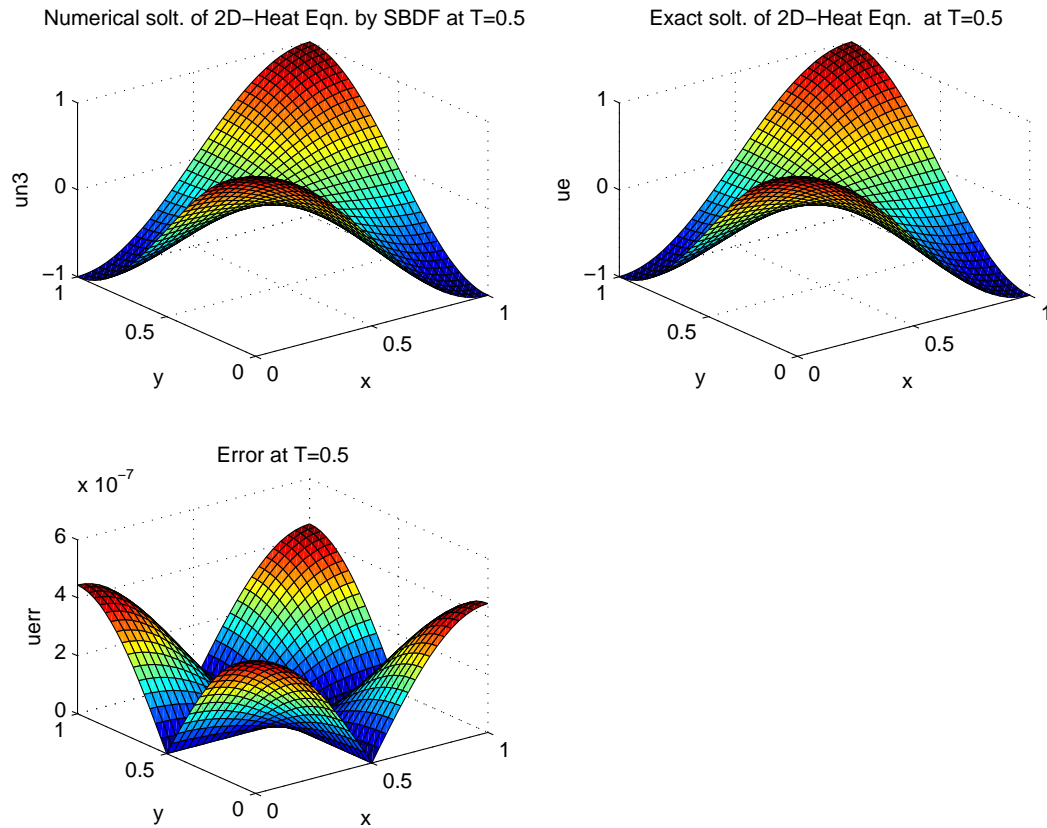


FIG. 5.3. Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $D_x = D_y = 0.01$  and  $dt = 0.0005$ .

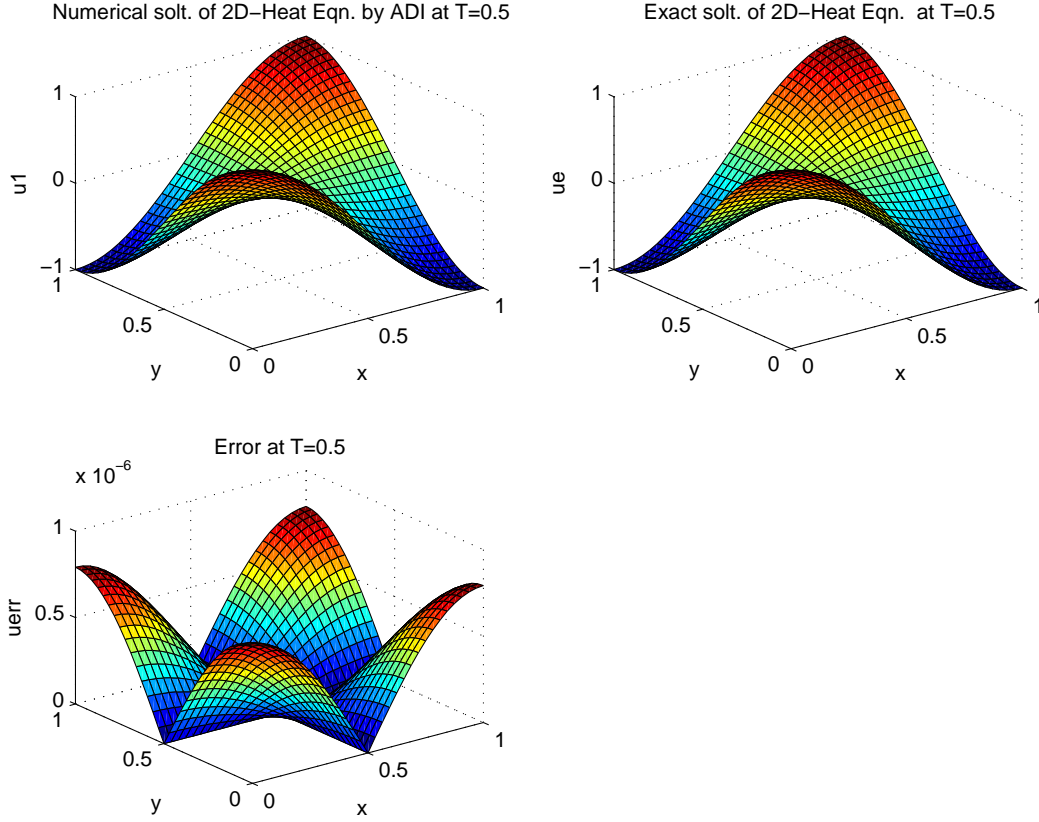


FIG. 5.4. Comparison of errors at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $D_x = D_y = 0.01$  and  $dt = 0.0005$ .

