Modified Jacobian Newton Iterative Method with Embedded Domain Decomposition Method

Jürgen Geiser [a,*], Christos Kravvaritis [b]

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

[b] *Department of Mathematics, University of Athens, Panepistimiopolis 15784, Athens, Greece*

Abstract

In this article a new approach is proposed for constructing a domain decomposition method based on the iterative operator-splitting method for nonlinear differential equations. The convergence properties of such a method are studied. The main feature of the proposed idea are the linearization of the nonlinear equations and the application of iterative splitting methods. We present iterative operator-splitting method with embedded Newton methods to solve the 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.

Key words: numerical analysis; operator-splitting method; initial value problems; iterative solver method; nonlinear equations

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 [20] and [17]. Further important research was done to obtain a higher order for the splitting methods (see [21]). For this reason, the iterative splitting methods became more important for linear and nonlinear

* Corresponding author.

Email addresses: geiser@mathematik.hu-berlin.de (Jürgen Geiser), ckrav@math.uoa.gr (Christos Kravvaritis).

differential equations, while simple increasing of iteration steps affects the order of the scheme (see [24]).

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 discuss 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 [12] and convection-diffusion-reaction-equations [8,15,16,13].

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

$$\partial_t u(t) = A(u(t))\, u(t) \; + \; B(u(t))\, u(t) \; , \quad t \in (0, T), \tag{1}$$

where the initial condition is $u(0) = u_0$. The operators $A(u)$ and $B(u)$ can be spatially discretized operators, i.e. 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:

$$\partial_t u(t) = A\, u(t) \; + \; B\, u(t) \; , \quad t \in (0, T), \tag{2}$$

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 (2) with the subproblems

$$\begin{aligned} \frac{\partial u^*(t)}{\partial t} &= A u^*(t) \; , \quad \text{with } u^*(t^n) = u^n \; , \\ \frac{\partial u^{**}(t)}{\partial t} &= B u^{**}(t) \; , \quad \text{with } u^{**}(t^n) = u^*(t^{n+1}) \; , \end{aligned}$$

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 some error, called *splitting error*. The splitting error of the sequential operator-splitting method can be derived as (cf. e.g. [17], [20])

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

$$= \begin{cases} 0 & , \text{ for } [A,B]=0 \;, \\ O(\tau_n) & , \text{ for } [A,B]\neq 0 \;, \end{cases}$$

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 $i = 0, 2, \ldots 2m$.

$$\frac{\partial u_i(x,t)}{\partial t} = Au_i(x,t) \;+\; Bu_{i-1}(x,t), \text{ with } \; u_i(t^n) = u^n \tag{3}$$

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

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

$$\frac{\partial u_{i+1}(x,t)}{\partial t} = \; Au_i(x,t) \;+\; Bu_{i+1}(x,t), \tag{4}$$

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

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

where u^n is the known split approximation at the time level $t = t^n$ (see [7]).

Remark 1 *We can generalize the iterative splitting method to a multi-iterative splitting method by introducing new splitting operators, e.g. 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, e.g. quadratic nonlinearity, the fixpoint iteration method, where our iterative operator splitting method is one, see [24]. For stronger nonlinearities, e.g. cubic or higher order polynomial nonlinearities, the modified Jacobian method with embedded iterative-splitting methods.

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

5.1. *The altered Jacobian-Newton iterative methods with embedded sequential splitting methods*

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

$$\frac{dc}{dt} = A(c(t))c(t) + B(c(t))c(t), \text{with } c(t^n) = c^n, \tag{5}$$

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 [27]. 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:

Newton's method:

$F(c) = \frac{dc}{dt} - A(c(t))c(t) - B(c(t))c(t)$ 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 : $|c^{(k+1)} - c^{(k)}| \leq err$, where err is an error bound, e.g. $err = 10^{-4}$.

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

$$F(c) = \begin{pmatrix} F(c_1) \\ F(c_2) \\ \vdots \\ F(c_m) \end{pmatrix} \tag{6}$$

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 :

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

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

The modified Jacobian is given as :

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

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

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

$$F_1(u_1) = \partial_t u_1 - A(u_1)u_1 = 0 \quad \text{with} \quad u_1(t^n) = c^n, \tag{7}$$

$$F_2(u_2) = \partial_t u_2 - B(u_2)u_2 = 0 \quad \text{with} \quad u_2(t^n) = u_1(t^{n+1}), \tag{8}$$

