Iterative operator-splitting methods for nonlinear differential equations and applications of deposition processes.

Jürgen Geiser * Lena Noack †

April 24, 2008

Abstract

In this article we consider iterative operator-splitting methods for nonlinear differential equations. The main feature of the proposed idea is the embedding of Newton's method for solving the split parts of the nonlinear equation at each step. The convergence properties of such a mixed method are studied and demonstrated. We confirm with numerical applications the effectiveness of the proposed scheme in comparison with the standard operator-splitting methods by providing improved results and convergence rates. We apply our results to deposition processes.

Keyword numerical analysis, operator-splitting method, initial value problems, iterative solver method, stability analysis, convection-diffusion-reaction equation.

AMS subject classifications. 35J60, 35J65, 65M99, 65N12, 65Z05, 74S10, 76R50.

1 Introduction

Our study is motivated by complex models with coupled processes, e.g. transport and reaction equations with nonlinear parameters. The ideas for these models came from the simulation of heat transport in an engineering apparatus, e.g. crystal growth, cf. [13], or the simulation of chemical reaction and transport, e.g. in bio-remediation or waste disposals, cf. [11]. In the past many software tools have been developed for multi-dimensional and multi-physical problems, e.g. for the multi-dimensional transport reaction based on different PDE and ODE

*Department of Mathematics, Humboldt Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany, E-mail: geiser@mathematik.hu-berlin.de
†Department of Mathematics, Humboldt Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany, E-mail: noackl@math.hu-berlin.de
solvers. In the future a coupling between various software tools with different solver methods will be of interest and could be done using the fractional splitting method.

The paper is organized as follows. A mathematical model based on the nonlinear convection-diffusion equation is introduced in Section 2. The iterative splitting method for the differential equation is given in Section 3. The error analysis is discussed in Section 4. We introduce the numerical results in Section 5. Finally we discuss our future works in the area of splitting and decomposition methods.

2 Mathematical model

When gas transport is physically more complex because of combined flows in three dimensions, the fundamental equations of fluid dynamics become the starting points of the analysis. For our models with small Knudsen numbers, we can assume a continuum flow, and the fluid equations can be treated with a Navier-Stokes or especially with a convection-diffusion equation.

Three basic equations describe the conservation of mass, momentum, and energy, that are sufficient to describe the gas transport in the reactors, see [26].

1. Continuity: The conservation of mass requires the net rate of the mass accumulation in a region to be equal to the difference between the inflow and outflow rate.

2. Navier-Stokes: Momentum conservation requires the net rate of momentum accumulation in a region to be equal to the difference between the in and out rate of the momentum, plus the sum of the forces acting on the system.

3. Energy: The rate of accumulation of internal and kinetic energy in a region is equal to the net rate of internal and kinetic energy in by convection, plus the net rate of heat flow by conduction, minus the rate of work done by the fluid.

We will concentrate on the conservation of mass and assume that the energy and momentum is conserved, see [14]. Therefore the continuum flow can be described as convection-diffusion equation given as:

\[ \frac{\partial c}{\partial t} + \nabla F - R_g = 0, \text{ in } \Omega \times [0,T] \]  

\[ F = -D \nabla c, \]  

\[ c(x, t) = c_0(x), \text{ on } \Omega, \]  

\[ c(x, t) = c_1(x, t), \text{ on } \partial \Omega \times [0,T], \]  

where \( c \) is the molar concentration and \( F \) the flux of the species. \( D \) is the diffusivity matrix and \( R_g \) is the reaction term. The initial value is given as \( c_0 \) and we assume a Dirichlet boundary with the function \( c_1(x, t) \) being sufficient smooth.
3 The iterative splitting method

The previously defined sequential operator-splitting methods have several drawbacks besides their benefits. For instance, for non-commuting operators there might be a very large constant in the splitting error which requires the use of an unrealistically small time step. Also, splitting the original problem into the different subproblems with one operator, i.e. neglecting the other components, is physically questionable.

In order to avoid these problems, one can use the iterative operator-splitting method on an interval $[0, T]$. This algorithm is based on the iteration with fixed splitting discretization step size $\tau$. On every time interval $[t^n, t^{n+1}]$ the method solves the following subproblems consecutively for $i = 0, 2, \ldots, 2m$.

\[
\frac{\partial u_i(x, t)}{\partial t} = Au_i(x, t) + Bu_{i-1}(x, t), \quad \text{with } u_i(t^n) = u^n
\]

(4)

$u_0(x, t^n) = u^n$, $u_{-1} = 0$,

and $u_i(x, t) = u_{i-1}(x, t) = u_1$, on $\partial \Omega \times (0, T)$,

\[
\frac{\partial u_{i+1}(x, t)}{\partial t} = Au_i(x, t) + Bu_{i+1}(x, t),
\]

(5)

with $u_{i+1}(x, t^n) = u^n$,

and $u_i(x, t) = u_{i-1}(x, t) = u_1$, on $\partial \Omega \times (0, T)$,

where $u^n$ is the known split approximation at time level $t = t^n$ (see [8]). This algorithm constitutes an iterative method which involves in each step both operators $A$ and $B$. Hence, there is no real separation of the different physical processes in these equations.

3.1 Iterative operator-splitting method as fixed-point scheme

The iterative operator-splitting method is used as a fixed-point scheme to linearize the nonlinear operators, see [12] and [17].

We concentrate again on nonlinear differential equations of the form

\[
\frac{du}{dt} = A(u(t))u(t) + B(u(t))u(t), \quad \text{with } u(t^n) = u^n,
\]

(6)

where $A(u), B(u)$ are matrices with nonlinear entries and densely defined, where we assume that the entries involve the spatial derivatives of $c$, see [33]. In the following we discuss the standard iterative operator-splitting method as a fixed-point iteration method to linearize the operators.

