Geiser, J.; Hueso Pagoaga, JL.; Martínez Molada, E. (2017). New versions of iterative splitting methods for the momentum equation. Journal of Computational and Applied Mathematics. 309:359-370. doi:10.1016/j.cam.2016.06.002

New versions of iterative splitting methods for the momentum equation

Jürgen Geiser [a], Jose L. Hueso [b], Eulalia Martínez [c]

[a] *Dept. of Electrical Engineering and Information Technology, Ruhr-University of Bochum, Germany*
[b] *Instituto de Matemática Multidisciplinar. Universitat Politècnica de València, Spain*
[c] *Instituto de Matemática Pura y Aplicada. Universitat Politècnica de València, Spain*

Abstract

In this paper we propose some modifications in the schemes for the iterative splitting techniques defined in [1] for partial differential equations and introduce the parallel version of these modified algorithms. Theoretical results related to the order of the iterative splitting for these schemes are obtained. In the numerical experiments we compare the obtained results by applying iterative methods to approximate the solutions of the nonlinear systems obtained from the discretization of the splitting techniques to the mixed convection-diffusion Burgers equation and a momentum equation that models a viscous flow. The differential equations in each splitting interval are solved by the back-Euler-Newton algorithm using sparse matrices.

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

AMS subject classifications: 35K45, 35K90, 47D60, 65M06, 65M55.

1. Introduction

Nowadays, iterative splitting methods are considered as excellent decomposition methods to obtain higher-order results and to embed nonlinearities. This is due to the advantage of this technique in combining iterative and splitting behavior for decoupling physics problems. In this paper, we develop new nonlinear solvers that are modifications of the iterative splitting schemes defined in [1] and [2].

⋆ This work has been supported by Ministerio de Ciencia e Innovación de España MTM2014-52016-C2-02.

Corresponding author: jlhueso@mat.upv.es

Iterative splitting schemes are used to solve nonlinear systems obtained from ordinary differential equations or spatially discretized partial differential equations. For their original scheme, one applies a Picard-iterative technique to solve the nonlinear systems, see [2] and [3]. The drawback is that such Picard's technique is a first order scheme, see [4]. Our benefit is embedding a nonlinear scheme, Newton's method, in the splitting methods. Such novel schemes are more accurate, higher order and accelerate the solver schemes.

The novelty consists in a modification based on the idea of using the solution of first equation of the split problem for updating operators in the second equation of the same iteration. We consider all possible variations for combining the nonlinear operators, getting six different schemes. We also derive parallel versions of such schemes in order to obtain additional faster schemes and an up to date nonlinear solver for large-scale computations. Further, we derive parallel versions of such schemes to obtain additional faster schemes and an up to date nonlinear solver for large scale computations, see [5].

The outline of the paper is as follows. In section 2 we introduce our mathematical model. The splitting techniques are presented in Section 3. In section 4, we discuss the theoretical results of the methods. Section 5 explains how to apply nonlinear solvers to the equations of the splitting and Section 6 applies the new splitting methods to a scalar and a vector valued PDE and compares the results with the ones of some classical splitting methods.

2. Mathematical Model

A great variety of natural phenomena can be described by an ordinary differential equation or a partial differential equation, the solution of which not always can be obtained by analytical methods. In fact, in the majority of cases, is much more practical using numerical methods in order to approximate the solution.

In the present paper we concentrate on partial differential equations given as:

$$\frac{du(v,t)}{dt} = f(t,u(v,t)), \quad u:\Omega\times\mathbb{R}\longrightarrow\mathbb{R}, \quad \Omega\subseteq\mathbb{R}^n$$
$$\text{Boundary condition: } u(v,t) = w(v,t), \quad (v,t)\in\partial\Omega\times[0,T]$$
$$\text{Initial condition: } u(v,0) = u_0(v), \quad v\in\Omega.$$

In case $n = 0$, we have an ordinary differential equation:

$$\frac{du(t)}{dt} = f(t,u(t)), \quad t\in[0,T], \quad u:[0,T]\longrightarrow\mathbb{R},$$

and in case $n > 0$, we will obtain a system of ordinary differential equations.

This is the case for convection-diffusion-reaction-equations, see [6–9], and for a computational simulation of heat-transfer [10], which, with the above notation, is a particular case with $n = 2$. Function $f(t,u(x,y,t))$ can contain partial derivatives up to second order u_x, u_y, u_{xx} and u_{yy}.

Specifically, we deal with a particular form for function $f(t,u(v,t))$, when it can be expressed in the following form:

$$\frac{du(v,t)}{dt} = A(u(v,t))u(v,t) + B(u(v,t))u(v,t) + g(v,t), \tag{1}$$

with $t \in [0,T]$, the initial condition is known $u(v,0) = u_0(v)$, and $A(u), B(u)$ are operators in a Banach space X involving only spatial derivatives of u, whereas $g(v,t)$ is an exterior perturbation.

3. Splitting techniques

Splitting techniques can be used with the aim of decomposing the original problem into a sequence of simpler problems when the size of the problem is big or maybe if we need to solve the problem taking into account physical properties of some parts of the equation. Sometimes, we want to separate the nonlinear part of the equation from the linear part or just our aim is using different numerical methods in each part of the equation, always with the final objective of building efficient methods with the usual properties of accuracy and stability.

The traditional method is the sequential operator splitting, but nowadays iterative splitting is being the objective of different studies, [11,12]. In all cases, we discretize the time interval $[0,T]$ in N subintervals by means of the partition $0 < t^1 < t^2 < \cdots < t^n < t^{n+1} < \cdots < T$ and solve a different problem consecutively in each of these subintervals.

3.1. *Classical operator splitting techniques*

Between the different splitting algorithms traditionally used, let us mention the following ones

3.1.1. *Sequential Operator Splitting*

In this scheme we solve two sub-problems in $[t^n, t^{n+1}]$ sequentially connected via the initial conditions. First we solve the problem considering only the first operator with the given initial condition and after that, the problem is solved considering the second operator with initial condition the solution obtained in the first problem, that is:

$$\begin{aligned} \frac{d\tilde{u}(t)}{dt} &= A(\tilde{u}(t))\tilde{u}(t), \quad \tilde{u}(t^n) = u(t^n) \\ \frac{du(t)}{dt} &= B(u(t))u(t), \quad u(t^n) = \tilde{u}(t^{n+1}), \end{aligned} \tag{2}$$

where $u(t^0) = u(0) = u_0$. The value of the split solution at the end of the subinterval, $\tilde{u}(t^{n+1})$ is the initial value for the next subinterval.

