Weighted Iterative Operator-Splitting Methods for stiff problems and applications

Jürgen Geiser[1] and Christos Kravvaritis[2]

[1] Weierstrass Institute of Applied Analysis and Stochastics,
Mohrenstrasse 39, D-10117 Berlin, Germany
geiser@wias-berlin.de
[2] Department of Mathematics, University of Athens,
Panepistimiopolis 15784, Athens, Greece
ckrav@math.uoa.gr

Abstract. In this paper we describe advanced operator-splitting methods for more accurate and exact decoupling of stiff systems. We deal with 2 stiff operators and contribute for both stiff operators a new weighted iterative operator-splitting method, decribed in [3]. Based on the stabilisation theory for stiff systems, given in [**?**] and [9], we developed our weighting method. The idea behind is a combination of first order and higher order operator splitting methods. We discuss our methods in welknown test-problems for systems of ordinary-differential equations and compare with the exact solutions of the sytsems. Finally we present an application to a convection-reaction equation with stiff reaction-term. We end with a discuss for modifing our methods for multi-dimensional and multi-physical problems.

1 Introduction

We motivate our studying on the methods for solving complicate coupled and stiff equations coming from models for transport-reaction-, bio-remediation- and nonlinear diffusion-reaction-problems. The main advantage of the operator-splitting methods is decoupling the mixed processes into simpler synchroneous physical processes. Because of the symmetry we could decouple such processes. To be more accurate and to come to higher order splitting methods, a effective a new group of methods is presented, based on higher order splitting methods. To stabilise the delicate initial process of such higher order methods, we present a stabilisation with initial presteps and weighting factors for the iterative method. We discuss the weighted splitting method based on higher iterative splitting method and apply the results to stiff systems of reaction- and convection-reaction-equations.

The paper is organised as follows. The mathematical model is presented in the section 2. The family of operator-splitting methods are discussed in section 3. In the section 4 we present the stability analysis of the weighted iterative operator splitting method. The numerical applications of systems of ODE's and PDE's are discussed in section 5. The conclusions and future works are introduced in section 6 .

2 Mathematical Model

The motivation for the study presented below is coming from a computational simulation of bio-remediation [1] or radioactive contaminants [5], [4].

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.

In the following we describe the Operator-Splitting methods as a basic method for solving our equation.

3 Operator-Splitting Methods

The operator-splitting methods are used to solve complex models in the geophysical and environmental physics, they are developed and applied in [13], [12], [14] and [15]. The ideas in this article are based on solving simpler equations with respect to receive higher order discretization methods for the remaining equations. For this aim we use the operator-splitting method and decouple the equation as following described.

3.1 Splitting methods of first order for linear equations (A-B-splitting)

First we describe the simplest operator-splitting, which is called *sequential splitting*, for the following system of ordinary linear differential equations:

$$\partial_t c(t) = A \, c(t) \; + \; B \, c(t) \, , \tag{4}$$

where the initial-conditions are $c^n = c(t^n)$. The operators A and B are bounded operators in a Banach-space. For example we could also discretised the spatial variable and get operators (matrices) in an ODE context. Hence, they can be considered as bounded operators.

The sequential operator-splitting method is introduced as a method which solves the two sub-problems sequentially, where the different sub-problems are

connected via the initial conditions. This means that we replace the original problem (4) with the sub-problems

$$\frac{\partial c^*(t)}{\partial t} = Ac^*(t) \ , \quad \text{with } c^*(t^n) = c^n \ , \tag{5}$$
$$\frac{\partial c^{**}(t)}{\partial t} = Bc^{**}(t) \ , \quad \text{with } c^{**}(t^n) = c^*(t^{n+1}) \ ,$$

whereby the splitting time-step is defined as $\tau_n = t^{n+1} - t^n$. The approximated split solution is defined as $c^{n+1} = c^{**}(t^{n+1})$.

Clearly, the change of the original problems with the sub-problems usually results some error, called *splitting error*. Obviously, the splitting error of the sequential splitting method can be derived as follows (cf. e.g.[6])

$$\rho_n = \frac{1}{\tau}(\exp(\tau_n(A+B)) - \exp(\tau_n B)\exp(\tau_n A))\ c(t^n)$$

$$= \frac{1}{2}\tau_n[A,B]\ c(t^n) + O(\tau^2) \ . \tag{6}$$

whereby $[A,B] := AB - BA$ is the commutator of A and B. Consequently, the splitting error is $O(\tau_n)$ when the operators A and B do not commute, otherwise the method is exact. Hence, by definition, the sequential splitting is called *first order splitting method* .

In the next subsection we present the iterative-splitting method.

3.2 Iterative splitting method

The following algorithm is based on the iteration with fixed splitting discretization step-size τ. On the time interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for $i = 0, 2, \ldots 2m$. (cf. [11] and [3].)

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \ + \ Bc_{i-1}(t), \text{ with } \ c_i(t^n) = c^n \tag{7}$$
$$\frac{\partial c_{i+1}(t)}{\partial t} = Ac_i(t) \ + \ Bc_{i+1}(t), \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{8}$$

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

In the following we will analyze the convergence and the rate of the convergence of the method (7)–(8) 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* $\mathbf{X}$

$$\begin{aligned} \partial_t c(t) = Ac(t) + Bc(t), \quad 0 < t \leq T \\ c(0) = c_0 \end{aligned} \tag{9}$$

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 (7)–(8) is convergent and the and the rate of the convergence is of second order.*

The proof could be found in [8].

Remark 1. When A and B are matrices (i.e. (7)–(8) is a system of ordinary differential equations), for the growth estimation we can use the concept of the logarithmic norm. (See e.g.[10].) 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 second order to the exact solution.

In the next subsection we present a modification of the iterative-splitting method with weighting of the operators.

3.3 Weighted Iterative splitting method

We assume an improved iterative splitting method with respect to more stable behaviour in the continuous case.

As a first method the unsammetric weighted iterative splitting method is introduced. The algorithm is based on the iteration with fixed splitting discretization step-size τ. On the time interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for $i = 0, 2, \ldots 2m$.

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \ + \ \omega \ Bc_{i-1}(t), \text{ with } \ c_i(t^n) = c^n \tag{10}$$

$$\begin{aligned} &\text{and } c_0(t^n) = c^n \ , \ c_{-1} = 0.0, \\ &\frac{\partial c_{i+1}(t)}{\partial t} = \omega \ Ac_i(t) \ + \ Bc_{i+1}(t), \end{aligned} \tag{11}$$

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

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 $c^{n+1} = c_{2m+1}(t^{n+1})$. Our parameter $\omega \in [0, 1]$. For $\omega = 0$ we have the A-B-splitting and for $\omega = 1$ we have the iterative splitting method, see [3].

In the same manner the initial conditions of the weighted iterative splitting method are weighted between the sequential splitting and iterative splitting method.

4 Stability Theory

In the following we present the stability analysis for the continuous case with commutative operators. First we apply the recussion for the general case and then we concentrate us to the commutative case.

4.1 Recursion

We study our stability for the linear system (10) and (11).

We consider the suitable vector norm $||\cdot||$ on $\mathbb{R}^M$, together with its induced operator norm.

The matrix exponential of $Z \in \mathbb{R}^{M\times M}$ is denoted by $\exp(Z)$.

We assume that

$||\exp(\tau\ A) \leq 1||$ and $||\exp(\tau\ B) \leq 1||$, for all $\tau > 0$.

It can be shown that the system (4) implies $\exp(\tau\ (A+B)) \leq 1$ and is ifself stable.

For the linear problem (10) and (11) it follows be integration that

$$c_i(t) = \exp((t-t^n)A)c^n + \int_{t^n}^{t} \exp((t-s)A)\ \omega\ Bc_{i-1}(s)\ ds\ , \tag{12}$$

$$c_{i+1}(t) = \exp((t-t^n)B)c^n + \int_{t^n}^{t} \exp((t-s)B)\ \omega\ Ac_{i}(s)\ ds\ , \tag{13}$$

With the elimination of c_i we get

$$\begin{aligned} c_{i+1}(t) = \exp((t-t^n)B)c^n\ &+\ \omega\ \int_{t^n}^{t} \exp((t-s)B)\ A\ \exp((t-s)A)\ ds\ c^n \\ &+ \omega^2\ \int_{s=t^n}^{t}\int_{s'=t^n}^{s} \exp((t-s)B)\ A\ \exp((s-s')A)\ B\ c_{i-1}(s')\ ds'\ ds\ , \end{aligned} \tag{14}$$

For the following commution case we could evaluate the double integral $\int_{s=t^n}^{t}\ \int_{s'=t^n}^{s}$ as $\int_{s'=t^n}^{t}\ \int_{s=s'}^{t}$ and could derive the weighted stability-theory.