We split our nonlinear differential equation (6) by applying

\[
\frac{du_i(t)}{dt} = A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i-1}(t), \quad \text{with } u_i(t^n) = c^n,
\]

(7)

\[
\frac{du_{i+1}(t)}{dt} = A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i+1}(t), \quad \text{with } u_{i+1}(t^n) = c^n,
\]

(8)

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \ldots, 2m + 1$. $u_0(t) = c_n$ is the starting solution, where we assume that the solution $c^{n+1}$ is
near $c^n$, or $u_0(t) = 0$. So we have to solve the local fixed-point problem. $c^n$ is
the known split approximation at time level $t = t^n$.

The split approximation at time level $t = t^{n+1}$ is defined as $c^{n+1} = u_{2m+2}(t^{n+1})$.
We assume that the operators $A(u_{i-1}(t^{n+1}))$, $B(u_{i-1}(t^{n+1}))$ are constant for
$i = 1, 3, \ldots, 2m + 1$. Here the linearization is done with respect to the itera-
tions, such that $A(u_{i-1}(t^{n+1}))$, $B(u_{i-1}(t^{n+1}))$ are at least non-dependent operators in the
iterative equations, and we can apply the linear theory. For the linearization we
assume at least in the first equation $A(u_{i-1}(t)) \approx A(u_i(t))$, and in the second
equation $B(u_{i-1}(t)) \approx B(u_{i+1}(t))$, for small $t$.

We have
$$||A(u_{i-1}(t^{n+1}))-A(u^{n+1})u(t^{n+1})|| \leq \epsilon,$$
for sufficient iterations $i \in \{1, 3, \ldots, 2m + 1\}$.

**Remark 3.1** The linearization with the fixed-point scheme can be used for
smooth or weak nonlinear operators, otherwise we lose the convergence behavior,
while we did not converge to the local fixed point, see [17].

### 3.2 Operator-splitting method with embedded Jacobian

**Newton iterative method**

The Newton’s method is used to solve the nonlinear parts of the iterative
operator-splitting method, see the linearization techniques in [17],[18]. We apply
the iterative operator-splitting method and obtain:

$F_1(u_i) = \partial u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0,$
with $u_i(t^n) = c^n$,

$F_2(u_{i+1}) = \partial u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0,$
with $u_{i+1}(t^n) = c^n$,

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \ldots, 2m + 1$.
$c_0(t) = 0$ is the starting solution and $c^n$ is the known split approximation at
time level $t = t^n$. The results of the methods are $c(t^{n+1}) = u_{2m+2}(t^{n+1})$. The
splitting method with embedded Newton’s method is given as

$u_i^{(k+1)} = u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial u_i^{(k)} - A(u_i^{(k)}))u_i^{(k)} - B(u_{i-1}^{(k)})u_{i-1}^{(k)},$

with $D(F_1(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}}u_i^{(k)}),$

and $k = 0, 1, 2, \ldots, K$, with $u_i(t^n) = c^n$,

$u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D(F_2(u_{i+1}^{(l)}))^{-1}(\partial u_{i+1}^{(l)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i+1}^{(k)})u_{i+1}^{(k)})c_2^{(l)}$, 

with $D(F_2(u_{i+1}^{(l)})) = -(B(u_{i+1}^{(l)}) + \frac{\partial B(u_{i+1}^{(l)})}{\partial u_{i+1}^{(l)}}u_{i+1}^{(l)},$

and $l = 0, 1, 2, \ldots, L$, with $u_{i+1}(t^n) = c^n$. 

Remark 3.2 For the iterative operator-splitting method with Newton’s method we have two iteration procedures. The first iteration is Newton’s method for computing the solution of the nonlinear equations, the second iteration is the iterative splitting method, which computes the resulting solution of the coupled equation systems. The embedded method is used for strong nonlinearities.

3.3 Stabilization of the initial values for the Newton iterative method

To stabilize the initial conditions for the Newton’s method we could apply different ideas:

1.) Apply the implicit value for \( B(u^{k-1}_{i-1})u^{k}_{i-1} \) to stabilize the diagonal of the matrix.

2.) Apply first the linear operator, if one operator is linear.

3.) Apply an iterated prestep for the first value.

ad.1.)

We use a first step to initialize our nonlinear scheme. For the stabilization, we additionally use the \( B \) operator and balance the diagonal entries of the matrices.

For \( i = 0 \):
\[
\begin{align*}
u_0^{(k+1)} &= u_0^{(k)} - D(F_1(u_0^{(k)}))^{-1}(\partial_t u_0^{(k)} - A(u_0^{(k)})u_0^{(k)} - B(u_0^{(k+1)})u_0^{(k+1)}), \\
\text{with } D(F_1(u_0^{(k)})) &= -(A(u_0^{(k)}) + \frac{\partial A(u_0^{(k)})}{\partial u_0^{(k)}} u_0^{(k)}),
\end{align*}
\]
and \( k = 0, 1, 2, \ldots, K \), with \( u_0(t^n) = c^n \).

