Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 5995–6010
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations

Volker John [a], Gunar Matthies [b,*], Joachim Rang [c]

[a] FR 6.1—Mathematik, Universität des Saarlandes, Postfach 15 11 50, 66041 Saarbrücken, Germany
[b] Fakultät für Mathematik, Ruhr-Universität Bochum, Universitätsstr. 150, 44780 Bochum, Germany
[c] Institut für Mathematik, TU Clausthal, Erzstr. 1, 38678 Clausthal-Zellerfeld, Germany

Received 9 August 2004; received in revised form 29 June 2005; accepted 12 October 2005

Abstract

This paper presents a numerical study of two ways for discretizing and linearizing the time-dependent incompressible Navier–Stokes equations. One approach consists in first applying a semi-discretization in time by a fully implicit θ-scheme. Then, in each discrete time, the equations are linearized by a fixed point iteration. The number of iterations to reach a given stopping criterion is a priori unknown in this approach. In the second approach, Rosenbrock schemes with s stages are used as temporal discretization. The non-linearity of the Navier–Stokes equations is treated internally in the Rosenbrock methods. In each discrete time, exactly s linear systems of equations have to be solved. The numerical study considers five two-dimensional problems with distinct features. Four implicit time stepping schemes and five Rosenbrock methods are involved.

Keywords: Incompressible Navier–Stokes equations; Rosenbrock methods; Implicit θ-schemes; Fixed point iteration

1. Introduction

The motion of an incompressible fluid is governed by the incompressible Navier–Stokes equations given (in dimensionless form) by

$$\begin{aligned} &\mathbf{u}_t - Re^{-1}\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} && \text{in } (0,T]\times\Omega,\\ &\nabla\cdot\mathbf{u} = 0 && \text{in } [0,T]\times\Omega,\\ &\mathbf{u} = \mathbf{g} && \text{on } [0,T]\times\partial\Omega,\\ &\mathbf{u}(0,\cdot) = \mathbf{u}_0 && \text{in } \Omega,\\ &\int_\Omega p\,\mathrm{d}\mathbf{x} = 0 && \text{in } [0,T]. \end{aligned} \tag{1}$$

Here, $\mathbf{u}$ is the velocity, $\mathbf{u}_0$, the initial velocity, p, the pressure, $\mathbf{f}$, represents body forces, $\mathbf{g}$, the given Dirichlet boundary data, $[0,T]$ is a given time interval and $\Omega\subset\mathbb{R}^d$, $d\in\{2,3\}$, a domain. The Navier–Stokes equations posses as parameter the Reynolds number Re. Depending on Re, different flow regimes are described with (1). If Re is sufficiently small and the data in (1) do not depend on the time, $(\mathbf{u},p)$ represent a stationary flow field. For larger Re, the flow is time-dependent and laminar;

* Corresponding author.
E-mail addresses: john@math.uni-sb.de (V. John), gunar.matthies@ruhr-uni-bochum.de (G. Matthies), major@math.tu-clausthal.de (J. Rang).
URLs: http://www.math.uni-sb.de/ag/john (V. John), http://www.ruhr-uni-bochum.de/jpnum (G. Matthies), http://www.math.tu-clausthal.de/Personen/rang.html (J. Rang).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2005.10.007

and for large Re, the flow becomes turbulent. The accurate and fast solution of the Navier–Stokes equations is the core of many numerical simulations, e.g., in the simulation of crystal growth [1] or of fuel cells [2].

The numerical solution of the time-dependent Navier–Stokes equations requires their discretization in time and space as well as a linearization. There are much different approaches for all of these components. A large number of them can be found in [3].

This paper presents numerical studies for the Navier–Stokes equations in the case of two-dimensional laminar time-dependent flows. The concentration on laminar flows (instead of turbulent ones) avoids the use of a turbulence model. The turbulence model would be an additional factor which influences the computational results. Even for the two-dimensional laminar regime, the question of an optimal discretization approach is not yet answered.

The most important requirements for the numerical solution of the Navier–Stokes equations are accuracy and efficiency. It turned out in the last decade that for accurate results temporal discretizations of at least second order [4,5] and spatial discretizations with at least second-order velocity and first-order pressure [4,6,7,5] should be used. In the numerical studies presented in this paper, always the Q_2/P_1^{disc} finite element discretization is applied which fulfills this order condition. The use of even higher order discretizations in space leads to severe difficulties in the solution of the arising discrete problems, e.g., see [7,8]. Thus, the second requirement, a fast solution of (1), will not be met for such discretizations. Also the use of explicit time stepping schemes, whose length of the time step is restricted by the CFL condition results in general in an inefficient solution process, [4]. There are various other schemes for solving time-dependent (partial) differential equations like the generalized-α method, the linear continuous in time finite element scheme, and constant and linear discontinuous in time finite element schemes. An analytic study of these methods applied to a stabilized formulation of (1) can be found in [9].

This paper presents a numerical comparison of two approaches:

- applying first a semi-discretization in time with an implicit θ-scheme; applying second a fixed point iteration as linearization in each discrete time; discretizing third the linear problem in space with a finite element method;
- applying first a Rosenbrock scheme with s stages as semi-discretization; the solution of the non-linear problem requires then s solutions of linear systems in each discrete time; the linear systems are discretized with a finite element method.

These approaches are described in detail in Sections 2 and 3, respectively. Five examples with distinct features have been chosen for the numerical studies which are presented in Section 4. The results are summarized in Section 5.