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 at least we have to solve two Newton methods, each in one equations system. The contribution is to reduce the Jacobian matrix into a diagonal entries, e.g. with a weighted Newton method, see [26]. The splitting method with embedded Newton method is given for the continuous method as:

$$u_1^{(k+1)} = u_1^{(k)} - D(F_1(u_1^{(k)}))^{-1}(\partial_t u_1^{(k)} - A(u_1^{(k)})u_1^{(k)}), \tag{9}$$

$$\text{with } D(F_1(u_1^{(k)})) = \frac{\partial}{\partial u_1^{(k)}}(\partial_t u_1^{(k)} - A(u_1^{(k)}) - \frac{\partial A(u_1^{(k)})}{\partial u_1^{(k)}} u_1^{(k)}), \tag{10}$$

$$u_1^{(k)}(t^n) = c^n \text{ and } k = 0, 1, 2, \dots, K, \tag{11}$$

$$u_2^{(l+1)} = u_2^{(l)} - D(F_2(u_2^{(l)}))^{-1}(\partial_t u_2^{(l)} - B(u_2^{(l)})u_2^{(l)}), \tag{12}$$

$$\text{with } D(F_2(u_2^{(l)})) = \frac{\partial}{\partial u_1^{(k)}}(\partial_t u_2^{(k)} - B(u_2^{(l)}) - \frac{\partial B(u_2^{(l)})}{\partial u_2^{(l)}} u_2^{(l)}), \tag{13}$$

$$u_2^{(l)}(t^n) = u_1^K(t^{n+1}) \text{ and } l = 0, 1, 2, \dots, L. \tag{14}$$

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:

$$u_1^{(k+1)} = u_1^{(k)} - (D(F_1(u_1^{(k)})) + \delta_1(u_1^{(k)}))^{-1}(F_1(u_1^{(k)}) + \epsilon\, u_1^{(k)}), \tag{15}$$

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

Remark 2 *If we assume that we discretize the equation (7) and (8) with the backward-Euler method, e.g.:*

$$F_1(u_1(t^{n+1})) = u_1(t^{n+1}) - u_1(t^n) - \Delta t A(u_1(t^{n+1}))u_1(t^{n+1}) = 0 \quad \textit{with} \quad u_1(t^n) = c^n,$$
$$F_2(u_2) = u_2(t^{n+1}) - u_2(t^n) - \Delta t B(u_2(t^{n+1}))u_2(t^{n+1}) = 0 \quad \textit{with} \quad u_2(t^n) = u_1(t^{n+1}),$$

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

$$D(F_1(u_1(t^{n+1}))) = 1 - \Delta t(A(u_1(t^{n+1})) + \frac{\partial A(u_1(t^{n+1}))}{\partial u_1(t^{n+1})} u_1(t^{n+1})),$$
$$D(F_2(u_2)) = 1 - \Delta t(B(u_2(t^{n+1})) + \frac{\partial B(u_2(t^{n+1}))}{\partial u_2(t^{n+1})} u_2(t^{n+1})),$$

We can apply the equation (15) 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 [23] and [24].

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

$$\frac{du}{dt} = A(u(t))u(t) + B(u(t))u(t), \text{with } u(t^n) = c^n, \tag{16}$$

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 [27]. 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 (16) by applying:

$$\frac{du_i(t)}{dt} = A(u_{i-1}(t))u_i(t) \ + \ B(u_{i-1}(t))u_{i-1}(t), \text{ with } \ u_i(t^n) = c^n, \tag{17}$$
$$\frac{du_{i+1}(t)}{dt} = A(u_{i-1}(t))u_i(t) \ + \ B(u_{i-1}(t))u_{i+1}, \text{ with } \ u_{i+1}(t^n) = c^n, \tag{18}$$

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \dots, 2m+1$. $u_0(t) = c_n$ 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 $i = 1, 3, \dots, 2m+1$.
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

$$||A(u_{i-1}(t^{n+1}))u_i(t^{n+1}) - A(u^{n+1})u(t^{n+1})|| \leq \epsilon,$$

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

Remark 3 *The linearization with the fixpoint scheme can be used for smooth or weak nonlinear operators, otherwise we loose the convergence behavior, while we did not converge to the local fixpoint, see [24].*

The second ideas is based on the Newton method.

5.3. *Jacobian-Newton iterative method with embedded operator-splitting method*

Newton method is used to solve the nonlinear parts of the iterative operator-splitting method, see the linearization techniques in [24],[25].