4.2 Commution operators

For more transparency of the formula (14) we consider the eigenvalues λ_1 of A and λ_2 of B.

Replacing the operators A and B we obtain after some calculations

$$\begin{aligned} c_{i+1}(t) = c^n \frac{1}{\lambda_1-\lambda_2}\left(\omega\lambda_1 \exp((t-t^n)\lambda_1) + ((1-\omega)\lambda_1 - \lambda_2)\exp((t-t^n)\lambda_2)\right) \\ + c^n\ \omega^2\ \frac{\lambda_1\lambda_2}{\lambda_1-\lambda_2}\ \int_{s=t^n}^{t} \left(\exp((t-s)\lambda_1)\ -\ \exp((t-s)\lambda_2)\right) ds\ , \end{aligned} \tag{15}$$

We denote that this relation is symmetric in λ_1 and λ_2.

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

$$c_{2m}(t^{n+1}) = S_m(z_1, z_2)\ c^n\ , \tag{16}$$

where S_m is the stability function of the scheme with m-iterations. We use (15) and obtain after some calculations

$$\begin{aligned} S_1(z_1, z_2) = \omega^2\ c^n\ +\ \frac{\omega\ z_1 + \omega^2\ z_2}{z_1 - z_2}\ \exp(z_1)\ c^n \\ +\ \frac{(1 - \omega - \omega^2)\ z_1 - z_2}{z_1 - z_2}\ \exp(z_2)\ c^n \end{aligned} \tag{17}$$

$$\begin{aligned} S_2(z_1, z_2) = \omega^4\ c^n\ +\ \frac{\omega\ z_1 + \omega^4\ z_2}{z_1 - z_2}\ \exp(z_1)\ c^n \\ +\ \frac{(1 - \omega - \omega^4)\ z_1 - z_2}{z_1 - z_2}\ \exp(z_2)\ c^n \\ +\ \frac{\omega^2\ z_1\ z_2}{(z_1 - z_2)^2}\ ((\omega z_1 + \omega^2 z_2)\exp(z_1) \\ +(-(1 - \omega - \omega^2)z_1 + z_2)\exp(z_2))\ c^n \\ +\ \frac{\omega^2\ z_1\ z_2}{(z_1 - z_2)^3}\ ((-\omega z_1 - \omega^2 z_2)(\exp(z_1) - \exp(z_2)) \\ +((1 - \omega - \omega^2)z_1 - z_2)(\exp(z_1) - \exp(z_2)))\ c^n \end{aligned} \tag{18}$$

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

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

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

The stability of the two iterations is given in the following theorem with A and $A(\alpha)$-stability

Theorem 2. *We have the following stability :*

For S_1 we have the A-stability
$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_1(z_1, z_2)| \leq 1\ ,\ \forall\ \alpha \in [0, \pi/2]$ *with* $\omega = \frac{\sqrt{2}}{2}$

For S_2 we have the $A(\alpha)$-stability
$\max_{z_1 \leq 0, z_2 \in W_\alpha} |S_2(z_1, z_2)| \leq 1\ ,\ \forall\ \alpha \in [0, \pi/2]$ *with* $\omega \leq \left(\frac{1}{8\ \tan^2(\alpha)+1}\right)^{1/8}$

Proof. We consider a fixed $z_1 = z$ and $z_2 \to -\infty$. Then we obtain

$$S_1(z, \infty) = \omega^2(1 - e^z) \tag{19}$$

and

$$S_2(z,\infty) = \omega^4(1-(1-z)e^z) \tag{20}$$

If $z = x + iy$ then :
1.) For S_1

$$|S_1(z,\infty)|^2 = \omega^4(1-\exp(x)cosy + \exp(2x)) \leq 1 \tag{21}$$

after some rearrangements

$$\exp(2x) \leq \frac{1}{\omega^4} - 1 + 2\exp(x)\ \cos y \tag{22}$$

because of $x < 0$ and $y \in \mathbb{R}$ we could estimate $-2 \leq 2\exp(x)\ \cos(y)$ and $\exp(2x) \leq 1$

We could estimate (22) as $\omega \leq \frac{\sqrt{2}}{2}$.

2.) For S_2

$$\begin{aligned} |S_2(z,\infty)|^2 = \omega^8(1 - 2\exp(x)((1-x)\cos y + y\sin y) \\ + \exp(2x)((1-x)^2 + y^2)) \leq 1 \end{aligned} \tag{23}$$

after some calculations we could follow

$$\exp(x) \leq (\frac{1}{\omega^8} - 1)\frac{\exp(-x)}{(1-x)^2+y^2} - 2\frac{|1-x|+|y|}{(1-x)^2+y^2} \tag{24}$$

we could estimate for $x < 0$ and $y \in \mathbb{R}$ $\frac{|1-x|+|y|}{(1-x)^2+y^2} \leq 3/2$ and $\frac{1}{2\tan^2(\alpha)} < \frac{\exp(-x)}{(1-x)^2+y^2}$ where $\tan(\alpha) = y/x$.

And we get the bound for the ω
$\omega \leq \left(\frac{1}{8\tan^2(\alpha)+1}\right)^{1/8}$.

In the next section we apply our methods to test-problems.

5 Numerical Results

We deal with some applications in this section to verify our theoretical results, done in the previous sections.

Our motivation for using the operator splitting in applications is the possibility to decouple complex operators and handle them in separate equations with adapted methods. We simplify to a system of first order ordinary differential equations, which are simpler to compute.

To improve the sensibility and the stability of the iterative higher order splitting methods, we apply pre-steps to reach improved initial conditions or weight our methods between first order splitting and higher order splitting method.

We start with a first example to use the stabile first order splitting as a pre-step method and then start with the higher order iterative method.

5.1 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 :

$$\partial_t u_1 = -\lambda_1 u_1 + \lambda_2 u_2 \ , \tag{25}$$
$$\partial_t u_2 = \lambda_1 u_1 - \lambda_2 u_2 \ , \tag{26}$$
$$u_1(0) = u_{10} \ , \ u_2(0) = u_{20} \text{ (initial conditions)} \ , \tag{27}$$

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 (51)–(53) in operator notation, and end up with the following equations :

$$\partial_t u = Au + Bu \ , \tag{28}$$
$$u(0) = (u_{10}, u_{20})^T, \tag{29}$$

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

Our spitted operators are

$$A = \begin{pmatrix} -\lambda_1 & \lambda_2 \\ 0 & 0 \end{pmatrix} \ , \ B = \begin{pmatrix} 0 & 0 \\ \lambda_1 & -\lambda_2 \end{pmatrix} \ . \tag{30}$$

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 the complex equation (51)–(53) we could derive the analytical solution by integrating the system of ODE's :

$$u_1(t) = u_{10} + u_{20} \exp(-(\lambda_1 + \lambda_2)t) \ , \tag{31}$$
$$u_2(t) = \frac{\lambda_1}{\lambda_2} \ u_{10} - u_{20} \exp(-(\lambda_1 + \lambda_2)t) \ , \tag{32}$$

We apply first the sequential splitting and the iterative operator-splitting, further we combine them be using the pre-step based methods to see the improved results.

For the time-steps Δt we have $\Delta t = 1$ for 1 time-partition, $\Delta t = 0.1$ for 10 time-partitions etc.

To validate the various methods and obtain the optimal results we apply all the methods on this example with parameters $\lambda_1 = 0.25$, $\lambda_2 = 0.5$, initial conditions $u_{10} = u_{20} = 1$, $u_1^{-1}(0) = u_2^{-1}(0) = 0$ and the end-time $\Delta t = 1$. With these values we get from the analytical solution of our equation: $u_{1,exact} = 1$ and $u_{2,exact} = 0.73618$. We compute u_2 as an exact solution with Laplace-Transformation, see [6], [7].