2. Implicit θ-schemes as semi-discretization in time followed by a fixed point iteration

This approach uses for the discretization of (1) the following strategy:

(i) *Semi-discretization of (1) in time.* An implicit time stepping scheme will be applied first. The semi-discretization in time leads in each discrete time step to a non-linear system of equations.
(ii) *Variational formulation and linearization.* The non-linear system of equations is reformulated as variational problem and the non-linear variational problem is linearized by a fixed point iteration.
(iii) *Discretization of the linear systems in space.* The linear system of equations arising in each step of the fixed point iteration is discretized by a finite element method using an inf–sup stable pair of finite element spaces.

The individual steps in this strategy are described in detail now.

Let Δt_n be the current time step from t_{n-1} to t_n, i.e., $\Delta t_n = t_n - t_{n-1}$. We denote quantities at time level t_k by a subscript k. To describe the time stepping scheme for the incompressible Navier–Stokes equations (1), a general time step of the form

$$\begin{aligned} &\mathbf{u}_k + \theta_1 \Delta t_n [-Re^{-1}\Delta \mathbf{u}_k + (\mathbf{u}_k \cdot \nabla)\mathbf{u}_k] + \Delta t_k \nabla p_k \\ &\quad = \mathbf{u}_{k-1} - \theta_2 \Delta t_n [-Re^{-1}\Delta \mathbf{u}_{k-1} + (\mathbf{u}_{k-1} \cdot \nabla)\mathbf{u}_{k-1}] + \theta_3 \Delta t_n \mathbf{f}_{k-1} + \theta_4 \Delta t_n \mathbf{f}_k, \\ &\nabla \cdot \mathbf{u}_k = 0 \end{aligned} \tag{2}$$

is introduced, with the parameters $\theta_1, \ldots, \theta_4$. The time step (2) allows the implementation of a number of time stepping schemes by one single formula and the choice between the schemes by setting four parameters.

Three well-known one-step θ-schemes are obtained by appropriate choices of these parameters, see Table 1.

The fractional-step θ-scheme, [10,11], is obtained by three suitable steps of form (2). We want to present two variants of this scheme. Let

$$\theta = 1 - \frac{\sqrt{2}}{2}, \quad \tilde{\theta} = 1 - 2\theta, \quad \tau = \frac{\tilde{\theta}}{1-\theta}, \quad \eta = 1 - \tau.$$

Table 1
One-step θ-schemes

	θ_1	θ_2	θ_3	θ_4	t_{k-1}	t_k	Δt_k
Forward Euler scheme	0	1	1	0	t_{n-1}	t_n	Δt_n
Backward Euler scheme (BWE)	1	0	0	1	t_{n-1}	t_n	Δt_n
Crank–Nicolson scheme (CN)	0.5	0.5	0.5	0.5	t_{n-1}	t_n	Δt_n

Table 2
The two variants of the fractional-step θ-schemes

	θ_1	θ_2	θ_3	θ_4	t_{k-1}	t_k	Δt_k
FS0	$\tau\theta$	$\eta\theta$	$\eta\theta$	$\tau\theta$	t_{n-1}	$t_{n-1}+\theta\Delta t_n$	$\theta\Delta t_n$
	$\eta\tilde{\theta}$	$\tau\tilde{\theta}$	$\tau\tilde{\theta}$	$\eta\tilde{\theta}$	$t_{n-1}+\theta\Delta t_n$	$t_n-\theta\Delta t_n$	$\tilde{\theta}\Delta t_n$
	$\tau\theta$	$\eta\theta$	$\eta\theta$	$\tau\theta$	$t_n-\theta\Delta t_n$	t_n	$\theta\Delta t_n$
FS1	$\tau\theta$	$\eta\theta$	θ	0	t_{n-1}	$t_{n-1}+\theta\Delta t_n$	$\theta\Delta t_n$
	$\eta\tilde{\theta}$	$\tau\tilde{\theta}$	0	$\tilde{\theta}$	$t_{n-1}+\theta\Delta t_n$	$t_n-\theta\Delta t_n$	$\tilde{\theta}\Delta t_n$
	$\tau\theta$	$\eta\theta$	θ	0	$t_n-\theta\Delta t_n$	t_n	$\theta\Delta t_n$

The two variants, FS0 and FS1, are presented in Table 2. FS1 requires the evaluation of $\mathbf{f}$ only at the times t_{n-1} and $t_n-\theta\Delta t_n$ whereas FS0 needs the evaluation of $\mathbf{f}$ in addition at $t_{n-1}+\theta\Delta t_n$ and at t_n. Both variants are second-order schemes but FS1 does not integrate second-order polynomials (with respect to t) exactly. However, most of other fundamental properties, like stability, are the same for both variants, [12].

There are a number of investigations of the time discretizations introduced above applied to the Navier–Stokes equations, see Gresho and Sani [3, Section 3.16] or Emmrich [13, Section 4.1] for a survey of the present state of art. The Crank–Nicolson scheme was studied by Temam [14], Heywood and Rannacher [15] and Bause [16] for the already spatially discretized Navier–Stokes equations (with a finite element method). One can prove, under a number of assumptions on the smoothness of the data, that the error between the time discrete and the time-continuous finite element velocity in $L^\infty(0,T;L^2(\Omega))$ behaves like $(\Delta t)^2$ for the equidistant time step Δt. The fractional-step θ-scheme was investigated analytically by Klouček and Rys [17] and Müller-Urbaniak [12]. A second-order error estimate similar to the Crank–Nicolson scheme was proved in [12].

