Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2009, Article ID 307298, 24 pages
doi:10.1155/2009/307298

Research Article

Modified Jacobian Newton Iterative Method: Theory and Applications

Jürgen Geiser

Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany

Correspondence should be addressed to Jürgen Geiser, geiser@mathematik.hu-berlin.de

Received 3 September 2008; Revised 10 November 2008; Accepted 4 December 2008

Recommended by José Roberto Castilho Piqueira

This article proposes a new approach to the construction of a linearization method based on the iterative operator-splitting method for nonlinear differential equations. The convergence properties of such a method are studied. The main features of the proposed idea are the linearization of nonlinear equations and the application of iterative splitting methods. We present an iterative operator-splitting method with embedded Newton methods to solve nonlinearity. We confirm with numerical applications the effectiveness of the proposed iterative operator-splitting method in comparison with the classical Newton methods. We provide improved results and convergence rates.

1. Introduction

In this paper we propose a modified Jacobian-Newton iterative method to solve nonlinear differential equations. In the first paper we concentrate on ordinary differential equations, but numerical results are also obtained for partial differential equations. Basic studies of the operator-splitting methods are found in [1, 2]. Further important research was carried out to obtain a higher order for the splitting methods (see [3]). For this reason, the iterative splitting methods became more important for linear and nonlinear differential equations, while simple increasing of iteration steps affects the order of the scheme (see [4]). An interesting topic is Newton's methods for nonlinear problems with specifications for numerical implementations, (see [5]). Efficient modifications of Newton's methods, with regard to the computation of the Jacobian matrices are discussed in [6]. In our paper we discuss the benefit of the combination of splitting and linearization methods (see theoretical frameworks [7, 8]).

The outline of the paper is as follows. For our mathematical model we describe the convection-diffusion-reaction equation in Section 2. The fractional splitting is introduced in Section 3. We present the iterative splitting methods in Section 4. Section 5 discusses the Newton methods and their modifications. In Section 6 we present the numerical results from the solution of selected model problems. We end the article in Section 7 with a conclusion and comments.

2. Mathematical Model

The motivation for the study presented below originates from a computational simulation of heat-transfer [9] and convection-diffusion-reaction-equations [10–13].

In the present paper we concentrate on ordinary differential equations, given as

$$\partial_t u(t) = A(u(t))\, u(t) + B(u(t))u(t), \quad t \in (0, T), \tag{2.1}$$

where the initial condition is $u(0) = u_0$. The operators $A(u)$ and $B(u)$ can be spatially discretized operators, that is, they can correspond to the discretized in space convection and diffusion operators (matrices). In the following, we deal with bounded nonlinear operators.

The aim of this paper is to present a new method based on Newton and iterative schemes.

In the next section we discuss the decoupling of the time-scale with a first-order fractional splitting method.

3. Fractional-Splitting Methods of First-Order for Linear Equations

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

$$\partial_t u(t) = Au(t) + Bu(t), \quad t \in (0, T), \tag{3.1}$$

where the initial condition is $u(0) = u_0$. The operators A and B are linear and bounded operators in a Banach space (see also Section 2).

The sequential operator-splitting method is introduced as a method that solves two subproblems sequentially, where the different subproblems are connected via the initial conditions. This means that we replace the original problem (3.1) with the subproblems

$$\begin{aligned} \frac{\partial u^*(t)}{\partial t} &= Au^*(t), \quad \text{with } u^*(t^n) = u^n, \\ \frac{\partial u^{**}(t)}{\partial t} &= Bu^{**}(t), \quad \text{with } u^{**}(t^n) = u^*(t^{n+1}), \end{aligned} \tag{3.2}$$

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

Clearly, the replacement of the original problem with the subproblems usually results in an error, called *splitting error*. The splitting error of the sequential operator-splitting method can be derived as (cf., e.g., [1, 2]).

$$\rho_n = \frac{1}{\tau_n}\left(\exp\left(\tau_n(A+B)\right) - \exp\left(\tau_n B\right)\exp\left(\tau_n A\right)\right)u(t^n)$$
$$= \begin{cases} 0, & \text{for } [A,B]=0, \\ O(\tau_n), & \text{for } [A,B]\neq 0, \end{cases} \tag{3.3}$$

where $[A,B] := AB - BA$ 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 operator-splitting is called the *first-order splitting method*.

4. The 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 subproblems consecutively for $i = 0, 2, \ldots 2m$:

$$\begin{aligned}
\frac{\partial u_i(x,t)}{\partial t} &= Au_i(x,t) + Bu_{i-1}(x,t), \quad \text{with } u_i(t^n) = u^n, \\
u_0(x,t^n) &= u^n, \qquad u_{-1} = 0, \\
u_i(x,t) &= u_{i-1}(x,t) = u_1, \quad \text{on } \partial\Omega \times (0,T), \\
\frac{\partial u_{i+1}(x,t)}{\partial t} &= Au_i(x,t) + Bu_{i+1}(x,t), \quad \text{with } u_{i+1}(x,t^n) = u^n, \\
u_i(x,t) &= u_{i-1}(x,t) = u_1, \quad \text{on } \partial\Omega \times (0,T),
\end{aligned} \tag{4.1}$$

where u^n is the known split approximation at the time level $t = t^n$ (see [14]).

Remark 4.1. We can generalize the iterative splitting method to a multi-iterative splitting method by introducing new splitting operators, for example, spatial operators. Then we obtain multi-indices to control the splitting process; each iterative splitting method can be solved independently, while connecting with further steps to the multi-splitting methods. In the following we introduce the multi-iterative splitting method for a combined time-space splitting method.

5. The Modified Jacobian-Newton Methods and Fixpoint-Iteration Methods

In this section we describe the modified Jacobian-Newton methods and Fixpoint-iteration methods.

We propose for weak nonlinearities, for example, quadratic nonlinearity, the fixpoint iteration method, where our iterative operator splitting method is one, see [4]. For stronger nonlinearities, for example, cubic or higher order polynomial nonlinearities, the modified Jacobian method with embedded iterative-splitting methods is suggested.

The point of embedding the splitting methods into the Newton methods is to decouple the equation systems into simpler equations. Such simple equation systems can be solved with scalar Newton methods.

5.1. The Altered Jacobian-Newton Iterative Methods with Embedded Sequential Splitting Methods

We confine our attention to time-dependent partial differential equations of the form

$$\frac{dc}{dt} = A(c(t))c(t) + B(c(t))c(t), \quad \text{with } c(t^n) = c^n, \tag{5.1}$$

where $A(c), B(c) : \mathbf{X} \to \mathbf{X}$ are linear and densely defined in the real Banach space $\mathbf{X}$, involving only spatial derivatives of c, see [8]. We assume also that we have a weak nonlinear operator with $A(c)c = \lambda_1 c$ and $B(c)c = \lambda_2 c$, where λ_1 and λ_2 are constant factors.

In the following we discuss the embedding of a sequential splitting method into the Newton method.

The altered Jacobian-Newton iterative method with an embedded iterative splitting method is given as follows.

Newton's Method

$F(c) = dc/dt - A(c(t))c(t) - B(c(t))c(t)$ and we can compute $c^{(k+1)} = c^{(k)} - D(F(c^{(k)}))^{-1} F(c^{(k)})$, where $D(F(c))$ is the Jacobian matrix and $k = 0, 1, \ldots$.

We stop the iterations when we obtain: $|c^{(k+1)} - c^{(k)}| \leq \text{err}$, where err is an error bound, for example, $\text{err} = 10^{-4}$.

We assume the spatial discretization, with spatial grid points, $i = 1, \ldots, m$ and obtain the differential equation system:

$$F(c) = \begin{pmatrix} F(c_1) \\ F(c_2) \\ \vdots \\ F(c_m) \end{pmatrix}, \tag{5.2}$$

where $c = (c_1, \ldots, c_m)T$ and m is the number of spatial grid points.

The Jacobian matrix for the equation system is given as:

$$DF(c) = \begin{pmatrix} \frac{\partial F(c_1)}{c_1} & \frac{\partial F(c_1)}{c_2} & \cdots & \frac{\partial F(c_1)}{c_m} \\ \frac{\partial F(c_2)}{c_1} & \frac{\partial F(c_2)}{c_2} & \cdots & \frac{\partial F(c_2)}{c_m} \\ \vdots & & & \\ \frac{\partial F(c_m)}{c_1} & \frac{\partial F(c_m)}{c_2} & \cdots & \frac{\partial F(c_m)}{c_m} \end{pmatrix}, \tag{5.3}$$

where $c = (c_1, \ldots, c_m)$.