The sequential (A-B) splitting method For the sequential (A-B) splitting method, we define the following splitting equations of our system of ODE's in an A- and B-step as following

A-step

$$\begin{aligned} \partial_t u_1^* &= -\lambda_1 u_1^* + \lambda_2 u_2^* \,, \\ \partial_t u_2^* &= 0 \,, \\ u_1^*(0) &= u_{10} \,,\; u_2^*(0) = u_{20} \,, \end{aligned}$$

B-step

$$\begin{aligned} \partial_t u_1^{**} &= 0 \,, \\ \partial_t u_2^{**} &= \lambda_1 u_1^{**} - \lambda_2 u_2^{**} \,, \\ u_1^{**}(0) &= u_{10}^*(\Delta t) \,,\; u_2^{**}(0) = u_{20}^*(\Delta t) \,, \end{aligned}$$

where $t \in [0, \Delta t]$ and the result of the computation is $u(\Delta t) = (u_1^{**}(\Delta t), u_2^{**}(\Delta t))^t$.

For these systems of equations we can derive analytical solutions and apply them in our numerical scheme, leading to

$$\begin{aligned} u_1^*(t) &= u_{10}\exp(-\lambda_1 t) + u_{20}\frac{\lambda_2}{\lambda_1} \\ u_2^*(t) &= u_{20}, \end{aligned}$$

and

$$\begin{aligned} u_1^{**}(t) &= u_{10}^{**}, \\ u_2^{**}(t) &= u_{20}^{**}\exp(-\lambda_2 t) + u_{10}^{**}\frac{\lambda_1}{\lambda_2}, \end{aligned}$$

and $u_1^{**}(0) = u_1^*(t),\; u_2^{**}(0) = u_2^*(t)$.
We compute with our given scheme and the numerical results are presented in Table 1.

The iterative splitting method For the iterative splitting method, we have the following splitting equations of our system of ODE's. We divide in step i and $i+1$ as following

Step i

Number of time-partitions	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	1.2211992169	0.8467828848	2.211992×10^{-1}	1.105996×10^{-1}
5	1.1847412811	0.8285539169	1.847413×10^{-1}	9.237064×10^{-2}
10	1.1802926209	0.8263295868	1.802926×10^{-1}	9.014631×10^{-2}
100	1.1763176930	0.8243421229	1.763177×10^{-1}	8.815885×10^{-2}

Table 1. Numerical results for the first example with the sequential (A-B) splitting method.

$$\begin{aligned} \partial_t u_1^i &= -\lambda_1 u_1^i + \lambda_2 u_2^i \ , \\ \partial_t u_2^i &= \lambda_1 u_1^{i-1} - \lambda_2 u_2^{i-1} \ , \\ u_1^i(0) = u_{10} \ , \ u_2^i(0) &= u_{20} \ , \end{aligned}$$

Step $i+1$

$$\begin{aligned} \partial_t u_1^{i+1} &= -\lambda_1 u_1^i + \lambda_2 u_2^i \ , \\ \partial_t u_2^{i+1} &= \lambda_1 u_1^{i+1} - \lambda_2 u_2^{i+1} \ , \\ u_1^{i+1}(0) = u_{10} \ , \ u_2^{i+1}(0) &= u_{20} \ , \end{aligned}$$

where $t \in [0, \Delta t]$.

For the steps i and $i+1$ we can derive analytical solutions and apply them in our numerical scheme. The analytical solutions are given as

$$\begin{aligned} u_1^i(t) &= u_{10}\exp(-\lambda_1 t) + u_{20}\frac{\lambda_2}{\lambda_1} \\ &\quad + u_1^{i-1}(t)(\lambda_2 t - \frac{\lambda_2}{\lambda_1}) + u_2^{i-1}(t)(-\frac{\lambda_2^2}{\lambda_1}t - \frac{\lambda_2^2}{\lambda_1^2}) \ , \\ u_2^i(t) &= u_1^{i-1}(t)\lambda_1 t - u_2^{i-1}(t)\lambda_2 t + u_{20}, \end{aligned}$$

and

$$\begin{aligned} u_1^{i+1}(t) &= -u_1^i(t)\lambda_1 t + u_2^i(t)\lambda_2 t + u_{10}, \\ u_2^{i+1}(t) &= u_{20}\exp(-\lambda_2 t) + u_{10}\frac{\lambda_1}{\lambda_2} \\ &\quad + u_1^i(t)(\lambda_1 t - \frac{\lambda_1}{\lambda_2}) + u_2^i(t)(-\frac{\lambda_1^2}{\lambda_2}t - \frac{\lambda_1^2}{\lambda_2^2}) \ , \end{aligned}$$

We compute with our given scheme and the numerical results are presented in Table 2.

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	1	1.1126340828	0.7916998145	1.126341×10^{-1}	5.551654×10^{-2}
1	2	1.1126340828	0.7916998145	1.126341×10^{-1}	5.551654×10^{-2}
1	4	0.9499679893	0.7743708882	5.003201×10^{-2}	3.818761×10^{-2}
1	10	0.9344708685	0.7727199697	6.552913×10^{-2}	3.653669×10^{-2}
1	50	0.9344606445	0.7727188806	6.553936×10^{-2}	3.653560×10^{-2}
10	1	1.0005094994	0.8009580357	5.094994×10^{-4}	6.477476×10^{-2}
10	2	1.0005094994	0.8009580357	5.094994×10^{-4}	6.477476×10^{-2}
10	4	0.9993139424	0.8005987078	6.860576×10^{-4}	6.441543×10^{-2}
10	10	0.9993125029	0.8005982751	6.874971×10^{-4}	6.441500×10^{-2}
10	50	0.9993125029	0.8005982751	6.874971×10^{-4}	6.441500×10^{-2}
100	1	1.0000055350	0.8030208642	5.534966×10^{-6}	6.683759×10^{-2}
100	2	1.0000055350	0.8030208642	5.534966×10^{-6}	6.683759×10^{-2}
100	4	0.9999930886	0.8030171866	6.911412×10^{-6}	6.683391×10^{-2}
100	10	0.9999930884	0.8030171866	6.911567×10^{-6}	6.683391×10^{-2}
100	50	0.9999930884	0.8030171866	6.911567×10^{-6}	6.683391×10^{-2}

Table 2. Numerical results for the first example with the iterative splitting method.

5.2 The pre-stepping method

Further we combine the A-B and iterative splitting method, as proposed in [8]. The new method project the iterative solution into the correct solution-space. This we called the pre-stepping method. Actually, we perform on the beginning once the A-B splitting and then we perform on the results the iterative splitting. For the pre-stepping method we get the results in Table 3.

The weighted splitting method According to the weighted splitting method, we divide our system of ODE's in step i and $i+1$ as following

Step i

$$\begin{aligned} \partial_t u_1^i &= -\lambda_1 u_1^i + \lambda_2 u_2^i \ , \\ \partial_t u_2^i &= \omega\lambda_1 u_1^{i-1} - \omega\lambda_2 u_2^{i-1} \ , \\ u_1^i(0) = u_{10} \ , \ u_2^i(0) &= u_{20} \ , \end{aligned}$$

Step $i+1$

$$\begin{aligned} \partial_t u_1^{i+1} &= -\omega\lambda_1 u_1^i + \omega\lambda_2 u_2^i \ , \\ \partial_t u_2^{i+1} &= \lambda_1 u_1^{i+1} - \lambda_2 u_2^{i+1} \ , \\ u_1^{i+1}(0) = u_{10} \ , \ u_2^{i+1}(0) &= u_{20} \ , \end{aligned}$$