For \( i \geq 1 \) we have:
\[
\begin{align*}
u_i^{(k+1)} &= u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_i^{(k+1)})u_i^{(k+1)}), \\
\text{with } D(F_1(u_i^{(k)})) &= -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}} u_i^{(k)}),
\end{align*}
\]
and \( k = 0, 1, 2, \ldots, K \), with \( u_i(t^n) = c^n \).

\[
\begin{align*}
u_{i+1}^{(l+1)} &= u_{i+1}^{(l)} - D(F_2(u_{i+1}^{(l)}))^{-1}(\partial_t u_{i+1}^{(l)} - A(u_{i+1}^{(l+1)})u_{i+1}^{(l+1)} - B(u_{i+1}^{(l)})u_{i+1}^{(l+1)}), \\
\text{with } D(F_2(u_{i+1}^{(l)})) &= -(B(u_{i+1}^{(l)}) + \frac{\partial B(u_{i+1}^{(l)})}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}),
\end{align*}
\]
and \( l = 0, 1, 2, \ldots, L, \) with \( u_{i+1}(t^n) = c^n \).

Here we stabilize the Newton’s method with further entries to the diagonals.

ad.2.)

(pre-step)
\[
\begin{align*}
\partial_t u_{i-1} &= A(u_{i-2})u_{i-2} + Bu_{i-1} \\
\text{with } u_{i-1}(t^n) &= c^n.
\end{align*}
\]
\[ u_i^{(k+1)} = u_i^{(k)} - D(F_i(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_i^{(k+1)})u_i^{(k)}), \]

with \( D(F_i(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}}u_i^{(k)}), \)

and \( k = 0, 1, 2, \ldots, K, \) with \( u_i(t^n) = c^n, \)

\[ u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D(F_2(u_{i+1}^{(l)}))^{-1}(\partial_t u_{i+1}^{(l)} - A(u_i^{(k)})u_i^{(k)} - B(u_i^{(k)})u_i^{(k+1)})c_2^{(i)}, \]

with \( D(F_2(u_{i+1}^{(l)})) = -(B(u_{i+1}^{(l)}) + \frac{\partial B(u_{i+1}^{(l)})}{\partial u_i^{(k)}}u_i^{(k)}), \)

and \( l = 0, 1, 2, \ldots, L, \) with \( u_{i+1}(t^n) = c^n. \)

ad3.) with iterative prestep method.

**Remark 3.3** For the iterative operator-splitting method with Newton’s method we have two iteration procedures. The first iteration is Newton’s method for computing the solution of the nonlinear equations, the second iteration is the iterative splitting method, which computes the resulting solution of the coupled equation systems. The embedded method is used for strong nonlinearities.

## 4 Error analysis

In the sequel we demonstrate the error analysis for the linear and nonlinear decomposition methods. In this section we designate as \( e_i(t) := c(t) - c_i(t) \) the error between the exact solution and the approximated solution after \( i \) iterations.

### 4.1 Error analysis of the linear method

We present the convergence and the rate of convergence of method (4)–(5), where \( m \) tends to infinity.

**Theorem 4.1** Let us consider the abstract Cauchy problem in a Banach space \( X \)

\[ \partial_t c(t) = Ac(t) + Bc(t), \quad 0 < t \leq T, \]

\[ c(0) = c_0, \]

where \( A, B, A + B : X \to X \) are given linear bounded operators being generators of a \( C_0 \)-semigroup and \( c_0 \in X \) is a given element.

Then the iteration process (4)–(5) for \( i = 0, 2, \ldots, 2m \) is consistent with order \( O(\tau_n^{2m}). \)

The estimate is given as:

\[ \| e_{i+1} \| = K_1 \tau_n^2 \| e_{i-1} \| + O(\tau_n^4). \]
A proof can be found in [8].
The a priori error expression is given in the following result (see [17]).

**Corollary 4.2** Equation (9) shows that after one more iteration step \((i = 2m+1)\) we have the estimate

\[
\|e_{2m+1}\| = K_m \|e_0\| \tau_n^{2m} + O(\tau_n^{2m+1}),
\]

(10)

where \(c_0(t)\) is the initial guess.

The global error is given in the next Theorem (see [17]).

**Theorem 4.3** We assume the local error of the estimate (9) and a \(k\)-th order discretization method for the time. After \(i = 2m + 1\) iteration steps there holds

\[
\|u(t_n) - u_{2m+1}(t_n)\| = t_n^k K_m \|e_0\| \tau_n^{2m} + t_n^k O(\tau_n^{2m+1}),
\]

(11)

where \(c_0(t)\) is the initial guess.

The proof uses classical operator splitting methods (see [30]).

**Remark 4.4** When \(A\) and \(B\) are matrices, we obtain a system of ordinary differential equations. To estimate the growth of the matrices, we can use the concept of the logarithmic norm and obtain more detailed results, see [16].

**Remark 4.5** We note that a huge class of important differential operators generate a 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.

### 4.2 Error analysis for the nonlinear method

Here we discuss the linearization techniques and their approximations.

#### 4.2.1 Linearization by Iterative Splitting Method

**Theorem 4.6** Let us consider the following problem

\[
\frac{\partial}{\partial t} c(t) = A(c(t))c(t) + B(c(t))c(t), \quad 0 < t \leq T,
\]

\[
c(0) = c_0,
\]

where \(A, B\) are nonlinear differentiable bounded operators \(A, B\) in a Banach space \(X\).

Linearizing the nonlinear operators yields the linearized equation
\[ \partial_t c(t) = \tilde{A}c(t) + \tilde{B}c(t) + R(c_i)c_i, \quad 0 < t \leq T, \]
\[ \tilde{A} = A(c_i) + \frac{\partial A(c_i)}{\partial c}c_i, \quad \tilde{B} = B(c_i) + \frac{\partial B(c_i)}{\partial c}c_i, \quad R(c_i) = \frac{\partial A(c_i)}{\partial c}c_i + \frac{\partial B(c_i)}{\partial c}c_i, \]

where \( \tilde{A}, \tilde{B}, \tilde{A} + \tilde{B} : X \to X \) are given, linear bounded operators being generators of the \( C_0 \)-semigroup and \( c_0 \in X \) is a given element. The linearization is of the form \( A(c)c \approx A(c_i)c_i + (\frac{\partial A(c_i)}{\partial c})_i(c - c_i) \) where \( c_i \in X \) is a linearized solution, we further assume \( (\frac{\partial A(c_i)}{\partial c})_i \) is a constant Jacobian matrix.