The Crank–Nicolson and the fractional-step θ-scheme are widely used in the numerical solution of the incompressible Navier–Stokes equations, [18,5]. The Crank–Nicolson scheme is A-stable whereas the fractional-step θ-scheme is even strongly A-stable. That means, the Crank–Nicolson scheme may lead to numerical oscillations in problems with rough initial data or boundary conditions. These oscillations are damped out only if sufficiently small time steps are used. Compared to the fractional-step θ-scheme, a smaller time step might be necessary for the Crank–Nicolson scheme to ensure robustness.

The solution of (2) will be approximated by a finite element method. Finite element methods are popular and successful spatial discretizations used in computational fluid dynamics. The basis of the finite element method is a variational formulation of (2). For simplicity, we will consider the case that (1) is equipped with no-slip or homogeneous Dirichlet boundary conditions since in this case ansatz and test spaces are the same in the variational formulation. Let $V=(H_0^1(\Omega))^2$, $Q=L_0^2(\Omega)$. The derivation of the variational problem is done in the usual way by multiplying the equations in (2) with test functions, integrating over Ω and applying integration by parts. The variational problem is to find $(\mathbf{u}_k,p_k)\in(V,Q)$ such that for all $(\mathbf{v},q)\in(V,Q)$

$$\begin{aligned}&(\mathbf{u}_k,\mathbf{v})+\theta_1\Delta t_n[(Re^{-1}\nabla\mathbf{u}_k,\nabla\mathbf{v})+((\mathbf{u}_k\cdot\nabla)\mathbf{u}_k,\mathbf{v})]-\Delta t_k(p_k,\nabla\cdot\mathbf{v})\\&\quad=(\mathbf{u}_{k-1},\mathbf{v})+\theta_3\Delta t_n(\mathbf{f}_{k-1},\mathbf{v})+\theta_4\Delta t_n(\mathbf{f}_k,\mathbf{v})-\theta_2\Delta t_n[(Re^{-1}\nabla\mathbf{u}_{k-1},\nabla\mathbf{v})+((\mathbf{u}_{k-1}\cdot\nabla)\mathbf{u}_{k-1},\mathbf{v})],\\&0=(\nabla\cdot\mathbf{u}_k,q),\end{aligned}\tag{3}$$

where $(\mathbf{v},\mathbf{w})=\int_\Omega\mathbf{v}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}$.

The non-linear system (3) is solved iteratively starting with an initial guess $(\mathbf{u}_k^0,p_k^0)$. Given $(\mathbf{u}_k^m,p_k^m)$, the iterate $(\mathbf{u}_k^{m+1},p_k^{m+1})$ is computed by solving

$$\begin{aligned}&(\mathbf{u}_k^{m+1},\mathbf{v})+\theta_1\Delta t_n[(Re^{-1}\nabla\mathbf{u}_k^{m+1},\nabla\mathbf{v})+((\mathbf{u}_k^m\cdot\nabla)\mathbf{u}_k^{m+1},\mathbf{v})]-\Delta t_k(p_k^{m+1},\nabla\cdot\mathbf{v})\\&\quad=(\mathbf{u}_{k-1},\mathbf{v})+\theta_3\Delta t_n(\mathbf{f}_{k-1},\mathbf{v})+\theta_4\Delta t_n(\mathbf{f}_k,\mathbf{v})-\theta_2\Delta t_n[Re^{-1}(\nabla\mathbf{u}_{k-1},\nabla\mathbf{v})+((\mathbf{u}_{k-1}\cdot\nabla)\mathbf{u}_{k-1},\mathbf{v})],\\&0=(\nabla\cdot\mathbf{u}_k^{m+1},q)\end{aligned}\tag{4}$$

for all $(\mathbf{v},q)\in(V,Q)$, $m=0,1,2,\ldots$ That means, the linearization is done by a fixed point iteration. In a recent numerical study for laminar time-dependent Navier–Stokes equations, [8], this fixed point has been proven to be superior to a Newton type method for linearizing systems of type (3). The initial guess is chosen to be the solution of the previous time step $(\mathbf{u}_k^0,p_k^0)=(\mathbf{u}_{k-1},p_{k-1})$. Eqs. (4) are called Oseen equations. For larger Reynolds numbers, they are convection dominated.

Eqs. (4) are discretized by a finite element method (FEM). Let (V^h,Q^h) be a pair of finite element spaces which fulfill the inf–sup stability condition. Then, the finite element problem has the following form (indices $k,m,m+1$ will be neglected): Find $(\mathbf{u}^h,p^h)\in V^h\times Q^h$ such that

$$\begin{aligned}&(\mathbf{u}^h,\mathbf{v}^h)+\theta_1\Delta t_n[(Re^{-1}\nabla\mathbf{u}^h,\nabla\mathbf{v}^h)+((\mathbf{u}^h_{\text{old}}\cdot\nabla)\mathbf{u}^h,\mathbf{v}^h)]-\Delta t_k(p^h,\nabla\cdot\mathbf{v}^h)\\&\quad=(\mathbf{u}^h_{k-1},\mathbf{v}^h)+\theta_3\Delta t_n(\mathbf{f}^h_{k-1},\mathbf{v}^h)+\theta_4\Delta t_n(\mathbf{f}^h_k,\mathbf{v}^h)-\theta_2\Delta t_n[Re^{-1}(\nabla\mathbf{u}^h_{k-1},\nabla\mathbf{v}^h)+((\mathbf{u}^h_{k-1}\cdot\nabla)\mathbf{u}^h_{k-1},\mathbf{v})],\\&0=(\nabla\cdot\mathbf{u}^h,q^h)\end{aligned}\tag{5}$$