Operator splitting methods can be viewed as time-discretization methods, which define exponential approximations to the exact solution at the mesh-points, $t_i,\ i = 1, \ldots, N$. Hence, as for any arbitrary time-discretization method, we can introduce the notion of the discretization error, called local splitting error, as

$$e_n(t) = u_n(t) - U_n(t),$$

where $u_n(t)$ is the splitting solution and $U_n(t)$ is the analytical solution.

The splitting is said to be of local order p in $[t^n, t^{n+1}]$ if:

$$e_n(t) = O(\tau_n^{p+1}), \text{ with } \tau_n = t^{n+1} - t^n.$$

It is well known, by using Baker-Cambell-Hausdorff formula, that the Sequential Operator-Splitting error equation has the following form, [17]:

$$\begin{aligned} e_n &= \frac{1}{2}\tau_n^2[AB - BA]u(t^n) + O(\tau_n^3) \\ &= \begin{cases} 0, & \text{for } [A,B] = 0, \\ O(\tau_n), & \text{for } [A,B] \neq 0 \end{cases} \end{aligned}$$

where $[A, B] = AB - BA$ is the commutator operator. So, this method is in general of order 1 and it is exact if operators A and B commute.

In order to improve the local order of the splitting techniques, different alternatives have been proposed, see the study presented in [13]. We will consider the additive splitting and the Strang-Marchuk splitting for purposes of comparison with the algorithms introduced in this paper. The first one is of first order and the second one has order two.

3.1.2. *Additive Splitting*

In this method, one independently solves for each operator and then, combines the two solutions before passing to the next interval, according to:

$$\begin{aligned} \frac{d\tilde{u}(t)}{dt} &= A(\tilde{u}(t))\tilde{u}(t), \text{ with } \tilde{u}(t^n) = u(t^n), \\ \frac{d\tilde{\tilde{u}}(t)}{dt} &= B(\tilde{\tilde{u}}(t))\tilde{\tilde{u}}(t), \text{ with } \tilde{\tilde{u}}(t^n) = u(t^n), \end{aligned} \tag{3}$$

and we take the approximated solution $u(t^{n+1}) = \tilde{u}(t^{n+1}) + \tilde{\tilde{u}}(t^{n+1}) - u(t^n)$.

3.1.3. *Strang-Marchuk Splitting*

In this method, first we solve for operator A in the left half of interval t^n, t^{n+1}, then we solve for B in the whole interval, and again for A in the second half of the interval. The three subproblems are connected by the initial conditions, according to

$$\begin{aligned} \frac{d\tilde{u}(t)}{dt} &= A(\tilde{u}(t))\tilde{u}(t), \text{ with } \tilde{u}(t^n) = u(t^n) \text{and } t \in [t^n, t^n + \tau/2], \\ \frac{d\tilde{\tilde{u}}(t)}{dt} &= B(\tilde{\tilde{u}}(t))\tilde{\tilde{u}}(t), \text{ with } \tilde{\tilde{u}}(t^n) = \tilde{u}(t^n + \tau/2), \\ \frac{du(t)}{dt} &= A(u(t))u(t), \text{ with } u(t^n + \tau/2) = \tilde{\tilde{u}}(t^n) \text{and } t \in [t^n + \tau/2, t^n]. \end{aligned} \tag{4}$$

3.2. *Iterative Splitting methods*

Operator splitting methods have been overcome by the iterative splitting method defined in [11]. Our aim now is to introduce some modifications in the recent iterative splitting schemes appearing in [1] and develop parallel versions of these algorithms. In the iterative splitting methods, the splitting is iteratively applied in each subinterval before passing to the next subinterval. There are several options for the iterative splitting method:

– Solve each equation sequentially or independently.
– Linearize or not the differential equation.
– Exchange the splitting parts in each iteration.

Let us now examine different versions of the method that derive from these options.

Let us consider algorithm (5.13) of [1], that can be written as follows:

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),\\ \frac{du_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_i(t), \text{ with } u_i(t^n) = u(t^n).\end{aligned}$$

We propose a modification based on the idea of using the solution of first equation $\tilde{u}_i(t)$ for updating operators $A(u)$ and $B(u)$ in the second equation of the same iteration, that is, substituting $A(u_{i-1})$ and $B(u_{i-1})$ by $A(\tilde{u}_i)$ and $B(\tilde{u}_i)$. Moreover we consider all possible variations for combining the nonlinear operators, getting six different schemes.

3.2.1. *Linear Serial Iterative Splitting (LS)*

The linear serial iterative splitting solves the first equation for the linear part of operator A, and then, solves the second equation for the linear part of operator B, using the solution of part A. The process is iterated m times before passing to the next interval.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),\\ \frac{du_i(t)}{dt} &= A(\tilde{u}_i(t))\tilde{u}_i(t) + B(\tilde{u}_i(t))u_i(t), \text{ with } u_i(t^n) = u(t^n),\end{aligned} \tag{5}$$

for $i = 1, 2, \ldots, m$. One starts with a fixed function $u_0(t)$ that verifies the initial condition $u_0(0) = u_0$. Once m iterations have been performed, we take the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$.

3.2.2. *Linear Parallel Iterative Splitting (LP)*

The linear parallel iterative splitting solves the first equation for the linear part of operator A and the second equation for the linear part of operator B without using the solution of part A. The next iterate is the average of these solutions.

$$\begin{aligned}\frac{d\tilde{u}_i(t)}{dt} &= A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),\\ \frac{d\tilde{\tilde{u}}_i(t)}{dt} &= A(u_{i-1}(t))u_{i-1}(t) + B(u_{i-1}(t))\tilde{\tilde{u}}_i(t), \text{ with } \tilde{\tilde{u}}_i(t^n) = u(t^n),\end{aligned} \tag{6}$$

and the next iteration is

$$u_i(t) = \frac{\tilde{u}_i(t) + \tilde{\tilde{u}}_i(t)}{2} \tag{7}$$

for $i = 1, 2, \ldots, m$.Once m iterations have been performed, we take the approximated solution $u(t^{n+1}) = u_m(t^{n+1})$.