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	0	1.2211992169	0.8467828848	2.211992×10^{-1}	1.105996×10^{-1}
1	1	1.0603905271	0.7861342740	6.039053×10^{-2}	4.995100×10^{-2}
1	2	0.9454209657	0.7738864908	5.457903×10^{-2}	3.770321×10^{-2}
1	3	0.9345436701	0.7727277253	6.545633×10^{-2}	3.654445×10^{-2}
1	10	0.9344606449	0.7727188806	6.553936×10^{-2}	3.653560×10^{-2}
5	0	1.1847412811	0.8285539169	1.847413×10^{-1}	9.237064×10^{-2}
5	1	1.18432628	0.7988842196	1.843263×10^{-3}	6.270094×10^{-2}
5	2	1.18448500	0.7986054507	1.844850×10^{-3}	6.242217×10^{-2}
5	3	0.9972861333	0.7975848110	2.713867×10^{-3}	6.140153×10^{-2}
5	10	0.9972649862	0.7975800703	2.735014×10^{-3}	6.139679×10^{-2}
10	0	1.1802926209	0.8263295868	1.802926×10^{-1}	9.014631×10^{-2}
2	10	1.0005095181	0.8009440338	5.095181×10^{-4}	6.476076×10^{-2}
10	2	1.0005095627	0.8009105675	5.095627×10^{-4}	6.472729×10^{-2}
10	3	0.9993139425	0.8005986507	6.860575×10^{-4}	6.441537×10^{-2}
10	10	0.9993125029	0.8005982751	6.874971×10^{-4}	6.441500×10^{-2}
100	0	1.1763176930	0.8243421229	1.763177×10^{-1}	8.815885×10^{-2}
100	1	1.0000055350	0.8030208507	5.534966×10^{-6}	6.683757×10^{-2}
100	2	1.0000055350	0.8030208183	5.534967×10^{-6}	6.683754×10^{-2}
100	3	0.9999930886	0.8030171866	6.911412×10^{-6}	6.683391×10^{-2}
100	10	0.9999930884	0.8030171866	6.911567×10^{-6}	6.683391×10^{-2}

Table 3. Numerical results for the first example with the pre-stepping method: first iteration with A-B splitting (denoted with 0 iterative steps) and then iterative splitting.

where $t \in [0, \Delta t]$.

For the steps i and $i+1$ we can derive analytical solutions and apply them in our numerical scheme. The analytical solutions are given as

$$\begin{aligned} u_1^i(t) &= u_{10}\exp(-\lambda_1 t) + u_{20}\frac{\lambda_2}{\lambda_1} \\ &\quad + \omega u_1^{i-1}(t)(\lambda_2 t - \frac{\lambda_2}{\lambda_1}) + \omega u_2^{i-1}(t)(-\frac{\lambda_2^2}{\lambda_1}t + \frac{\lambda_2^2}{\lambda_1^2}) \,, \\ u_2^i(t) &= \omega u_1^{i-1}(t)\lambda_1 t - \omega u_2^{i-1}(t)\lambda_2 t + u_{20}, \end{aligned}$$

and

$$\begin{aligned} u_1^{i+1}(t) &= -\omega u_1^i(t)\lambda_1 t + \omega u_2^i(t)\lambda_2 t + u_{10}, \\ u_2^{i+1}(t) &= u_{20}\exp(-\lambda_2 t) + u_{10}\frac{\lambda_1}{\lambda_2} \\ &\quad + \omega u_1^i(t)(\lambda_1 t - \frac{\lambda_1}{\lambda_2}) + \omega u_2^i(t)(-\frac{\lambda_1^2}{\lambda_2}t - \frac{\lambda_1^2}{\lambda_2^2}) \,, \end{aligned}$$

We compute with our given scheme and the numerical results are presented in Table 8.

A variation of this method is to perform a specific number of iterations for various values of ω, which until now was considered as fixed. In our application, we performed 3 iterations: for the first iteration we have chosen $\omega = 0$, for the second $\omega = 0.5$ and for the third $\omega = 1$. The results are presented in Table 4.

Number of time-partitions	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	1.0020344744	0.7799175652	2.034474×10^{-3}	2.332271×10^{-2}
5	1.0018440081	0.7987533072	1.844008×10^{-3}	4.486969×10^{-3}
10	1.0005095389	0.8009283936	5.095389×10^{-4}	2.311883×10^{-3}
100	1.0000055350	0.8030208356	5.534966×10^{-6}	2.194407×10^{-4}

Table 4. Numerical results for the first example with the weighted splitting method, for 3 iterations with omega=0,0.5,1.

5.3 Weighted splitting method with pre-stepping

A variation of the traditional weighted splitting method described above, is to combine it with the A-B splitting, so we have again a pre-stepping method. The first idea is to apply on the beginning the A-B splitting and then perform a number of iterations (4 iterations for our example) of the weighted splitting method. The second idea works like the first one, but after the 4 iterations we apply again the idea of the A-B splitting and then again 4 iterations of the weighted splitting method. The second application of A-B splitting is done as follows: we set as initial approximation for the weighted method the value

$$\tilde{u} = \frac{u_{AB} + u_i}{2},$$

where u_{AB} is the result of the A-B splitting and u_i the approximation after the first 4 iterations with the weighted splitting.

The results of these variations are presented in Tables 5 and 6, respectively.

In the following figures we attempt depict graphically the results of our work on the specific example described in Section 5. Figure 1 shows the behavior of the error for the solution u_1 as a function of the number of time partitions, for the five methods presented in this article. We see clearly that the A-B splitting method gives a significantly larger error, while the error of the other methods is very small, especially if we have 10 or more time partitions. Figure 2 is a zoomed picture of Figure 1 and shows clearly that the fifth method (the variation of the weighted splitting method) provides the lowest error for this example, while the other three methods give still a relative small error of approximately the same order. Figure 3 shows the behavior of the error for the solution u_1 as a function

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	0	1.2211992169	0.8467828848	2.211992×10^{-1}	1.105996×10^{-1}
1	4	0.9322267543	0.7755099753	6.777325×10^{-2}	2.773030×10^{-2}
5	0	1.1847412811	0.8285539169	1.847413×10^{-1}	9.237064×10^{-2}
5	4	0.9972633758	0.7980951300	2.736624×10^{-3}	5.145146×10^{-3}
10	0	1.1802926209	0.8263295868	1.802926×10^{-1}	9.014631×10^{-2}
10	4	0.9993111141	0.8008502014	6.888859×10^{-4}	2.390075×10^{-3}
100	0	1.1763176930	0.8243421229	1.763177×10^{-1}	8.815885×10^{-2}
100	4	0.9999930858	0.8030418451	6.914230×10^{-6}	1.984313×10^{-4}

Table 5. Numerical results for the first example with the pre-stepped weighted splitting method: first iteration with A-B splitting (denoted with 0 iterative steps) and then 4 iterative steps with the weighted method, $\omega = 0.9$.

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	0	1.2211992169	0.8467828848	2.211992×10^{-1}	1.105996×10^{-1}
1	4	0.9322267543	0.7755099753	6.777325×10^{-2}	2.773030×10^{-2}
1	0	0.5762400752	0.8110953742	4.237599×10^{-1}	7.491210×10^{-2}
1	4	0.9369903968	0.7760242631	6.300960×10^{-2}	2.721601×10^{-2}
5	0	1.1847412811	0.8285539169	1.847413×10^{-1}	9.237064×10^{-2}
5	4	0.9972633758	0.7980951300	2.736624×10^{-3}	5.145146×10^{-3}
5	0	0.5909917577	0.8133221481	4.090082×10^{-1}	7.713887×10^{-2}
5	4	0.9972633728	0.7980957754	2.736627×10^{-3}	5.144501×10^{-3}
10	0	1.1802926209	0.8263295868	1.802926×10^{-1}	9.014631×10^{-2}
10	4	0.9993111141	0.8008502014	6.888859×10^{-4}	2.390075×10^{-3}
10	0	0.5898011477	0.8135897061	4.101989×10^{-1}	7.740643×10^{-2}
10	4	0.9993111141	0.8008502216	6.888859×10^{-4}	2.390055×10^{-3}
100	0	1.1763176930	0.8243421229	1.763177×10^{-1}	8.815885×10^{-2}
100	4	0.9999930858	0.8030418451	6.914230×10^{-6}	1.984313×10^{-4}
100	0	0.5881553893	0.8136919840	4.118446×10^{-1}	7.750871×10^{-2}
100	4	0.9999930858	0.8030418451	6.914230×10^{-6}	1.984313×10^{-4}

Table 6. Numerical results for the first example with the pre-stepped weighted splitting method: first iteration with A-B splitting (denoted with 0 iterative steps) and then 4 iterative steps with the weighted method, $\omega = 0.9$, then again an A-B approximation and 4 iterations with the weighted method afterwards.

of the number of time partitions, for various numbers of iterations and confirms that, even if we have a small number of iterations, we obtain a good result, which means actually an error of small order. Figure 4 is the same as Figure 1, but the results are taken from application of the five methods on a stiff problem with $\lambda_1 = 1000$, $\lambda_2 = 0.5$. We observe that for a number of 5 time partitions or more, we have a relatively small error and the variation of the first weighted splitting method gives generally the smallest error. Figure 5 is the same as Figure 1, but the data are taken from the results for the solution u_2. In this case we

see again that the A-B splitting method gives a significantly larger error, but also the iterative and the pre-stepping splitting method give relative high errors. The best methods for this solution are the weighted methods, and especially the weighted method with $\omega = 0.5$ fixed and 4 iterations.