The modified Jacobian is given as:

$$DF(c) = \begin{pmatrix} \dfrac{\partial F(c_1)}{c_1} + F(c_1) & \dfrac{\partial F(c_1)}{c_2} & \dots & \dfrac{\partial F(c_1)}{c_m} \\ \dfrac{\partial F(c_2)}{c_1} & \dfrac{\partial F(c_2)}{c_2} F(c_2) & \dots & \dfrac{\partial F(c_2)}{c_m} \\ \vdots & & & \\ \dfrac{\partial F(c_m)}{c_1} & \dfrac{\partial F(c_m)}{c_2} & \dots & \dfrac{\partial F(c_m)}{c_m} + F(c_m) \end{pmatrix}, \tag{5.4}$$

where $c = (c_1, \dots, c_n)$.

By embedding the sequential splitting method we obtain the following algorithm. We decouple into two equation systems:

$$\begin{aligned} F_1(u_1) = \partial_t u_1 - A(u_1)u_1 = 0 \quad \text{with } u_1(t^n) = c^n, \\ F_2(u_2) = \partial_t u_2 - B(u_2)u_2 = 0 \quad \text{with } u_2(t^n) = u_1(t^{n+1}), \end{aligned} \tag{5.5}$$

where the results of the methods are $c(t^{n+1}) = u_2(t^{n+1})$, and $u_1 = (u_{11}, \dots, u_{1n})$, $u_2 = (u_{21}, \dots, u_{2n})$.

Thus we have to solve two Newton methods, each in one equations system. The contribution is to reduce the Jacobian matrix into a diagonal entry, for example, with a weighted Newton method, see [15]. The splitting method with embedded Newton method is given as follows.

Algorithm 5.1. We assume the spatial operators A and B are discretized, for example, finite difference or finite element methods; further all initial conditions and boundary conditions are discrete given. Then we can apply the Newton's method in its discrete form as:

$$\begin{gathered} u_1^{(k+1)} = u_1^{(k)} - D(F_1(u_1^{(k)}))^{-1} (\partial_t u_1^{(k)} - A(u_1^{(k)}) u_1^{(k)}), \\ \text{with } D(F_1(u_1^{(k)})) = \frac{\partial}{\partial u_1^{(k)}} \left(\partial_t u_1^{(k)} - A(u_1^{(k)}) - \frac{\partial A(u_1^{(k)})}{\partial u_1^{(k)}} u_1^{(k)} \right), \\ u_1^{(k)}(t^n) = c^n, \quad k = 0, 1, 2, \dots, K, \\ u_2^{(l+1)} = u_2^{(l)} - D(F_2(u_2^{(l)}))^{-1} (\partial_t u_2^{(l)} - B(u_2^{(l)}) u_2^{(l)}), \\ \text{with } D(F_2(u_2^{(l)})) = \frac{\partial}{\partial u_1^{(k)}} \left(\partial_t u_2^{(k)} - B(u_2^{(l)}) - \frac{\partial B(u_2^{(l)})}{\partial u_2^{(l)}} u_2^{(l)} \right), \\ u_2^{(l)}(t^n) = u_1^K(t^{n+1}), \quad l = 0, 1, 2, \dots, L. \end{gathered} \tag{5.6}$$

where k and l are the iteration indices, K and L the maximal iterative steps for each part of the Newton's method. The maximal iterative steps allow us to have at least an error of:

$$\begin{aligned} \left|u_1^{(K)(t^{n+1})} - u_1^{(K-1)(t^{n+1})}\right| &\leq \text{err}, \\ \left|u_2^{(L)(t^{n+1})} - u_2^{(L-1)(t^{n+1})}\right| &\leq \text{err}, \end{aligned} \tag{5.7}$$

where err is the error bound, for example, err $= 10^{-6}$.

The approximated solution is given as:

$$u(t^{n+1}) = u_2^{(L)(t^{n+1})}. \tag{5.8}$$

For the improvement method, we can apply the weighted Newton method. We try to skip the delicate outer diagonals in the Jacobian matrix and apply:

$$u_1^{(k+1)} = u_1^{(k)} - \left(D\left(F_1\left(u_1^{(k)}\right)\right) + \delta_1\left(u_1^{(k)}\right)\right)^{-1}\left(F_1\left(u_1^{(k)}\right) + \epsilon\, u_1^{(k)}\right), \tag{5.9}$$

where the function δ can be applied as a scalar, for example, $\delta = 10^{-6}$, and the same with ϵ. It is important to ensure that δ is small enough to preserve the convergence.

Remark 5.2. If we assume that we discretize (5.5) with the backward-Euler method, for example,

$$\begin{aligned} &F_1(u_1(t^{n+1})) = u_1(t^{n+1}) - u_1(t^n) - \Delta t A(u_1(t^{n+1}))u_1(t^{n+1}) = 0 \quad \text{with } u_1(t^n) = c^n, \\ &F_2(u_2) = u_2(t^{n+1}) - u_2(t^n) - \Delta t B(u_2(t^{n+1}))u_2(t^{n+1}) = 0 \quad \text{with } u_2(t^n) = u_1(t^{n+1}), \end{aligned} \tag{5.10}$$

then we obtain the derivations $D(F_1(u_1(t^{n+1})))$ and $D(F_2(u_2(t^{n+1})))$

$$\begin{aligned} D(F_1(u_1(t^{n+1}))) &= 1 - \Delta t\left(A(u_1(t^{n+1})) + \frac{\partial A(u_1(t^{n+1}))}{\partial u_1(t^{n+1})}u_1(t^{n+1})\right), \\ D(F_2(u_2)) &= 1 - \Delta t\left(B(u_2(t^{n+1})) + \frac{\partial B(u_2(t^{n+1}))}{\partial u_2(t^{n+1})}u_2(t^{n+1})\right). \end{aligned} \tag{5.11}$$

We can apply the equation (5.9) analogously $u_2^{(l+1)}$.

5.2. Iterative Operator-Splitting Method as a Fixpoint Scheme

The iterative operator-splitting method is used as a fixpoint scheme to linearize the nonlinear operators, see [4, 16].

We confine our attention to time-dependent partial differential equations of the form:

$$\frac{du}{dt} = A(u(t))u(t) + B(u(t))u(t), \quad \text{with } u(t^n) = c^n, \tag{5.12}$$

where $A(u), B(u) : \mathbf{X} \to \mathbf{X}$ are linear and densely defined in the real Banach space $\mathbf{X}$, involving only spatial derivatives of c, see [8]. In the following we discuss the standard iterative operator-splitting methods as a fixpoint iteration method to linearize the operators.

We split our nonlinear differential equation (5.12) by applying:

$$\begin{aligned} \frac{du_i(t)}{dt} &= A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i-1}(t), \quad \text{with } u_i(t^n) = c^n, \\ \frac{du_{i+1}(t)}{dt} &= A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i+1}, \quad \text{with } u_{i+1}(t^n) = c^n, \end{aligned} \tag{5.13}$$

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 the solution c^{n+1} is near c^n, or $u_0(t) = 0$. So we have to solve the local fixpoint problem. c^n is the known split approximation at the 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 the operators $A(u_{i-1}), B(u_{i-1}) : \mathbf{X} \to \mathbf{X}$ to be linear and densely defined on the real Banach space $\mathbf{X}$, for $i = 1, 3, \ldots, 2m+1$.

Here the linearization is done with respect to the iterations, such that $A(u_{i-1}), B(u_{i-1})$ are at least non-dependent operators in the iterative equations, and we can apply the linear theory.

The linearization is at least in the first equation $A(u_{i-1}) \approx A(u_i)$, and in the second equation $B(u_{i-1}) \approx B(u_{i+1})$.

We have

$$\left\| A(u_{i-1}(t^{n+1}))u_i(t^{n+1}) - A(u^{n+1})u(t^{n+1}) \right\| \le \epsilon, \tag{5.14}$$

with sufficient iterations $i = \{1, 3, \ldots, 2m+1\}$.

Remark 5.3. The linearization with the fixpoint scheme can be used for smooth or weak nonlinear operators, otherwise we lose the convergence behavior, while we did not converge to the local fixpoint, see [4].

The second idea is based on the Newton method.

5.3. Jacobian-Newton Iterative Method with Embedded Operator-Splitting Method

The Newton method is used to solve the nonlinear parts of the iterative operator-splitting method (see the linearization techniques in [4, 17]).

Newton Method

The function is given as:

$$F(c) = \frac{\partial c}{\partial t} - A(c(t))c(t) - B(c(t))c(t) = 0, \tag{5.15}$$

The iteration can be computed as:

$$c^{(k+1)} = c^{(k)} - D\left(F\left(c^{(k)}\right)\right)^{-1} F\left(c^{(k)}\right), \tag{5.16}$$

where $D(F(c))$ is the Jacobian matrix and $k = 0, 1, \ldots.$ and $c = (c_1, \ldots, c_m)$ is the solution vector of the spatial discretized nonlinear equation.