Newton method:

The function is given as:

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

The iteration can be computed as:

$c^{(k+1)} = c^{(k)} - D(F(c^{(k)}))^{-1}F(c^{(k)}),$

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

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

$$F_1(u_i) = \partial_t u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0, \tag{19}$$

$$\text{with} \quad u_i(t^n) = c^n, \tag{20}$$

$$F_2(u_{i+2}) = \partial_t u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0, \tag{21}$$

$$\text{with} \quad u_{i+1}(t^n) = c^n, \tag{22}$$

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 at least we have to solve two Newton methods and the contributions will be to reduce the Jacobian matrix, e.g. skip the diagonal entries. The splitting method with the embedded Newton method is given as:

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

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

$$\text{and } k = 0, 1, 2, \ldots, K,$$

$$\text{with} \quad u_i(t^n) = c^n,$$

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

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

$$\text{and } l = 0, 1, 2, \ldots, L,$$

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

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 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 to skip 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 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, the second iteration is the iterative splitting method, which compute 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*

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

$$\frac{\partial u(t)}{\partial t} = (\lambda_1 + \lambda_3)u(t) + (\lambda_2 + \lambda_4)(u(t))^p,\ t \in [0, T], \text{with } u(0) = 1, \tag{23}$$

where the analytical solution can be derived, see [23], as:

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

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

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

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:

$$u(t)^{1-p} = u_0 \exp((1-p)(\lambda_1 + \lambda_3)t) + \frac{\lambda_2 + \lambda_4}{\lambda_1 + \lambda_3}(\exp((1-p)(\lambda_1 + \lambda_3)t) - 1)\ . \tag{24}$$

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

0.) The sequential operator-splitting method with analytical solutions is given as:

We apply the quasilinear iterative operator-splitting method:

$$\frac{du_1(t)}{dt} = A(u_1(t))u_1(t), \text{ with } u_1(t^n) = u^n,$$
$$\frac{du_2(t)}{dt} = B(u_2(t))u_2, \text{ with } u_2(t^n) = u_1^{n+1},$$

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, as Backward Euler or higher Runge-Kutta methods.

The analytical results for each equation part is given as:

$$u_1(t)^{1-p} = u(t^n)\exp((1-p)(\lambda_1)t) + \frac{\lambda_2}{\lambda_1}(\exp((1-p)(\lambda_1)t) - 1) \ . \tag{25}$$

$$u_2(t^{n+1})^{1-p} = u_1(t^{n+1})\exp((1-p)(\lambda_3)t) + \frac{\lambda_4}{\lambda_3}(\exp((1-p)(\lambda_3)t) - 1) \ . \tag{26}$$

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.

1.) The sequential operator-splitting method with embedded Newton method is given as:

We apply the quasilinear iterative operator-splitting method:

$$\frac{du_1(t)}{dt} = A(u_1(t))u_1(t), \text{ with } u_1(t^n) = u^n,$$
$$\frac{du_2(t)}{dt} = B(u_2(t))u_2, \text{ with } u_2(t^n) = u_1^{n+1},$$

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, as Backward Euler or higher Runge-Kutta methods.

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

$$u_1^{(k+1)} = u_1^{(k)} - D(F_1(u_1^{(k)}))^{-1}(\partial_t u_1^{(k)} - A(u_1^{(k)})u_1^{(k)}), \tag{27}$$

$$\text{with } D(F_1(u_1^{(k)})) = -(A(u_1^{(k)}) + \frac{\partial A(u_1^{(k)})}{\partial u_1^{(k)}} u_1^{(k)}), \tag{28}$$

$$u_1^{(k)}(t^n) = c^n \text{ and } k = 0, 1, 2, \ldots, K, \tag{29}$$

$$u_2^{(l+1)} = u_2^{(l)} - D(F_2(u_2^{(l)}))^{-1}(\partial_t u_2^{(l)} - B(u_2^{(l)})u_2^{(l)}), \tag{30}$$

$$\text{with } D(F_2(u_2^{(l)})) = -(B(u_2^{(l)}) + \frac{\partial B(u_2^{(l)})}{\partial u_2^{(l)}} u_2^{(l)}), \tag{31}$$

$$u_2^{(l)}(t^n) = u_1^K(t^{n+1}) \text{ and } l = 0, 1, 2, \ldots, L. \tag{32}$$

where we discretize the equations and obtain the discretised operators :

$$\partial_t u_1^{(k)} - A(u_1^{(k)})u_1^{(k)} = 0 \tag{33}$$

as

$$F_1(u_1(t^{n+1}))u_1^{(k)}(t^{n+1}) - u_1(t^n) - \Delta t A(u_1^{(k)}(t^{n+1}))u_1^{(k)}(t^{n+1}) = 0 \ , \tag{34}$$