3.2.3. *Quasilinear Serial Iterative Splitting (QS)*

The quasilinear versions solves each equation for a whole part of the operators, A or B, not necessarily linear.

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

with $i = 1, 2, \ldots, m$.

3.2.4. *Quasilinear Parallel Iterative Splitting (QP)*

On the other hand, we can choose a parallel version, which is given as:

$$\frac{d\tilde{u}_i(t)}{dt} = A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),$$
$$\frac{d\tilde{\tilde{u}}_i(t)}{dt} = A(u_{i-1}(t))u_{i-1}(t) + B(\tilde{\tilde{u}}_i(t))\tilde{\tilde{u}}_i(t), \text{ with } \tilde{\tilde{u}}_i(t^n) = u(t^n),$$

and the next iteration is

$$u_i(t) = \frac{\tilde{u}_i(t) + \tilde{\tilde{u}}_i(t)}{2}$$

for $i = 1, 2, \ldots, m$.

3.2.5. *Modified Linear Parallel Iterative Splitting algorithm (MLP)*

In order accelerate the convergence of the parallel algorithms, it is sometimes useful to perform the splitting twice in each iteration. In the modified parallel versions, there are two processors. Each processor applies the splitting in different order at each iterative step, and then both share their results before the next iteration.

Processor 1:

$$\frac{d\tilde{u}_i(t)}{dt} = A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),$$
$$\frac{d\tilde{\tilde{u}}_i(t)}{dt} = A(u_{i-1}(t))\tilde{u}_i(t) + B(u_{i-1}(t))\tilde{\tilde{u}}_i(t), \text{ with } \tilde{\tilde{u}}_i(t^n) = u(t^n),$$

Processor 2:

$$\frac{d\tilde{v}_i(t)}{dt} = A(v_{i-1}(t))v_{i-1}(t) + B(v_{i-1}(t))\tilde{v}_i(t), \text{ with } \tilde{v}_i(t^n) = v(t^n),$$
$$\frac{d\tilde{\tilde{v}}_i(t)}{dt} = A(v_{i-1}(t))\tilde{\tilde{v}}_i(t) + B(v_{i-1}(t))\tilde{v}_i(t), \text{ with } \tilde{\tilde{v}}_i(t^n) = v(t^n),$$

and the next iteration is

$$u_i(t) = v_i(t)) = \frac{\tilde{\tilde{u}}_i(t) + \tilde{\tilde{v}}_i(t)}{2},$$

for $i = 1, 2, \ldots$. The starting solution is the same for both processors $u_0(t) = v_0(t)$. Once m iterations have been performed, we take the approximated solution $u(t^{n+1}) = v(t^{n+1}) = u_m(t^{n+1})$.

3.2.6. *Modified Quasilinear Parallel Iterative Splitting algorithm (MQP)*
Processor 1:

$$\frac{d\tilde{u}_i(t)}{dt} = A(\tilde{u}_i(t))\tilde{u}_i(t) + B(u_{i-1}(t))u_{i-1}(t), \text{ with } \tilde{u}_i(t^n) = u(t^n),$$
$$\frac{d\tilde{\tilde{u}}_i(t)}{dt} = A(\tilde{u}_i(t))\tilde{u}_i(t) + B(\tilde{\tilde{u}}_i(t))\tilde{\tilde{u}}_i(t), \text{ with } \tilde{\tilde{u}}_i(t^n) = u(t^n),$$

Processor 2:

$$\frac{d\tilde{v}_i(t)}{dt} = A(v_{i-1}(t))v_{i-1}(t) + B(\tilde{v}_i(t))\tilde{v}_i(t), \text{ with } \tilde{v}_i(t^n) = v(t^n),$$
$$\frac{d\tilde{\tilde{v}}_i(t)}{dt} = A(\tilde{\tilde{u}}_i(t))\tilde{\tilde{u}}_i(t) + B(\tilde{v}_i(t))\tilde{v}_i(t), \text{ with } \tilde{\tilde{v}}_i(t^n) = v(t^n),$$

and the next iteration is

$$u_i(t) = v_i(t) = \frac{\tilde{\tilde{u}}_i(t) + \tilde{\tilde{v}}_i(t)}{2}.$$

4. Theoretical results

In this section we are going to obtain the convergence order for the different algorithms, showing the expressions for the local error.

Theorem 1 *We assume to have bounded nonlinear operators A and B, while given as $||A(u)|| \leq A_{max}$ and $||B(u)|| \leq B_{max}$ for all $u \in \mathbf{X}$.*

Then the numerical errors are given for the different schemes as:

(i) Serial version, the numerical error is given as:

$$||u - u_m|| \leq ||A_{max}||^m ||B_{max}||^m \mathcal{O}(\tau^{2m}),$$

where u is the exact solution and u_m the iterative solution of (5).

(ii) Parallel version, the numerical error is given as:

$$||u - u_m|| \leq \left(\frac{||A_{max}||}{2} + \frac{||B_{max}||}{2}\right)^m \mathcal{O}(\tau^m),$$

where u is the exact solution and u_m the iterative solution of (6).

Proof:

(i) For the serial version, we have the following proof:

Let us consider the iteration (5) on the subinterval $[t^n, t^{n+1}]$. For the local error function $\tilde{e}_i(t) = u(t) - \tilde{u}_i(t)$ and $e_i(t) = u(t) - u_i(t)$ and u is the analytical solution. We have the following linearized relations based on the assumption of maximal operators $A(u)$ and $B(u)$:

$$\begin{aligned} &\partial_t \tilde{e}_i(t) = A_{max}\tilde{e}_i(t) + B_{max}e_{i-1}(t), \quad t \in (t^n, t^{n+1}], \\ &\tilde{e}_i(t^n) = 0, \end{aligned} \tag{8}$$

and

$$\partial_t e_i(t) = A_{max}\tilde{e}_i(t) + B_{max}e_i(t), \quad t \in (t^n, t^{n+1}], \tag{9}$$
$$e_i(t^n) = 0.$$