TABLE 5.7

Numerical results for the second problem with the Iterative Operator-Splitting method and SBDF3 with  $\Delta x = \frac{2}{5}$ .

$\epsilon$	$\Delta t$	$err_{L_\infty}$	$err_{L_1}$	CPU times
0.1	2/5	0.0542	0.0675	0.422000
	2/25	0.0039	0.0051	0.547000
	2/125	5.6742e-005	5.9181e-005	0.657000
0.001	2/5	7.6601e-004	8.6339e-004	0.610000
	2/25	4.3602e-005	5.4434e-005	0.625000
	2/125	8.0380e-006	1.0071e-005	0.703000

time discretization schemes accelerates the solving process with more or less than 2 or 3 iterative steps. Such stiff problems can be solved by an efficient combination of the iterative operator-splitting scheme and SBDF methods with more accuracy than the SBDF method without splitting.

**5.4. Fourth Experiment: Advection-diffusion equation.** We tackle the 2-dimensional advection-diffusion equation with periodic boundary conditions

$$(5.28) \quad \partial_t u = -\mathbf{v} \nabla u + D \Delta u,$$

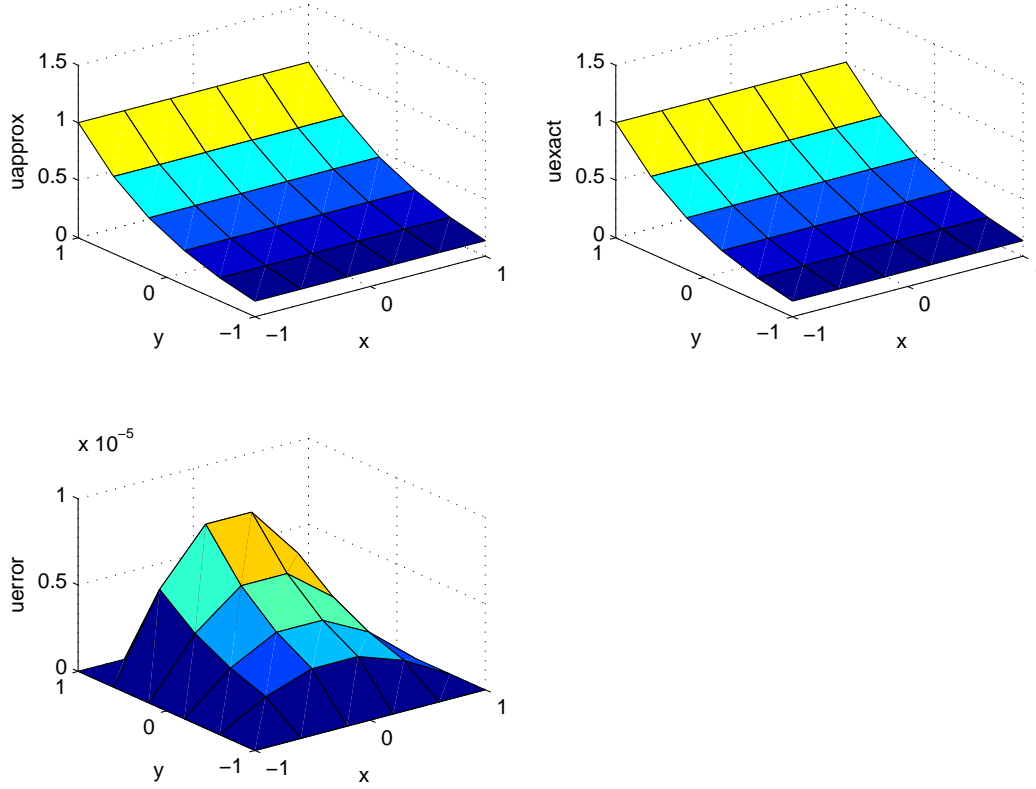


FIG. 5.5. Exact solution, numerical solution and error at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $\Delta x = \frac{2}{5}$  and  $\Delta t = \frac{2}{125}$

TABLE 5.8

Numerical results for the second problem with the Iterative Operator-Splitting method by the Trapezoidal rule, SBDF2 and SBDF3 with  $\Delta x = \frac{2}{5}$