for all $(\mathbf{v}^h,q^h)\in(V^h\times Q^h)$. Here, $\mathbf{u}^h_{\text{old}}\in V^h$ is the current approximation of the velocity while $\mathbf{f}^h_{k-1}$ and $\mathbf{f}^h_k$ are finite element representations of the right-hand side at the times t_{k-1} and t_k, respectively. Numerical experiences show that it is in general much more efficient to solve (5) only approximately in each step of the fixed point iteration instead of solving it always accurately [18,7]. In the computations presented in this paper, the solution of (5) was stopped after having reduced the Euclidean norm of the initial residual by the factor 10.

3. The vertical method of lines

The Navier–Stokes equations (1) can be written in the abstract form

$$M\dot{\mathbf{w}}=F(t,\mathbf{w}),\quad \mathbf{w}(t_0)=\mathbf{w}_0\tag{6}$$

with $\mathbf{w}=(\mathbf{u},p)^{\mathrm{T}}$ and $\dot{\mathbf{w}}$ denotes the derivative of $\mathbf{w}$ with respect to time. If the square matrix M is non-singular, (6) is an implicit ordinary differential equation (ODE). However, due to the incompressible constraint, M is singular for the Navier–Stokes equations and (6) is a differential algebraic equation (DAE).

There are many solvers known for solving implicit ODEs or DAEs of form (6). However, (6) is often stiff. Thus, explicit solvers will fail generally when they are applied. The use of implicit methods becomes necessary. Fully implicit Runge–Kutta methods (IRK-methods) are one attempt to approximate the solution of (6). They have very good stability properties, but require in general the solution of non-linear equations. Hence, IRK-methods are rather expensive schemes. There are approaches which require only the solution of one linear system in each time step like linearly implicit RK-methods (LIRK-methods), adaptive RK-methods or the popular Rosenbrock methods. The price of the smaller numerical costs in these schemes is a reduced stability in comparison to IRK-methods. Furthermore, the Jacobian has to be computed. If so-called W-methods are used then only an approximation of the Jacobian matrix is needed. It is possible to integrate (6) with a fixed Jacobian calculated at some previous time step. Of course, the convergence of a W-method can be accelerated if this matrix is a good approximation of the Jacobian, for example Ostermann created Rosenbrock methods with $J=\partial_u F+\mathrm{O}(\Delta t_n)$ (i.e., [19]). Standard books on this topic are [20,21]. In [21], one can find many information about the classification of W-methods.

We will concentrate in our numerical studies on Rosenbrock methods. It is well known that solvers for (6) may have order reduction if they are applied to a semi-discretized parabolic problem. Rosenbrock methods can avoid this order reduction if some additional conditions are fulfilled. However, many Rosenbrock solvers need the Jacobian and the time derivative of the right-hand side F in each discrete time.

In the remainder of this section, we will describe the application of Rosenbrock methods to the Navier–Stokes equations (1) in detail.

3.1. The semi-discretization of the Navier–Stokes equations in space

First, the Navier–Stokes equations (1) are semi-discretized in space with a finite element method. For simplicity of presentation, we consider only the case of homogeneous Dirichlet boundary conditions. Writing (1) component-wise

$$\dot{u}_1-Re^{-1}\Delta u_1+u_1\partial_x u_1+u_2\partial_y u_1+\partial_x p=f_1,\tag{7}$$

$$\dot{u}_2-Re^{-1}\Delta u_2+u_1\partial_x u_2+u_2\partial_y u_2+\partial_y p=f_2,\tag{8}$$

$$\partial_x u_1+\partial_y u_2=0\tag{9}$$

with $\mathbf{u}=(u_1,u_2)^{\mathrm{T}}$, multiplying (7)–(9) by test functions $\mathbf{v}=(v_1,v_2)^{\mathrm{T}}\in V$, $q\in Q$, integrating over Ω and applying integration by parts, we obtain the following weak problem: Find $(\mathbf{u},p)\in L^2(0,T;V)\times L^2(0,T;Q)$ with $u=(u_1,u_2)^{\mathrm{T}}$ such that a.e. in $(0,T)$

$$
\begin{aligned}
&(\dot{u}_1, v_1) + Re^{-1}(\nabla u_1, \nabla v_1) + (u_1 \partial_x u_1 + u_2 \partial_y u_1, v_1) + (\partial_x p, v_1) = (f_1, v_1),\\
&(\dot{u}_2, v_2) + Re^{-1}(\nabla u_2, \nabla v_2) + (u_1 \partial_x u_2 + u_2 \partial_y u_2, v_2) + (\partial_y p, v_2) = (f_2, v_2),\\
&(u_1, \partial_x q) + (u_2, \partial_y q) = 0
\end{aligned}
$$

holds for all $(\mathbf{v}, q) \in V \times Q$. The finite element method approximates the solution of the weak problem in some finite dimensional subspaces $V^h \subset V$ and $Q^h \subset Q$. Let $V^h = W^h \times W^h$, $\{\varphi_1, \ldots, \varphi_{N_u}\}$ be a basis of W^h and $\{\psi_1, \ldots, \psi_{N_p}\}$ be a basis of Q^h. The following matrices and vectors are defined:

