# Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations


Jürgen Geiser[1] and Joscha Gedicke[2]

Humboldt Univeristät zu Berlin, D- Berlin, Germany
[1]geiser@mathematik.hu-berlin.de
[2]gedicke@mathematik.hu-berlin.de



**Abstract.** In this paper we design higher order time integrators for systems of stiff ordinary differential equations. We could combine implicit Runge-Kutta- and BDF-methods with iterative operator-splitting methods to obtain higher order methods. The motivation of decoupling each complicate operator in simpler operators with an adapted time-scale allow us to solve more efficiently our problems. We compare our new methods with the higher order Fractional-Stepping Runge-Kutta methods, developed for stiff ordinary differential equations. The benefit will be the individual handling of each operators with adapted standard higher order time-integrators. The methods are applied to convection-diffusion-reaction equations and we could obtain higher order results. Finally we discuss the iterative operator-splitting methods for the applications to multi-physical problems.



**Keywords.** Operator Splitting method, Iterative Solver methods, Runge-Kutta methods, Fractional-Stepping Runge-Kutta methods, Convection-Diffusion-Reaction-equation.


## 1 Introduction

We motivate our studying on combining explicit and implicit time-discretization methods with iterative Operator-Splitting methods as efficient discretization- and solver-methods.

The iterative operator-splitting methods have their main advantage in combining iterative and splitting behavior. On the one hand the iterative behavior allows higher order results and on the other hand the splitting behavior allows to decompose in simpler operator. For these simpler operator-equation one could use standard implicit and explicit Runge-Kutta or BDF-method for solving the stiff and non-stiff parts. The stability analysis is discussed for the commutative and non-commutative case of operators. Based on these contributions we compare our proposed decoupling method with the explicit-implicit-methods

(IMEX), that are used without splitting. We present that the results of our proposed iterative operator-splitting method is more efficient and accurate as the use of complicate combined time-stepping methods.

## 2 Mathematical Model

Our model equations are coming from a computational simulation of bio-remediation [2] or radioactive contaminants [9], [8].

The mathematical equations are given by

$$\partial_t \, R \, c + \nabla \cdot (\mathbf{v}c - D \nabla c) = f(c) \; , \tag{1}$$

$$f(c) = c^p \; , \text{ chemical-reaction and } p > 0 \tag{2}$$

$$f(c) = \frac{c}{1-c} \; , \text{ bio-remediation} \tag{3}$$

The unknown $c = c(x,t)$ is considered in $\Omega \times (0,T) \subset \mathbb{R}^d \times \mathbb{R}$, the space-dimension is given by $d$ . The Parameter $R \in \mathbb{R}^+$ is constant and is named as retardation factor. The other parameters $f(c)$ are nonlinear functions, for example bio-remediation or chemical reaction. $D$ is the Scheidegger diffusion-dispersion tensor and $\mathbf{v}$ is the velocity.

The aim of this paper is to present a new iterative method based on operator-splitting methods for partial differential equations. In a first paper, we focus on ordinary differential equations and discuss the theory and application for a weighted method.

## 3 Iterative splitting method

The following algorithm is based on the iteration with fixed splitting discretization step-size $\tau$. On the time interval $[t^n, t^{n+1}]$ we solve the following subproblems consecutively for $i = 1, 3, \dots, 2m+1$. (cf. [15] and [7].)

$$\frac{\partial c_i(t)}{\partial t} = A c_i(t) \; + \; B c_{i-1}(t), \text{ with } \; c_i(t^n) = c^n \tag{4}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = A c_i(t) \; + \; B c_{i+1}(t), \text{ with } \; c_{i+1}(t^n) = c^n \; , \tag{5}$$

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

In the following we will analyze the convergence and the rate of the convergence of the method (4)–(5) for $m$ tends to infinity for the linear operators $A, B : \mathbf{X} \to \mathbf{X}$ where we assume that these operators and their sum are generators of the $C_0$ semigroups. We emphasize that these operators aren't necessarily bounded, so, the convergence is examined in general Banach space setting.

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

$$\begin{aligned} \partial_t c(t) &= Ac(t) + Bc(t), \quad 0 < t \leq T \\ c(0) &= c_0 \end{aligned} \tag{6}$$

*where* $A, B, A + B : \mathbf{X} \to \mathbf{X}$ *are given linear operators being generators of the* $C_0$*-semigroup and* $c_0 \in \mathbf{X}$ *is a given element. Then the iteration process (4)–(5) is convergent and the rate of the convergence is of higher order.*

The proof could be found in [12].

*Remark 1.* When $A$ and $B$ are matrices (i.e. (4)–(5) is a system of ordinary differential equations), for the growth estimation we can use the concept of the logarithmic norm. (See e.g.[14].) Hence, for many important classes of matrices we can prove the validity.