We assume that the iteration process (4)–(5) is convergent and the convergence is of second order.

It holds
\[ \| e_i \| = K \tau_n \| e_{i-1} \| + O(\tau_n^2), \]  
where \( K \) is an estimation of the residual \( \| R(\tilde{c}) \| \leq R_{\max} \in \mathbb{R}^+ \) for all \( \tilde{c} \in X \) and \( \| \tilde{B} \| \leq \tilde{K} \).

One could also obtain the result with Lipschitz-constants.

We now prove the argument using the semi-group theory.

**Proof.**

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

The linearized splitting method is given as :
\[ \frac{\partial c_i(t)}{\partial t} = \tilde{A}c_i(t) + \tilde{B}c_{i-1}(t) + R(c_{i-1})c_{i-1}(t), \]
with \( c_i(t^n) = e^n \) \[ c_0(t^n) = e^n, \quad c_{-1} = 0, \]
\[ \frac{\partial c_{i+1}(t)}{\partial t} = \tilde{A}c_i(t) + \tilde{B}c_{i+1}(t) + R(c_{i-1})c_{i-1}(t), \]
with \( c_{i+1}(t^n) = e^n \),

where \( e^n \) is the known split approximation at the time level \( t = t^n \). We solve the subproblems consecutively for \( i = 0, 2, \ldots, 2m \).

For the error function \( e_i(t) = c(t) - c_i(t) \) we have the relations
\[ \partial t e_i(t) = \tilde{A}(e_i(t)) + \tilde{B}(e_i(t)) + R(e_{i-1})c_{i-1}(t), \quad t \in (t^n, t^{n+1}], \]
\[ e_i(t^n) = 0, \]
and
\[ \partial t e_{i+1}(t) = \tilde{A}(e_i(t)) + \tilde{B}(e_{i+1}(t)) + R(e_{i-1})c_{i-1}(t), \quad t \in (t^n, t^{n+1}], \]
\[ e_{i+1}(t^n) = 0, \]
for \( m = 0, 2, 4, \ldots \), with \( c_0(0) = 0 \) and \( c_{-1}(t) = c(t) \) and
\[
\dot{A} = A(e_{i-1}) + \frac{\partial A(e_{i-1})}{\partial c} e_{i-1}, \quad \dot{B} = B(e_{i-1}) + \frac{\partial B(e_{i-1})}{\partial c} e_{i-1},
\]
\[
R(e_{i-1}) = \frac{\partial A(e_{i-1})}{\partial c} e_{i-1} + \frac{\partial B(e_{i-1})}{\partial c} e_{i-1}.
\]

In the following we derive the linearized equations. We use the notation \( X^2 \)
for the product space \( X \times X \) enabled with the norm \( \| (u, v) \| = \max \{ \| u \|, \| v \| \} \)
\((u, v) \in X\). The elements \( E_i(t), F_i(t) \in X^2 \) and the linear operator \( A : X^2 \to X^2 \)
are defined as follows
\[
E_i(t) = \begin{bmatrix} e_i(t) \\ e_{i+1}(t) \end{bmatrix}; \quad A = \begin{bmatrix} \dot{A} & 0 \\ \dot{A} & \dot{B} \end{bmatrix},
\]
\[
F_i(t) = \begin{bmatrix} R(e_{i-1})e_{i-1} + \dot{B}e_{i-1} \\ R(e_{i-1})e_{i-1} \end{bmatrix}.
\]

where have the bounded and linearized operators \( \dot{A}, \dot{B} \) and \( R(e_{i-1}) \).

Using notation (18) and (19), the relations (17)–(18) can be written in the form
\[
\partial_t E_i(t) = AE_i(t) + F_i(t), \quad t \in [t^n, t^{n+1}].
\]
\[
E_i(t^n) = 0.
\]

Due to our assumptions that \( A \) and \( B \) are bounded and differentiable and that
we have a lipschitzian domain, \( A \) is a generator of the one-parameter \( C_0 \) semi-group \( (A(t))_{t \geq 0} \). We also assume the estimate of our term \( F_i(t) \) with the growth conditions.

We can estimate the right hand side \( F_i(t) \) with the help of Lemma 1 presented after this proof. Hence, using the variations of constants formula, the solution of the abstract Cauchy problem (21) with homogeneous initial condition can be written as (cf. e.g. [7])
\[
E_i(t) = \int_{t^n}^t \exp(A(t - s))F_i(s)ds, \quad t \in [t^n, t^{n+1}].
\]

Hence, using the denotation
\[
\| E_i \|_\infty = \sup_{t \in [t^n, t^{n+1}]} \| E_i(t) \|,
\]
and taking into account Lemma 1, we have
\[
\| E_i(t) \|_\infty \leq \| F_i \|_\infty \int_{t^n}^t \| \exp(A(t - s)) \| ds \leq C \| e_{i-1}(t) \| \int_{t^n}^t \| \exp(A(t - s)) \| ds, \quad t \in [t^n, t^{n+1}].
\]