For the iteration (8) and (9) we have:

$$\tilde{e}_i(t) = \int_{t^n}^{t} \exp(A_{max}(t-s))B_{max}e_{i-1}(s)ds, \quad t \in [t^n, t^{n+1}],$$

and

$$e_i(t) = \int_{t^n}^{t} \exp(B_{max}(t-s))A_{max}\tilde{e}_i(s)ds, \quad t \in [t^n, t^{n+1}].$$

We estimate:

$$\begin{aligned}\|\tilde{e}_i\| &\leq \|B_{max}\|||e_{i-1}|| \int_{t^n}^{t} \|\exp(A_{max}(t-s))\|ds \\ &\leq \tilde{K}\|B_{max}\|\|e_{i-1}\|\,\tau_n + \mathcal{O}(\tau_n^2)),\end{aligned}$$

and

$$\begin{aligned}\|e_i\| &\leq \|A_{max}\|||\tilde{e}_i|| \int_{t^n}^{t} \|\exp(B_{max}(t-s))\|ds \\ &\leq K\|A_{max}\|\|\tilde{e}_i\|\,\tau_n + \mathcal{O}(\tau_n^2)).\end{aligned}$$

From these inequalities we obtain:

$$\begin{aligned}\|e_i\| &\leq K\|A_{max}\|\,(\tilde{K}\|B_{max}\|e_{i-1}\|\|\tau_n + \mathcal{O}(\tau_n^2))\,\tau_n \\ &\leq K_2\|A_{max}\|\|B_{max}\|\|e_{i-1}\|\tau_n^2 + \mathcal{O}(\tau_n^3).\end{aligned}$$

The recursion of our errors is applied and we obtain:

$$\|e_m\| \leq \|A_{max}\|^m\|B_{max}\|^m\mathcal{O}(\tau_n^{2m}).$$

Thus, for $2m$ serial iterations, m over operator A and m over operator B, we have $\mathcal{O}(\tau_n^{2m})$ of convergence order.

(ii) For the parallel version, we have the following proof:

Let us consider the iteration (6) on the subinterval $[t^n, t^{n+1}]$. For the local error function $\tilde{e}_i(t) = u(t) - \tilde{u}_i(t)$ and $\tilde{\tilde{e}}_i(t) = u(t) - \tilde{\tilde{u}}_i(t)$ and u is the analytical solution. We have the following linearized relations based on the assumption of maximal operators $A(u)$ and $B(u)$:

$$\partial_t\tilde{e}_i(t) = A_{max}\tilde{e}_i(t) + B_{max}e_{i-1}(t), \quad t \in (t^n, t^{n+1}], \tag{10}$$
$$\tilde{e}_i(t^n) = 0,$$

and

$$\partial_t\tilde{\tilde{e}}_i(t) = A_{max}e_{i-1}(t) + B_{max}\tilde{\tilde{e}}_i(t), \quad t \in (t^n, t^{n+1}], \tag{11}$$
$$\tilde{\tilde{e}}_i(t^n) = 0.$$

For the iteration (10) and (11) we have:

$$\tilde{e}_i(t) = \int_{t^n}^{t} \exp(A_{max}(t-s))B_{max}e_{i-1}(s)ds, \quad t \in [t^n, t^{n+1}],$$

and

$$\tilde{\tilde{e}}_i(t) = \int_{t^n}^{t} \exp(B_{max}(t-s))A_{max}e_{i-1}(s)ds, \quad t \in [t^n, t^{n+1}].$$

We estimate to:

$$\begin{aligned}\|\tilde{e}_i\| &\leq \|B_{max}\|||e_{i-1}|| \int_{t^n}^{t} \|\exp(A_{max}(t-s))\|ds \\ &\leq \tilde{K}\|B_{max}\|\|e_{i-1}\|\,\tau_n + \mathcal{O}(\tau_n^2)),\end{aligned}$$

and

$$\begin{aligned}\|\tilde{\tilde{e}}_i\| &\leq \|A_{max}\|||e_{i-1}|| \int_{t^n}^{t} \|\exp(B_{max}(t-s))\|ds \\ &\leq K\|A_{max}\|\|e_{i-1}\|\,\tau_n + \mathcal{O}(\tau_n^2)).\end{aligned}$$

We insert both into (7) and obtain:

$$\|e_i\| \leq \frac{(\|A_{max}\| + \|B_{max}\|)}{2}\,\|e_{i-1}\|\,\tau_n + \mathcal{O}(\tau_n^2).$$

The recursion of our errors is applied and we obtain:

$$\|e_m\| \leq \left(\frac{(\|A_{max}\| + \|B_{max}\|)}{2}\right)^m \mathcal{O}(\tau_n^m).$$

Thus, for $2m$ serial iterations, m over operator A and m over operator B, we have $\mathcal{O}(\tau_n^m)$ of convergence order.

□

According to [15,16], the results about the order of splitting methods on bounded operators, still remain valid in a setting of unbounded operators, without requiring any additional order condition. This is achieved by basing the analysis on the abstract framework of (semi)groups. The convergence analysis also includes generalizations to splittings consisting of more than two operators, and to variable time steps.

5. Solving the problem

By applying one of these splitting techniques to the main problem (1), written as (12), we have divided it into different simpler problems, but in any case we have to solve the resulting systems of ordinary differential equations by using numerical methods. We will construct a grid $(x_i, y_j) \in \Omega,\ i = 1, 2, \ldots, n_x,\ j = 1, 2, \ldots, n_y$ and use divided differences for approximating the spatial derivatives. By using the notation, $u_{ij}(t) \approx u(x_i, y_j, t)$ the problem becomes a system of ordinary differential equations:

$$\frac{du_{ij}(t)}{dt} = f_{ij}(t, u(t)), \quad t \in [0, T], \quad i = 1, 2, \ldots, n_x, \; j = 1, 2, \ldots, n_y. \tag{12}$$

If we express this system of ODE's in its integral form, we have:

$$u_{ij}(t^{n+1}) = u_{ij}(t^n) + \int_{t^n}^{t^{n+1}} f_{ij}(s, u(s))ds. \tag{13}$$

By applying the left-hand rectangle method to approximate the integral, we obtain Euler's method, defined as follows:

$$u_{ij}(t^{n+1}) = u_{ij}(t^n) + \tau_n f_{ij}(t^n, u(t^n)),$$

where $\tau_n = t^{n+1} - t^n$.