*Remark 2.* We note that a huge class of important differential operators generate contractive semigroup. This means that for such problems -assuming the exact solvability of the split sub-problems- the iterative splitting method is convergent in higher order to the exact solution.

## 4 Stability Theory for the iterative splitting method with analytical initialization

We consider in the following the linear problem :

$$\partial_t c(t) = Ac(t) \; + \; Bc(t) \; , \tag{7}$$

whereby the initial-conditions are $c^n = c(t^n)$. The operators $A$ and $B$ are spatially discretised operators, e.g. they correspond to the discretised in space convection and diffusion operators (matrices). Hence, they can be considered as bounded operators.

We distinguish in the following the 2 cases : commutative and non-commutative operators.

### 4.1 Commutative part with continuous equation

In the following we discuss the improved and stable iterative method. One could stabilize the methods by using initial conditions, that are approximations of the solutions. One could show that the method alone is not stable enough, see [13]. In that case the improved stability theory is presented as follows.

**Theorem 2.** *Let us consider the iterative method, that starts with the initial-value* $c_1$*, done from a A-B splitting method or Strang-Splitting method.*

*Then we could proof, that holds*

$$c_{i+3}(z, -\infty) = 0 \leq 1 \; , i = 1, 3, \ldots \; , \tag{8}$$

*where* $c_1 \in U$*.*

*Proof.* First the stability of the A-B splitting :

We have the equations :

$$\frac{\partial c_0}{\partial t} = A c_0 \ , \ c_0(0) = c_n \ , \frac{\partial c_1}{\partial t} = B c_1 \ , \ c_1(0) = c_0(\tau) \ , \tag{9}$$

We insert the operators : $A = \lambda_1$ and $B = \lambda_2$

We could derive the analytical solution and get the solution :

$$c_1(t) = \exp((\lambda_1 + \lambda_2) t) \ c_n \ , \tag{10}$$

We have further the stability, we denote $z_1 = \lambda_1 \tau$ and $z_2 = \lambda_2 \tau$

$$c_1(z_1, z_2) = \exp(z_1 + z_2) \ c_n \ , \tag{11}$$

For the stiff-case : $z_2 \to -\infty$

We have :

$$\lim_{z_2 \to -\infty} c_1(z_1, z_2) = 0 \ , \tag{12}$$

and therefore we have the stability :

$$||c_1(z_1, -\infty)|| \leq 1 \ , \tag{13}$$

is fulfilled.

The value $c_1$ is a start-value of the iterative method.

We have for the iterative method the following stability :

$$\frac{\partial c_{i+1}}{\partial t} = A c_{i+1} + B c_i \ , \ c_{i+1}(0) = c_n \ , \tag{14}$$

$$\frac{\partial c_{i+2}}{\partial t} = A c_{i+1} + B c_{i+2} \ , \ c_{i+2}(0) = c_n \ , \tag{15}$$

We insert the operators : $A = \lambda_1$ and $B = \lambda_2$

We could derive the analytical solution for $c_{i+1}$ and get the solution :

$$c_{i+1}(t) = \exp(\lambda_1 (t - t_n)) \ (\int_{t_n}^t \exp(-\lambda_1 (s - t_n)) \lambda_2 c_i \ ds + c_n) \ , \tag{16}$$

$$c_{i+2}(t) = \exp(\lambda_2 (t - t_n)) \ (\int_{t_n}^t \exp(-\lambda_2 (s - t_n)) \lambda_1 c_{i+1} \ ds + c_n) \ , \tag{17}$$

If we compute the $c_2$ by inserting $c_1$ we get :

$$\begin{aligned} c_2(t) = \exp(\lambda_1 (t - t_n)) \ (\int_{t_n}^t \exp(-\lambda_1 (s - t_n)) \lambda_2 \qquad & (18) \\ \exp((\lambda_1 + \lambda_2)(s - t_n)) c_n \ ds + c_n) \ , & \\ c_2(t) = \exp((\lambda_1 + \lambda_2) t) c_n \ , \qquad & (19) \end{aligned}$$

We insert the result for the next iteration to compute $c_3$

$$c_3(t) = \exp(\lambda_2(t-t_n)) \; (\int_{t_n}^{t} \exp(-\lambda_2(s-t_n))\lambda_1 \qquad (20)$$

$$\exp((\lambda_1+\lambda_2)(s-t_n))c_n \; ds + c_n) \; ,$$

$$c_3(t) = \exp((\lambda_1+\lambda_2)t)c_n \; , \qquad (21)$$

The stability result for the $c_3$ is also given as :

$$c_3(z_1, z_2) = \exp(z_1 + z_2) \; c_n \; , \qquad (22)$$

For the stiff-case : $z_2 \rightarrow -\infty$
We have :

$$\lim_{z_2 \rightarrow -\infty} c_3(z_1, z_2) = 0 \; , \qquad (23)$$

and therefore we have the stability :

$$||c_3(z_1, -\infty)|| \leq 1 \; , \qquad (24)$$

is fulfilled.
And also for arbitrary iteration-steps :

$$||c_{i+2}(z_1, \infty)|| \leq 1 \; , \; i = 1, 3, \ldots \qquad (25)$$

This shows that for arbitrary $i = 1, 3, \ldots$ the iterative method is stable.