We then have to apply the iterative operator-splitting method and obtain:

$$\begin{aligned} F_1(u_i) &= \partial_t u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0, \quad \text{with } u_i(t^n) = c^n, \\ F_2(u_{i+2}) &= \partial_t u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0, \quad \text{with } u_{i+1}(t^n) = c^n, \end{aligned} \tag{5.17}$$

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 the time-level $t = t^n$. The results of the methods are $c(t^{n+1}) = u_{2m+2}(t^{n+1})$.

Thus we have to solve two Newton methods and the contribution will be to reduce the Jacobian matrix, for example, skip the diagonal entries. The splitting method with the embedded Newton method is given as:

$$\begin{gathered} u_i^{(k+1)} = u_i^{(k)} - D\left(F_1\left(u_i^{(k)}\right)\right)^{-1}\left(\partial_t u_i^{(k)} - A\left(u_i^{(k)}\right)u_i^{(k)} - B\left(u_{i-1}^{(k)}\right)u_{i-1}^{(k)}\right), \\ \text{with } D\left(F_1\left(u_i^{(k)}\right)\right) = -\left(A\left(u_i^{(k)}\right) + \frac{\partial A\left(u_i^{(k)}\right)}{\partial u_i^{(k)}} u_i^{(k)}\right), \quad k = 0, 1, 2, \ldots, K, \quad \text{with } u_i(t^n) = c^n, \\ u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D\left(F_2\left(u_{i+1}^{(l)}\right)\right)^{-1}\left(\partial_t u_{i+1}^{(l)} - A\left(u_i^{(k)}\right)u_i^{(k)} - B\left(u_{i+1}^{(k)}\right)u_{i+1}^{(k)}\right)c_2^{(l)}\right), \\ \text{with } D\left(F_2\left(u_{i+1}^{(l)}\right)\right) = -\left(B\left(u_{i+1}^{(l)}\right) + \frac{\partial B\left(u_{i+1}^{(l)}\right)}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}\right), \quad l = 0, 1, 2, \ldots, L, \quad \text{with } u_{i+1}(t^n) = c^n, \end{gathered} \tag{5.18}$$

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 the time-level $t = t^n$. The results of the methods are $c(t^{n+1}) = u_{2m+2}(t^{n+1})$.

For the improvement by skipping the delicate outer diagonals in the Jacobian matrix, we apply $u_i^{(k+1)} = u_i^{(k)} - (D(F_1(u_i^{(k)})) + \delta_1(u_i^{(k)}))^{-1}(F_1(u_i^{(k)}) + \epsilon\, u_i^{(k)})$, and analogously $u_{i+1}^{(l+1)}$.

Remark 5.4. For the iterative operator-splitting method with the Newton iteration we have two iteration procedures. The first iteration is the Newton method to compute the solution of the nonlinear equations, and 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.

6. Numerical Results

In this section, we present the numerical results for nonlinear differential equation using several variations of the proposed Newton and iterative schemes as solvers.

6.1. First Numerical Example: Bernoulli Equation

As a nonlinear differential example, we choose the Bernoulli equation:

$$\frac{\partial u(t)}{\partial t} = (\lambda_1 + \lambda_3)u(t) + (\lambda_2 + \lambda_4)(u(t))^p, \quad t \in [0, T], \text{ with } u(0) = 1, \tag{6.1}$$

where the analytical solution can be derived as (see also [16]):

$$u(t) = \exp((\lambda_1 + \lambda_3)t)\left[-\frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3}\exp((\lambda_1 + \lambda_3)(p-1)t) + c\right]^{1/(1-p)}. \tag{6.2}$$

Using $u(0) = 1$ we find that $c = 1 + (\lambda_2 + \lambda_4)/(\lambda_1 + \lambda_3)$, so

$$u(t) = \exp((\lambda_1 + \lambda_3)t)\left\{1 + \frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3}[1 - \exp((\lambda_1 + \lambda_3)(p-1)t)]\right\}^{1/(1-p)}. \tag{6.3}$$

We choose $p = 2$, $\lambda_1 = -1$, $\lambda_2 = -0.5$, $\lambda_3 = -100$, $\lambda_4 = -20$ and, for example, $\Delta t = 10^{-2}$. The analytical solutions can be given as:

$$u(t)^{1-p} = u_0 \exp((1-p)(\lambda_1 + \lambda_3)t) + \frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3}(\exp((1-p)(\lambda_1 + \lambda_3)t) - 1). \tag{6.4}$$

We divide the time interval $[0, T]$, with $T = 1$, in n intervals with length $\tau_n = T/n$.

(1) The sequential operator-splitting method with analytical solutions is given as follows.

We apply the quasilinear iterative operator-splitting method:

$$\begin{aligned} \frac{du_1(t)}{dt} &= A(u_1(t))u_1(t), \quad \text{with } u_1(t^n) = u^n, \\ \frac{du_2(t)}{dt} &= B(u_2(t))u_2, \quad \text{with } u_2(t^n) = u_1^{n+1}, \end{aligned} \tag{6.5}$$

with the nonlinear operators $A(u)u = \lambda_1 u(t) + \lambda_2 (u(t))^{p-1}u$, $B(u)u = \lambda_3 u(t) + \lambda_4 (u(t))^{p-1}u$. The result is given as $u_2(t^{n+1}) = u^{n+1}$.

We apply the Newton method and discretize the operators with time discretization methods such as backward-Euler or higher Runge-Kutta methods.

The analytical result for each equation part is given as:

$$
\begin{aligned}
u_1(t)^{1-p} &= u(t^n)\exp\left((1-p)(\lambda_1)t\right) + \frac{\lambda_2}{\lambda_1}\left(\exp\left((1-p)(\lambda_1)t\right) - 1\right),\\
u_2\left(t^{n+1}\right)^{1-p} &= u_1\left(t^{n+1}\right)\exp\left((1-p)(\lambda_3)t\right) + \frac{\lambda_4}{\lambda_3}\left(\exp\left((1-p)(\lambda_3)t\right) - 1\right),
\end{aligned}
\tag{6.6}
$$

where the result is given as $u(t^{n+1}) = u_2(t^{n+1})$.

We can apply the simpler equations and solve the sequential operator-splitting method.

(2) The sequential operator-splitting method with embedded Newton method is given as follows.

We apply the quasilinear iterative operator-splitting method:

$$
\begin{aligned}
\frac{du_1(t)}{dt} &= A(u_1(t))u_1(t), \quad \text{with } u_1(t^n) = u^n,\\
\frac{du_2(t)}{dt} &= B(u_2(t))u_2, \quad \text{with } u_2(t^n) = u_1^{n+1},
\end{aligned}
\tag{6.7}
$$

with the nonlinear operators $A(u)u = \lambda_1 u(t) + \lambda_2 (u(t))^{p-1} u$, $B(u)u = \lambda_3 u(t) + \lambda_4 (u(t))^{p-1} u$. The result is given as $u_2(t^{n+1}) = u^{n+1}$.

We apply the Newton method and discretize the operators with time discretization methods such as backward-Euler or higher Runge-Kutta methods.

The splitting method with embedded Newton's method is given as

$$
\begin{aligned}
&u_1^{(k+1)} = u_1^{(k)} - D\left(F_1\left(u_1^{(k)}\right)\right)^{-1}\left(\partial_t u_1^{(k)} - A\left(u_1^{(k)}\right)u_1^{(k)}\right),\\
&\text{with } D\left(F_1\left(u_1^{(k)}\right)\right) = -\left(A\left(u_1^{(k)}\right) + \frac{\partial A\left(u_1^{(k)}\right)}{\partial u_1^{(k)}} u_1^{(k)}\right),\\
&u_1^{(k)}(t^n) = c^n, \quad k = 0, 1, 2, \ldots, K,\\
&u_2^{(l+1)} = u_2^{(l)} - D\left(F_2\left(u_2^{(l)}\right)\right)^{-1}\left(\partial_t u_2^{(l)} - B\left(u_2^{(l)}\right)u_2^{(l)}\right),\\
&\text{with } D\left(F_2\left(u_2^{(l)}\right)\right) = -\left(B\left(u_2^{(l)}\right) + \frac{\partial B\left(u_2^{(l)}\right)}{\partial u_2^{(l)}} u_2^{(l)}\right),\\
&u_2^{(l)}(t^n) = u_1^K(t^{n+1}), \quad l = 0, 1, 2, \ldots, L,
\end{aligned}
\tag{6.8}
$$

where we discretize the equations and obtain the discretized operators:

$$
\partial_t u_1^{(k)} - A\left(u_1^{(k)}\right)u_1^{(k)} = 0,
\tag{6.9}
$$

as

$$F_1(u_1(t^{n+1}))u_1^{(k)}(t^{n+1}) - u_1(t^n) - \Delta t A(u_1^{(k)}(t^{n+1}))u_1^{(k)}(t^{n+1}) = 0, \tag{6.10}$$

where we have the initialization of the Newton's method as $u_1^{(0)}(t^{n+1}) = 0$ or $u_1^{(0)}(t^{n+1}) = u_1(t^n)$.