$$
\begin{aligned}
&(M)_{ij} = (\varphi_j, \varphi_i), && i,j = 1, \ldots, N_u,\\
&(A_{11})_{ij} = Re^{-1}(\nabla \varphi_j, \nabla \varphi_i) + (u_1 \partial_x \varphi_j + u_2 \partial_y \varphi_j, \varphi_i), && i,j = 1, \ldots, N_u,\\
&(A_{22})_{ij} = (A_{11})_{ij}, && i,j = 1, \ldots, N_u,\\
&(B_1)_{ij} = (\partial_x \psi_j, \varphi_i), && i = 1, \ldots, N_u,\ \ j = 1, \ldots, N_p,\\
&(B_2)_{ij} = (\partial_y \psi_j, \varphi_i), && i = 1, \ldots, N_u,\ \ j = 1, \ldots, N_p,\\
&(f_k)_i = (f_k, \varphi_i), && i = 1, \ldots, N_u,\ \ k = 1, 2.
\end{aligned}
$$

With these settings, we get the so-called MOL-DAE (MOL—method of lines)

$$
\begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{pmatrix}
\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{p} \end{pmatrix}
=
\begin{pmatrix} f_1 \\ f_2 \\ 0 \end{pmatrix}
-
\begin{pmatrix} A_{11} & 0 & B_1 \\ 0 & A_{22} & B_2 \\ B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & 0 \end{pmatrix}
\begin{pmatrix} u_1 \\ u_2 \\ p \end{pmatrix}
$$

or shorter

$$
H\dot{\mathbf{w}} = -A(\mathbf{w})\mathbf{w} + \mathbf{f} =: F(t, \mathbf{w}). \tag{10}
$$

The differentiation index of the MOL-DAE (10) is 2, see [22]. In [23], it is shown that the perturbation index equals also to 2.

3.2. General Rosenbrock methods

Let $\Delta t_n > 0$ be the size of the time step from t_{n-1} to t_n. An s-stage Rosenbrock method applied to (10) has the form

$$
\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta t_n \sum_{i=1}^{s} b_i \mathbf{K}_{ni}
$$

with

$$
(H - \Delta t_n \gamma A_n)\mathbf{K}_{ni} = F\left(t_n + \alpha_i \Delta t_n, \mathbf{w}_n + \Delta t_n \sum_{j=1}^{i-1} \alpha_{ij} \mathbf{K}_{nj}\right) + \Delta t_n A_n \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_{nj} + \Delta t_n \gamma_i C_n, \tag{11}
$$

where

$$
C_n := \frac{\partial F}{\partial t}(t, \mathbf{w})\bigg|_{t=t_n, \mathbf{w}=\mathbf{w}_n}
$$

and A_n is the Jacobian of F with respect to $\mathbf{w}$ at $t = t_n$ and $\mathbf{w} = \mathbf{w}_n$. The coefficients of the method are γ, α_i, γ_i, γ_{ij}, α_{ij} and b_i. In each time step, s linear systems with the same matrix have to be solved to compute $\mathbf{K}_{n1}, \ldots, \mathbf{K}_{ns}$. However, the right-hand side of the $(i+1)$th system depends on the solution of the first to the ith system, see (11). The parameters in (11) should be chosen in such a way that some order conditions are fulfilled to obtain a sufficient consistency order. A derivation of these conditions with Butcher series can be found in [20]. Here we only summarize the conditions up to order 3 for $s = 3$

$$
\begin{cases}
b_1 + b_2 + b_3 = 1,\\
b_2\beta_2 + b_3\beta_3 = \frac{1}{2} - \gamma,\\
b_2\alpha_2^2 + b_3\alpha_3^2 = \frac{1}{3},\\
b_3\beta_2\beta_{32} = \frac{1}{6} - \gamma + \gamma^2,
\end{cases} \tag{12}
$$

where we use the abbreviations $\beta_{ij} := \alpha_{ij} + \gamma_{ij}$ and $\beta_i := \sum_{j=1}^{i-1} \beta_{ij}$. We get an additional consistency condition if we assume that A_n is an $O(\Delta t_n)$ disturbance of the Jacobian

$$
b_2\alpha_2 + b_3\alpha_3 = \frac{1}{2}. \tag{13}
$$

If the Jacobian is replaced by an arbitrary matrix W we get

$$\begin{cases} b_3\alpha_{32}\alpha_2 = \frac{1}{6}, \\ b_3\alpha_{32}\beta_2 = \frac{1}{6} - \frac{\gamma}{2}, \\ b_3\beta_{32}\alpha_2 = \frac{1}{6} - \frac{\gamma}{2}. \end{cases} \tag{14}$$

For stability reasons, the matrix W should be an approximation of the Jacobian. If a Rosenbrock method is applied to semi-discretized PDEs, the following condition has to be satisfied to avoid order reduction:

$$b^{\mathrm{T}}B^j(2B^2e - \alpha^2) = 0, \qquad 1 \leqslant j \leqslant 2 \tag{15}$$

with $B := (\beta_{ij})_{i,j=1}^{s}$, $\alpha^2 := (\alpha_1^2, \ldots, \alpha_s^2)^{\mathrm{T}}$ and $e := (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^s$. With (12), the conditions (15) simplify to

$$\begin{cases} b_3\beta_{32}\alpha_2^2 = \frac{1}{6} - \frac{2}{3}\gamma, \\ \gamma = \frac{1}{2} + \frac{1}{6}\sqrt{3}. \end{cases} \tag{16}$$

For a third-order method with three stages, we get the algebraic condition

$$b_2\omega_{22}\alpha_2^2 + b_3(\omega_{32}\alpha_2^2 + \omega_{33}\alpha_3^2) = 1, \tag{17}$$

where $(\omega_{ij})_{i,j=1}^{s} = B^{-1}$. A Rosenbrock method which satisfies (12) and (16) fulfills (17), see [24].