*Remark 3.* The iterative operator-splitting method is invariant to the analytical solution and therefore stable. So it is enough to guaranty a prestepping method, that shift the solution into the exact solution space.

### 4.2 Non-commutative part with continuous equation

In the following we discuss the improved and stable iterative method.
We discuss the noncommutative part for the operators.

**Theorem 3.** *Let us consider the iterative method, that starts with the initial-value $c_1$, that is of n-th order exact.*
*Then we could proof, that holds*

$$c_{i+3}(z, -\infty) = 0 \leq 1 \; , i = 1, 3, \ldots, \qquad (26)$$

*where $c_1 \in U$.*

*Proof.* We have the stability of an analytical solution, that is exact or have at least order $n$ and get the solution :

$$c_1(t) = \exp((\lambda_1 + \lambda_2)t) \; c_n \; , \qquad (27)$$

We have further the stability, we denote $z_1 = \lambda_1 \tau$ and $z_2 = \lambda_2 \tau$

$$c_1(z_1, z_2) = \exp(z_1 + z_2)\ c_n\ , \tag{28}$$

For the stiff-case : $z_2 \rightarrow -\infty$
We have :

$$\lim_{z_2 \rightarrow -\infty} c_1(z_1, z_2) = 0\ , \tag{29}$$

and therefore we have the stability :

$$||c_1(z_1, -\infty)|| \leq 1\ , \tag{30}$$

is fulfilled.
The value $c_1$ is a start-value of the iterative method.
We have for the iterative method the following stability :

$$\frac{\partial c_{i+1}}{\partial t} = A c_{i+1} + B c_i\ ,\ c_{i+1}(0) = c_n\ , \tag{31}$$

$$\frac{\partial c_{i+2}}{\partial t} = A c_{i+1} + B c_{i+2}\ ,\ c_{i+2}(0) = c_n\ , \tag{32}$$

We insert the operators : $A = \lambda_1$ and $B = \lambda_2$
We could derive the analytical solution for $c_{i+1}$ and get the solution :

$$c_{i+1}(t) = \exp(\lambda_1(t - t_n))\ (\int_{t_n}^{t} \exp(-\lambda_1(s - t_n))\lambda_2 c_i\ ds + c_n)\ , \tag{33}$$

$$c_{i+2}(t) = \exp(\lambda_2(t - t_n))\ (\int_{t_n}^{t} \exp(-\lambda_2(s - t_n))\lambda_1 c_{i+1}\ ds + c_n)\ , \tag{34}$$

If we compute the $c_2$ by inserting $c_1$ we get :

$$c_2(t) = \exp(\lambda_1(t - t_n))\ (\int_{t_n}^{t} \exp(-\lambda_1(s - t_n))\lambda_2 \tag{35}$$

$$\exp((\lambda_1 + \lambda_2)(s - t_n))c_n\ ds + c_n)\ , \tag{36}$$

Now we compute the noncommutative case let us do it till the order 2 and we get :

$$\begin{aligned} c_2(t) &= (1 + \lambda_1\tau + \lambda_1^2\tau^2/2! + O(\tau^3)) && (37)\\ &\quad (1 + \lambda_2\tau - \lambda_1\lambda_2\tau^2/2! + \lambda_2\lambda_1\tau^2/2! + \lambda_2^2\tau^2/2! + O(\tau^3)) \\ &= (1 + \lambda_1\tau + \lambda_2\tau + \lambda_1^2\tau^2/2! + \lambda_1\lambda_2\tau^2/2! + \lambda_2\lambda_1\tau^2/2! \\ &\quad + \lambda_2^2\tau^2/2! + O(\tau^3)) \\ &\approx \exp((\lambda_1 + \lambda_2)(t - t^n)) && (38) \end{aligned}$$

with $\tau = t - t^n$.
The stability result for the $c_2$ is also given as :

$$c_2(z_1, z_2) = \exp(z_1 + z_2)\ c_n\ , \tag{39}$$

For the stiff-case : $z_2 \to -\infty$
We have :

$$\lim_{z_2 \to -\infty} c_2(z_1, z_2) = 0 \ , \tag{40}$$

and therefore we have the stability.