For the second iteration equation we have

$$\partial_t u_2^{(l)} - B(u_2^{(l)})u_2^{(l)}, \tag{6.11}$$

as

$$F_2(u_2(t^{n+1})) = u_2^{(l)}(t^{n+1}) - u_2(t^n) - \Delta t B(u_2^{(l)}(t^{n+1}))u_2^{(l)}(t^{n+1}) = 0, \tag{6.12}$$

where we have the initialization of the Newton's method as $u_2^{(0)}(t^{n+1}) = 0$ or $u_2^{(0)}(t^{n+1}) = u_1(t^n)$.

The derivations are given as:

$$\begin{gathered} D(F_1(u_1(t^{n+1}))) = 1 - \Delta t\left(A(u_1(t^{n+1})) + \frac{\partial A(u_1(t^{n+1}))}{\partial u_1(t^{n+1})} u_1(t^{n+1}) \right), \\ D(F_2(u_2)) = 1 - \Delta t\left(B(u_2(t^{n+1})) + \frac{\partial B(u_2(t^{n+1}))}{\partial u_2(t^{n+1})} u_2(t^{n+1}) \right). \end{gathered} \tag{6.13}$$

(3) The standard iterative operator-splitting method is given as follows.

We apply the quasilinear iterative operator-splitting method:

$$\begin{aligned} \frac{du_i(t)}{dt} &= A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i-1}(t), \quad \text{with } u_i(t^n) = u^n, \\ \frac{du_{i+1}(t)}{dt} &= A(u_{i-1}(t))u_i(t) + B(u_{i-1}(t))u_{i+1}, \quad \text{with } u_{i+1}(t^n) = u^n, \end{aligned} \tag{6.14}$$

with the nonlinear operators $A(u)u = \lambda_1 u(t) + \lambda_2(u(t))^{p-1}u$, $B(u)u = \lambda_3 u(t) + \lambda_4(u(t))^{p-1}u$. The initialization of the fixpoint iteration is $u_0 = u^n$ or $u_0 = 0$ with $A(u_0) = \lambda_1$ and $B(u_0) = \lambda_3$.

For the iterations we can apply the analytical solution of each equation:

$$\begin{aligned} u_i(t) &= u^n \exp\left(A(u_{i-1}(t))t\right) + \left(A(u_{i-1}(t))\right)^{-1}\left(B(u_{i-1}(t))u_{i-1}(t)\right)\left(1 - \exp\left(A(u_{i-1}(t))t\right)\right), \\ u_{i+1}(t) &= u^n \exp\left(B(u_{i-1}(t))t\right) + \left(B(u_{i-1}(t))\right)^{-1}\left(A(u_{i-1}(t))u_{i-1}(t)\right)\left(1 - \exp\left(B(u_{i-1}(t))t\right)\right). \end{aligned} \tag{6.15}$$

Further the iterative steps can be done.

(4) The Newton iterative method with embedded iterative operator-splitting method is given as follows.

We apply the quasilinear iterative operator-splitting method:

$$
\begin{aligned}
\frac{du_i(t)}{dt} &= A(u_i(t))u_i(t) + B(u_{i-1}(t))u_{i-1}(t), \quad \text{with } u_i(t^n) = u^n, \\
\frac{du_{i+1}(t)}{dt} &= A(u_i(t))u_i(t) + B(u_{i+1}(t))u_{i+1}, \quad \text{with } u_{i+1}(t^n) = u^n,
\end{aligned}
\tag{6.16}
$$

with the nonlinear operators $A(u)u = \lambda_1 u(t) + \lambda_2 (u(t))^{p-1} u$, $B(u)u = \lambda_3 u(t) + \lambda_4 (u(t))^{p-1} u$. The initialization of the fixpoint iteration is $u_0(t^{n+1}) = u^n$ or $u_0(t^{n+1}) = 0$.

The discretization of the nonlinear ordinary differential equation is performed with higher-order Runge-Kutta methods.

The Newton method is applied as:

$$
\begin{aligned}
u_i^{(k+1)} &= u_i^{(k)} - D\left(F_1\left(u_i^{(k)}\right)\right)^{-1}\left(\partial_t u_i^{(k)} - A\left(u_i^{(k)}\right)u_i^{(k)} - B(u_{i-1})u_{i-1}\right), \\
D\left(F_1\left(u_i^{(k)}\right)\right) &= -\left(A\left(u_i^{(k)}\right) + \frac{\partial A\left(u_i^{(k)}\right)}{\partial u_i^{(k)}} u_i^{(k)}\right), \quad k = 0,1,2,\ldots,K, \\
& u_i(t^n) = c^n, \\
u_i(t^{n+1}) &= u_i(t^{n+1})^{K+1}, \quad \text{where } |u_i(t^{n+1})^{K+1} - u_i(t^{n+1})^{K}| \leq \text{err}, \\
u_{i+1}^{(l+1)} &= u_{i+1}^{(l)} - D\left(F_2\left(u_{i+1}^{(l)}\right)\right)^{-1}\left(\partial_t u_{i+1}^{(l)} - A(u_i)u_i - B\left(u_{i+1}^{(l)}\right)u_{i+1}^{(l)} c_2^{(l)}\right), \\
D\left(F_2\left(u_{i+1}^{(l)}\right)\right) &= -\left(B\left(u_{i+1}^{(l)}\right) + \frac{\partial B\left(u_{i+1}^{(l)}\right)}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}\right), \quad l = 0,1,2,\ldots,L, \\
& u_{i+1}(t^n) = c^n, \\
u_{i+1}(t^{n+1}) &= u_{i+1}(t^{n+1})^{L+1}, \quad \text{where } |u_{i+1}(t^{n+1})^{L+1} - u_{i+1}(t^{n+1})^{L}| \leq \text{err}.
\end{aligned}
\tag{6.17}
$$

Here the time-step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1,3,\ldots,2m+1$. $u_0(t) = 0$ is the starting solution and c^n is the known split approximation at the time-level $t = t^n$. The results of the methods are $u(t^{n+1}) = u_{2m+2}(t^{n+1})$.

We apply the discretization methods for the iteration steps.

We discretize the equations

$$
\partial_t u_i^{(k)} - A\left(u_i^{(k)}\right)u_i^{(k)} - B(u_{i-1})u_{i-1} = 0\,,
\tag{6.18}
$$

as

$$
F_1\left(u_i^{(k)}(t^{n+1})\right) = u_i^{(k)}(t^{n+1}) - u_i(t^n) - \Delta t\left(A\left(u_i^{(k)}(t^{n+1})\right)u_i^{(k)}(t^{n+1}) + B\left(u_{i-1}(t^{n+1})\right)u_{i-1}(t^{n+1})\right),
\tag{6.19}
$$

where we have the initialization of the Newton's method as $u_i^{(0)}(t^{n+1}) = 0$ or $u_i^{(0)}(t^{n+1}) = u(t^n)$.

Table 1: Numerical results for the Bernoulli equation with sequential operator-splitting method.

Time partitions	Approximated solution	Error
1	0.0000000000	6.620107e-044
2	0.0000000000	5.874983e-023
3	0.0000000000	6.351951e-016
4	0.0000000000	1.917794e-012
5	0.0000000002	2.232302e-010
10	0.0000023626	2.362646e-006
50	0.0015822287	1.582229e-003

Table 2: Numerical results for the Bernoulli equation with sequential operator-splitting method with embedded Newton's method.

Time partitions	Approximated solution	Error
1	0.4705129443	4.705129e-001
4	0.0546922483	5.469225e-002
5	0.0269954419	2.699544e-002
10	0.0008034713	8.034713e-004
15	0.0000000000	1.137634e-044
100	0.0000000000	1.137634e-044

For the second iteration equation we have:

$$\partial_t u_2^{(l)} - A(u_i)u_i - B(u_2^{(l)})u_2^{(l)} = 0, \tag{6.20}$$

as

$$F_2(u_{i+1}^{(l)}(t^{n+1})) = u_{i+1}^{(l)}(t^{n+1}) - u_{i+1}(t^n) - \Delta t\left(A(u_i(t^{n+1}))u_i(t^{n+1}) + B(u_{i+1}^{(l)}(t^{n+1}))u_{i+1}^{(l)}(t^{n+1})\right), \tag{6.21}$$

where we have the initialization of the Newton's method as $u_{i+1}^{(0)}(t^{n+1}) = 0$ or $u_{i+1}^{(0)}(t^{n+1}) = u(t^n)$.