The stability function is given by

$$R_0(z) = 1 + zb^{\mathrm{T}}(I - zB)^{-1}e.$$

A Rosenbrock method satisfying

$$\beta_{si} = b_i, \quad i = 1, \ldots, s, \quad \text{and} \quad \alpha_s = 1 \tag{18}$$

is called stiffly accurate. Methods satisfying (18) yield asymptotically exact results for the problem $\dot{u} = \lambda(u - \varphi(t)) + \dot{\varphi}(t)$.

In the same way as for Runge–Kutta methods, the step size control can be done with an embedded formula. Therefore the coefficients b_i in (11) are replaced by different coefficients $\hat{b}_i$ which yield a second solution $\hat{\mathbf{w}}_n$ of inferior order, in general one order less. There are at least two possibilities to estimate the numerical error $\|\mathbf{w}_n - \hat{\mathbf{w}}_n\|$ which can be found in the books [25,20]. In our numerical tests, we will not exploit the ability of the step length control since we like to compare the Rosenbrock methods with the implicit θ-schemes on the same meshes in space and time.

3.3. The Rosenbrock methods tested in numerical studies

We will study Rosenbrock methods with 3 and 4 stages in our numerical tests.

Lang and Verwer constructed in [24] the solver ROS3P. This method is of order 3, A-stable with $R_0(\infty) = \sqrt{3} - 1 \approx 0.73$ and fulfills the order condition (17). The coefficients are

$\gamma = 7.886751346999999\text{e}{-01}$	
$\alpha_{21} = 1.000000000000000\text{e}{+00}$	$\gamma_{21} = -1.000000000000000\text{e}{+00}$
$\alpha_{31} = 1.000000000000000\text{e}{+00}$	$\gamma_{31} = -7.886751346999999\text{e}{-01}$
$\alpha_{32} = 0.000000000000000\text{e}{+00}$	$\gamma_{32} = -1.077350269000000\text{e}{+00}$
$b_1 = 6.666666667000000\text{e}{-01}$	$\hat{b}_1 = 3.333333333000000\text{e}{-01}$
$b_2 = 0.000000000000000\text{e}{+00}$	$\hat{b}_2 = 3.333333333000000\text{e}{-01}$
$b_3 = 3.333333333000000\text{e}{-01}$	$\hat{b}_3 = 3.333333333000000\text{e}{-01}$

ROWDAIND2 is a stiffly accurate solver of order 3 designed for index 2 problems, see [26], with the coefficients

$\gamma = 3.000000000000000\text{e}{-01}$	
$\alpha_{21} = 5.000000000000000\text{e}{-01}$	$\gamma_{21} = -1.121794871794876\text{e}{-01}$
$\alpha_{31} = 2.800000000000000\text{e}{-01}$	$\gamma_{31} = 2.540000000000000\text{e}{+00}$
$\alpha_{32} = 7.200000000000000\text{e}{-01}$	$\gamma_{32} = -3.840000000000000\text{e}{+00}$
$\alpha_{41} = 2.800000000000000\text{e}{-01}$	$\gamma_{41} = 3.866666666666667\text{e}{-01}$
$\alpha_{42} = 7.200000000000000\text{e}{-01}$	$\gamma_{42} = -7.200000000000000\text{e}{-01}$
$\alpha_{43} = 0.000000000000000\text{e}{+00}$	$\gamma_{43} = 3.333333333333333\text{e}{-02}$
$b_1 = 6.666666666666666\text{e}{-01}$	$\hat{b}_1 = 4.799002800355166\text{e}{-01}$
$b_2 = 0.000000000000000\text{e}{+00}$	$\hat{b}_2 = 5.176203811215082\text{e}{-01}$

$b_3 = 3.333333333333333e{-}02$	$\hat{b}_3 = 2.479338842975209e{-}03$
$b_4 = 3.000000000000000e{-}01$	$\hat{b}_4 = 0.000000000000000e{+}00$

The third-order method ROS3Pw is a strongly A-stable method with $R_0(\infty) = \sqrt{3} - 1 \approx 0.73$, see [27], and the coefficients

$\gamma = 7.8867513459481287e{-}01$	
$\alpha_{21} = 1.5773502691896257e{+}00$	$\gamma_{21} = -1.5773502691896257e{+}00$
$\alpha_{31} = 5.0000000000000000e{-}01$	$\gamma_{31} = -6.7075317547305480e{-}01$
$\alpha_{32} = 0.0000000000000000e{+}00$	$\gamma_{32} = -1.7075317547305482e{-}01$
$b_1 = 1.0566243270259355e{-}01$	$\hat{b}_1 = -1.7863279495408180e{-}01$
$b_2 = 4.9038105676657971e{-}02$	$\hat{b}_2 = 3.3333333333333333e{-}01$
$b_3 = 8.4529946162074843e{-}01$	$\hat{b}_3 = 8.4529946162074843e{-}01$