*Remark 4.* The iterative operator-splitting method is invariant to the analytical solution and therefore stable. So it is enough to guaranty a prestepping method exists, that could have a high order of accuracy.

In the next subsection we present the used time-discretization methods.

# 5 Runge-Kutta, BDF-method and IMEX-methods

For the time-discretization of the splitted equation, the combination of accurate methods that will fit in the higher order context of the iterative operator splitting methods are important.

Based on the iterative methods the start-solution for the first iteration-step is important to obtain higher order results. For the next iteration steps the order have to increased till the proposed order of the time-discretization.

Therefore we propose the Runge-Kutta and BDF-methods as adapted time-discretization methods to reach higher order results.

For the time-discretization we use the following higher order discretization methods.

## 5.1 Runge-Kutta method

We use the implicit trapezoidal rule:

$$\begin{array}{c|cc} 0 & & \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{41}$$

Further more we use the following Gauß Runge-Kutta method :

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{42}$$

To use this Runge-Kutta methods with our operator-splitting method we have to take into account that we solve in each iteration step equations of the form $\partial_t u_i = A u_i + b$. Where $b = B u_{i-1}$ is a discrete function as we only have a discrete solution for $u_{i-1}$.
For the implicit trapezoidal rule this is no problem, because we do not need

the values at any sub-points. Where on the other hand for the Gauß method we need to now the values of b at the sub-points $t_0 + c_1 h$ and $t_0 + c_2 h$ with $c = (\frac{1}{2} - \frac{\sqrt{3}}{6}, \frac{1}{2} + \frac{\sqrt{3}}{6})^T$. Therefor we must interpolate b. To do so we choose the cubic spline functions.
Numerical experiments show that this works properly with non-stiff problems, but worth with stiff-problems.

### 5.2 BDF method

Because the higher order Gauß Runge-Kutta method combined with cubic spline interpolation does not work properly with stiff problems we use the following BDF method of order 3 which does not need any sub-points and therefor no interpolation is needed.

BDF3

$$1/k(11/6u^{n+2} - 3u^{n+1} + 3/2u^n - 1/3u^{n-1} = A(u^{n+3}) \tag{43}$$

For the prestepping, i.e. to obtain $u_1, u_2$, we use the above implicit trapezoidal rule.

### 5.3 Implicit-explicit methods

The implicit-explicit (IMEX) schemes have been widely developed for time integration of spatial discretised partial differential equations of diffusion-convection type. These methods are applied to decouple the implicit and explicit terms. So for example the convection-diffusion equation, one use the explicit part for the convection term and the implicit part for the diffusion. In our application we divide between the stiff and nonstiff term, so we apply the implicit part for the stiff operators and the explicit part for the nonstiff operators.

**FS-RK-method** We propose the A-stable FSRK-scheme, see [1], of first and second order for our applications.
The tableau in the Butcher-form is given as

$$\begin{array}{c|cccc|cccc}
1 & 1 & & & & 0 & & & \\
1 & 1 & 0 & & & 0 & 1 & & \\
\frac{4}{9} & -\frac{88}{45} & 0 & \frac{12}{5} & 0 & 0 & \frac{5}{9} & 0 & \\
\frac{1}{3} & -\frac{407}{75} & 0 & \frac{144}{25} & 0 & 0 & -\frac{31}{15} & 0 & \frac{12}{5} \\
\hline
\text{order1} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\hline
\text{order2} & \frac{1}{10} & 0 & \frac{9}{10} & 0 & 0 & \frac{1}{4} & 0 & \frac{3}{4}
\end{array} \tag{44}$$

To obtain second order convergence in numerical examples it is important to split the operator in the right way as we will show later.

**SBDF-Method** We use the following SBDF method which is a modification of the BDF3 method.
As prestepping method we use again the implicit trapezoidal rule.

$$1/k(11/6u^{n+1} - 3u^n + 3/2u^{n-1} - 1/3u^{n-2}) \tag{45}$$
$$= 3A(u^n) - 3A(u^{n-1}) + A(u^{n-2}) + B(u^{n+1}) \tag{46}$$

Again it is important to split the operator in the right way.

## 6 Numerical Results

We start with a first example of the higher order iterative method.

### 6.1 First test-example of an ODE

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

We deal with the following equation :

$$\partial_t u_1 = -\lambda_1 u_1 + \lambda_2 u_2 \ , \tag{47}$$
$$\partial_t u_2 = \lambda_1 u_1 - \lambda_2 u_2 \ , \tag{48}$$
$$u_1(0) = u_{10} \ , \ u_2(0) = u_{20} \text{ (initial conditions) } , \tag{49}$$

where $\lambda_1 \in \mathbb{R}^+$ and $\lambda_2 \in \mathbb{R}^+$ 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-system (47)–(49) in operator notation, and end up with the following equations :

$$\partial_t u = Au + Bu \ , \tag{50}$$
$$u(0) = (u_{10}, u_{20})^T , \tag{51}$$

where $u(t) = (u_1(t), u_2(t))^T$ for $t \in [0, T]$.