Since \( (A(t))_{t \geq 0} \) is a semigroup, the so called growth estimate is
\[
\| \exp(A(t)) \| \leq K \exp(\omega t), \quad t \geq 0,
\]
with some numbers \( K \geq 0 \) and \( \omega \in \mathbb{R} \) (see [7]).
• Assume that \((A(t))_{t \geq 0}\) is a bounded or exponentially stable semigroup, i.e. that (25) holds with some \(\omega \leq 0\). Then obviously the inequality
\[
\| \exp(A t) \| \leq K; \quad t \geq 0,
\]
holds, and hence from (24), we have
\[
\| E_i(t) \|_\infty \leq K \tau_n \| e_{i-1}(t) \|, \quad t \in (0, \tau_n).
\] (27)

• Assume that \((A(t))_{t \geq 0}\) has exponential growth with some \(\omega > 0\). From (25) we have
\[
\int_{t^n}^{t^{n+1}} \| \exp(A(t-s)) \| ds \leq K_{\omega}(t), \quad t \in [t^n, t^{n+1}],
\]
where
\[
K_{\omega}(t) = \frac{K}{\omega} (\exp(\omega(t-t^n)) - 1), \quad t \in [t^n, t^{n+1}],
\] (28)
and hence
\[
K_{\omega}(t) \leq \frac{K}{\omega} (\exp(\omega \tau_n) - 1) = K \tau_n + O(\tau_n^2),
\] (29)
where \(\tau_n = t^{n+1} - t^n\). The estimations (27) and (30) result in
\[
\| E_i \|_\infty = K \tau_n \| e_{i-1} \| + O(\tau_n^2).
\] (31)
Taking into the account the definition of \(E_i\) and the norm \(\| \cdot \|_\infty\), that results to have the estimation \(\| e_{i+1} \| \leq \| e_i \|\), we obtain
\[
\| e_i \| = K \tau_n \| e_{i-1} \| + O(\tau_n^2),
\] which proves our statement.

Lemma 4.7 The term \(F_i(t)\) given by (20) can be estimated as
\[
\| F_i(t) \| \leq C \| e_{i-1} \|.
\] (32)
where we assume the boundedness of \(R(e_{i-1})\) and \(\tilde{B}\), see Theorem 4.6.

Proof. We have the norm \(\| F_i(t) \| = \max\{ F_{i_1}(t), F_{i_2}(t) \} \).
Each term can be bounded as follows.
\[
\| F_{i_1}(t) \| \leq \| (R(e_{i-1}(t)) + \tilde{B}) e_{i-1}(t) \| \leq (R_{\max} + \tilde{K}) \| e_{i-1}(t) \|,
\] (33)
\[
\| F_{i_2}(t) \| \leq \| R(e_{i-1}(t)) e_{i-1}(t) \| \leq R_{\max} \| e_{i-1}(t) \|.
\] (34)
where \(R_{\max}\) and \(\tilde{K}\) are constants and defined in Theorem 4.6.
So we obtain the estimate
\[
\| F_i(t) \| \leq C \| e_{i-1}(t) \|,
\] where \(C = R_{\max} + \tilde{K}\).
4 ERROR ANALYSIS

4.2.2 Linearization by Newton’s Method

In this approach we use Newton’s method for linearization. Here we have two steps in the proof of the error analysis.

1.) Error of Newton’s method;
2.) Error of the iterative or non-iterative Operator Splitting method.

Theorem 4.8 Consider the problem

\[ \frac{\partial c(t)}{\partial t} = A(c(t)) + B(c(t)), \quad 0 < t \leq T, \]

\[ c(0) = c_0, \]

where \( A, B \) are nonlinear differentiable bounded operators in a Banach space \( X \).

We apply the Newton’s method to solve the nonlinear equations and obtain

\[ u_{i+1}^{(k+1)} = u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_i^{(k)})u_i^{(k)}), \]

with \( D(F_1(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}}u_i^{(k)}), \)

with \( u_i(t^n) = c^n, \)

\[ u_{i+1}^{(k+1)} = u_{i+1}^{(k)} - D(F_2(u_{i+1}^{(k)}))^{-1}(\partial u_{i+1}^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_i^{(k)})u_i^{(k)})c_2^{(k)}, \]

with \( D(F_2(u_{i+1}^{(k)})) = -(B(u_i^{(k)}) + \frac{\partial B(u_i^{(k)})}{\partial u_i^{(k)}}u_i^{(k)}), \)

and \( k = 0, 1, 2, \ldots, \)

with \( u_{i+1}(t^n) = c^n. \)

The iterations are \( i = 0, 2, \ldots, 2m. \) \( c_0(t) = 0 \) is the starting solution and \( c^n \) is the known split approximation at the time level \( t = t^n. \) The result of the schemes is \( c(t^{n+1}) = u_{2m+2}(t^{n+1}). \)

The following inequality holds

\[ \| e_i(t)_{(k+1)} \| \leq K \tau_n^2 \| e_i(t)_{(k)} \|^2, \]

where \( \tau_n = t^{n+1} - t^n, \) \( K \) is a constant and \( k \) is the index for the Newton iteration.

Proof.

The sketch of the proof is outlined in two parts. The first part gives the approximation error of Newton’s method and the second the approximation error of the iterative operator splitting method.

First Part:
The error for Newton’s method can be derived as
\[ \| e_i^{(k+1)} \| \leq K_1 \| e_i^{(k)} \|^2, \] (37)
where \( e_i^{(k+1)} = e_i^{(k)} - c^* \), and \( c^* \) is the exact solution of the nonlinear problem and \( K_1 \) is a constant, see [21].