	$\epsilon$	$\Delta t$	$err_{L_\infty}$	$err_{L_1}$	CPU times
Trapezoidal	1	2/25	0.9202	1.6876	0.141600
		2/125	0.8874	1.6156	0.298121
SBDF2	1	2/25	0.0473	0.0917	0.131938
		2/125	0.0504	0.0971	0.204764
SBDF3	1	2/25	0.0447	0.0878	0.149792
		2/125	0.0501	0.0964	0.537987

$$(5.29) \quad = -v_x \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + D \frac{\partial^2 u}{\partial y^2},$$

$$(5.30) \quad u(\mathbf{x}, \mathbf{t}_0) = u_0(\mathbf{x}),$$

with parameters

$$(5.31) \quad v_x = v_y = 1,$$

$$(5.32) \quad D = 0.01,$$

$$(5.33) \quad t_0 = 0.25.$$

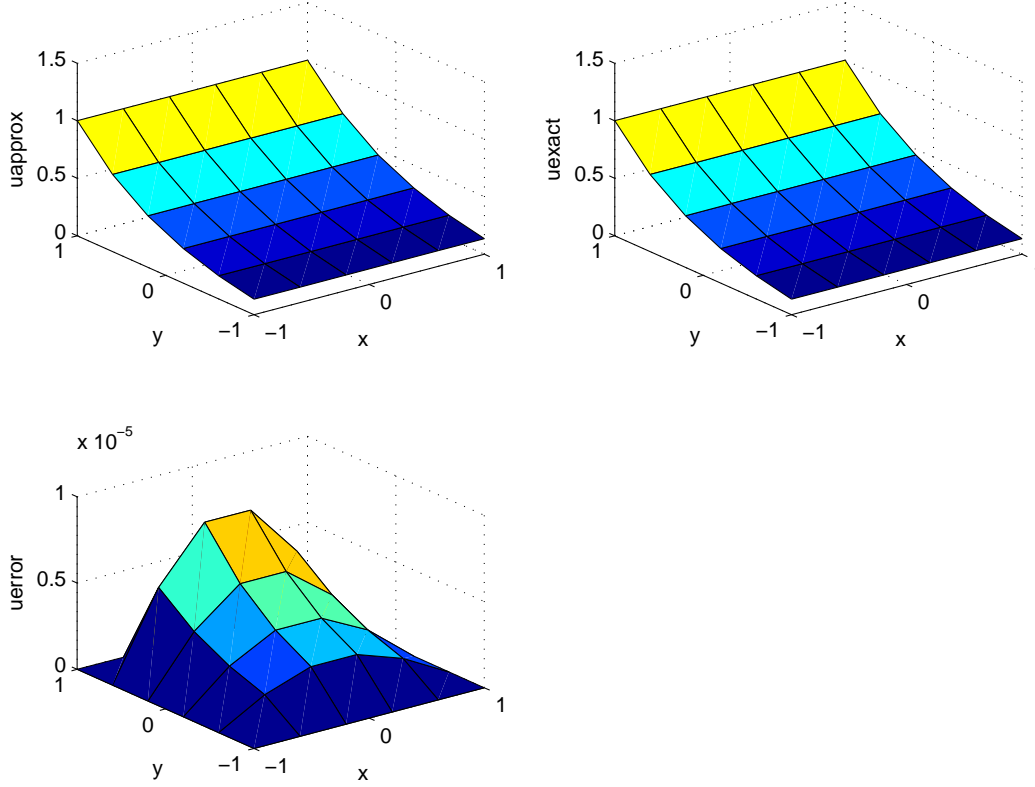


FIG. 5.6. Exact solution, numerical solution and error at  $T=0.5$  with various  $\Delta x$  and  $\Delta y$  for  $\Delta x = \frac{2}{5}$  and  $\Delta t = \frac{2}{125}$

The advection-diffusion problem has an analytical solution

$$(5.34) \quad u_a(\mathbf{x}, t) = \frac{1}{t} \exp\left(\frac{-(\mathbf{x} - \mathbf{v}t)^2}{4Dt}\right),$$

which we will use as a convenient initial function:

$$(5.35) \quad u(\mathbf{x}, t_0) = u_a(\mathbf{x}, t_0).$$

We apply dimensional splitting to our problem:

$$(5.36) \quad \frac{\partial u}{\partial t} = A_x u + A_y u,$$

where:

$$(5.37) \quad A_x = -v_x \frac{\partial u}{\partial x} + D \frac{\partial^2 u}{\partial x^2}$$

$$(5.38) \quad A_y = -v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial y^2}.$$

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

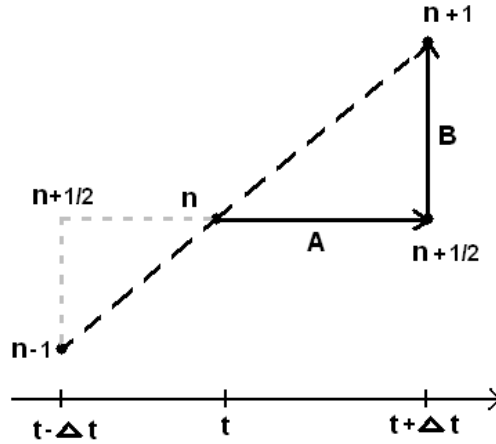


FIG. 5.7. Principle of AB-Splitting.