This method can be improved by applying instead the trapezoidal rule or Simpson's formula obtaining Heun's method or Runge-Kutta's method, in their explicit forms. In order to avoid instability problems we consider implicit methods, such as backward Euler's method, obtained by applying the right-hand rectangle method to approximate the integral (13), that is:

$$u_{ij}(t^{n+1}) = u_{ij}(t^n) + \tau_n f_{ij}(t^{n+1}, u(t^{n+1})).$$

In this way, the system of ODE's has been approximated by a nonlinear system, which can be expressed as a fixed point problem with function iteration:

$$G(u_{ij}(t^{n+1})) = u_{ij}(t^n) + \tau_n f_{ij}(t^{n+1}, u(t^{n+1})),$$

where we choose a starting guess and iterate for $k = 1, 2, \ldots$

$$u_{ij}^{(0)}(t^{n+1}) = u_{ij}(t^n),$$
$$u_{ij}^{(k+1)}(t^{n+1}) = G(u_{ij}^{(k)}(t^{n+1})),$$

until a tolerance is reached:

$$\left\| u_{ij}^{(k+1)}(t^{n+1}) - u_{ij}^{(k)}(t^{n+1}) \right\| \leq tol.$$

Otherwise, we can apply Newton's method, which converges quadratically. At each step, we have to solve a linear system with $n_x \times n_y = S$ unknowns. In order to use matrix operations, we order the unknowns as a vector $u_{ij}(t^{n+1}) = u_s(t^{n+1}), \quad s = 1, \ldots, n_x \times n_y = S$, so that, the nonlinear system is expressed as:

$$F(u_s(t^{n+1})) = u_s(t^{n+1}) - u_s(t^n) - \tau_n f_s(t^{n+1}, u(t^{n+1})).$$

The Jacobian matrix needed for Newton's method is:

$$F'(u_s(t^{n+1})) = I_{S \times S} - \tau_n f_s'(t^{n+1}, u(t^{n+1})),$$

where I is the identity matrix and $f_s'(t^{n+1}, u(t^{n+1}))$ represents $\dfrac{\partial f_s(t^{n+1}, u(t^{n+1}))}{du(t^{n+1})}$. Then, we choose a starting guess and iterate for $k = 1, 2, \ldots$

$$u_s^{(0)}(t^{n+1}) = u_s(t^n)$$
$$u_s^{(k+1)}(t^{n+1}) = u_s^{(k)}(t^{n+1}) - (F'(u_s^{(k)}(t^{n+1})))^{-1} F(u_s^{(k)}(t^{n+1}))$$

to achieve the desired tolerance if the method converges.

We illustrate the above procedure for the Linear Serial Iterative Splitting (5). The two differential equations occurring in each step of the splitting method are solved by applying Newton's method to the nonlinear functions

$$\tilde{F}(\tilde{u}_i(t^{n+1})) = \tilde{u}_i(t^{n+1}) - \tilde{u}_i(t^n) - \tau_n A(u_{i-1}(t^{n+1}))\tilde{u}_i(t^{n+1}) + B(u_{i-1}(t^{n+1}))u_{i-1}(t^{n+1}),$$
$$F(u_i(t^{n+1}) = u_i(t^{n+1}) - u_i(t^n) - \tau_n A(\tilde{u}_i(t^{n+1}))\tilde{u}_i(t^{n+1}) + B(\tilde{u}_i(t^{n+1}))u_i(t^{n+1}).$$

Observe that the order of the numerical integrator will affect the outcome of the splitting method, limiting its accuracy, see [13], as well as the convergence order of the nonlinear solver. One has also to consider the problem of order reduction, see [18].

6. Numerical Results

In order to assess the performance of the splitting methods methods, we compare the values $u_\tau(x_i, y_j, T)$ at the final time T of the solution obtained using a constant temporal step $\tau = t^{n+1} - t^n$ with the corresponding ones of the analytical solution $U(x_i, y_j, T)$ or a fine reference. If the spatial discretization is fixed, the numerical solution converges to a limit different from the analytical solution when the temporal step tends to 0. In this case, it is better to refer the error to this limit instead of U.

Then we consider the following estimations for the error of a discrete solution. The *numerical error*

$$e_\tau = \max_{i,j} \|u_\tau(x_i, y_j, T) - U(x_i, y_j, T)\|,$$

and the *estimated numerical error*

$$\tilde{e}_\tau = \max_{i,j} \|u_\tau(x_i, y_j, T) - u_{\tau/2}(x_i, y_j, T)\|,$$

where $u_{\tau/2}(x_i, y_j, T)$ is the result obtained taking twice the number of temporal steps.

The *convergence rate* is computed from the estimated numerical errors corresponding to different time steps such as

$$\rho_\tau = \frac{\log(\tilde{e}_\tau) - \log(\tilde{e}_{\tau/2)}}{\log(2)}.$$

We also measure the convergence rate of the method with respect to the number of iterations of the splitting algorithm, by comparing the errors obtained with different number of iterations according to

$$\rho_{m_1,m_2} = \frac{\log(\tilde{e}_{m_1}) - \log(\tilde{e}_{m_2})}{\log(m_2) - \log(m_1)},$$

where $\tilde{e}_{m_k}$ denotes the estimated numerical error of the solution obtained performing m_k iterations per step of the splitting method.

We compare different splitting schemes applied to two examples, the mixed convection-diffusion and Burgers' equation and a equation that models a viscous flow. The differential equations in each splitting interval are solved by using the back-Euler-Newton algorithm.

It is worth mentioning that one step of Newton's method suffices to reach an approximate solution of the implicit Euler method in each splitting interval. Additional steps do not produce a significant error reduction. Thus, in the numerical examples, only one Newton's step is performed.