Second Part:
For the iterative operator splitting method, we obtain the approximation error:
\[ \| e_i^{(k)} \| = K_2 \tau_n \| e_{i-1}^{(k)} \| + \mathcal{O}(\tau_n^2), \] (38)
where \( K_2 \) is an estimation of the residual, see Theorem 4.6, and \( \tau_n = t^{n+1} - t^n \).

We insert the result of equation 1 into the equation 38 and obtain the error of the nonlinear splitting scheme, which is given as:
\[ \| e_i^{(k+1)}(t) \| \leq K \tau_n^2 \| e_{i-1}^{(k)}(t) \|^2, \]
where \( K \) is a combination of the constants \( K_1 \) and \( K_2 \).

\[ \square \]

5 Numerical examples

In the next experiments we deal with nonlinear differential equations. Because of the regularity assumptions to our splitting method we apply 2-4 iteration steps.

5.1 Test example 1: Burgers equation

We deal with a 2D example where we can derive an analytical solution.
\[ \partial_t u = -u \partial_x u - u \partial_y u + \mu (\partial_{xx} u + \partial_{yy} u) + f(x, y, t), \] (39)
\( (x, y, t) \in \Omega \times [0, T] \)
\[ u(x, y, 0) = u_{\text{ana}}(x, y, 0), \ (x, y) \in \Omega \] (40)
with \( u(x, y, t) = u_{\text{ana}}(x, y, t) \) on \( \partial \Omega \times [0, T] \), (41)
where \( \Omega = [0, 1] \times [0, 1], \) \( T = 1.25 \), and \( \mu \) is the viscosity.

The analytical solution is given as
\[ u_{\text{ana}}(x, y, t) = (1 + \exp(\frac{x + y - t}{2\mu}))^{-1}, \] (42)
where \( f(x, y, t) = 0. \)

The operators are given as:
A(u)u = -u\partial_x u - u\partial_y u, \text{ hence } A(u) = -u\partial_x - u\partial_y \text{ (the nonlinear operator)},
Bu = \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t) \text{ (the linear operator)}.

We apply the nonlinear Algorithm 7 to the first equation and obtain
\[ A(u_{i-1})u_i = -u_{i-1}\partial_x u_i - u_{i-1}\partial_y u_i \quad \text{and} \]
\[ Bu_{i-1} = \mu(\partial_{xx} + \partial_{yy})u_{i-1} + f, \]
and we obtain linear operators, because \( u_{i-1} \) is known from the previous time step.

In the second equation we obtain by using Algorithm 8:
\[ A(u_{i-1})u_i = -u_{i-1}\partial_x u_i - u_{i-1}\partial_y u_i \quad \text{and} \]
\[ Bu_{i+1} = \mu(\partial_{xx} + \partial_{yy})u_{i+1} + f, \]
and we have also linear operators.

The maximal error at end time \( t = T \) is given as
\[ \text{err}_{max} = |u_{num} - u_{ana}| = \max_{i=1}^{\rho} |u_{num}(x_i, t) - u_{ana}(x_i, t)|, \]
the numerical convergence rate is given as
\[ \rho = \log(\text{err}_{h/2}/\text{err}_h)/\log(0.5). \]

We have the following results, see Tables 1 and 2, for different steps in time and space and different viscosities.

<table>
<thead>
<tr>
<th>( \Delta x = \Delta y )</th>
<th>( \Delta t )</th>
<th>( \text{err}_{L1} )</th>
<th>( \text{err}_{max} )</th>
<th>( \rho_{L1} )</th>
<th>( \rho_{max} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>1/10</td>
<td>0.0549</td>
<td>0.1867</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>1/10</td>
<td>0.0468</td>
<td>0.1599</td>
<td>0.2303</td>
<td>0.2234</td>
</tr>
<tr>
<td>1/40</td>
<td>1/20</td>
<td>0.0418</td>
<td>0.1431</td>
<td>0.1630</td>
<td>0.1608</td>
</tr>
<tr>
<td>1/10</td>
<td>1/20</td>
<td>0.0447</td>
<td>0.1626</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>1/20</td>
<td>0.0331</td>
<td>0.1215</td>
<td>0.4353</td>
<td>0.4210</td>
</tr>
<tr>
<td>1/40</td>
<td>1/20</td>
<td>0.0262</td>
<td>0.0943</td>
<td>0.3352</td>
<td>0.3645</td>
</tr>
<tr>
<td>1/10</td>
<td>1/40</td>
<td>0.0405</td>
<td>0.1551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>1/40</td>
<td>0.0265</td>
<td>0.1040</td>
<td>0.6108</td>
<td>0.5768</td>
</tr>
<tr>
<td>1/40</td>
<td>1/40</td>
<td>0.0181</td>
<td>0.0695</td>
<td>0.5517</td>
<td>0.5804</td>
</tr>
</tbody>
</table>

Table 1: Numerical results for the Burgers equation with viscosity \( \mu = 0.05 \), initial condition \( u_0(t) = c_n \), and two iterations per time step.

Figure 1 presents the profile of the 2D nonlinear Burgers equation.
Table 2: Numerical results for the Burgers equation with viscosity $\mu = 5$, initial condition $u_0(t) = c_n$, and two iterations per time step.