Our spitted operators are

$$A = \begin{pmatrix} -\lambda_1 & \lambda_2 \\ 0 & 0 \end{pmatrix} \ , \ B = \begin{pmatrix} 0 & 0 \\ \lambda_1 & -\lambda_2 \end{pmatrix} \ . \tag{52}$$

We chose such an example to have $AB \neq BA$, and therefore we have a splitting error of first order for the usual sequential splitting methods, called A-B splitting.

For a first non-stiff example we chose $\lambda_1 = 0.25$ and $\lambda_2 = 0.5$ on the time interval [0,1].
Our numerical results based on the above described RK-methods of second and fourth order are presented in the following table 1. We chose a constant time-stepsize $h = 10^{-4}$ to make sure that we do not influence the convergence rate of

our iterative operator-splitting by the Runge-Kutta methods.
The numerical results show that the splitting error decreases as long as the used Runge-Kutta method allows it. Therefore you can say that more iterations are only sufficient when you use a method of higher order. You can also see that the iterative operator-splitting method is of order $(i-1)$ as long as the Runge-Kutta method is good enough.

| Number of time-partitions | Iterative Steps | $err_1$ (2th order) | $err_2$ (2th order) | $err_1$ (4th order) | $err_2$ (4th order) |
|---|---|---|---|---|---|
| 2 | 1 | 4.5321e-002 | 3.6077e-003 | 4.5321e-002 | 3.6077e-003 |
| 2 | 10 | 3.9664e-003 | 4.7396e-004 | 3.9664e-003 | 4.7397e-004 |
| 2 | 100 | 3.9204e-004 | 4.8078e-005 | 3.9204e-004 | 4.8083e-005 |
| 3 | 1 | 4.6126e-004 | 3.6077e-003 | 4.6126e-004 | 3.6077e-003 |
| 3 | 10 | 7.8129e-006 | 2.9285e-005 | 7.8069e-006 | 2.9289e-005 |
| 3 | 100 | 8.5988e-008 | 2.8270e-007 | 8.0050e-008 | 2.8682e-007 |
| 4 | 1 | 4.6126e-004 | 2.2459e-005 | 4.6126e-004 | 2.2464e-005 |
| 4 | 10 | 4.1883e-007 | 4.2629e-008 | 4.1321e-007 | 4.8154e-008 |
| 4 | 100 | 5.9521e-009 | 5.4846e-009 | 4.0839e-010 | 4.9968e-011 |
| 5 | 1 | 1.9096e-006 | 2.2459e-005 | 1.9040e-006 | 2.2464e-005 |
| 5 | 10 | 6.0151e-009 | 3.7052e-009 | 4.7929e-010 | 1.8295e-009 |
| 5 | 100 | 5.5356e-009 | 5.5354e-009 | 5.0404e-014 | 1.7830e-013 |
| 6 | 1 | 1.9096e-006 | 6.1224e-008 | 1.9040e-006 | 6.6759e-008 |
| 6 | 10 | 5.5528e-009 | 5.5336e-009 | 1.7198e-011 | 1.9820e-012 |
| 6 | 100 | 5.5355e-009 | 5.5355e-009 | 2.4425e-015 | 4.4409e-016 |

**Table 1.** Numerical results for the first example with the iterative splitting method and 2th and 4th order RK method.

As a stiff example we chose $\lambda_1 = 1$ and $\lambda_2 = 10^4$ on the time interval [0,1]. And test the Implicit-explicit methods.

**FS Runge-Kutta**

We use the above presented FS Runge-Kutta method of first and second order. The results are presented in table 2.

| time steps | $err_1$ (1th order) | $err_2$(1th order) | $err_1$ (2th order) | $err_2$(2th order) |
|---|---|---|---|---|
| 10 | 9.0883e+002 | 9.0883e-002 | 4.6630e+002 | 4.6631e-002 |
| 100 | 9.8980e+001 | 9.8980e-003 | 4.2007e+001 | 4.2007e-003 |
| 1000 | 9.9870e+000 | 9.9870e-004 | 4.0068e+000 | 4.0068e-004 |
| 10000 | 9.9960e-001 | 9.9960e-005 | 3.0767e-001 | 3.0767e-005 |