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:

$$\partial_t u_2^{(l)} - B(u_2^{(l)})u_2^{(l)} \tag{35}$$

as

$$F_2(u_2(t^{n+1})) = u_2^{(l)}(t^{n+1}) - u_2(t^n) - \Delta t B(u_2^{(l)}(t^{n+1}))u_2^{(l)}(t^{n+1}) = 0 \ , \tag{36}$$

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 :

$$D(F_1(u_1(t^{n+1}))) = 1 - \Delta t(A(u_1(t^{n+1})) + \frac{\partial A(u_1(t^{n+1}))}{\partial u_1(t^{n+1})}u_1(t^{n+1})),$$
$$D(F_2(u_2)) = 1 - \Delta t(B(u_2(t^{n+1})) + \frac{\partial B(u_2(t^{n+1}))}{\partial u_2(t^{n+1})}u_2(t^{n+1})).$$

2.) The standard iterative operator-splitting method is given as :

We apply the quasilinear iterative operator-splitting method:

$$\frac{du_i(t)}{dt} = A(u_{i-1}(t))u_i(t) \ + \ B(u_{i-1}(t))u_{i-1}(t), \text{ with } \ u_i(t^n) = u^n,$$
$$\frac{du_{i+1}(t)}{dt} = A(u_{i-1}(t))u_i(t) \ + \ B(u_{i-1}(t))u_{i+1}, \text{ with } \ u_{i+1}(t^n) = u^n,$$

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 equations:

$$u_i(t) = u^n \exp(A(u_{i-1}(t))t)$$
$$+(A(u_{i-1}(t)))^{-1}(B(u_{i-1}(t))u_{i-1}(t))(1 - \exp(A(u_{i-1}(t))t)), \tag{37}$$
$$u_{i+1}(t) = u^n \exp(B(u_{i-1}(t))t)$$
$$+(B(u_{i-1}(t)))^{-1}(A(u_{i-1}(t))u_{i-1}(t))(1 - \exp(B(u_{i-1}(t))t)), \tag{38}$$

Further the iterative steps can be done.

3.) Newton iterative method with embedded iterative operator-splitting method is given as :

We apply the quasilinear iterative operator-splitting method:

$$\frac{du_i(t)}{dt} = A(u_i(t))u_i(t) \ + \ B(u_{i-1}(t))u_{i-1}(t), \text{ with } \ u_i(t^n) = u^n,$$
$$\frac{du_{i+1}(t)}{dt} = A(u_i(t))u_i(t) \ + \ B(u_{i+1}(t))u_{i+1}, \text{ with } \ u_{i+1}(t^n) = u^n,$$

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

$$
\begin{aligned}
& u_i^{(k+1)} = u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i-1})u_{i-1}), \\
& \text{with } D(F_1(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}} u_i^{(k)}), \\
& \text{and } k = 0, 1, 2, \ldots, K, \\
& \text{with } \quad u_i(t^n) = c^n, \\
& \text{with } \quad u_i(t^{n+1}) = u_i(t^{n+1})^{K+1}, \text{where } |u_i(t^{n+1})^{K+1} - u_i(t^{n+1})^K| \leq err
\end{aligned}
$$

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

where 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 :

$$\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i-1})u_{i-1} = 0 \ , \tag{39}$$

as

$$F_1(u_i^{(k)}(t^{n+1})) = u_i^{(k)}(t^{n+1}) - u_i(t^n) - \Delta t(A(u_i^{(k)}(t^{n+1}))u_i^{(k)}(t^{n+1}) + B(u_{i-1}(t^{n+1}))u_{i-1}(t^{n+1})), \tag{40}$$

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

For the second iteration equation we have:

$$\partial_t u_2^{(l)} - A(u_i)u_i - B(u_2^{(l)})u_2^{(l)} = 0 \tag{41}$$

as

$$F_2(u_{i+1}^{(l)}(t^{n+1})) = u_{i+1}^{(l)}(t^{n+1}) - u_{i+1}(t^n)$$
$$-\Delta t \left(A(u_i(t^{n+1}))u_i(t^{n+1}) + B(u_{i+1}^{(l)}(t^{n+1}))u_{i+1}^{(l)}(t^{n+1}) \right) \quad (42)$$