2nd order finite difference scheme

$$(5.39) \quad L_x u(x) = -v_x \frac{u(x) - u(x - \Delta x)}{\Delta x} + D_x \frac{u(x + \Delta x) + u(x) + u(x - \Delta x)}{\Delta x^2},$$

because the new diffusion constant eliminates the first order error (i.e. 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>1</sup>

$$(5.40) \quad L_t u(t) = \frac{1}{\Delta t} \left( \frac{137}{60} u(t + \Delta t) - 5u(t) + 5u(t - \Delta t) - \frac{10}{3} u(t - 2\Delta t) + \frac{5}{4} u(t - 3\Delta t) - \frac{1}{5} u(t - 4\Delta t) \right).$$

Our aim is to compare the iterative splitting method with AB splitting (Lie-Trotter splitting, see [6]). Since  $[A_x, A_y] = 0$ , there is no splitting error for 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 iterative splitting out-competes AB splitting in terms of computational effort and round-off errors. But first some remarks need to be made about the special behaviour of both methods when combined with high order Runge-Kutta and BDF methods.

**5.4.1. Splitting and Schemes of High Order in Time. Concerning AB-Splitting:** The principle of AB-splitting is well-known and simple. The equation

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

$\frac{du}{dt} = Au + Bu$  is broken down to:

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

which are connected via  $u^{n+1}(t) = u^{n+1/2}(t + \Delta t)$ . This is pointed out in figure (5.7). AB splitting works very well for any given one-step method like the Crank-Nicholson-Scheme. When the splitting error (which is an error in time) is not taken into account, it is also compatible with high order schemes such as explicit/implicit Runge-Kutta-schemes.

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

Choose for instance a BDF2 method which, in the 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 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 (5.7) 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 increases dramatically when we move to a multi-step method of higher order.

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

**5.4.2. Remarks about iterative splitting:** The BDF methods apply very well to 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 performed in one direction only <sup>2</sup>.

In particular, we make a subdivision of our existing time-discretization  $t_j = t_0 + j\Delta t$  into  $I$  parts. So we have sub-intervals  $t_{j,i} = t_j + i\Delta t/I$ ,  $0 \leq i \leq I$  on which we solve the following equations iteratively:

$$(5.41) \quad \frac{du^{i/I}}{dt} = Au^{i/I} + Bu^{(i-1)/I}$$

$$(5.42) \quad \frac{du^{(i+1)/I}}{dt} = Au^{i/I} + Bu^{(i+1)/I}$$

$u^{-1/I}$  is either 0 or a reasonable approximation<sup>3</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

<sup>2</sup>As we will see there is an exception to this.

<sup>3</sup>In fact the order of the approximation is not of much importance if we complete a sufficient number of iterations. In the case  $u^{-1/I} = 0$ , we have the exception that a step in the A-direction is done while B is left out. The error of this step certainly vanishes after a few iterations, but mostly after only one iteration



TABLE 5.9

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

TABLE 5.10

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

happen not to be the times at which a Runge-Kutta (RK) method needs to know them. Therefore, in the 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 can now summarize our results in table 5.9 which shows which methods are practicable for each kind of splitting scheme.<sup>4</sup>

**5.4.3. Numerical results.** After resolving the technical aspects of this issue, we can now proceed to the actual computations. A question which arises is which of the splitting methods requires the least computational effort since we can expect all of them to solve the problem with more or less the same accuracy if we use practicable methods with equal order, as  $[A_x, B_x] = 0$ . We have tested the dimensional splitting of the 2-dimensional advection-diffusion equation with AB splitting combined with a 5th order RK method after Dormand and Prince, and with iterative splitting in conjunction with a BDF5 scheme. We used  $40 \times 40$  and  $80 \times 80$  grids and completed  $n_t$  time steps each subdivided into 10 smaller steps until we reached a time  $t_{end} = 0.6$  which is sufficient to see the main effects. Iterative splitting was performed with 2 iterations which was already sufficient to attain the desired order. In tables 5.10 and 5.11, the errors at time  $t_{end}$  and the computation times are shown.

REMARK 5.4. *As we can see, the error in the iterative splitting scheme reaches the same value as the AB splitting error after a certain number of time steps and remains below it for all additional steps we accomplish. Of course, the error cannot drop below a certain value which is governed by the spatial discretization increments. It can be noted that, while the computation time used for iterative splitting is always about 20-40% less than that for AB splitting<sup>5</sup> the accuracy is, with a sufficient number of time steps, slightly better than that of AB splitting. This is due to the roundoff*

<sup>4</sup>Something in favour of the iterative splitting scheme is that it also takes into the account the fact that AB splitting may be used alongside the high order methods alluded to but cannot maintain the order if  $[A, B] \neq 0$ , while the iterative splitting scheme re-establishes the maximum order of the scheme after a sufficient number of iterations have been completed.

<sup>5</sup>The code for both methods is kept in the simplest possible form.

TABLE 5.11  
*Errors and computation times of AB splitting and iterative splitting for a 80x80 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

error which is higher for the Runge-Kutta method because of the greater amount of basic operations needed to compute RK steps.

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

**6. Conclusions and Discussions .** We have presented an iterative operator-splitting method as a competitive method to compute splittable differential equations. On the basis of an integral formulation of the iterative scheme, we have presented an error analysis and the local error for bounded operators. Numerical examples confirm the method's application to ordinary differential and partial differential equations. Here, an optimal balance of time, space and number of iteration steps is necessary and a one order reduction is obtained. In the future, we will focus on the development of improved operator-splitting methods for application to nonlinear differential equations.

**7. Appendix: Extension to unbounded operators and proof ideas.** 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. [10, 14].):

$$(7.1) \quad \frac{\partial c_i(t)}{\partial t} = Ac_i(t) + Bc_{i-1}(t), \text{ with } c_i(t^n) = c^n$$

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

$$(7.2) \quad \frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) + Bc_{i+1}(t),$$

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  $n$ .)