6.1. *Mixed convection-diffusion and Burgers equation*

Our first numerical example is a partial differential equation which is a a mix of the convection-diffusion equation and the Burgers' equation in 2D:

$$\begin{aligned}\partial_t u = &-\frac{1}{2}u(\partial_x u + \partial_y u) - \frac{1}{2}(\partial_x u + \partial_y u)\\ &+ \mu(\partial_{xx}u + \partial_{yy}u) + f(x,y,t),\ (x,y,t) \in \Omega \times [0,T],\\ &u(x,y,0) = u_{\text{ana}}(x,y,0),\ (x,y) \in \Omega,\\ &u(x,y,t) = u_{\text{ana}}(x,y,t), (x,y,t) \in \partial\Omega \times [0,T],\end{aligned} \tag{14}$$

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

The analytical solution is

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

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

By considering the following operators

$$\begin{aligned}A(u)v &= -\frac{1}{2}u(\partial_x v + \partial_y v) + \frac{1}{2}\mu(\partial_{xx}v + \partial_{yy}v),\\ Bv &= -\frac{1}{2}(\partial_x v + \partial_y v) + \frac{1}{2}\mu(\partial_{xx}v + \partial_{yy}v) + f(x,y,t),\end{aligned}$$

we split (14) in the the Burgers' term, A and the convection-diffusion term, B

$$\partial_t u = A(u)u + Bu.$$

The spatial domain is discretized taking a rectangular mesh with $n_x = n_y = 10$ intervals. Tables 1 and 2 compare the errors and the convergence rate of methods *LS*, *LP*, *QS* and *QP* with 8 iterations of the splitting method for the mixed convection-diffusion and Burgers' equation with different viscosities.

For $\mu = 0.5$, the behaviors of the methods are very similar, but for higher viscosities, $\mu = 5$, the serial methods overcome the parallel ones. The numerical error with respect to the analytical solution decreases with the temporal step reaching a minimum at a certain value of τ but then starts increasing for smaller values of the time step. This is because the analytical solution is not the 'exact' of the spatially discretized problem. The common final value of the numerical errors is related to this difference. However, the estimated numerical errors decrease steadily and the convergence rate is very stable for the considered methods. The serial schemes are better than the parallel ones for small N, but for big N the differences fade away.

The methods behave better for $\mu = 5$. The increase of the numerical error appears at a higher number of time intervals, the estimated errors are smaller than for $\mu = 0.5$ and the convergence rate clearly increases with the number of time intervals.

The accuracy of the solutions depends on the number of iterations of the splitting algorithm. The more iterations are performed, the more accurate the solution is. As before, the error measured with respect to the analytical solution does not decrease after a number of iterations. For the parallel schemes, it decreases mainly when the number of iterations is odd, oscillating in some cases. However, the estimated numerical error decreases more regularly.

Method	Linear Serial			Linear Parallel			Quasilinear Serial			Quasilinear Parallel		
N	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ
25	3.09e–3	2.60e–3	–	1.23e–2	1.06e–2	–	3.40e–3	2.84e–3	–	1.05e–2	9.07e–3	–
50	1.05e–3	7.68e–4	1.76	1.61e–3	1.43e–3	2.90	1.04e–3	7.58e–4	1.90	1.45e–3	1.28e–3	2.83
100	2.85e–4	3.76e–4	1.03	2.87e–4	3.84e–4	1.89	2.84e–4	3.75e–4	1.01	2.92e–4	3.87e–4	1.73
200	9.64e–5	1.87e–4	1.01	9.61e–5	1.88e–4	1.03	9.65e–5	1.87e–4	1.00	9.50e–5	1.89e–4	1.04
400	2.84e–4	9.35e–5	1.00	2.84e–4	9.36e–5	1.00	2.84e–4	9.35e–5	1.00	2.84e–4	9.38e–5	1.01
800	3.77e–4	4.67e–5	1.00	3.77e–4	4.67e–5	1.00	3.77e–4	4.67e–5	1.00	3.77e–4	4.68e–5	1.00
1600	4.24e–4	2.34e–5	1.00	4.24e–4	2.34e–5	1.00	4.24e–4	2.34e–5	1.00	4.24e–4	2.34e–5	1.00
3200	4.47e–4	1.17e–5	1.00	4.47e–4	1.17e–5	1.00	4.47e–4	1.17e–5	1.00	4.47e–4	1.17e–5	1.00

Table 1
Numerical error, estimated error and convergence rate for $n_x = 10$ and $\mu = 0.5$.

Method	Linear Serial			Linear Parallel			Quasilinear Serial			Quasilinear Parallel		
N	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ
25	5.02e–3	4.04e–3	–	1.15e–2	8.93e–3	–	5.01e–3	4.03e–3	–	1.15e–2	8.91e–3	–
50	9.81e–4	8.23e–4	2.30	2.57e–3	2.04e–3	2.13	9.79e–4	8.21e–4	2.30	2.56e–3	2.04e–3	2.13
100	1.58e–4	1.40e–4	2.55	5.24e–4	4.33e–4	2.24	1.58e–4	1.40e–4	2.55	5.23e–4	4.32e–4	2.24
200	1.79e–5	1.68e–5	3.06	9.12e–5	7.92e–5	2.45	1.79e–5	1.68e–5	3.06	9.11e–5	7.91e–5	2.45
400	1.12e–6	1.09e–6	3.95	1.21e–5	1.10e–5	2.84	1.12e–6	1.09e–6	3.95	1.20e–5	1.10e–5	2.84
800	5.70e–8	5.61e–8	4.28	1.03e–6	9.85e–7	3.48	5.70e–8	5.61e–8	4.28	1.03e–6	9.85e–7	3.48
1600	9.41e–10	2.77e–8	1.02	4.76e–8	5.17e–8	4.25	9.41e–10	2.77e–8	1.02	4.75e–8	5.17e–8	4.25
3200	2.68e–8	1.39e–8	1.00	2.68e–8	1.39e–8	1.90	2.68e–8	1.39e–8	1.00	2.68e–8	1.39e–8	1.90

Table 2
Numerical error, estimated error and convergence rate for $n_x = 10$ and $\mu = 5$.

The convergence rates of each iteration with respect to the first one increase with the number of iterations, being higher for the serial schemes and for small viscosities. Table 3 analyses the behaviour of the splitting methods (4), (2) and (3) in the conditions of table (2) for comparison with the methods introduced in this paper. The classical methods present convergence order about 1 and higher errors, specially for big N.

Table 4 shows the evolution of the numerical error, the estimated error and the numerical convergence rate with the number of iterations of the splitting for the linear serial and parallel schemes, LS and LP, with 100 time steps and the same values of the viscosities. The quasilinear schemes give similar results.