The derivations are given as:

$$\begin{aligned} D(F_1(u_i^{(k)}(t^{n+1}))) &= 1 - \Delta t\left(A(u_{i+1}^{(k)}(t^{n+1})) + \frac{\partial A(u_{i+1}^{(k)}(t^{n+1}))}{\partial u_{i+1}^{(k)}(t^{n+1})} u_{i+1}^{(k)}(t^{n+1}) \right), \\ D(F_2(u_{i+1}^{(l)}(t^{n+1}))) &= 1 - \Delta t\left(B(u_{i+1}^{(l)}(t^{n+1})) + \frac{\partial B(u_{i+1}^{(l)}(t^{n+1}))}{\partial u_{i+1}^{(l)}(t^{n+1})} u_{i+1}^{(l)}(t^{n+1}) \right). \end{aligned} \tag{6.22}$$

Our numerical results for the different methods are presented in Tables 1, 2, 3, and 4. The errors of the methods are shown in Figures 1, 2, and 3. We chose different iteration steps and time partitions. The error between the analytical and numerical solution is shown with the supremum norm at time $T = 1.0$.

Table 3: Numerical results for the Bernoulli equation with iterative operator-splitting method.

Time partitions	Number of iter.	Approximated solution	Error
1	2	0.0125000000	1.250000e-002
1	4	0.2927814810	2.927815e-001
1	10	0.0109667158	1.096672e-002
1	50	0.0109556732	1.095567e-002
5	2	0.0109913109	1.099131e-002
5	4	0.3152826900	3.152827e-001
5	10	0.0108511723	1.085117e-002
5	50	0.0108509643	1.085096e-002
10	2	0.0108995483	1.089955e-002
10	4	0.2437741856	2.437742e-001
10	10	0.0108426328	1.084263e-002
10	50	0.0108426158	1.084262e-002
50	2	0.0149667882	1.496679e-002
50	4	0.0166913971	1.669140e-002
50	10	0.0157464111	1.574641e-002
50	50	0.0159933864	1.599339e-002
100	2	0.0154572223	1.545722e-002
100	4	0.0160048071	1.600481e-002
100	10	0.0158481781	1.584818e-002
100	50	0.0158673179	1.586732e-002

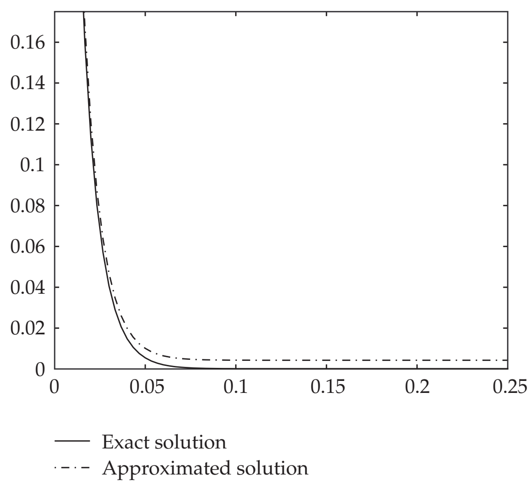

Figure 1: Analytical and approximated solution with sequential operator-splitting method.

The experiments show the reduced errors for more iteration steps and more time partitions. Because of the time-discretization method for ODEs, we restrict the number of iteration steps to a maximum of five. If we restrict the error bound to 10^{-3}, two iteration steps and five time partitions give the most effective combination.

Table 4: Numerical results for the Bernoulli equation with iterative operator-splitting method with embedded Newton's method.

Time partitions	Number of iter.	Approximated solution	Error
1	2	0.0000000000	1.137634e-044
1	4	0.0000000000	1.137634e-044
1	10	0.0000000000	1.137634e-044
1	20	0.0000000000	1.137634e-044
2	2	0.0000000000	1.137634e-044
2	4	0.0000000000	1.137634e-044
2	10	0.0000000000	1.137634e-044
2	20	0.0000000000	1.137634e-044

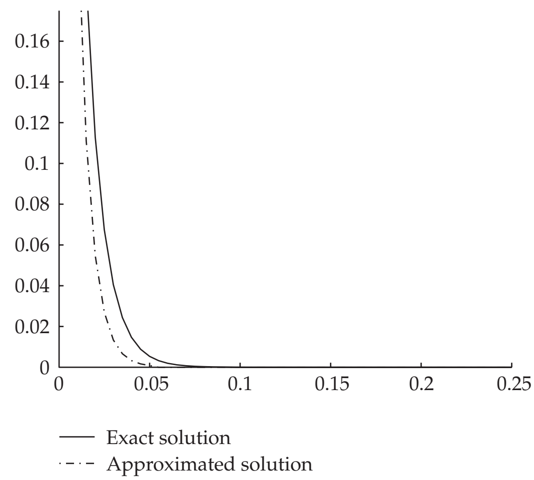

Figure 2: Analytical and approximated solution with sequential operator-splitting method with embedded Newton's method.

6.2. Second Numerical Example: 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:

$$\begin{aligned} \partial_t u &= -\frac{1}{2}u\partial_x u - \frac{1}{2}u\partial_y u - \frac{1}{2}\partial_x u - \frac{1}{2}\partial_y u \\ &\quad + \mu\left(\partial_{xx}u + \partial_{yy}u\right) + f(x,y,t), \quad (x,y,t) \in \Omega \times [0,T], \\ u(x,y,0) &= u_{\text{ana}}(x,y,0), \quad (x,y) \in \Omega, \\ u(x,y,t) &= u_{\text{ana}}(x,y,t) \quad \text{on } \partial\Omega \times [0,T], \end{aligned} \tag{6.23}$$

where $\Omega = [0,1] \times [0,1]$, $T = 1.25$, and μ is the viscosity.

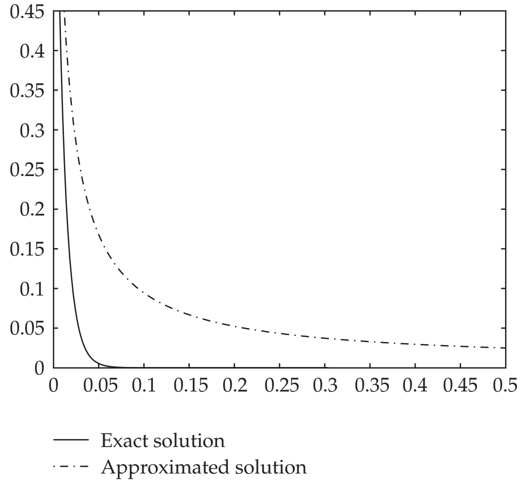

Figure 3: Analytical and approximated solution with iterative operator-splitting method.

The analytical solution is given as

$$u_{\text{ana}}(x, y, t) = \left(1 + \exp\left(\frac{x + y - t}{2\mu}\right)\right)^{-1} + \exp\left(\frac{x + y - t}{2\mu}\right), \tag{6.24}$$

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

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

$$A(u)u = -\frac{1}{2}u\partial_x u - \frac{1}{2}u\partial_y u + \frac{1}{2}\mu(\partial_{xx}u + \partial_{yy}u), \tag{6.25}$$

hence

$$\begin{aligned} A(u) &= \frac{1}{2}\left(-u\partial_x - u\partial_y + \mu(\partial_{xx} + \partial_{yy})\right) \quad \text{(the Burgers term)}, \\ Bu &= -\frac{1}{2}\partial_x u - \frac{1}{2}\partial_y u + \frac{1}{2}\mu(\partial_{xx}u + \partial_{yy}u) + f(x, y, t) \quad \text{(the convection-diffusion term)}. \end{aligned} \tag{6.26}$$

For the first equation we apply the nonlinear Algorithm 5.1 and obtain

$$\begin{aligned} A(u_{i-1})u_i &= -\frac{1}{2}u_{i-1}\partial_x u_i - \frac{1}{2}u_{i-1}\partial_y u_i + \frac{1}{2}\mu(\partial_{xx}u_i + \partial_{yy}u_i), \\ Bu_{i-1} &= \frac{1}{2}\left(-\partial_x - \partial_y + \mu(\partial_{xx} + \partial_{yy})\right)u_{i-1}, \end{aligned} \tag{6.27}$$

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

Table 5: 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.

$\Delta x = \Delta y$	Δt	err_{L_1}	$\mathrm{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/5	1/20	0.0137	0.0354		
1/10	1/20	0.0055	0.0139	1.3264	1.3499
1/20	1/20	0.0017	0.0043	1.6868	1.6900
1/40	1/20	$8.8839 \cdot 10^{-5}$	$3.8893 \cdot 10^{-4}$	4.2588	3.4663
1/5	1/40	0.0146	0.0377		
1/10	1/40	0.0064	0.0160	1.1984	1.2315
1/20	1/40	0.0026	0.0063	1.3004	1.3375
1/40	1/40	$8.2653 \cdot 10^{-4}$	0.0021	1.6478	1.6236

Table 6: 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.