**7.1. Two unbounded Operators.** THEOREM 7.1. *Let us consider the abstract Cauchy problem in a Banach space  $\mathbf{X}$*

$$(7.3) \quad \begin{aligned} \partial_t c(x, t) &= Ac(x, t) + Bc(x, t), \quad 0 < t \leq T \text{ and } x \in \Omega, \\ c(x, 0) &= c_0(x), \quad x \in \Omega, \\ c(x, t) &= c_1(x, t), \quad x \in \partial\Omega \times [0, T], \end{aligned}$$

where  $A, B : D(\mathbf{X}) \rightarrow \mathbf{X}$  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  $\text{dom}(A) = \text{dom}(B)$ .

Further, we assume the following bounds:

$$\begin{aligned} (7.4) \quad & \|B^\alpha \exp(B\tau_n)\| \leq \kappa \tau_n^{-\alpha}. \\ (7.5) \quad & \|B^\alpha \exp((A+B)\tau_n)\| \leq \kappa \tau_n^{-\alpha}, \\ (7.6) \quad & \|\exp(A\tau_n)B^{1-\alpha}\| \leq \tilde{\kappa} \tau_n^{p(1-\alpha)}, \\ (7.7) \quad & \|A^\beta \exp(A\tau_n)\| \leq \kappa \tau_n^{-\beta}. \\ (7.8) \quad & \|A^\beta \exp((A+B)\tau_n)\| \leq \kappa \tau_n^{-\beta}, \\ (7.9) \quad & \|\exp(B\tau_n)A^{1-\beta}\| \leq \tilde{\kappa} \tau_n^{q(1-\beta)}, \end{aligned}$$

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  $n = 1, \dots, N$ . Then the iteration process (7.1)–(7.2) for  $i = 1, 3, \dots, 2m + 1$  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 (7.1)–(7.2) on the sub-interval  $[t^n, t^{n+1}]$ .

For the first iterations we have:

$$(7.10) \quad \partial_t c_1(t) = A c_1(t), \quad t \in (t^n, t^{n+1}],$$

and for the second iteration we have:

$$(7.11) \quad \partial_t c_2(t) = A c_1(t) + B c_2(t), \quad t \in (t^n, t^{n+1}],$$

In general we have:

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

$$(7.12) \quad \partial_t c_i(t) = A c_i(t) + B c_{i-1}(t), \quad t \in (t^n, t^{n+1}],$$

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

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

$$(7.13) \quad \partial_t c_i(t) = A c_{i-1}(t) + B c_i(t), \quad t \in (t^n, t^{n+1}],$$

We have the following solutions for the iterative scheme:

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

$$(7.14) \quad c_1(t) = \exp(A(t - t^n))c(t^n), \quad t \in (t^n, t^{n+1}],$$

$$(7.15) \quad \begin{aligned} c_2(t) &= \exp(B(t - t^n))c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A c_1(s) ds, \quad t \in (t^n, t^{n+1}]. \end{aligned}$$

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

$$(7.16) \quad \partial_t c_i(t) = \exp(A(t - t^n))c(t^n) + \int_{t^n}^t \exp((t - s)A)B c_{i-1}(s) ds, \quad t \in (t^n, t^{n+1}],$$

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

$$(7.17) \quad \phi_i(t) = \exp(B(t - t^n))c(t^n) + \int_{t^n}^t \exp((t - s)B)Ac_{i-1}(s) ds, \quad t \in (t^n, t^{n+1}],$$

**The consistency is given as:**

For  $e_1$  we have:

$$(7.18) \quad c_1(t^{n+1}) = \exp(A\tau_n)c(t^n),$$

$$(7.19) \quad \begin{aligned} c(t^{n+1}) &= \exp((A + B)\tau_n)c(t^n) = \exp(A\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \exp((s - t^n)(A + B))c(t^n) ds. \end{aligned}$$

We obtain:

$$(7.20) \quad \begin{aligned} \|e_1\| &= \|c - c_1\| \leq \|\exp((A + B)\tau_n)c(t^n) - \exp(A\tau_n)c(t^n)\| \\ &\leq \left\| \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B \exp((s - t^n)(A + B))c(t^n) ds \right\| \\ &\leq \left\| \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1} - s))B^{1-\alpha}B^\alpha \exp((s - t^n)(A + B))c(t^n) ds \right\| \\ &\leq \int_{t^n}^{t^{n+1}} \|\exp(A(t^{n+1} - s))B^{1-\alpha}\| \|B^\alpha \exp((s - t^n)(A + B))\| ds \|c(t^n)\| \\ &\leq \int_{t^n}^{t^{n+1}} \frac{1}{(t^{n+1} - s)^{p(1-\alpha)}} \frac{\kappa}{(s - t^n)^\alpha} ds \|c(t^n)\| \\ &\leq \int_{t^n}^{t^{n+1/2}} \left( \frac{\kappa}{(s - t^n)^\alpha} + \frac{C}{\tau^{p(1-\alpha)}} \right) ds \\ &+ \int_{t^{n+1/2}}^{t^{n+1}} \left( \frac{C}{\tau^\alpha} + \frac{C}{(t^{n+1} - s)^{p(1-\alpha)}} \right) ds \\ &\leq C(\tau^{1-\alpha} + \tau^{p\alpha} + \tau^\alpha + \tau^{p\alpha}) \\ (7.21) \quad &\leq C\tau^{\min((1-\alpha), p\alpha)} \|c(t^n)\| \end{aligned}$$