**Table 2.** Numerical results for the first example with the FS RK method of order 1 and 2.

Due to the bad results we try another splitting.
Our spitted operators are now

$$A = \begin{pmatrix} -\lambda_1 & 0 \\ \lambda_1 & 0 \end{pmatrix} \; , \; B = \begin{pmatrix} 0 & \lambda_2 \\ 0 & -\lambda_2 \end{pmatrix} \; . \tag{53}$$

We then get better results presented in table 3.

| time steps | $err_1$ (1th order) | $err_2$(1th order) | $err_1$ (2th order) | $err_2$(2th order) |
|---|---|---|---|---|
| 10 | 1.8178e-005 | 1.8178e-005 | 1.1831e-002 | 1.1831e-002 |
| 100 | 1.9798e-006 | 1.9798e-006 | 1.6173e-004 | 1.6173e-004 |
| 1000 | 1.9976e-007 | 1.9976e-007 | 1.8797e-006 | 1.8797e-006 |
| 10000 | 1.9994e-008 | 1.9994e-008 | 3.4345e-008 | 3.4345e-008 |

**Table 3.** Numerical results for the first example with the FS RK method of order 1 and 2.

**SBDF**

Now we use the described SBDF method. The results are presented in table 4.

| time steps | $err_1$ | $err_2$ |
|---|---|---|
| 10 | 1.8767e-010 | 1.8762e-010 |
| 100 | 2.1316e-014 | 2.1142e-018 |
| 1000 | 8.8818e-016 | 1.0842e-019 |
| 10000 | 4.8850e-015 | 5.6921e-019 |

**Table 4.** Numerical results for the first example with the SBDF method of 3th order.

In the following we compare this results with our iterative operator splitting with the described BDF3 method.

**BDF3**

The advantage of the BDF3 method is that it allows us to use bigger time step sizes.
We first chose a time step size $h = 10^{-2}$ for the BDF3 method. The numerical results are presented in table 5.
We can even get good results when we only chose a time step size $h = 10^{-1}$ for the BDF3 method. The results are presented in table 6.

Based on the presented tables our proposed iterative operator-splitting methods with standard higher order methods have more accurate results and are more efficient as the complicate IMEX-methods.

| Number of time-partitions | Iterative Steps | $err_1$ | $err_2$ |
|---|---|---|---|
| 5 | 1 | 3.4434e-001 | 3.4434e-001 |
| 5 | 10 | 3.0907e-004 | 3.0907e-004 |
| 10 | 1 | 2.2600e-006 | 2.2600e-006 |
| 10 | 10 | 1.5397e-011 | 1.5397e-011 |
| 15 | 1 | 9.3025e-005 | 9.3025e-005 |
| 15 | 10 | 5.3002e-013 | 5.4205e-013 |
| 20 | 1 | 1.2262e-010 | 1.2260e-010 |
| 20 | 10 | 2.2204e-014 | 2.2768e-018 |

**Table 5.** Numerical results for the first example with the iterative splitting method and BDF3.

| Number of time-partitions | Iterative Steps | $err_1$ | $err_2$ |
|---|---|---|---|
| 5 | 1 | 3.4433e-001 | 3.4433e-001 |
| 10 | 1 | 2.2591e-006 | 2.2591e-006 |
| 15 | 1 | 1.0039e-004 | 1.0039e-004 |
| 20 | 1 | 6.3926e-010 | 6.3943e-010 |
| 25 | 1 | 1.3385e-009 | 1.3385e-009 |
| 30 | 1 | 4.8302e-010 | 4.8307e-010 |

**Table 6.** Numerical results for the first example with the iterative splitting method and BDF3.

### 6.2 Second Example

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

$$\partial_t u = D \partial_{xx} u \tag{54}$$

$$u(x, 0) = \sin(\pi x) \tag{55}$$

with vanishing Dirichlet-boundary conditions.
We have the time-interval $t \in [0, T]$ and the space-interval $x \in [0, X]$.
We chose $D = 0.025$ and $T = X = 1$.
The analytical solution of the equation is given by

$$u_{\text{exact}}(x, t) = \sin(\pi x) \exp(-D\pi^2 t) \ . \tag{56}$$

For the spatial discretization we use an upwind finite difference discretization given as :

$$\partial^- \partial^+ u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \ . \tag{57}$$

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

$$A = \frac{D}{\Delta x^2} \cdot \begin{pmatrix} 0 & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 0 \end{pmatrix} \quad (58)$$

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

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} := A \ . \quad (59)$$

We now solve the problem

$$\partial_t u = A_1 u + A_2 u \quad (60)$$

$$(61)$$

with our iterative operator-splitting together with the BDF3 method with time step size $h = 10^{-2}$. As the error occurs mostly at the point where we split the interval, we present values around this point in table 7.