$\Delta x = \Delta y$	Δt	err_{L_1}	$\mathrm{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/5	1/20	$1.3166 \cdot 10^{-5}$	$2.9819 \cdot 10^{-5}$		
1/10	1/20	$5.6944 \cdot 10^{-6}$	$1.3541 \cdot 10^{-5}$	1.2092	1.1389
1/20	1/20	$1.6986 \cdot 10^{-6}$	$4.5816 \cdot 10^{-6}$	1.7452	1.5634
1/40	1/20	$7.8145 \cdot 10^{-7}$	$2.0413 \cdot 10^{-6}$	1.1201	1.1663
1/5	1/40	$1.4425 \cdot 10^{-5}$	$3.2036 \cdot 10^{-5}$		
1/10	1/40	$7.2343 \cdot 10^{-6}$	$1.5762 \cdot 10^{-5}$	0.9957	1.0233
1/20	1/40	$3.0776 \cdot 10^{-6}$	$6.7999 \cdot 10^{-6}$	1.2330	1.2129
1/40	1/40	$9.8650 \cdot 10^{-7}$	$2.3352 \cdot 10^{-6}$	1.6414	1.5420

In the second equation we obtain by using Algorithm 5.1:

$$\begin{aligned} A(u_{i-1})u_i &= -\frac{1}{2}u_{i-1}\partial_x u_i - \frac{1}{2}u_{i-1}\partial_y u_i + \frac{1}{2}\mu(\partial_{xx}u_i + \partial_{yy}u_i), \\ Bu_{i+1} &= \frac{1}{2}\big(-\partial_x - \partial_y + \mu(\partial_{xx} + \partial_{yy})\big)u_{i+1}, \end{aligned} \tag{6.28}$$

and we have linear operators.

We deal with different viscosities μ as well as different step-sizes in time and space. We have the following results (see Tables 5 and 6).

Figure 4 presents the profile of the 2D linear and nonlinear convection-diffusion equation.

Remark 6.1. In the examples, we deal with more iteration steps to obtain higher-order convergence results. In the first test we have four iterative steps but a smaller viscosity ($\mu = 0.5$) such 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 two iteration steps. Here we see the loss of differentiability, that becomes stiff equation parts. To obtain the same results, we have to increase the number of iteration steps. Therefore we can show an improvement of the convergence order with respect to the iteration steps.

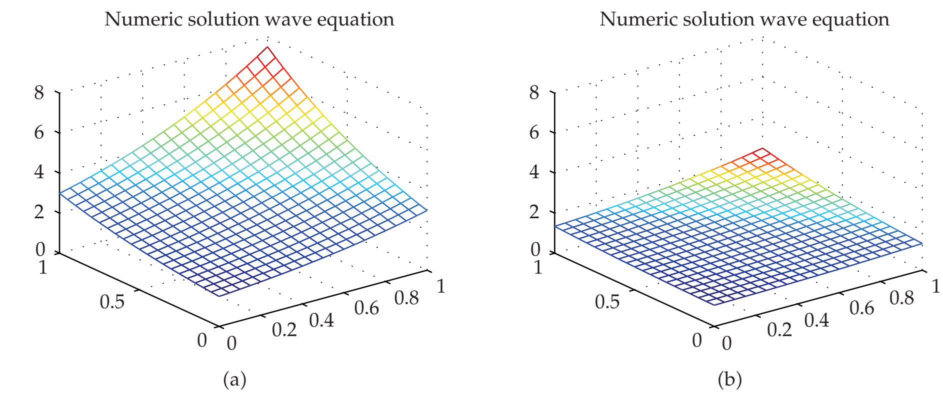

Figure 4: Mixed convection-diffusion and Burgers equation at initial time $t = 0.0$ (a) and end time $t = 1.25$ (b) for viscosity $\mu = 0.5$.

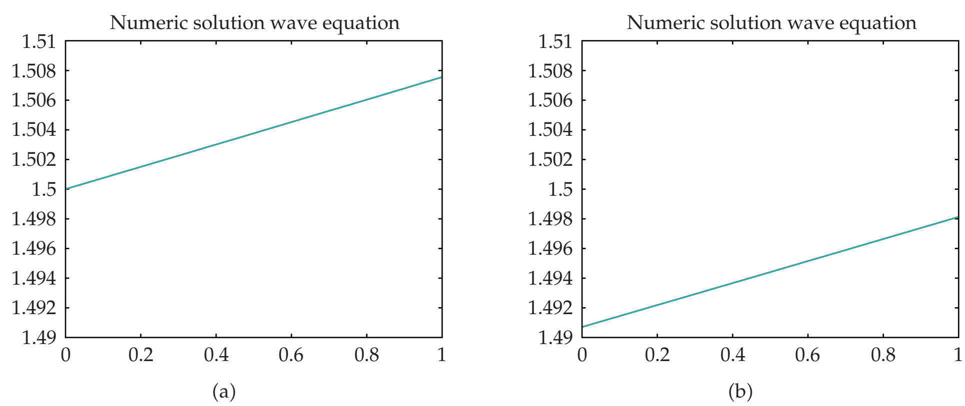

Figure 5: 1D momentum equation at initial time $t = 0.0$ (a) and end time $t = 1.25$ (b) for $\mu = 5$ and $v = 0.001$.

6.3. Third Numerical Example: Momentum Equation (Molecular Flow)

We deal with an example of a momentum equation, that is used to model the viscous flow of a fluid.

$$\begin{gathered}
\partial_t \mathbf{u} = -\mathbf{u} \cdot \nabla \mathbf{u} + 2\mu \nabla \left(D(\mathbf{u}) + \frac{1}{3} \nabla \mathbf{u} \right) + \mathbf{f}(x, y, t), \quad (x, y, t) \in \Omega \times [0, T], \\
\mathbf{u}(x, y, 0) = \mathbf{g}_1(x, y), \quad (x, y) \in \Omega, \\
\mathbf{u}(x, y, t) = \mathbf{g}_2(x, y, t) \quad \text{on } \partial\Omega \times [0, T], \text{ (enclosed flow)},
\end{gathered} \tag{6.29}$$

where $\mathbf{u} = (u_1, u_2)^t$ is the solution and $\Omega = [0, 1] \times [0, 1]$, $T = 1.25$, $\mu = 5$, and $\mathbf{v}= (0.001, 0.001)^t$ are the parameters and I is the unit matrix.

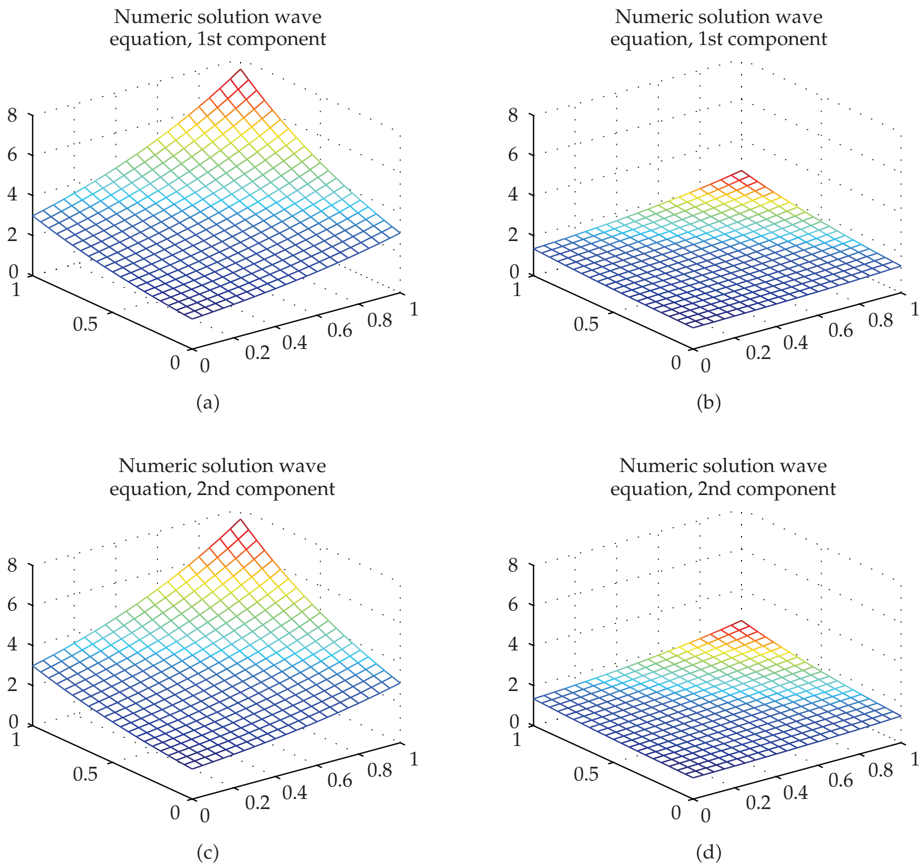

Figure 6: 2D momentum equation at initial time $t = 0.0$ ((a), (c)) and end time $t = 1.25$ ((b), (d)) for $\mu = 0.5$ and $v = (1,1)^t$ for the first and second component of the numerical solution.