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

See assumption to the interval see Figure 7.1:

For  $e_2$  we have:

$$(7.22) \quad \begin{aligned} c_2(t^{n+1}) &= \exp(B\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A \exp((s - t^n)A)c(t^n) ds, \end{aligned}$$

$$(7.23) \quad \begin{aligned} c(t^{n+1}) &= \exp(B\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A \exp((s - t^n)A)c(t^n) ds \\ &+ \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A \\ &\quad \int_{t^n}^s \exp(A(s - \rho))B \exp((\rho - t^n)(A + B))c(t^n) d\rho ds. \end{aligned}$$

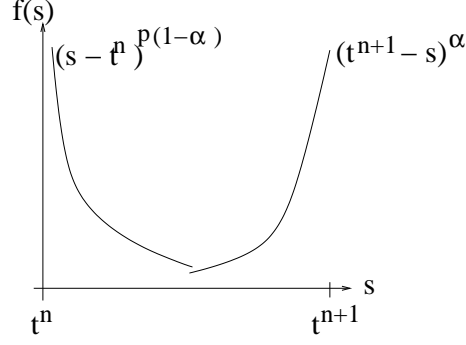


FIG. 7.1. Function of the estimations.

We obtain:

$$(7.24) \quad \|e_2\| \leq \|\exp((A+B)\tau_n)c(t^n) - c_2\|$$

$$(7.25) \quad = \left\| \int_{t^n}^{t^{n+1}} \exp(B(t^{n+1} - s))A \int_{t^n}^s \exp(A(s - \rho))B \exp((\rho - t^n)(A+B))c(t^n) d\rho ds \right\|$$

$$(7.26) \quad = \int_{t^n}^{t^{n+1}} \|\exp(B(t^{n+1} - s))A^{1-\alpha}\|$$

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

$$(7.27) \quad = \int_{t^n}^{t^{n+1/2}} \left( \frac{\kappa_1}{(t^{n+1} - s)^{p_1(1-\alpha_1)}} + \frac{\kappa_2}{(t^{n+1} - s)^{p_2(1-\alpha_2)}} + \frac{C_1}{\tau^{1-\alpha_2}} \right) ds$$

$$(7.28) \quad + \int_{t^{n+1/2}}^{t^{n+1}} \left( \frac{C_2}{\tau^{1-\alpha_1}} + \frac{C_1}{\tau^{1-\alpha_2}} + \frac{\kappa_2}{(s - t^n)^{\alpha_2}} \right) ds$$

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

For odd and even iterations, the recursive proof is given in the following. In the next steps, we shift  $t^n \rightarrow 0$  and  $t^{n+1} \rightarrow \tau_n$  for simpler calculations, see [13]. The initial conditions are given with  $c(0) = c(t^n)$ .

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

$$(7.29) \quad c_i(\tau_n) = \exp(A\tau_n)c(0)$$

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

$$+ \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2B)A \exp((\tau_n - s_1 - s_2)A)c(0) ds_2 ds_1$$

$$+ \dots +$$

$$+ \int_0^{\tau_n} \exp(As_1)B \int_0^{\tau_n - s_1} \exp(s_2A)B \int_0^{\tau_n - s_1 - s_2} \exp(s_3A)B \dots$$

$$\int_0^{\tau_n - \sum_{j=1}^{i-1} s_j} \exp(As_i)B \exp((\tau_n - \sum_{j=1}^{i-1} s_j)A)c(0) ds_i \dots ds_1,$$

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

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

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

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

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

□

REMARK 7.1. 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

$$\begin{aligned}
(7.32) \quad \|e_1\| &= \tilde{C}\tau_n^{\min(1 - \alpha_1, p_1 \alpha_1)} \|c(t^n)\| + \mathcal{O}(\tau_n^{1 + \alpha_1}) \\
&\text{and hence} \\
\|e_2\| &= \tilde{C}\|e_0\| \tau_n^{\min(1 - \alpha_1, p_1 \alpha_1) + \min(1 - \beta_1, q_1 \beta_1)} \\
(7.33) \quad &+ \mathcal{O}(\tau_n^{1 + \min(1 - \alpha_1, p_1 \alpha_1) + \min(1 - \beta_1, q_1 \beta_1)}),
\end{aligned}$$

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

REMARK 7.2. 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  $\mathcal{O}(\tau_n^{m-1})$ .

**7.2. Exponential Runge-Kutta Method.** The computations of the methods can be done as:

$$(7.34) \quad c_1(t^{n+1}) = \exp(A\tau)c(t^n)$$

$$(7.35) \quad c_2(t^{n+1}) = (\phi_0(B\tau) + \phi_1(B\tau)A^\alpha \phi_0(A\tau))c(t^n),$$

where we assume  $\|A^\alpha \phi_0(A\tau)\| \leq \tau^{-\alpha}C$  and  $\phi_0(z) = \exp(z)$ ,  $\phi_1(z) = \frac{\phi_0(z) - 1}{z}$ .