If we are interested in approximating the analytical solution, we can look for a time step that minimizes the numerical error for a given size of the spatial discretization. Table 5 shows that the number of temporal steps needed for the method LS to attain the least numerical error is approximately proportional to the number of spatial discretization points, $n_x \times n_y$. The minimum error is inversely proportional to $n_x \times n_y$. For $\mu = 5$ the obtained accuracy is much better and the optimal number of time intervals is higher. The other methods have a similar behavior.

Method	Operator splitting			Additive splitting			Strang-Marchuk		
n_t	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ
25	2.04e–004	6.97e–005	NaN	2.37e–003	1.36e–003	NaN	5.78e–005	1.20e–004	NaN
50	1.36e–004	5.66e–005	0.30	1.00e–003	5.00e–004	1.45	1.71e–004	7.79e–005	0.62
100	8.05e–005	3.70e–005	0.61	5.05e–004	2.52e–004	0.99	2.49e–004	4.16e–005	0.91
200	4.36e–005	2.12e–005	0.80	2.52e–004	1.26e–004	1.00	2.91e–004	2.11e–005	0.98
400	2.31e–005	1.13e–005	0.90	1.26e–004	6.31e–005	1.00	3.11e–004	1.08e–005	0.97
800	1.19e–005	5.86e–006	0.95	6.31e–005	3.16e–005	1.00	3.21e–004	5.44e–006	0.98
1600	6.15e–006	3.03e–006	0.95	3.16e–005	1.58e–005	1.00	3.27e–004	2.75e–006	0.99
3200	3.12e–006	1.54e–006	0.98	1.58e–005	7.89e–006	1.00	3.29e–004	1.38e–006	1.00

Table 3
Results of some classical methods for Burger equation with $n_x = 10$ and $\mu = 5$.

Method	Linear Serial $\mu = 0.5$			Linear Parallel $\mu = 0.5$			Linear Serial $\mu = 5$			Linear Parallel $\mu = 5$		
Iter	e	$\tilde{e}$	$\rho_{1,iter}$	e	$\tilde{e}$	$\rho_{1,iter}$	e	$\tilde{e}$	$\rho_{1,iter}$	e	$\tilde{e}$	$\rho_{1,iter}$
1	6.68e–3	5.41e–3	–	9.84e–3	1.62e–2	–	3.19e–3	1.82e–3	–	5.04e–4	3.69e–3	–
2	1.27e–3	8.71e–4	2.64	6.36e–3	8.13e–3	0.99	1.38e–3	5.77e–4	1.65	3.18e–3	3.55e–3	0.06
3	3.96e–4	2.09e–4	2.96	1.77e–3	2.97e–3	1.54	8.00e–4	2.75e–4	1.72	3.65e–4	1.74e–3	0.68
4	2.88e–4	5.94e–5	3.25	1.20e–3	1.59e–3	1.68	5.26e–4	1.56e–4	1.77	1.37e–3	1.66e–3	0.58
5	2.84e–4	1.86e–5	3.52	3.86e–4	7.64e–4	1.90	3.69e–4	9.84e–5	1.81	2.86e–4	1.08e–3	0.76
6	2.84e–4	6.28e–6	3.77	3.78e–4	4.21e–4	2.04	2.71e–4	6.61e–5	1.85	7.98e–4	1.03e–3	0.71
7	2.85e–4	2.22e–6	4.01	2.84e–4	2.25e–4	2.20	2.05e–4	4.65e–5	1.88	2.31e–4	7.55e–4	0.82
8	2.85e–4	8.22e–7	4.23	2.87e–4	1.28e–4	2.33	1.58e–4	3.38e–5	1.92	5.24e–4	7.12e–4	0.79

Table 4
Numerical error, estimated error and convergence rate of the linear serial and parallel iterative splitting methods for $n_x = 10$ and $n_t = 100$.

6.2. *Momentum equation*

We consider an example of momentum equation used to model the viscous flow of a fluid, see [1].

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

where 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.

In the 2D case, the analytical solution $\mathbf{u} = (u_1, u_2)^t$ is given by

N	$n_x = n_y = 5$	$n_x = n_y = 10$	$n_x = n_y = 20$	$n_x = n_y = 40$
25	0.001308	0.013238	0.118062	1.188981
50	**0.000340**	0.001696	0.018422	0.126009
100	0.001083	0.000288	0.002873	0.023147
200	0.001453	**0.000096**	0.000354	0.004382
400	0.001637	0.000284	0.000072	0.000705
800	0.001728	0.000377	**0.000024**	0.000084
1600	0.001774	0.000424	0.000071	0.000018
3200	0.001797	0.000447	0.000095	**0.000006**

Table 5
Numerical error e_τ of the linear serial scheme in terms of the spatial and temporal discretization points for $\mu = 0.5$ and 4 iterations per time interval.

$$u_1(x,y,t) = \left(1 + \exp\left(v_1 \frac{x+y-t}{2\mu}\right)\right)^{-1} + \exp\left(v_1 \frac{x+y-t}{2\mu}\right)$$
$$u_2(x,y,t) = \left(1 + \exp\left(v_2 \frac{x+y-t}{2\mu}\right)\right)^{-1} + \exp\left(v_2 \frac{x+y-t}{2\mu}\right).$$

In order to apply splitting methods, the equation is decomposed in

$$A(\mathbf{u})\mathbf{u} = -\mathbf{u}\nabla\mathbf{u} + 2\mu\nabla D(\mathbf{u})$$
$$B\mathbf{u} = \frac{2}{3}\Delta\mathbf{u}.$$

The parallel versions of the splitting schemes above considered, *LP* and *QP*, require a very small temporal step to converge for this example, so that they are not competitive. Thus, we use the modified versions MLP and MQP for recovering the convergence with the same time steps as the serial schemes.

Table 6 shows the numerical error, the estimated error and the convergence rate of the considered methods with 4 iterations of the splitting method for the momentum equation with $\Omega = [0,1] \times [0,1]$, $T = 1.25$, $\mu = 2$, and $v = (1,1)^t$. The values correspond to the first component of the solution, but the results are roughly the same for the second one. The methods present a behaviour similar to that of the former example. The numerical error e_τ does not decrease beyond about $n_t = 200$, but the estimated error $\tilde{e}_\tau$ always decreases, showing the convergence of the method.