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 :

$$D(F_1(u_i^{(k)}(t^{n+1}))) = 1 - \Delta t(A(u_{i+1}^{(k)}(t^{n+1})) + \frac{\partial A(u_{i+1}^{(k)}(t^{n+1}))}{\partial u_{i+1}^{(k)}(t^{n+1})} u_{i+1}^{(k)}(t^{n+1})),$$
$$D(F_2(u_{i+1}^{(l)}(t^{n+1}))) = 1 - \Delta t(B(u_{i+1}^{(l)}(t^{n+1})) + \frac{\partial B(u_{i+1}^{(l)}(t^{n+1}))}{\partial u_{i+1}^{(l)}(t^{n+1})} u_{i+1}^{(l)}(t^{n+1})),$$

Our numerical results for the different methods are presented in the Tables 1-4. The errors of the methods are shown in Figure 1-3. 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 1
Numerical results for the Bernoulli equation with sequential operator-splitting method.

time part.	approx. 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 part.	approx. 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

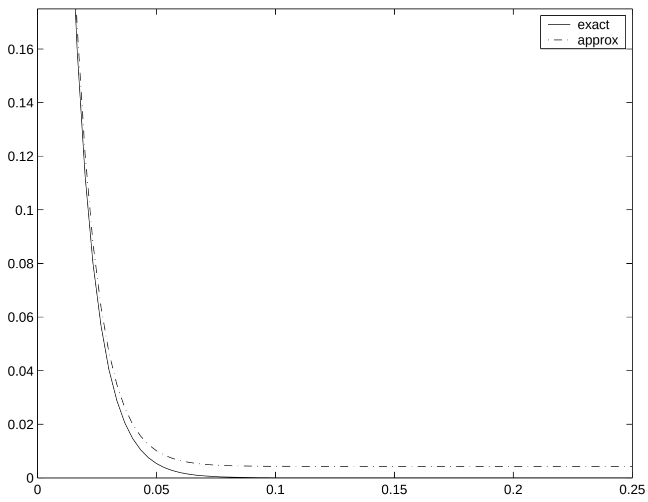

Fig. 1. Analytical and approximated solution with sequential operator-splitting method.

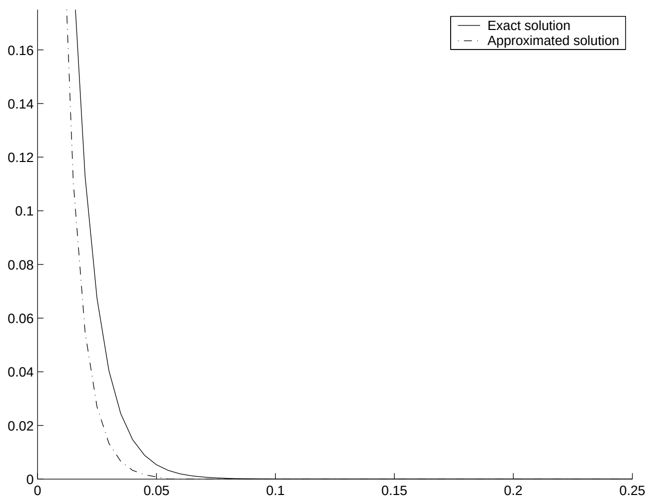

Fig. 2. Analytical and approximated solution with sequential operator-splitting method with embedded Newton's method.

The experiments result in showing 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 5 iteration steps. If we restrict the error bound to 10^{-3}, 2 iteration steps and 5 time partitions give the most effective combination.

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

time part.	number of iter.	approx. 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

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

time part.	number of iter.	approx. 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

7. Conclusions and Discussions

We present decomposition methods for differential equations based on iterative and non-iterative methods. The nonlinear equations are solved with embedded Newton meth-

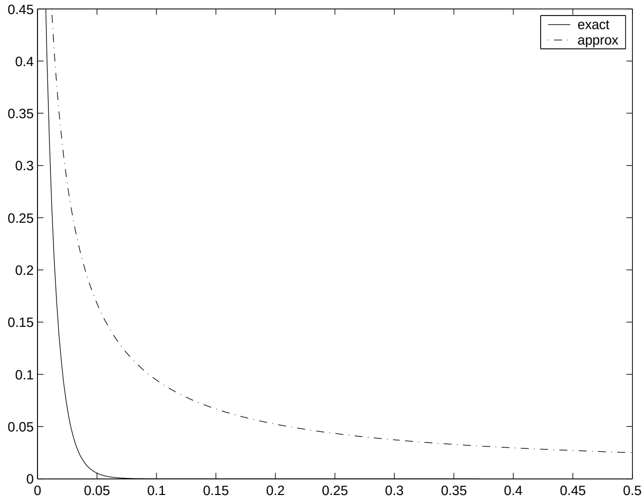

Fig. 3. Analytical and approximated solution with iterative operator-splitting method.

ods. We present new ideas of the linearization to obtain more accurate results. The superiority of the new embedded Newton methods over the traditional sequential methods are demonstrated. The results show more accurate solutions with respect to the 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 a next paper we discuss the error analysis of the nonlinear methods.