The nonlinear function $D(\mathbf{u}) = \mathbf{u} \cdot \mathbf{u} + \mathbf{v} \cdot \mathbf{u}$ is the viscosity flow, and $\mathbf{v}$ is a constant velocity.

We can derive the analytical solution with respect to the first two test examples with the functions:

$$\begin{aligned} u_1(x,y,t) &= \left(1 + \exp\left(\frac{x+y-t}{2\mu}\right)\right)^{-1} + \exp\left(\frac{x+y-t}{2\mu}\right), \\ u_2(x,y,t) &= \left(1 + \exp\left(\frac{x+y-t}{2\mu}\right)\right)^{-1} + \exp\left(\frac{x+y-t}{2\mu}\right). \end{aligned} \tag{6.30}$$

For the splitting method our operators are given as:

$$\begin{aligned} A(\mathbf{u})\mathbf{u} &= -\mathbf{u}\nabla\mathbf{u} + 2\mu\nabla D(\mathbf{u}) \quad \text{(the nonlinear operator)}, \\ B\mathbf{u} &= \frac{2}{3}\mu\Delta\mathbf{u} \quad \text{(the linear operator)}. \end{aligned} \tag{6.31}$$

We deal first with the one-dimensional case,

$$\begin{aligned}
\partial_t u &= -u \cdot \partial_x u + 2\mu \partial_x \left(D(u) + \frac{1}{3} \partial_x u \right) + f(x,t), \quad (x,t) \in \Omega \times [0,T], \\
u(x,0) &= g_1(x), \quad (x) \in \Omega, \\
u(x,t) &= g_2(x,t) \quad \text{on } \partial\Omega \times [0,T], \text{ (enclosed flow)},
\end{aligned} \tag{6.32}$$

where u is the solution and $\Omega = [0,1]$, $T = 1.25$, $\mu = 5$, and $v = 0.001$ are the parameters.

Then the operators are given as:

$$\begin{aligned}
A(u)u &= -u\partial_x u + 2\mu \partial_x D(u) \quad \text{(the nonlinear operator)}, \\
Bu &= \frac{2}{3}\mu \partial_{xx} u \quad \text{(the linear operator)}.
\end{aligned} \tag{6.33}$$

For the iterative operator-splitting as fixed point scheme, we have the following results (see Tables 7 and 9).

Figure 5 presents the profile of the 1D momentum equation.

We have the following results for the 2D case (see Tables 10, 11, and 12).

Figure 6 presents the profile of the 2D momentum equation.

For the Newton operator-splitting method we obtain the following functional matrices for the one-dimensional case:

$$\begin{aligned}
DF(u) &= (4\mu - 1)\partial_x u, \\
D(F(u)) &= -\begin{pmatrix} \partial_{u_1} F_1(u) & \partial_{u_2} F_1(u) \\ \partial_{u_1} F_2(u) & \partial_{u_2} F_2(u) \end{pmatrix} \\
&= -\begin{pmatrix} -\partial_x u_1 + 4\mu \partial_x u_1 & -\partial_x u_2 + 4\mu \partial_x u_2 \\ -\partial_y u_1 + 4\mu \partial_y u_1 & -\partial_y u_2 + 4\mu \partial_y u_2 \end{pmatrix} \\
&= (4\mu - 1)\nabla u.
\end{aligned} \tag{6.34}$$

For the two-dimensional case, we use:

$$\begin{aligned}
A(\mathbf{u})\mathbf{u} &= -\mathbf{u}\nabla\mathbf{u} + 2\mu \nabla D(\mathbf{u}) \\
&= -\begin{pmatrix} u_1 \partial_x u_1 + u_2 \partial_x u_2 \\ u_1 \partial_y u_1 + u_2 \partial_y u_2 \end{pmatrix} + 2\mu \begin{pmatrix} 2u_1 \partial_x u_1 + 2u_2 \partial_x u_2 + v_1 \partial_x u_1 + v_2 \partial_x u_2 \\ 2u_1 \partial_y u_1 + 2u_2 \partial_y u_2 + v_1 \partial_y u_1 + v_2 \partial_y u_2 \end{pmatrix}.
\end{aligned} \tag{6.35}$$

Here, we do not need the linearization and apply the standard iterative splitting method.

We only linearize the first split step and therefore we can relax this step with the second linear split step. Therefore we obtain stable methods, see [15]. For the Newton iterative method, we have the following results, see Table 8.

Table 7: Numerical results for the 1D momentum equation with $\mu = 5, v = 0.001$, initial condition $u_0(t) = c_n$, and two iterations per time-step.

Δx	Δt	err_{L_1}	$\text{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/10	1/20	0.0213	0.0495		
1/20	1/20	0.0203	0.0470	0.0689	0.0746
1/40	1/20	0.0198	0.0457	0.0401	0.0402
1/80	1/20	0.0195	0.0450	0.0216	0.0209
1/10	1/40	0.0134	0.0312		
1/20	1/40	0.0117	0.0271	0.1957	0.2009
1/40	1/40	0.0108	0.0249	0.1213	0.1211
1/80	1/40	0.0103	0.0238	0.0682	0.0674
1/10	1/80	0.0094	0.0217		
1/20	1/80	0.0073	0.0169	0.3591	0.3641
1/40	1/80	0.0062	0.0143	0.2451	0.2448
1/80	1/80	0.0056	0.0129	0.1478	0.1469

Table 8: Numerical results for the 1D momentum equation with $\mu = 5, v = 0.001$, initial condition $u_0(t) = c_n$, two iterations per time-step and $K = 1$ using Newton iterative method.

Δx	Δt	err_{L_1}	$\text{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/10	1/20	0.0180	0.0435		
1/20	1/20	0.0120	0.0276	0.5867	0.6550
1/40	1/20	0.0095	0.0227	0.3311	0.2870
1/80	1/20	0.0085	0.0208	0.1706	0.1231
1/10	1/40	0.0172	0.0459		
1/20	1/40	0.0125	0.0305	0.4652	0.5884
1/40	1/40	0.0108	0.0253	0.2366	0.2698
1/80	1/40	0.0097	0.0235	0.1191	0.1111
1/10	1/80	0.0166	0.0475		
1/20	1/80	0.0132	0.0338	0.3327	0.4917
1/40	1/80	0.0119	0.0280	0.1640	0.2734
1/80	1/80	0.0112	0.0265	0.0802	0.0779

Remark 6.2. In the more realistic examples of 1D and 2D momentum equations, we can also observe the stiffness problem, which we obtain with a more hyperbolic behavior. In the 1D experiments we deal with a more hyperbolic behavior and can obtain at least first-order convergence with two iterative steps. In the 2D experiments we obtain nearly second-order convergence results with two iterative steps, if we increase the parabolic behavior, for example, larger μ and $\mathbf{v}$ values. For such methods, we have to balance the usage of the iterative steps with refinement in time and space with respect to the hyperbolicity of the equations. At least we can obtain a second-order method with more than two iterative steps. Therefore the stiffness influences the number of iterative steps.

Table 9: Numerical results for the 1D momentum equation with $\mu = 50$, $v = 0.1$, initial condition $u_0(t) = c_n$, and two iterations per time-step.

Δx	Δt	err_{L_1}	$\text{err}_{\max}$	ρ_{L_1}	$\rho_{\max}$
1/10	1/20	$2.7352 \cdot 10^{-6}$	$6.4129 \cdot 10^{-6}$		
1/20	1/20	$2.3320 \cdot 10^{-6}$	$5.4284 \cdot 10^{-6}$	0.2301	0.2404
1/40	1/20	$2.1144 \cdot 10^{-6}$	$4.9247 \cdot 10^{-6}$	0.1413	0.1405
1/80	1/20	$2.0021 \cdot 10^{-6}$	$4.6614 \cdot 10^{-6}$	0.0787	0.0793
1/10	1/40	$2.1711 \cdot 10^{-6}$	$5.2875 \cdot 10^{-6}$		
1/20	1/40	$1.7001 \cdot 10^{-6}$	$4.1292 \cdot 10^{-6}$	0.3528	0.3567
1/40	1/40	$1.4388 \cdot 10^{-6}$	$3.4979 \cdot 10^{-6}$	0.2408	0.2394
1/80	1/40	$1.3023 \cdot 10^{-6}$	$3.1694 \cdot 10^{-6}$	0.1438	0.1423
1/10	1/80	$1.6788 \cdot 10^{-6}$	$4.1163 \cdot 10^{-6}$		
1/20	1/80	$1.1870 \cdot 10^{-6}$	$2.9138 \cdot 10^{-6}$	0.5001	0.4984
1/40	1/80	$9.1123 \cdot 10^{-7}$	$2.2535 \cdot 10^{-6}$	0.3814	0.3707
1/80	1/80	$7.6585 \cdot 10^{-7}$	$1.9025 \cdot 10^{-6}$	0.2507	0.2443

Table 10: Numerical results for the 2D momentum equation with $\mu = 2$, $v = (1, 1)^t$, initial condition $u_0(t) = c_n$, and two iterations per time-step.