Method	Linear Serial			Modified Linear Parallel			Quasilinear Serial			Modified Quasilinear Parallel		
n_t	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ
50	2.00e–3	1.83e–3	–	2.57e–3	2.31e–3	–	2.23e–3	2.07e–3	–	2.72e–3	2.52e–3	–
100	2.72e–4	2.59e–4	2.82	3.34e–4	3.22e–4	2.84	2.95e–4	2.87e–4	2.86	3.29e–4	3.20e–4	2.97
200	5.61e–5	2.09e–5	3.63	4.52e–5	3.43e–5	3.23	5.73e–5	2.26e–5	3.66	4.70e–5	3.38e–5	3.24
400	4.36e–5	2.44e–6	3.10	4.20e–5	2.54e–6	3.76	4.37e–5	2.54e–6	3.16	4.22e–5	2.81e–6	3.59
800	4.13e–5	1.07e–6	1.19	4.06e–5	1.14e–6	1.15	4.13e–5	1.08e–6	1.23	4.06e–5	1.17e–6	1.26

Table 6
Numerical error, estimated error and convergence rate for $n_x = n_y = 10$, $\mu = 2$ and $v = (1,1)^t$.

Table 7 shows the corresponding results for the classical splitting method above considered. One observes that the sequential iterative methods have better convergence properties and less error than the order one classical methods.

Method	Operator splitting			Additive splitting			Strang-Marchuk		
n_t	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ	e_τ	$\tilde{e}_\tau$	ρ_τ
50	9.46e–002	4.65e–002	–	Diverg.	Diverg.	–	1.70e–001	2.13e–002	–
100	4.83e–002	2.49e–002	0.90	Diverg.	Diverg.	–	1.69e–001	1.20e–002	0.83
200	2.35e–002	1.24e–002	1.00	6.99e–002	3.70e–002	–	1.74e–001	7.86e–003	0.61
400	1.20e–002	5.91e–003	1.07	3.29e–002	1.77e–002	1.07	1.79e–001	4.66e–003	0.75
800	6.04e–003	3.01e–003	0.97	1.61e–002	8.03e–003	1.14	1.83e–001	2.51e–003	0.89

Table 7
Numerical error, estimated error and convergence rate for $n_x = n_y = 10$, $\mu = 2$ and $v = (1, 1)^t$.

Under more demanding values of the parameters, $\mu = 50$, and $v = (100, 0.001)^t$, a very small time step is required for convergence, as shown in table 8. Even when the methods converge, the solution of the discretized problem is quite different from the analytical solution. Thus, the numerical error is not very sensitive to the number of time steps. Nevertheless, the estimated error shows the convergence of the methods. Due to the parameters asymmetry, the errors of the first and the second components are quite different, as seen in Table 8.

Method	Linear Serial						Modified Linear Parallel					
n_t	$e_{\tau,x}$	$e_{\tau,y}$	$\tilde{e}_{\tau,x}$	$\tilde{e}_{\tau,y}$	$\rho_{\tau,x}$	$\rho_{\tau,y}$	$e_{\tau,x}$	$e_{\tau,y}$	$\tilde{e}_{\tau,x}$	$\tilde{e}_{\tau,y}$	$\rho_{\tau,x}$	$\rho_{\tau,y}$
1600	4.95e–3	5.73e–2	5.75e–4	1.88e–3	–	–	4.84e–3	7.01e–2	1.68e–3	2.85e–2	–	–
3200	4.86e–3	5.58e–2	1.41e–4	5.71e–4	2.02	1.72	4.91e–3	5.54e–2	1.93e–4	2.27e–3	3.13	3.65
6400	4.85e–3	5.57e–2	2.92e–5	4.66e–5	2.27	3.62	4.86e–3	5.58e–2	4.64e–5	1.92e–4	2.05	3.57

Table 8
Numerical error, estimated error and convergence rate for each variable with $n_x = n_y = 10$, $\mu = 50$ and $v = (100, 0.001)^t$.

7. Conclusion and Discussion

We present novel nonlinear solver methods, which are embedded to iterative splitting schemes. The nonlinear methods are based on Newton's method. We could see the advantage of dealing with higher accurate methods, while the linearization methods are effective and less time consuming. We apply the new iterative splitting methods to Burgers and momentum equations. For the serial and also parallel versions, we obtain higher order results of about $2 - 4$. The benefit of reducing the computational time for the full solver process was seen for the modified linear and quasilinear parallel versions. In future, such novel nonlinear iterative splitting schemes can be applied and tested for large scale computational problems. Another issue deserving a deeper study it the problem or order reduction for PDE problems with initial Dirichlet conditions.

References

[1] J. Geiser. *Modified Jacobian Newton Iterative Method: Theory and Applications.* Mathematical problems in Enginiering, Hindawi Publishing Corporation, Volume 2009 (2009), Article ID 307298.

[2] J. Geiser. *Iterative Splitting Methods for Differential Equations.* Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series, edited by Magoules and Lai, 2011.

[3] O. Nevanlinna, Remarks on Picard-Lindelöf Iteration: Part I, BIT 29 (1989) 328-346.

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

[5] M.J. Gander , 50 Years of Time Parallel Time Integration, in: T. Carraro, M. Geiger, S. Krkel, R. Rannacher (Eds.), to appear in Multiple Shooting and Time Domain Decomposition, Springer Verlag, 2015.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] J. Geiser, Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations, J. Comput. Appl. Math., 217 (2008) 227-242.

[13] I. Farago, A modified iterated operator splitting method
Applied Mathematical Modelling 32 (2008) 1542-1551.

[14] Multiple Shooting and Time Domain Decomposition Methods. Volume 9 of the series Contributions in Mathematical and Computational Sciences, pp 69-113

[15] Eskil Hansen, Alexander Osterman. Dimension splitting for evolution equations, Numerische Mathematik 108, (2008), 557-570

[16] Eskil Hansen, Alexander Osterman. Exponential splitting for unbounded operators, Mathematics of Computation. 78, (2009), 1485-1496.

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

[18] J.M. Sanz-Serna, J.G. Verwer, and W.H. Hundsdorfer, *Convergence and Order Reduction of Runge-Kutta Schemes Applied to Evolutionary Problems in Partial Differential Equations*, Numer. Math. 50, (1986) 405-418