| Number of time-partitions | Iterative Steps | error $x = 0.4$ | error $x = 0.5$ | error $x = 0.6$ |
|---|---|---|---|---|
| 1 | 10 | 1.0379e-001 | 2.1866e-001 | 2.0795e-001 |
| 5 | 10 | 1.6514e-002 | 3.4518e-002 | 1.6514e-002 |
| 10 | 10 | 2.0856e-003 | 3.8652e-003 | 1.8342e-003 |
| 15 | 10 | 2.6049e-004 | 6.0690e-004 | 2.6049e-004 |
| 20 | 10 | 3.9743e-005 | 6.4629e-005 | 3.6828e-005 |

**Table 7.** Numerical results for the second example with the iterative splitting method and BDF3.

### 6.3 Third example : Convection-diffusion-reaction equation

We consider the one-dimensional convection-diffusion-reaction equation given by

$$R\partial_t u + v\partial_x u - D\partial_{xx} u = -\lambda u \ , \text{ on } \Omega \times [t_0, t_{\text{end}}) \quad (62)$$

$$u(x, t_0) = u_{\text{exact}}(x, t_0) \ , \quad (63)$$

$$u(0, t) = u_{\text{exact}}(0, t) \ , \ u(L, t) = u_{\text{exact}}(L, t), \quad (64)$$

defined over $\Omega \times [t_0, t_{\text{end}})$ with $\Omega = [0, L]$, and $t_0 = 10^4$, $t_{\text{end}} = 2 \cdot 10^4$ and $L = 30$. Further we have $\lambda = 10^{-5}$, $v = 0.001$, $D = 0.0001$ and $R = 1.0$.

The analytical solution of the equation (62) considered on $\mathbb{R} \times (0, t_{\text{end}})$, with vanishing Dirichlet-boundary conditions and also using a $\delta$-function as initial value, can be derived by Laplace-Transformation, and is given by

$$u_{\text{exact}}(x, t) = \frac{\tilde{u}_0}{2\sqrt{D\pi t}} \exp\left(-\frac{(x - vt)^2}{4Dt}\right) \exp(-\lambda t) \ , \quad (65)$$

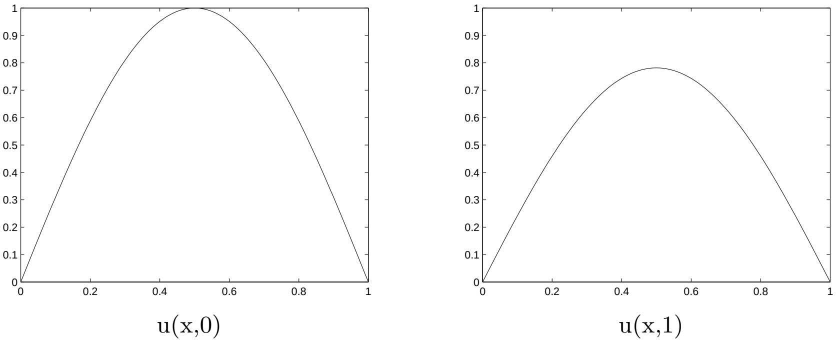


**Fig. 1.** Numerical result for the second example with the iterative splitting method and BDF3, left figure $t = 0$, right figure $t = 1$.

with $\tilde{u}_0 = 1$, the restriction of $u_{\text{exact}}$ to $\Omega \times (0, t_{\text{end}})$ is a solution to (62)-(64).

To be out of the singular point of the exact solution, we start from the time-point $t_0 = 10^4$.

Our spitted operators are

$$A = \frac{D}{R} \partial_{xx} u \ , \ B = -\frac{1}{R} (\lambda + v) \partial_x u \ . \tag{66}$$

For the spatial discretization we use the finite difference discretization method. For the operator A we use as above an upwind difference and for the operator B we use an backward difference and we set the space step size to $\Delta x = \frac{1}{300}$. We solve the problem with our iterative operator-splitting together with the BDF3 method with time step size $h = 10^{-2}$. Our numerical results are presented in table 8 and in figure 2.

| Number of time-partitions | Iterative Steps | error $x = 18$ | error $x = 20$ | error $x = 22$ |
|---|---|---|---|---|
| 1 | 10 | 9.8993e-002 | 1.6331e-001 | 9.9054e-002 |
| 2 | 10 | 9.5011e-003 | 1.6800e-002 | 8.0857e-003 |
| 3 | 10 | 9.6209e-004 | 1.9782e-002 | 2.2922e-004 |
| 4 | 10 | 8.7208e-004 | 1.7100e-002 | 1.5168e-003 |

**Table 8.** Numerical results for the third example with the iterative splitting method and BDF3.

## 7 Conclusions and Discussions

We present the application of the iterative Operator-Splitting methods with combination of adequate explicit or implicit methods on each operator-equation. The stability of such methods are shown and their consistency. The benefit of