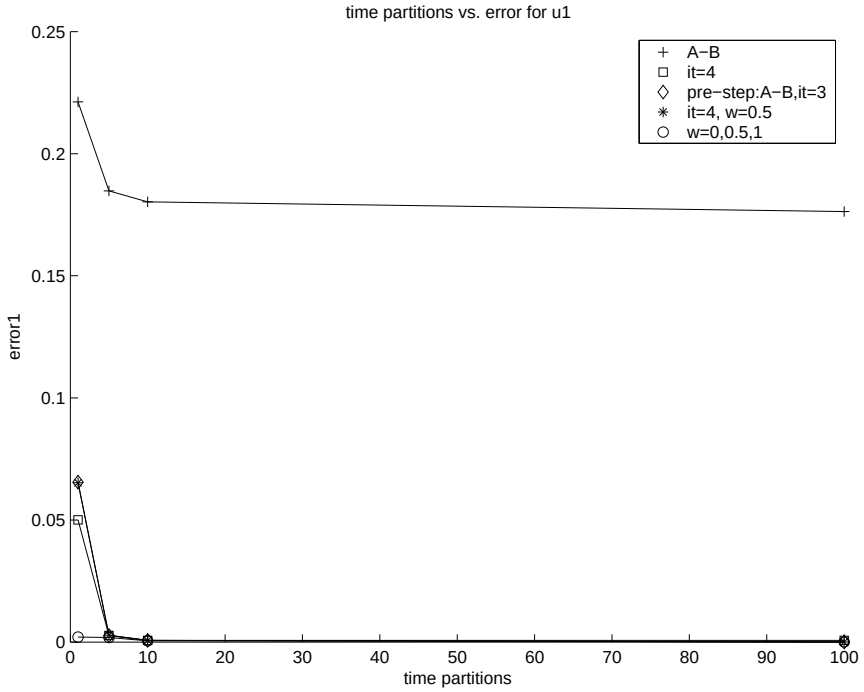

Fig. 1.

5.4 Second Example

We deal with a first order partial differential-equation given as a transport eqaution in the following:

$$\partial_t u_1 = -v_1 \partial_x u_1 - \lambda u_1 \ , \tag{33}$$

$$\partial_t u_2 = -v_2 \partial_x u_2 + \lambda u_1 \ , \tag{34}$$

$$u_1(x,0) = 1 \ , \text{for } 0.1 \leq x \leq 0.2 \ , \tag{35}$$

$$u_1(x,0) = 0 \ , \text{for else} \ ,$$

$$u_2(x,0) = 0 \ , \text{for } x \in [0,1] \ , \tag{36}$$

where $\lambda \in \mathbb{R}^+$ and $v \in \mathbb{R}^+$. We have the time-interval $t \in [0,T]$ and the space-interval $x \in [0,X]$.

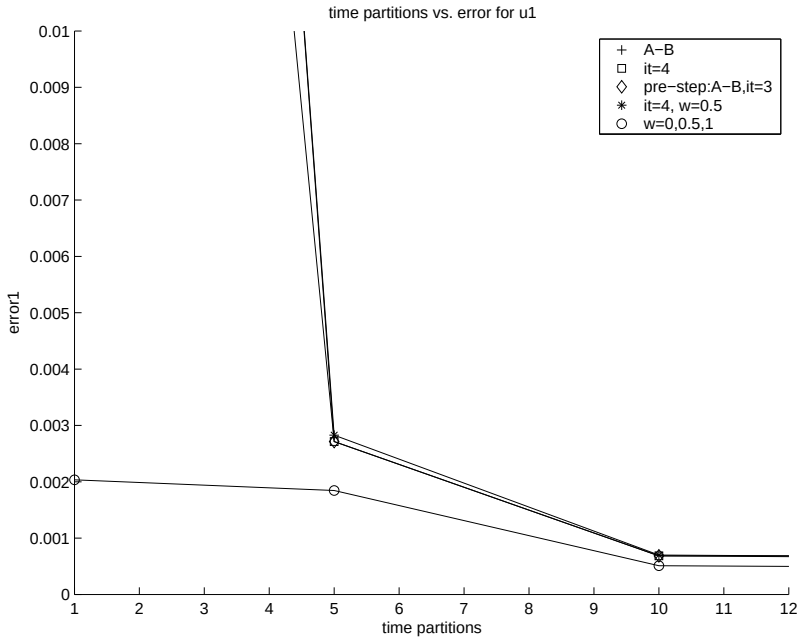

Fig. 2.

We rewrite the equation-system (47)–(50) in operator notation, and end up with the following equations :

$$\partial_t u = Au + Bu \ , \tag{37}$$

$$u_1(x,0) = 1 \ , \text{for } 0.1 \leq x \leq 0.2 \ , \tag{38}$$

$$u_1(x,0) = 0 \ , \text{for else} \ , \tag{39}$$

where $u = (u_1, u_2)$

Our spitted operators are

$$A = \begin{pmatrix} -v_1\partial_x & 0 \\ 0 & -v_2\partial_x \end{pmatrix} \ , \ B = \begin{pmatrix} -\lambda_1 & 0 \\ \lambda_1 & 0 \end{pmatrix} \ . \tag{40}$$

We use the finite difference method as spatial discretisation method and solve the time-discretisation analytically.

The spatial discretisation is done as follows, we concentrate us to an interval $x \in [0, 1.5]$ and $\Delta x = 0.1$.

For the transport-term we use an upwind finite difference discretisation given as :

$$\partial_x u_i = \frac{u_i - u_{i-1}}{\Delta x} \ . \tag{41}$$

We use for the initial-values are given as impulses as :

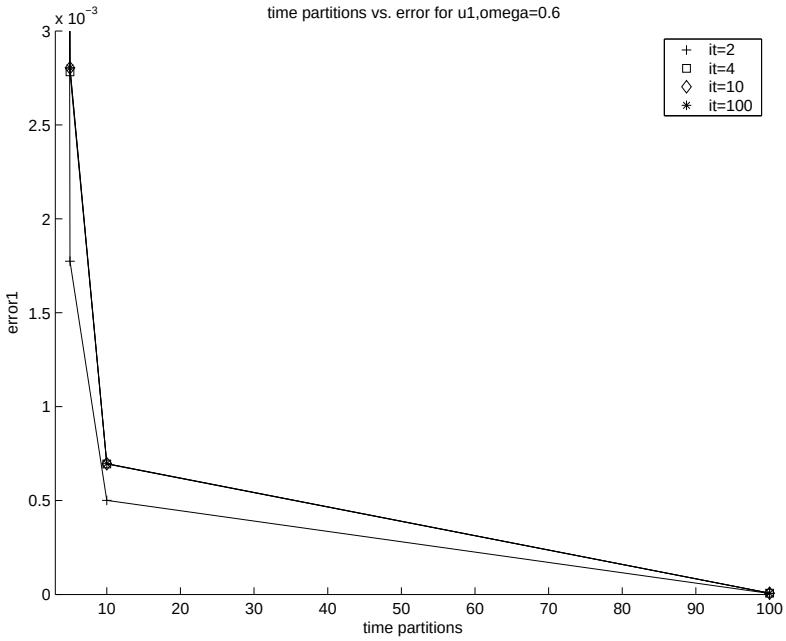
time partitions vs. error for u1,omega=0.6
x 10^-3
error1
time partitions
it=2
it=4
it=10
it=100

Fig. 3.