<table>
<thead>
<tr>
<th>$\Delta x = \Delta y$</th>
<th>$\Delta t$</th>
<th>$\text{err}_{L_1}$</th>
<th>$\text{err}_{\text{max}}$</th>
<th>$\rho_{L_1}$</th>
<th>$\rho_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>1/10</td>
<td>$1.1168 \times 10^{-7}$</td>
<td>$2.4390 \times 10^{-7}$</td>
<td>0.4439</td>
<td>0.5070</td>
</tr>
<tr>
<td>1/20</td>
<td>1/10</td>
<td>$8.2098 \times 10^{-8}$</td>
<td>$1.7163 \times 10^{-7}$</td>
<td>0.3479</td>
<td>0.3614</td>
</tr>
<tr>
<td>1/40</td>
<td>1/10</td>
<td>$6.4506 \times 10^{-8}$</td>
<td>$1.3360 \times 10^{-7}$</td>
<td>0.0383</td>
<td>0.0414</td>
</tr>
<tr>
<td>1/10</td>
<td>1/20</td>
<td>$3.8260 \times 10^{-8}$</td>
<td>$9.0093 \times 10^{-8}$</td>
<td>0.5733</td>
<td>0.6619</td>
</tr>
<tr>
<td>1/20</td>
<td>1/20</td>
<td>$2.5713 \times 10^{-8}$</td>
<td>$5.6943 \times 10^{-8}$</td>
<td>0.4565</td>
<td>0.5088</td>
</tr>
<tr>
<td>1/40</td>
<td>1/20</td>
<td>$1.8738 \times 10^{-8}$</td>
<td>$4.0020 \times 10^{-8}$</td>
<td>0.0568</td>
<td>0.0602</td>
</tr>
<tr>
<td>1/10</td>
<td>1/40</td>
<td>$1.9609 \times 10^{-9}$</td>
<td>$4.9688 \times 10^{-8}$</td>
<td>0.7250</td>
<td>0.8014</td>
</tr>
<tr>
<td>1/20</td>
<td>1/40</td>
<td>$1.1863 \times 10^{-8}$</td>
<td>$2.8510 \times 10^{-8}$</td>
<td>0.5934</td>
<td>0.6482</td>
</tr>
<tr>
<td>1/40</td>
<td>1/40</td>
<td>$7.8625 \times 10^{-9}$</td>
<td>$1.8191 \times 10^{-8}$</td>
<td>0.4565</td>
<td>0.5088</td>
</tr>
</tbody>
</table>

Remark 5.1 In the examples, we have to different cases of $\mu$, which smoothen our equation. In the first test we use a very small $\mu = 0.05$, so that we have a dominant hyperbolic behavior, due to this we have a loose in the regularity and sharp front. The iterative splitting method is loose one order. In the second test, we have increased the smoothness, while we get a more parabolic behavior. We could shown that the results are improved to higher accuracy.
5.2 Test example 2: mixed convection-diffusion and Burgers equation

We deal with a 2D example which is a mixture of a convection-diffusion and Burgers equation. We can derive an analytical solution.

\[
\frac{\partial u}{\partial t} = -\frac{1}{2}u \frac{\partial u}{\partial x} - \frac{1}{2}u \frac{\partial u}{\partial y} - \frac{1}{2}u + \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t), \quad (x, y, t) \in \Omega \times [0, T] \\
u(x, y, 0) = u_{\text{asym}}(x, y, 0), \quad (x, y) \in \Omega \\
\text{with } u(x, y, t) = u_{\text{asym}}(x, y, t) \text{ on } \partial \Omega \times [0, T],
\]

where \( \Omega = [0, 1] \times [0, 1] \), \( T = 1.25 \), and \( \mu \) is the viscosity.

The analytical solution is given as

\[
u_{\text{asym}}(x, y, t) = (1 + \exp(\frac{x + y - t}{2\mu}))^{-1} + \exp(\frac{x + y - t}{2\mu}),
\]

where we compute \( f(x, y, t) \) accordingly.

We split the convection-diffusion and the Burgers equation. The operators are given as:

\[
A(u)u = -\frac{1}{2}u \frac{\partial u}{\partial x} - \frac{1}{2}u \frac{\partial u}{\partial y} + \frac{1}{2}u(\partial_{xx} u + \partial_{yy} u), \quad \text{hence} \\
A(u) = 1/2(-u \frac{\partial u}{\partial x} - u \frac{\partial u}{\partial y} + \mu(\partial_{xx} u + \partial_{yy} u)) \quad \text{(the Burgers term), and}
\]

\[
Bu = -\frac{1}{2} \frac{\partial u}{\partial x} - \frac{1}{2} \frac{\partial u}{\partial y} + \frac{1}{2} \mu(\partial_{xx} u + \partial_{yy} u) + f(x, y, t) \quad \text{(the convection-diffusion term)}.
\]

For the first equation we apply the nonlinear Algorithm 7 and obtain

\[
A(u_{i-1})u_i = -\frac{1}{2}u_{i-1} \frac{\partial u_i}{\partial x} - \frac{1}{2}u_{i-1} \frac{\partial u_i}{\partial y} + \frac{1}{2}u_{i-1}(\partial_{xx} u_i + \partial_{yy} u_i) \quad \text{and}
\]

\[
Bu_{i-1} = 1/2(-\frac{\partial u_i}{\partial x} - \frac{\partial u_i}{\partial y} + \mu(\partial_{xx} u_i + \partial_{yy} u_i))u_{i-1},
\]

and we obtain linear operators, because \( u_{i-1} \) is known from the previous time step.

In the second equation we obtain by using Algorithm 8:

\[
A(u_{i-1})u_i = -\frac{1}{2}u_{i-1} \frac{\partial u_i}{\partial x} - \frac{1}{2}u_{i-1} \frac{\partial u_i}{\partial y} + \frac{1}{2}u_{i-1}(\partial_{xx} u_i + \partial_{yy} u_i) \quad \text{and}
\]

\[
Bu_{i+1} = 1/2(-\frac{\partial u_i}{\partial x} - \frac{\partial u_i}{\partial y} + \mu(\partial_{xx} u_i + \partial_{yy} u_i))u_{i+1},
\]

and we have also linear operators.