$\Delta x = \Delta y$	Δt	err_{L_1} 1st c.	$\text{err}_{\max}$ 1st c.	ρ_{L_1} 1st c.	$\rho_{\max}$ 1st c.	err_{L_1} 2nd c.	$\text{err}_{\max}$ 2nd c.	ρ_{L_1} 2nd c.	$\rho_{\max}$ 2nd c.
1/5	1/20	0.0027	0.0112			0.0145	0.0321		
1/10	1/20	0.0016	0.0039	0.7425	1.5230	0.0033	0.0072	2.1526	2.1519
1/20	1/20	0.0007	0.0022	1.2712	0.8597	0.0021	0.0042	0.6391	0.7967
1/5	1/40	0.0045	0.0148			0.0288	0.0601		
1/10	1/40	0.0032	0.0088	0.5124	0.7497	0.0125	0.0239	1.2012	1.3341
1/20	1/40	0.0014	0.0034	1.1693	1.3764	0.0029	0.0054	2.1263	2.1325
1/5	1/80	0.0136	0.0425			0.0493	0.1111		
1/10	1/80	0.0080	0.0241	0.7679	0.8197	0.0278	0.0572	0.8285	0.9579
1/20	1/80	0.0039	0.0113	1.0166	1.0872	0.0115	0.0231	1.2746	1.3058

Table 11: Numerical results for the 2D momentum equation for the first component with $\mu = 50$, $v = (100, 0.01)^t$, initial condition $u_0(t) = c_n$, and two iterations per time-step.

$\Delta x = \Delta y$	Δt	err_{L_1} 1st c.	$\text{err}_{\max}$ 1st c.	ρ_{L_1} 1st c.	$\rho_{\max}$ 1st c.
1/5	1/20	$1.5438 \cdot 10^{-5}$	$3.4309 \cdot 10^{-5}$		
1/10	1/20	$4.9141 \cdot 10^{-6}$	$1.0522 \cdot 10^{-5}$	1.6515	1.7052
1/20	1/20	$1.5506 \cdot 10^{-6}$	$2.9160 \cdot 10^{-6}$	1.6641	1.8513
1/5	1/40	$2.8839 \cdot 10^{-5}$	$5.5444 \cdot 10^{-5}$		
1/10	1/40	$1.3790 \cdot 10^{-5}$	$2.3806 \cdot 10^{-5}$	1.0645	1.2197
1/20	1/40	$3.8495 \cdot 10^{-6}$	$6.8075 \cdot 10^{-6}$	1.8408	1.8061
1/5	1/80	$3.1295 \cdot 10^{-5}$	$5.5073 \cdot 10^{-5}$		
1/10	1/80	$1.7722 \cdot 10^{-5}$	$2.6822 \cdot 10^{-5}$	0.8204	1.0379
1/20	1/80	$7.6640 \cdot 10^{-6}$	$1.1356 \cdot 10^{-5}$	1.2094	1.2400

Table 12: Numerical results for the 2D momentum equation for the second component with $\mu = 50$, $v = (100, 0.01)^t$, initial condition $u_0(t) = c_n$, and two iterations per time-step.

$\Delta x = \Delta y$	Δt	err_{L_1} 2nd c.	$\mathrm{err}_{\max}$ 2nd c.	ρ_{L_1} 2nd c.	$\rho_{\max}$ 2nd c.
1/5	1/20	$4.3543 \cdot 10^{-5}$	$1.4944 \cdot 10^{-4}$		
1/10	1/20	$3.3673 \cdot 10^{-5}$	$7.9483 \cdot 10^{-5}$	0.3708	0.9109
1/20	1/20	$2.6026 \cdot 10^{-5}$	$5.8697 \cdot 10^{-5}$	0.3717	0.4374
1/5	1/40	$3.4961 \cdot 10^{-5}$	$2.2384 \cdot 10^{-4}$		
1/10	1/40	$1.7944 \cdot 10^{-5}$	$8.9509 \cdot 10^{-5}$	0.9622	1.3224
1/20	1/40	$1.5956 \cdot 10^{-5}$	$3.6902 \cdot 10^{-5}$	0.1695	1.2783
1/5	1/80	$9.9887 \cdot 10^{-5}$	$3.3905 \cdot 10^{-4}$		
1/10	1/80	$3.5572 \cdot 10^{-5}$	$1.3625 \cdot 10^{-4}$	1.4896	1.3153
1/20	1/80	$1.0557 \cdot 10^{-5}$	$4.4096 \cdot 10^{-5}$	1.7525	1.6275

7. Conclusion and Discussion

We present decomposition methods for differential equations based on iterative and non-iterative methods. The nonlinear equations are solved with embedded Newton's methods. We present new ideas on linearization to obtain more accurate results. The superiority of the new embedded Newton's methods over the traditional sequential methods is demonstrated in examples, especially through their simple implementation. Further, we have the smoothing properties of the iterative scheme that allow a balance between the nonlinear and the linear terms. The results show more accurate solutions with respect to time decomposition. In the future the iterative operator-splitting method can be generalized for multi-dimensional problems and also for non-smooth and nonlinear problems in time and space. In next paper we discuss error analysis of nonlinear methods.

References

[1] G. I. Marchuk, "Some applicatons of splitting-up methods to the solution of problems in mathematical physics," *Aplikace Matematiky*, vol. 1, pp. 103–132, 1968.

[2] G. Strang, "On the construction and comparison of difference schemes," *SIAM Journal on Numerical Analysis*, vol. 5, no. 3, pp. 506–517, 1968.

[3] H. Yoshida, "Construction of higher order symplectic integrators," *Physics Letters A*, vol. 150, no. 5–7, pp. 262–268, 1990.

[4] J. F. Kanney, C. T. Miller, and C. T. Kelley, "Convergence of iterative split-operator approaches for approximating nonlinear reactive problems," *Advances in Water Resources*, vol. 26, no. 3, pp. 247–261, 2003.

[5] P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithm*, vol. 35 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2004.

[6] P. Deuflhard, "A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting," *Numerische Mathematik*, vol. 22, no. 4, pp. 289–315, 1974.

[7] M. Ruzicka, *Nichtlineare Funktionalanalysis*, Springer, Berlin, Germany, 2004.

[8] E. Zeidler, *Nonlinear Functional Analysis and Its Applications. II/B: Nonlinear Monotone Operators*, Springer, New York, NY, USA, 1990.

[9] J. Geiser, O. Klein, and P. Philip, "Anisotropic thermal conductivity in apparatus insulation: numerical study of effects on the temperature field during sublimation growth of silicon carbide single crystals," preprint no. 1034 of Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany, 2005.

[10] J. Geiser, "Discretization methods with embedded analytical solutions for convection dominated transport in porous media," in *Proceedings of the 3rd International Conference on Numerical Analysis and Its Applications (NAA '04)*, vol. 3401 of *Lecture Notes in Computer Science*, pp. 288–295, Springer, Rousse, Bulgaria, June-July 2004.

[11] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, vol. 33 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2003.

[12] K. H. Karlsen, K.-A. Lie, J. R. Natvig, H. F. Nordhaug, and H. K. Dahle, "Operator splitting methods for systems of convection-diffusion equations: nonlinear error mechanisms and correction strategies," *Journal of Computational Physics*, vol. 173, no. 2, pp. 636–663, 2001.

[13] K. H. Karlsen and N. H. Risebro, "Corrected operator splitting for nonlinear parabolic equations," *SIAM Journal on Numerical Analysis*, vol. 37, no. 3, pp. 980–1003, 2000.

[14] I. Faragó and J. Geiser, "Iterative operator-splitting methods for linear problems," *International Journal of Computational Science and Engineering*, vol. 3, no. 4, pp. 255–263, 2007.

[15] C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics, SIAM, Philadelphia, Pa, USA, 1995.

[16] J. Geiser, "Iterative operator-splitting methods with higher-order time integration methods and applications for parabolic partial differential equations," *Journal of Computational and Applied Mathematics*, vol. 217, no. 1, pp. 227–242, 2008.

[17] K. H. Karlsen and N. H. Risebro, "An operator splitting method for nonlinear convection-diffusion equations," *Numerische Mathematik*, vol. 77, no. 3, pp. 365–382, 1997.