ROS34PW2 is a stiffly accurate W-method of order 3, see [27], whose coefficients are

$\gamma = 4.3586652150845900e{-}01$	
$\alpha_{21} = 8.7173304301691801e{-}01$	$\gamma_{21} = -8.7173304301691801e{-}01$
$\alpha_{31} = 8.4457060015369423e{-}01$	$\gamma_{31} = -9.0338057013044082e{-}01$
$\alpha_{32} = -1.1299064236484185e{-}01$	$\gamma_{32} = 5.4180672388095326e{-}02$
$\alpha_{41} = 0.0000000000000000e{+}00$	$\gamma_{41} = 2.4212380706095346e{-}01$
$\alpha_{42} = 0.0000000000000000e{+}00$	$\gamma_{42} = -1.2232505839045147e{+}00$
$\alpha_{43} = 1.0000000000000000e{+}00$	$\gamma_{43} = 5.4526025533510214e{-}01$
$b_1 = 2.4212380706095346e{-}01$	$\hat{b}_1 = 3.7810903145819369e{-}01$
$b_2 = -1.2232505839045147e{+}00$	$\hat{b}_2 = -9.6042292212423178e{-}02$
$b_3 = 1.5452602553351020e{+}00$	$\hat{b}_3 = 5.0000000000000000e{-}01$
$b_4 = 4.3586652150845900e{-}01$	$\hat{b}_4 = 2.1793326075422950e{-}01$

ROS34PW3 is a strongly A-stable W-method of order 3 with $R_0(\infty) \approx 0.63$. The coefficients are given by

$\gamma = 1.0685790213016289e{+}00$	
$\alpha_{21} = 2.5155456020628817e{+}00$	$\gamma_{21} = -2.5155456020628817e{+}00$
$\alpha_{31} = 5.0777280103144085e{-}01$	$\gamma_{31} = -8.7991339217106512e{-}01$
$\alpha_{32} = 7.5000000000000000e{-}01$	$\gamma_{32} = -9.6014187766190695e{-}01$
$\alpha_{41} = 1.3959081404277204e{-}01$	$\gamma_{41} = -4.1731389379448741e{-}01$
$\alpha_{42} = -3.3111001065419338e{-}01$	$\gamma_{42} = 4.1091047035857703e{-}01$
$\alpha_{43} = 8.2040559712714178e{-}01$	$\gamma_{43} = -1.3558873204765276e{+}00$
$b_1 = 2.2047681286931747e{-}01$	$\hat{b}_1 = 3.1300297285209688e{-}01$
$b_2 = 2.7828278331185935e{-}03$	$\hat{b}_2 = -2.8946895245112692e{-}01$
$b_3 = 7.1844787635140066e{-}03$	$\hat{b}_3 = 9.7646597959903003e{-}01$
$b_4 = 7.6955588053404989e{-}01$	$\hat{b}_4 = 0.0000000000000000e{+}00$

3.4. The implementation of the Rosenbrock methods

In the case of Navier–Stokes equations, we have

$$A_n = \begin{pmatrix} J_{11} & J_{12} & B_1 \\ J_{21} & J_{22} & B_2 \\ B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & 0 \end{pmatrix}, \quad C_n = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix},$$

where

$$(J_{11})_{ij} = (A_{11})_{ij} + ((\partial_x u_1)\varphi_j, \varphi_i), \quad (C_1)_i = (\dot{f}_1, \varphi_i),$$
$$(J_{12})_{ij} = ((\partial_y u_1)\varphi_j, \varphi_i), \quad (C_2)_i = (\dot{f}_2, \varphi_i),$$
$$(J_{21})_{ij} = ((\partial_x u_2)\varphi_j, \varphi_i), \quad (C_3)_i = 0,$$
$$(J_{22})_{ij} = (A_{22})_{ij} + ((\partial_y u_2)\varphi_j, \varphi_i).$$

We will transform the Rosenbrock method (11) to reduce the number of needed matrix-vector operations. For this reason, the new variables

$$\mathbf{U}_{ni} = \Delta t_n \sum_{j=1}^{i} \gamma_{ij} \mathbf{K}_{nj}, \quad i = 1, \ldots, s$$

are introduced. Since $\gamma_{ii} = \gamma > 0$ and $\gamma_{ij} = 0$ for $j > i$, the matrix $\Gamma = (\gamma_{ij})_{i,j=1}^{s}$ is invertible and the $\mathbf{K}_{ni}$ can be recovered from the $\mathbf{U}_{ni}$ via

$$\mathbf{K}_{ni} = \frac{1}{\Delta t_n} \sum_{j=1}^{i} c_{ij} \mathbf{U}_{nj}, \quad (c_{ij})_{i,j=1}^{s} = \Gamma^{-1}.$$

Inserting this formula into (11) yields by a straightforward calculation

$$\left(\frac{1}{\Delta t_n \gamma} H - A_n\right) \mathbf{U}_{ni} = F\left(t_n + \alpha_i \Delta t_n, \mathbf{w}_n + \sum_{j=1}^{i-1} a_{ij} \mathbf{U}_{nj}\right) - H \sum_{j=1}^{i-1} \frac{c_{ij}}{\Delta t_n} \mathbf{U}_{nj} + \Delta t_n \gamma_i \partial_t F(t_n, \mathbf{w}_n) \tag{19}$$

and

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \sum_{i=1}^{s} m_i \mathbf{U}_{ni}$$

with the coefficients

$$(a_{ij})_{i,j=1}^{s} = (\alpha_{ij})_{i,j=1}^{s} \Gamma^{-1}, \quad (m_1, \ldots, m_s) = (b_1, \ldots, b_s)\Gamma^{-1}.$$

Because of the special form of the matrix H, see (11), the matrix-vector products with H in the right-hand side of (19) are cheaper than the matrix-vector products with A_n in (11).