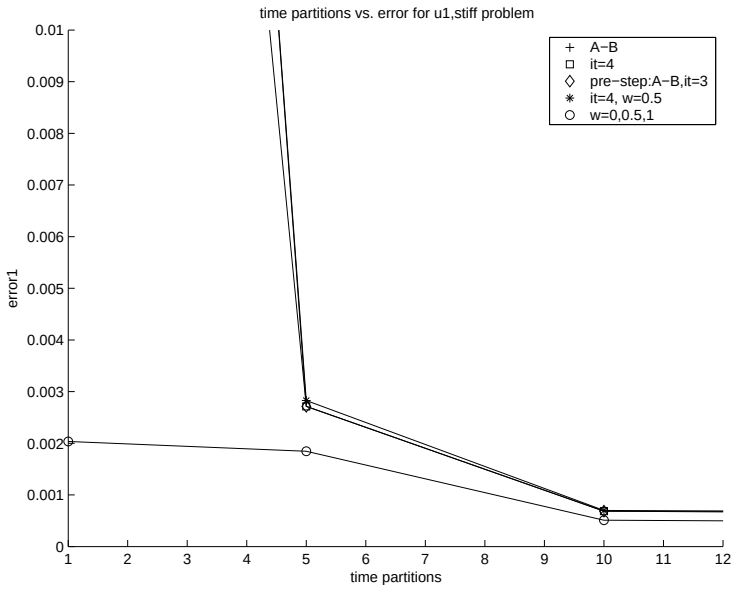
time partitions vs. error for u1,stiff problem
error1
time partitions
A−B
it=4
pre−step:A−B,it=3
it=4, w=0.5
w=0,0.5,1

Fig. 4.

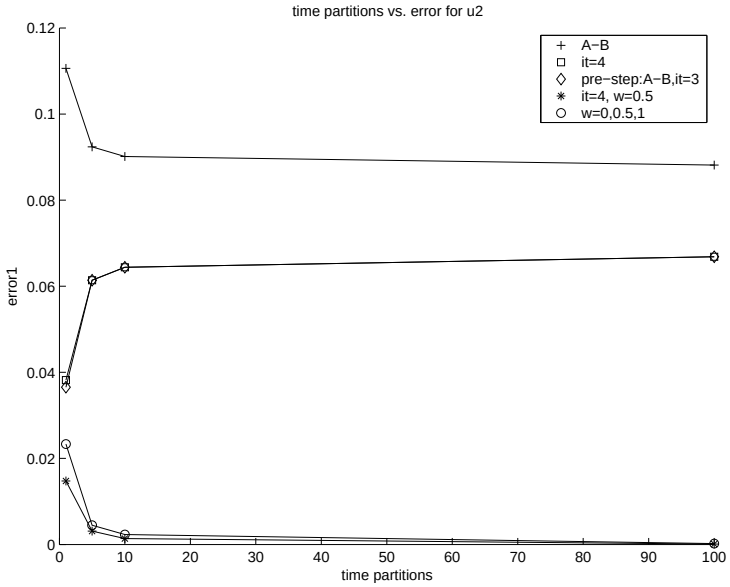

Fig. 5.

$$u_1(x) = \begin{cases} 1 \ , \ 0.1 \leq x \leq 0.3 \\ 0 \ , \ \text{else} \end{cases} \ . \tag{42}$$

and

$$u_2(x) = 0 \ , \ x \in [0, 1.5] \tag{43}$$

For the iterative operator-splitting method and the application to our transport-equaiton we deal for the discretised equation with two indices. The index i is the index for the spatial discretisation and the index j is for the iteration-steps.

We first solve all the equations with the index i, that means all 15 equation for each point. Then we did our iterative steps and we have the first time-step. We are finished for 1 partition and we repeat this 4 times more for the computations of 5 partitions.

In the following equation we write the discretisation in space and the iterative operator splitting algorithm. The time-discretisation is solved analytically.

We have the following algorithm :

$$\partial_t u_{1,i,j} = -v_1/\Delta x(u_{1,i,j} - u_{1,i-1,j}) - \lambda u_{1,i,j-1} \ , \tag{44}$$
$$\partial_t u_{2,i,j} = -v_2/\Delta x(u_{2,i,j} - u_{2,i-1,j}) + \lambda u_{1,i,j-1} \ , \tag{45}$$
$$\partial_t u_{1,i,j+1} = -v_1/\Delta x(u_{1,i,j} - u_{1,i-1,j}) - \lambda u_{1,i,j+1} \ , \tag{46}$$
$$\partial_t u_{2,i,j+1} = -v_2/\Delta x(u_{2,i,j} - u_{2,i-1,j}) + \lambda u_{1,i,j+1} \ , \tag{47}$$
$$u_{1,i,j}(t^n) = 1 \ , \text{for } i = 1, 2, 3, \tag{48}$$
$$u_{1,i,j}(t^n) = 0 \ , \text{else}, \tag{49}$$

$$u_{2,i,j}(t^n) = 0 \text{ , for } i = 0, \ldots, 15 \text{ ,} \tag{50}$$

where $\lambda = 0.5$ and $v_1 = 0.5$ and $v_2 = 1.0$.

For the time-interval we use $t \in [0, 1]$, we deal with 1 and 5 partitions.

The analytical solution of the equation-system (50)–(**??**) with the given values of parameters is

$$u_1(x,t) = \begin{cases} exp(-\lambda t) & \text{, for } 0.1 + v_1 t \leq x \leq 0.3 + v_1 t \text{ ,} \\ 0 & \text{, otherwise} \end{cases}$$

and

$$u_2(x,t) = \lambda(L_{1,2} + L_{2,2} + M_{12,2})$$

$$L_{1,2} = \begin{cases} -\frac{1}{\lambda} exp(-\lambda t) & \text{, for } 0.1 + v_1 t \leq x \leq 0.3 + v_1 t \text{ ,} \\ 0 & \text{, otherwise} \end{cases}$$

$$L_{2,2} = \begin{cases} \frac{1}{\lambda} & \text{, for } 0.1 + v_2 t \leq x \leq 0.3 + v_2 t \text{ ,} \\ 0 & \text{, otherwise} \end{cases}$$

$$M_{12,2} = \begin{cases} \frac{1}{\lambda} exp(-\lambda t) & \text{, for } 0.1 + v_1 t \leq x \leq 0.1 + v_2 t \text{ ,} \\ -\frac{1}{\lambda} exp(-\lambda t) exp(-(\frac{\lambda}{v_1 - v_2})(x - v_1 t - 0.3)) & \text{, for } 0.3 + v_1 t \leq x \leq 0.3 + v_2 t \text{ ,} \\ 0 & \text{, otherwise} \end{cases}$$

So, for the end-time $t_{end} = 1$, we check the results for the 2 end-points : $x_1 = v_1 t + 0.3$ and $x_2 = v_2 t + 0.3$. We get the exact solution of our equation:

$$u_1(x_1, t_{end}) = 0.60653 \text{ , } u_2(x_1, t_{end}) = 0$$

$$u_1(x_2, t_{end}) = 0 \text{ , } u_2(x_2, t_{end}) = 0.632105$$

For the steps j and $j + 1$, which are now actually ODE's, we can derive analytical solutions and apply them to our numerical scheme. The analytical solutions are given as

Christos please reset the exact solutions

$$u_{1,i,j} = u_{1,i-1,j} - \frac{\lambda \Delta x}{v_1} u_{1,i,j-1} + c_2 \exp(-v_1 t / \Delta x)$$
$$u_{2,i,j} = u_{2,i-1,j} + \frac{\lambda \Delta x}{v_1} u_{1,i,j-1} + c_1 \exp(-v_2 t / \Delta x),$$

and

$$u_{1,i,j+1} = \frac{v_1}{\lambda \Delta x}(u_{1,i-1,j} - u_{1,i,j}) + c_2 \exp(-\lambda t)$$
$$u_{2,i,j+1} = \frac{v_2}{\Delta x}(u_{2,i-1,j} - u_{2,i,j})t + \frac{v_1}{\Delta x}(u_{1,i-1,j} - u_{1,i,j})t + c_2 \exp(-\lambda t),$$

where c_1 and c_2 correspond to the initial values.