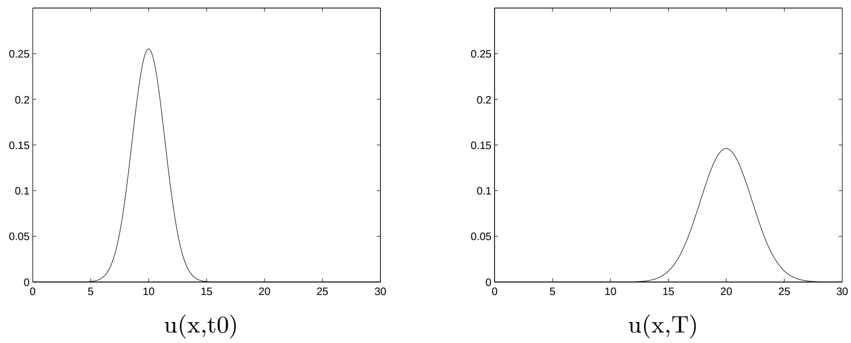


**Fig. 2.** Numerical result for the third example with the iterative splitting method and BDF3, left figure $t = 0$, right figure $t = T$.

such combination in comparing with pure time-discretization methods but with more complicate setting of the method is presented and efficiency is shown. The iterative operator-splitting methods are applied to multi-physics and the results are presented. We could apply such methods also for multi-dimensional problems, while splitting in simpler one-dimensional problems. In the future we focus us on the development of improved operator-splitting methods with respect to the application in nonlinear parabolic equations.

## References

1. J.C. Butcher. Numerical Methods for Ordinary Differential Equations. *John Wiley & Sons Ltd, Chichester*, (2003).
2. R.E. Ewing. Up-scaling of biological processes and multiphase flow in porous media. *IIMA Volumes in Mathematics and its Applications*, Springer-Verlag, 295 (2002), 195-215.
3. I. Farago, and Agnes Havasi. *On the convergence and local splitting error of different splitting schemes.* Eötvös Lorand University, Budapest, 2004.
4. P. Csomós, I. Faragó and A. Havasi. *Weighted sequential splittings and their analysis.* Comput. Math. Appl., (to appear)
5. K.-J. Engel, R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations.* Springer, New York, 2000.
6. I. Farago. *Splitting methods for abstract Cauchy problems.* Lect. Notes Comp.Sci. 3401, Springer Verlag, Berlin, 2005, pp. 35-45
7. I. Farago, J. Geiser. *Iterative Operator-Splitting methods for Linear Problems.* Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany, June 2005.
8. P. Frolkovič and J. Geiser. *Numerical Simulation of Radionuclides Transport in Double Porosity Media with Sorption.* Proceedings of Algorithmy 2000, Conference of Scientific Computing, 28-36, 2000.
9. J. Geiser. *Numerical Simulation of a Model for Transport and Reaction of Radionuclides.* Proceedings of the Large Scale Scientific Computations of Engineering and Environmental Problems, Sozopol, Bulgaria, 2001.

10. J. Geiser. *Gekoppelte Diskretisierungsverfahren für Systeme von Konvektions-Dispersions-Diffusions-Reaktionsgleichungen.* Doktor-Arbeit, Universität Heidelberg, 2003.
11. J. Geiser. $R^3T$ *: Radioactive-Retardation-Reaction-Transport-Program for the Simulation of radioactive waste disposals.* Proceedings: Computing, Communications and Control Technologies: CCCT 2004, The University of Texas at Austin and The International Institute of Informatics and Systemics (IIIS), to appear, 2004.
12. J. Geiser. *Iterative Operator-Splitting methods for Parabolic Differential Equations : Convergence theory.* Humboldt-Preprint, to be submitted, February 2006.
13. W. Hundsdorfer, L. Portero. A Note on Iterated Splitting Schemes. CWI Report MAS-E0404, Amsterdam, Netherlands, 2005.
14. W.H. Hundsdorfer, J. Verwer W. *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, Berlin, (2003).
15. 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.
16. R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics ,
17. G.I Marchuk. *Some applications of splitting-up methods to the solution of problems in mathematical physics.* Aplikace Matematiky, 1 (1968) 103-132.
18. G. Strang. *On the construction and comparision of difference schemes.* SIAM J. Numer. Anal., 5:506–517, 1968.
19. J.,G. Verwer and B. Sportisse. *A note on operator splitting in a stiff linear case.* MAS-R9830, ISSN 1386-3703, 1998.
20. Z. Zlatev. *Computer Treatment of Large Air Pollution Models.* Kluwer Academic Publishers, 1995.