**7.3. Unbounded / Bounded Scheme.** We propose 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  $i = 0, 1, \dots, m$ . (cf. [10, 14].):

$$(7.36) \quad \frac{\partial c_{i+1}(t)}{\partial t} = Ac_{i+1}(t) + Bc_i(t), \text{ with } c_i(t^n) = c^n$$

where  $c^n$  is the known split approximation at the time-level  $t = t^n$  and  $c_0(t) = 0$ .

We have the following assumptions:

ASSUMPTION 7.1.

Let:  $A : D(A) \rightarrow \mathbf{X}$  be sectorial, i.e.  $A$  is densely defined and closed linear operator on  $\mathbf{X}$  satisfying the resolvent condition:

$$(7.37) \quad \|(\lambda I - A)^{-1}\|_{\mathbf{X} \rightarrow \mathbf{X}} \leq \frac{M}{|\lambda - a|}$$

on the sector  $\{\lambda \in \mathbf{C} : \theta \leq |\arg(\lambda - a)| \leq \pi, \lambda \neq a\}$  for  $M \geq 1$ ,  $a \in \mathbb{R}$ , and  $0 < \theta \leq \pi/2$ .

Therefore  $A$  is the infinitesimal generator of an analytical semigroup  $\{\exp(tA)\}_{t \geq 0}$ . For  $\omega > -a$ , the fractional power of  $\tilde{A} = A + \omega I$  are well defined.

Further, we assume the following bounds for operator  $A$ :

$$(7.38) \quad \|\tau_n^\alpha A^\alpha \exp(A\tau_n)\| \leq \kappa,$$

$$(7.39) \quad \|hA \sum_{j=1}^{n-1} \exp(jhA)\| \leq \kappa,$$

where  $\alpha \geq 0$  and  $\tau_n = (t^{n+1} - t^n)$ .

The stability estimates allow to define the bounded operators:

$$(7.40) \quad \phi_j(tA) = \frac{1}{t^j} \int_0^t \exp((t - \tau)A) \frac{\tau^{j-1}}{(j-1)!} d\tau, \quad j \geq 1.$$

Further  $\phi_0(z) = \exp(z)$  and

$$(7.41) \quad \phi_{k+1}(z) = \frac{\phi_k(z) - 1/k!}{z}, \quad \phi_k(0) = \frac{1}{k!}, \quad k \geq 0.$$

**THEOREM 7.2.** Let us consider the abstract Cauchy problem in a Banach space  $\mathbf{X}$

$$(7.42) \quad \begin{aligned} \partial_t c(t) &= Ac(t) + Bc(t), \quad 0 < t \leq T, \\ c(0) &= c_0, \end{aligned}$$

where  $A : D(\mathbf{X}) \rightarrow \mathbf{X}$  are given linear operators which are generators of the  $C_0$ -semigroup and  $c_0 \in \mathbf{X}$  is a given element.  $B$  is a linear bounded operator in  $L^2$ .

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  $n = 1, \dots, N$ . Then the iteration process (7.36) for  $i = 1, 2, \dots, m$  is consistent with the order of the consistency  $\mathcal{O}(\tau_n^m)$ , where  $0 \leq \alpha < 1$ .

*Proof.* While  $B$  is bounded the proof can be done as:

For  $e_1$  we have:

$$(7.43) \quad c_1(t^{n+1}) = \exp(A\tau_n)c(t^n),$$

$$(7.44) \quad \begin{aligned} c(t^{n+1}) &= \exp((A+B)\tau_n)c(t^n) = \exp(A\tau_n)c(t^n) \\ &+ \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp((s-t^n)(A+B))c(t^n) ds. \end{aligned}$$

We obtain:

$$(7.45) \quad \begin{aligned} \|e_1\| &= \|c - c_1\| \leq \|\exp((A+B)\tau_n)c(t^n) - \exp(A\tau_n)c(t^n)\| \\ &\leq \left\| \int_{t^n}^{t^{n+1}} \exp(A(t^{n+1}-s))B \exp((s-t^n)(A+B))c(t^n) ds \right\| \\ &\leq \int_{t^n}^{t^{n+1}} \|\exp(A(t^{n+1}-s))\| \|B\| \|\exp((s-t^n)(A+B))\| ds \|c(t^n)\| \\ (7.46) \quad &\leq \tau \|B\| \|c(t^n)\|. \end{aligned}$$

Same idea can be done for the next iterations.

□

**7.4. Application to a numerical scheme.** The exponential Euler method is given as:

$$(7.47) \quad c_{i+1}(t^{n+1}) = \exp(A\tau)c(t^n) + \tau\phi_1(A\tau_n)Bc_i(t^n),$$

$$(7.48) \quad c_{i+1}(t^{n+1}) = \exp(A\tau)c(t^n) + \tau \frac{\exp(A\tau_n) - I}{A} Bc_i(t^n).$$

Convergence of the exponential Euler method:

**THEOREM 7.3.** *Let us consider the iterative scheme in a Banach space  $\mathbf{X}$*

$$(7.49) \quad \begin{aligned} \partial_t c_{i+1}(t) &= Ac_{i+1}(t) + Bc_i(t), \quad 0 < t \leq T, \\ c(0) &= c_0, \end{aligned}$$

where  $A : D(\mathbf{X}) \rightarrow \mathbf{X}$  are given linear operators which are generators of the  $C_0$ -semigroup.  $B$  is a linear bounded operator in  $L^2$  and  $c_0(t) = 0$ .