with the discretised

We have the following algorithm :

$$u_{1,i,j}(t^{n+1}) = (1 + \frac{\tau\ v_1}{\Delta x})^{-1}(u_{1,i,j}(t^n) + \frac{\tau v_1}{\Delta x}\ u_{1,i-1,j}(t^{n+1}) - \tau\ \lambda u_{1,i,j-1}(t^{n+1}))\ ,$$
$$u_{2,i,j}(t^{n+1}) = (1 + \frac{\tau\ v_2}{\Delta x})^{-1}(u_{2,i,j}(t^n) + \frac{\tau v_2}{\Delta x}\ u_{2,i-1,j}(t^{n+1}) + \tau\ \lambda u_{1,i,j-1}(t^{n+1}))\ ,$$
$$u_{1,i,j+1}(t^{n+1}) = (1 + \tau\lambda)^{-1}(u_{1,i,j+1}(t^n) - \frac{\tau v_1}{\Delta x}(u_{1,i,j}(t^{n+1}) - u_{1,i-1,j}(t^{n+1})))\ ,$$
$$u_{2,i,j+1}(t^{n+1}) = u_{2,i,j+1}(t^n) - \frac{\tau v_2}{\Delta x}(u_{2,i,j}(t^{n+1}) - u_{2,i-1,j}(t^{n+1})) + \tau\lambda u_{1,i,j+1}(t^{n+1})\ ,$$

$j = 1, 3, 5, \ldots$.
$u_{1,i,0}(t^{n+1}) = 1$, for $i = 1, 2, 3$,
$u_{1,i,0}(t^{n+1}) = 0$, else,
$u_{2,i,0}(t^{n+1}) = 0$, for $i = 0, \ldots, 15$,
$u_{1,0,j}(t^{n+1}) = 0$,
$u_{2,0,j}(t^{n+1}) = 0$,

In order to implement the algorithm on the computer, we tried to work similarly to the ODE example. In the implementation of the ODE example we used in our computer programm a vector a, in which we stored for every time partition the values of all the appearing u_1^i during the iterations. Similarly, vector b was used for u_2^i. Precisely, vector a was $[u_{10}\ \ u_1^{-1}(0)\ \ u_1^0\ \ u_1^1\ \ u_1^2 \ldots u_1^{iter}]$, where the first two coordinates are the initial values, which for our example were 1 and 0 respectively, and the rest of the coordinates are the solutions calculated during all the iterations. (total number of iterations=iter)

Now, in the case of a PDE we have two dimensions, so it makes sense to use a matrix A instead of a vector. Supposing we have a total number of iterations = iter and since we have 16 points in our spatial partition, the matrix A will be of the following form:

$$A = \left[\begin{array}{c|cccc} & & & & \\ * & u_{1,0,0} & u_{1,1,0} & \ldots & u_{1,15,0} \\ \hline u_{1,-1,1} & u_{1,0,1} & u_{1,1,1} & \ldots & u_{1,15,1} \\ u_{1,-1,2} & u_{1,0,2} & u_{1,1,2} & \ldots & u_{1,15,2} \\ \vdots & \vdots & & & \\ u_{1,-1,iter} & u_{1,0,iter} & u_{1,1,iter} & \ldots & u_{1,15,iter} \end{array}\right],$$

where the element $*$ does not play any role. The first row represents the given initial values for the 16 points of the partition (they are 0 or 1, according to x)

and they correspond to the initial values u_{10} and u_{20} in the ODE example. The first column also contains initial values, which correspond to the value $u_1^{-1}(0)$ in the case of the ODE, and they are equal to 0.

Considering the case of 1 time partition (timep=1), we have:
For $j = 1,\ i = 0$, the programm calculates $u_{1,0,1}$ using the initial values $u_{1,-1,1}$ and $u_{1,0,0}$.
For $j = 1,\ i = 1$, the programm calculates $u_{1,1,1}$ using the initial value $u_{1,1,0}$ and the previously calculated value $u_{1,0,1}$.
For $j = 1,\ i = 2$, the programm calculates $u_{1,2,1}$ using the initial value $u_{1,2,0}$ and the previously calculated value $u_{1,2,1}$. etc.

Similarly we obtain the values $u_{1,0,2}$, $u_{1,1,2}$, $u_{1,2,2}, \ldots$
For $j = 3,\ i = 0$, the programm calculates $u_{1,0,3}$ using the initial values $u_{1,-1,3}$ and the previously calculated value$u_{1,0,0}$.
For $j = 3,\ i = 1$, the programm calculates $u_{1,1,3}$ using the previously calculated values $u_{1,0,3}$ and $u_{1,1,2}$. etc

In table 8 we give the approximations and errors for the exact solutions at the end-time $t = 1$ and end-point $x = v_2 t + 0.3 = 1.3$.

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2
1	4	0.0000000000	0.0000000000	6.065307×10^{-1}	0
1	10	0.0000000000	0.0000000000	6.065307×10^{-1}	0
1	50	0.0000000000	0.0000000000	6.065307×10^{-1}	0
5	4	0.0000000000	0.0000000000	6.065307×10^{-1}	0
5	10	0.0000000000	0.0000000000	6.065307×10^{-1}	0
5	50	0.0000000000	0.0000000000	6.065307×10^{-1}	0

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

Please start with the third example as a complex nonlinear example Here it is enough to have only the numerical results in the table, we could show that we get better results for the more iteration or more partitions

5.5 Third test-example of an ODE

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

We deal with the following nonlinear equation :

$$\partial_t u_1 = -\lambda_1 u_1 + \lambda_2 u_2^2 \,, \tag{51}$$
$$\partial_t u_2 = \lambda_1 u_1 - \lambda_2 u_2^2 \,, \tag{52}$$

$$u_1(0) = 1 \ , \ u_2(0) = 1 \text{ (initial conditions)} \ , \tag{53}$$

where $\lambda_1 = 0.04$ and $\lambda_2 = 1\ 10^4$ are the decay factors. We have the time-interval $t \in [0, 1]$.

We rewrite the equation-system (51)–(53) in operator notation, and end up with the following equations :

$$\partial_t u = A(u) + B(u) \ , \tag{54}$$
$$u(0) = (u_{10}, u_{20})^T, \tag{55}$$

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

Our spitted operators are

$$A(u) = \begin{pmatrix} -\lambda_1 u_1 & 0 \\ \lambda u_1 & 0 \end{pmatrix} \ , \ B(u) = \begin{pmatrix} 0 & \lambda_2 u_2^2 \\ 0 & -\lambda_2 u_2^2 \end{pmatrix} \ . \tag{56}$$

We apply our iterative operator-splitting method and get :

Step i

$$\begin{aligned} \partial_t u_{1,i} &= -\lambda_1 u_{1,i} + \lambda_2 u_{2,i-1}^2 \ , \\ \partial_t u_{2,i} &= \lambda_1 u_{1,i} - \lambda_2 u_{2,i-1}^2 \ , \\ u_1^i(0) &= 1 \ , \ u_2^i(0) = 1 \ , \end{aligned}$$

Step $i + 1$

$$\begin{aligned} \partial_t u_{1,i+1} &= -\lambda_1 u_{1,i} + \lambda_2 u_{2,i+1}^2 \ , \\ \partial_t u_{2,i+1} &= \lambda_1 u_{1,i} - \lambda_2 u_{2,i+1}^2 \ , \\ u_1^{i+1}(0) &= 1 \ , \ u_2^{i+1}(0) = 1 \ , \end{aligned}$$

where $t \in [0, \Delta t]$.

We implement the discretised formular (implicit Euler) :

Step i

$$\begin{aligned} u_{1,i}(t^{n+1}) &= (1 + \lambda_1 \tau)^{-1} (u_{1,i}(t^n) + \tau \lambda_2 u_{2,i-1}^2(t^{n+1})) \ , \\ u_{2,i}(t^{n+1}) &= u_{2,i}(t^n) + \tau \lambda_1 u_{1,i}(t^{n+1}) - \tau \lambda_2 u_{2,i-1}^2(t^{n+1}) \ , \\ u_{1,i}(0) &= 1 \ , \ u_{2,i}(0) = 1 \ , \ \tau = t^{n+1} - t^n, \\ u_{2,0}^2(t^{n+1}) &= u_{2,0}^2(t^n) \end{aligned}$$

Step $i + 1$

$$\begin{aligned} u_{2,i+1}(t^{n+1}) &= (1 + \tau \lambda_2 u_{2,i+1}(t^n))^{-1} (u_{2,i+1}(t^n + \tau \lambda_1 u_{1,i}(t^{n+1})) \ , \\ u_{1,i+1}(t^{n+1}) &= u_{1,i+1}(t^n) - \tau \lambda_1 u_{1,i}(t^{n+1}) + \tau \lambda_2 u_{2,i}^2(t^{n+1}) \ , \\ u_{1,i+1}(0) &= 1 \ , \ u_{2,i+1}(0) = 1 \ , \ \tau = t^{n+1} - t^n, , \end{aligned}$$

where $t \in [0, 1]$.