We deal with different viscosities \( \mu \) as well as different time and space steps. We have the following results, see Tables 3 and 4.

Figure 2 presents the profile of the 2D linear and nonlinear convection-diffusion equation.
### Table 3: Numerical results for the mixed convection-diffusion and Burgers equation with viscosity $\mu = 0.5$, initial condition $u_0(t) = c_n$, and four iterations per time step.

<table>
<thead>
<tr>
<th>$\Delta x = \Delta y$</th>
<th>$\Delta t$</th>
<th>$\text{err}_{L_1}$</th>
<th>$\text{err}_{\text{max}}$</th>
<th>$\rho_{L_1}$</th>
<th>$\rho_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/5 1/20</td>
<td>0.0137</td>
<td>0.0354</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/10 1/20</td>
<td>0.0055</td>
<td>0.0139</td>
<td>1.3264</td>
<td>1.3499</td>
<td></td>
</tr>
<tr>
<td>1/20 1/20</td>
<td>0.0017</td>
<td>0.0043</td>
<td>1.6868</td>
<td>1.6900</td>
<td></td>
</tr>
<tr>
<td>1/40 1/20</td>
<td>8.8839 $\cdot 10^{-5}$</td>
<td>3.8893 $\cdot 10^{-4}$</td>
<td>4.2588</td>
<td>3.4663</td>
<td></td>
</tr>
<tr>
<td>1/5 1/40</td>
<td>0.0146</td>
<td>0.0377</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/10 1/40</td>
<td>0.0064</td>
<td>0.0160</td>
<td>1.1984</td>
<td>1.2315</td>
<td></td>
</tr>
<tr>
<td>1/20 1/40</td>
<td>0.0026</td>
<td>0.0063</td>
<td>1.3004</td>
<td>1.3375</td>
<td></td>
</tr>
<tr>
<td>1/40 1/40</td>
<td>8.2653 $\cdot 10^{-4}$</td>
<td>0.0021</td>
<td>1.6478</td>
<td>1.6236</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4: Numerical results for the mixed convection-diffusion and Burgers equation with viscosity $\mu = 5$, initial condition $u_0(t) = c_n$, and two iterations per time step.

<table>
<thead>
<tr>
<th>$\Delta x = \Delta y$</th>
<th>$\Delta t$</th>
<th>$\text{err}_{L_1}$</th>
<th>$\text{err}_{\text{max}}$</th>
<th>$\rho_{L_1}$</th>
<th>$\rho_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/5 1/20</td>
<td>1.3166 $\cdot 10^{-6}$</td>
<td>2.9819 $\cdot 10^{-5}$</td>
<td>1.2092</td>
<td>1.1389</td>
<td></td>
</tr>
<tr>
<td>1/10 1/20</td>
<td>5.6944 $\cdot 10^{-6}$</td>
<td>1.3541 $\cdot 10^{-5}$</td>
<td>1.7452</td>
<td>1.5634</td>
<td></td>
</tr>
<tr>
<td>1/20 1/20</td>
<td>1.6986 $\cdot 10^{-6}$</td>
<td>4.5816 $\cdot 10^{-6}$</td>
<td>1.1201</td>
<td>1.1663</td>
<td></td>
</tr>
<tr>
<td>1/40 1/20</td>
<td>7.8145 $\cdot 10^{-7}$</td>
<td>2.0413 $\cdot 10^{-6}$</td>
<td>0.9957</td>
<td>1.0233</td>
<td></td>
</tr>
<tr>
<td>1/5 1/40</td>
<td>1.4425 $\cdot 10^{-5}$</td>
<td>3.2036 $\cdot 10^{-5}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/10 1/40</td>
<td>7.343 $\cdot 10^{-6}$</td>
<td>1.5766 $\cdot 10^{-5}$</td>
<td>0.9957</td>
<td>1.0233</td>
<td></td>
</tr>
<tr>
<td>1/20 1/40</td>
<td>3.0776 $\cdot 10^{-6}$</td>
<td>6.7999 $\cdot 10^{-6}$</td>
<td>1.2330</td>
<td>1.2129</td>
<td></td>
</tr>
<tr>
<td>1/40 1/40</td>
<td>9.8650 $\cdot 10^{-7}$</td>
<td>2.3352 $\cdot 10^{-6}$</td>
<td>1.6414</td>
<td>1.5420</td>
<td></td>
</tr>
</tbody>
</table>

### Remark 5.2
In the examples, we deal with more iterative steps to obtain higher order convergence results. In the first test we have four iterative steps but a smaller viscosity ($\mu = 0.05$), so that we can reach at least a second order method. In the second test we use a high viscosity about $\mu = 5$ and get the second order result with 2 iterative steps. Here we see the loose of differentiability, that becomes stiff equation parts. To obtain the same results, we have to increased the iterative steps. So we could shown an improvement of the convergence order with respect to the iterative steps.

### 6 Conclusions and Discussions
We present a new method to solve nonlinear coupled partial differential equations. Based on a standard splitting method we derive different new methods, based on iterated and linearized ideas, to solve the decoupled equations more
efficiently. Such linearizations can reduce the decomposition error. The more hyperbolic behavior of the equations leads to an increase in the number of iterative steps of our method. At least we obtain a second order method. Such iterative splitting method can balance the different behavior of the underlying operators. So the one operator smoothen the solution process, while the other operator decrease the smoothness. Further a balance between the implicit and explicit discretization with the iterative splitting method is a new method that overcome to the mixed behavior in an unsplitted method.

References


REFERENCES