The application to exponential Euler method obtain the error  $O(\tau)$  for the first iterative step.

*Proof.*

By the variation of constants formula we obtain:

$$(7.50) \quad c_{i+1}(t^n + \theta h) = \exp(Ah\theta)c(t^n) + \int_0^{\theta h} \exp((\theta h - \tau)A)Bc_i(t_n + \tau) d\tau.$$

We expand in Taylor series but respect the bound of the operator  $A$ :

$$(7.51) \quad c_i(t^n + \tau) = c_i(t^n) + \int_0^\tau c'_i(t_n + \tau) d\tau,$$

We have the assumption 7.1 the  $c_i(t^n + \tau)$  is bounded and we have:

$$(7.52) \quad \begin{aligned} c_{i+1}(t^n + \theta h) &= \exp(Ah\theta)c(t^n) + \theta h\phi_1(\theta hA)Bc_i(t_n) \\ &+ r(c_i(t^n + \theta h)), \end{aligned}$$



and

$$r(c_i(t^n + \theta h)) = \int_0^{\theta h} \exp((\theta h - \tau)A) \int_0^\tau c'_i(t_n + \sigma) d\sigma d\tau.$$

So the error is given as:

$$\begin{aligned} (7.53) \quad & \|r(c_i(t^n + \theta h))\| \\ & \leq \left\| \int_0^{\theta h} \exp((\theta h - \tau)A) \int_0^\tau A \exp(A(t^n + \sigma)) c_i(t^n) d\sigma d\tau \right\|, \\ & \leq \int_0^{\theta h} \|\exp((\theta h - \tau)A)\| \int_0^\tau \|A \exp(A(t^n + \sigma))\| d\sigma d\tau \|c_i(t^n)\|, \\ & \leq \tau \kappa. \end{aligned}$$

□

#### REFERENCES

- [1] M. Bjorhus. *Operator splitting for abstract Cauchy problems*. IMA Journal of Numerical Analysis, 18, 419-443, 1998.
- [2] B. Davis. *Integral Transform and Their Applications*. Applied Mathematical Sciences, 25, Springer Verlag, New York, Heidelberg, Berlin, 1978 .
- [3] K.-J. Engel, R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*. Springer-Verlag, Heidelberg, New York, 2000.
- [4] I. Farago. *Splitting methods for abstract Cauchy problems*. Lect. Notes Comp.Sci. 3401, Springer Verlag, Berlin, 35-45, 2005.
- [5] I. Farago. *A modified Iterated operator-splitting method*. Applied Mathematical Modelling, Volume 32, Issue 8, 1542-1551, 2008.
- [6] I. Farago and J. Geiser. *Iterative Operator-Splitting Methods for Linear Problems*. Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, (2005) 1-18. International Journal of Computational Science and Engineering, accepted September 2007.
- [7] J. Geiser. *Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations*. Journal of Computational and Applied Mathematics, Elsevier, Amsterdam, The Netherlands, 217, 227-242, 2008.
- [8] 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.
- [9] J. Geiser. *Decomposition Methods for Differential Equations: Theory and Applications*. Chapman & Hall/CRC, Series: Numerical Analysis and Scientific Computing Series, edited by Magoules and Lai, 2009.
- [10] R. Glowinski. *Numerical methods for fluids*. Handbook of Numerical Analysis, Gen. eds. P.G. Ciarlet, J. Lions, Vol. IX, North-Holland Elsevier, Amsterdam, The Netherlands, 2003.
- [11] E. Hansen and A. Ostermann. *Exponential splitting for unbounded operators*. Mathematics of Computation, 78, 1485-1496, 2009.
- [12] E. Hille and R.G. Phillips *Functional Analysis and Semi-groups*. American Mathematical Society Colloquium Publications, Vol.XXXXI, revised edn. Providence, American Mathematical Society, 1957.
- [13] T. Jahnke and C. Lubich. *Error bounds for exponential operator splittings*. BIT Numerical Mathematics, 40:4, 735-745, 2000.
- [14] 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.
- [15] T. Kato. *Perturbation Theory for Linear Operators*. Springer, Berlin, Heidelberg, New-York, second edition, 1980.
- [16] C.T. Kelly. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 1995.
- [17] G.I Marchuk. *Some applications of splitting-up methods to the solution of problems in mathematical physics*. Aplikace Matematiky, 1, 103-132, 1968.

- [18] I. Najfeld and T.F. Havel. *Derivatives of the matrix exponential and their computation*. Adv. Appl. Math, <ftp://ftp.das.harvard.edu/pub/cheatham/tr-33-94.ps.gz>, 1995.
- [19] A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*. Applied Mathematical Sciences 44, Springer, Berlin, 1983.
- [20] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations* Series: Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1999.
- [21] G. Strang. *On the construction and comparison of difference schemes*. SIAM J. Numer. Anal., 5, 506-517, 1968.
- [22] S. Vandewalle. *Parallel Multigrid Waveform Relaxation for Parabolic Problems*. Teubner Skripten zur Numerik, B.G. Teubner Stuttgart, 1993.
- [23] K. Yosida. *Functional Analysis*. Classics in Mathematics, Springer-Verlag, Berlin-Heidelberg-New York, 1980.