6 Conclusions and Discussions

We present the new iterative operator-splitting methods with weighting factors. The mathematical background was the stabilisation of the pure iterative operator-splitting methods. We could obtain stable methods for linear and commutative operators for 2 iterations-steps. Numerically we could test commutative operators and enlarge our examples to noncommutative operators with the same behaviour of stability for many iterative steps. The numerical experiments show stability for examples in convection-reaction examples. In the future we focus us on the development of improved operator-splitting methods with respect to noncommutative and nonlinear operators and applications to stiff nonlinear parabilic equations.

References

1. 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.
2. I. Farago. *Splitting methods for abstract Cauchy problems.* Lect. Notes Comp.Sci. 3401, Springer Verlag, Berlin, 2005, pp. 35-45
3. 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.
4. 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.
5. 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.
6. J. Geiser. *Gekoppelte Diskretisierungsverfahren für Systeme von Konvektions-Dispersions-Diffusions-Reaktionsgleichungen.* Doktor-Arbeit, Universität Heidelberg, 2003.
7. 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.
8. J. Geiser. *Iterative Operator-Splitting methods for Parabolic Differential Equations : Convergence theory.* Humboldt-Preprint, to be submitted, February 2006.
9. W. Hundsdorfer, L. Portero. A Note on Iterated Splitting Schemes. CWI Report MAS-E0404, Amsterdam, Netherlands, 2005.
10. W.H. Hundsdorfer, J. Verwer W. *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, Berlin, (2003).
11. 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.
12. G.I Marchuk. *Some applicatons of splitting-up methods to the solution of problems in mathematical physics.* Aplikace Matematiky, 1 (1968) 103-132.
13. G. Strang. *On the construction and comparision of difference schemes.* SIAM J. Numer. Anal., 5:506–517, 1968.

14. J.,G. Verwer and B. Sportisse. *A note on operator splitting in a stiff linear case.* MAS-R9830, ISSN 1386-3703, 1998.
15. Z. Zlatev. *Computer Treatment of Large Air Pollution Models.* Kluwer Academic Publishers, 1995.

Number of time-partitions	Iterative Steps	$u_{1,num}$	$u_{2,num}$	err_1	err_2	Omega
1	2	1.1126340828	0.8160926567	1.126341×10^{-1}	1.285238×10^{-2}	0.25
1	2	1.1126340828	0.8079617093	1.126341×10^{-1}	4.721433×10^{-3}	0.5
1	2	1.1126340828	0.7998307619	1.126341×10^{-1}	3.409514×10^{-3}	0.75
1	2	1.1126340828	0.7949521935	1.126341×10^{-1}	8.288083×10^{-3}	0.9
1	4	0.9269442310	0.7959711040	7.305577×10^{-2}	7.269172×10^{-3}	0.25
1	4	0.9347359059	0.7884767539	6.526409×10^{-2}	1.476352×10^{-2}	0.5
1	4	0.9424104920	0.7812798004	5.758951×10^{-2}	2.196048×10^{-2}	0.75
1	4	0.9469590410	0.7771004869	5.304096×10^{-2}	2.613979×10^{-2}	0.9
1	10	0.9099112376	0.7939361815	9.008876×10^{-2}	9.304095×10^{-3}	0.25
1	10	0.9182845299	0.7865915672	8.171547×10^{-2}	1.664871×10^{-2}	0.5
1	10	0.9264682761	0.7795217846	7.353172×10^{-2}	2.371849×10^{-2}	0.75
1	10	0.9312910481	0.7754089555	6.870895×10^{-2}	2.783132×10^{-2}	0.9
5	2	1.0016709203	0.8028737927	1.670920×10^{-3}	3.664837×10^{-4}	0.25
5	2	1.0017282892	0.8015806799	1.728289×10^{-3}	1.659596×10^{-3}	0.5
5	2	1.0017855141	0.8002898909	1.785514×10^{-3}	2.950385×10^{-3}	0.75
5	2	1.0018197801	0.7995165304	1.819780×10^{-3}	3.723746×10^{-3}	0.9
5	4	0.9971147643	0.8014244641	2.885236×10^{-3}	1.815812×10^{-3}	0.25
5	4	0.9971720588	0.8001425295	2.827941×10^{-3}	3.097747×10^{-3}	0.5
5	4	0.9972291787	0.7988632567	2.770821×10^{-3}	4.377020×10^{-3}	0.75
5	4	0.9972633672	0.7980969674	2.736633×10^{-3}	5.143309×10^{-3}	0.9
5	10	0.9970936697	0.8014177382	2.906330×10^{-3}	1.822538×10^{-3}	0.25
5	10	0.9971509493	0.8001358573	2.849051×10^{-3}	3.104419×10^{-3}	0.5
5	10	0.9972080546	0.7988566373	2.791945×10^{-3}	4.383639×10^{-3}	0.75
5	10	0.9972422344	0.7980903793	2.757766×10^{-3}	5.149897×10^{-3}	0.9
10	2	1.0004882380	0.8028498596	4.882380×10^{-4}	3.904168×10^{-4}	0.25
10	2	1.0004953345	0.8022189267	4.953345×10^{-4}	1.021350×10^{-3}	0.5
10	2	1.0005024217	0.8015883188	5.024217×10^{-4}	1.651958×10^{-3}	0.75
10	2	1.0005066695	0.8012101100	5.066695×10^{-4}	2.030166×10^{-3}	0.9
10	4	0.9992926895	0.8024866900	7.073105×10^{-4}	7.535864×10^{-4}	0.25
10	4	0.9992997841	0.8018570159	7.002159×10^{-4}	1.383260×10^{-3}	0.5
10	4	0.9993111140	0.8008502586	6.888860×10^{-4}	2.390018×10^{-3}	0.9
10	10	0.9992912507	0.8024862527	7.087493×10^{-4}	7.540236×10^{-4}	0.25
10	10	0.9992983450	0.8018565802	7.016550×10^{-4}	1.383696×10^{-3}	0.5
10	10	0.9993054290	0.8012272545	6.945710×10^{-4}	2.013022×10^{-3}	0.75
10	10	0.9993096746	0.8008498253	6.903254×10^{-4}	2.390451×10^{-3}	0.9
100	2	1.0000055138	0.8032058078	5.513824×10^{-6}	3.446860×10^{-5}	0.25
100	2	1.0000055209	0.8031441595	5.520872×10^{-6}	9.611682×10^{-5}	0.5
100	2	1.0000055279	0.8030825117	5.527920×10^{-6}	1.577647×10^{-4}	0.75
100	2	1.0000055321	0.8030455231	5.532148×10^{-6}	1.947532×10^{-4}	0.9
100	4	0.9999930674	0.8032021267	6.932554×10^{-6}	3.814965×10^{-5}	0.25
100	4	0.9999930745	0.8031404797	6.925506×10^{-6}	9.979671×10^{-5}	0.5
100	4	0.9999930815	0.8030788330	6.918458×10^{-6}	1.614434×10^{-4}	0.75
100	4	0.9999930858	0.8030418451	6.914230×10^{-6}	1.984313×10^{-4}	0.9
100	10	0.9999930673	0.8032021267	6.932709×10^{-6}	3.814970×10^{-5}	0.25
100	10	0.9999930743	0.8031404796	6.925661×10^{-6}	9.979676×10^{-5}	0.5
100	10	0.9999930814	0.8030788329	6.918613×10^{-6}	1.614435×10^{-4}	0.75
100	10	0.9999930856	0.8030418451	6.914385×10^{-6}	1.984313×10^{-4}	0.9

Table 8. Numerical results for the first example with the weighted splitting method, for various values of omega fixed.