References

[1] I. Alonso-Mallo, B. Cano, and J.C. Jorge, Spectral-fractional step Runge-Kutta discretisations for initial boundary value problems with time dependent boundary conditions, Mathematics of Computation 73 (2004) 1801-1825.

[2] X.C. Cai, Additive Schwarz algorithms for parabolic convection-diffusion equations, Numer. Math. 60 (1991) 41-61.

[3] X.C. Cai, Multiplicative Schwarz methods for parabolic problems, SIAM J. Sci Comput. 15 (1994) 587-603.

[4] W. Cheney, Analysis for Applied Mathematics, Graduate Texts in Mathematics., 208, Springer, New York, Berlin, Heidelberg, 2001.

[5] C. N. Dawson, Q. Du, and D. F. Dupont, A finite Difference Domain Decomposition Algorithm for Numerical solution of the Heat Equation, Mathematics of Computation 57 (1991) 63-71.

[6] K.-J. Engel and R. Nagel, One-Parameter Semigroups for Linear Evolution Equations, Springer, New York, 2000.

[7] I. Farago and J. Geiser, Iterative Operator-Splitting methods for Linear Problems, Preprint No. 1043 of Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany, 2005.

[8] J. Geiser, Discretisation Methods with embedded analytical solutions for convection dominated transport in porous media, in: Proc. NA&A '04, Lecture Notes in Computer Science, Vol.3401, Springer, Berlin, 2005, pp. 288-295.

[9] J. Geiser, Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations, J. Comput. Appl. Math., accepted, June 2007.

[10] M.J. Gander and H. Zhao, Overlapping Schwarz waveform relaxation for parabolic problems in higher dimension, In A. Handlovičová, Magda Komorníkova, and Karol Mikula, editors, in: Proc. Algoritmy 14, Slovak Technical University, 1997, pp. 42-51.

[11] E. Giladi and H. Keller, Space time domain decomposition for parabolic problems. Technical Report 97-4, Center for research on parallel computation CRPC, Caltech, 1997.

[12] 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.

[13] W. Hundsdorfer and J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Series in Computational Mathematics Vol. 33, Springer Verlag, 2003.

[14] J. Kanney, C. Miller, and C.T. Kelley, Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems, Advances in Water Resources 26 (2003) 247-261.

[15] K.H. Karlsen and N.H. Risebro, Corrected operator splitting for nonlinear parabolic equations, SIAM J. Numer. Anal. 37 (2000) 980-1003.

[16] 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, J. Comput. Phys. 173 (2001) 636-663.

[17] G.I. Marchuk, Some applicatons of splitting-up methods to the solution of problems in mathematical physics, Aplikace Matematiky 1 (1968) 103-132.

[18] G.A. Meurant, Numerical experiments with a domain decomposition method for parabolic problems on parallel computers, in: R. Glowinski, Y.A. Kuznetsov, G.A. Meurant, J. Périaux and O. Widlund, (Ed.), Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA, 1991. SIAM.

[19] H.A. Schwarz, Über einige Abbildungsaufgaben, Journal für Reine und Angewandte Mathematik 70 (1869) 105-120.

[20] G. Strang, On the construction and comparision of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506-517.

[21] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A, Vol. 150, no. 5,6,7, 1990.

[22] J. Geiser. *Linear and Quasi-Linear Iterative Splitting Methods: Theory and Applications.* International Mathematical Forum, Hikari Ltd., Vol. 2, no. 49, 2391 - 2416, 2007.

[23] J. Geiser. *Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations.* Journal of Computational and Applied Mathematics, Elsevier, accepted, June 2007.

[24] 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.

[25] K.H. Karlsen and N. Risebro. *An Operator Splitting method for nonlinear convection-diffusion equation.* Numer. Math., 77, 3 , 365–382, 1997.

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

[27] E. Zeidler. Nonlinear Functional Analysis and its Applications. II/B Nonlinear montone operators Springer-Verlag, Berlin-Heidelberg-New York, 1990.