After rescaling, the system matrix of the Rosenbrock scheme (19) takes the form

$$\begin{pmatrix} M - \Delta t_n \gamma J_{11} & -\Delta t_n \gamma J_{12} & B_1 \\ -\Delta t_n \gamma J_{21} & M - \Delta t_n \gamma J_{22} & B_2 \\ B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & 0 \end{pmatrix},$$

whereas the system matrix for the implicit θ-schemes (5) looks like

$$\begin{pmatrix} M - \Delta t_n \theta_1 A_{11} & 0 & B_1 \\ 0 & M - \Delta t_n \theta_1 A_{11} & B_2 \\ B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & 0 \end{pmatrix}.$$

Thus, in the Rosenbrock methods one has to assemble and to store four sparse $N_u \times N_u$ matrices instead of only one such matrix in the implicit θ-schemes. This advantage of the implicit θ-schemes will disappear if for the viscous term instead of the gradient formulation $(Re^{-1}\nabla \mathbf{u}^h, \nabla \mathbf{v}^h)$ the deformation tensor formulation $(2Re^{-1}\mathbb{D}(\mathbf{u}^h), \mathbb{D}(\mathbf{v}^h))$ is used. The velocity deformation tensor is the symmetric part of the velocity gradient. In the deformation tensor formulation, all four blocks A_{ij}, $i,j \in \{1,2\}$, are non-zero and they are different (with $A_{21} = A_{12}^{\mathrm{T}}$). The use of the deformation tensor formulation becomes necessary if the Navier–Stokes equations are equipped with certain boundary conditions, e.g., slip boundary conditions.

4. The numerical studies

The discretizations of the Navier–Stokes equations (1) described in the previous sections are studied in a number of examples which possess different features:

Example 4.1. The solution is known and the discretization error in space dominates.

Example 4.2. The solution is known and the discretization error in time dominates.

Example 4.3. The solution is known and the problem is very stiff in one component.

Example 4.4. A vortex decay problem computed with different values of the Reynolds number.

Example 4.5. A flow around a cylinder where the quantities of interest are the drag and the lift coefficient at the cylinder as well as the difference of the pressure values in front and behind the cylinder.

In all numerical studies, the mapped Q_2/P_1^{disc} finite element discretization on quadrilateral grids was used. That means, the velocity is approximated by a continuous function which is piecewise biquadratic and the pressure by a discontinuous piecewise linear function. This pair of finite element spaces is considered currently among the best performing ones in the numerical simulation of incompressible flows, see [3,6,7]. The inf–sup stability condition for this pair of finite element spaces is proven in [28]. In the temporal discretizations, always a uniform time step $\Delta t_n = \Delta t$ for all n was used. The linear systems were solved with a preconditioned flexible GMRES method [29]. The preconditioner is a coupled multigrid method with Vanka smoother which is described in detail in [6,7,30]. All computations were performed with the code MooNMD, [31].

Examples 4.1–4.4 are defined on the unit square. In these examples, we used a grid consisting of squares of edge length h with $h = 1/64$. The Q_2/P_1^{disc} finite element discretization contains 33,282 degrees of freedom (d.o.f.) for the velocity and 12,288 d.o.f. for the pressure.

In Examples 4.1–4.4, we will study the velocity error in the semi-norm of $L^2(0,T;H^1(\Omega))$

$$\|\mathbf{u} - \mathbf{u}^h\|_{L^2(0,T;H^1(\Omega))} = \left(\int_0^T \|(\nabla \mathbf{u} - \nabla \mathbf{u}^h)(t)\|^2_{L^2(\Omega)} \, \mathrm{d}t \right)^{1/2} \tag{20}$$

and the pressure error in $L^2(0,T;L^2(\Omega))$

$$\|p - p^h\|_{L^2(0,T;L^2(\Omega))} = \left(\int_0^T \|(p - p^h)(t)\|^2_{L^2(\Omega)} \, \mathrm{d}t \right)^{1/2}. \tag{21}$$

Some general comments to the results of the computational tests are collected in Section 4.6.

4.1. An example with a dominating space error

Let $\Omega = (0,1)^2$ and the solution of (1) given by

$$u_1 = \sin t \sin(\pi x) \sin(\pi y),$$
$$u_2 = \sin t \cos(\pi x) \cos(\pi y),$$
$$p = \sin t \left(\sin(\pi x) + \cos(\pi y) - \frac{2}{\pi} \right)$$

with $\mathbf{u} = (u_1, u_2)^{\mathrm{T}}$. The right-hand side $\mathbf{f}$, the initial condition $\mathbf{u}_0$ and the non-homogeneous Dirichlet boundary conditions are chosen such that $(u_1, u_2, p)^{\mathrm{T}}$ is the closed form solution of (1) for the given Reynolds number. We will present computations with different Reynolds numbers. The final time is set to be $T = 1$. The spatial grid consisted of squares with edge length $h = 1/64$. The fineness of the spatial grid is chosen in such a way that the discretization error in space will dominate the discretization error in time already for rather large time steps (for the most methods). The computations were done for the time steps $\Delta t = 0.1 \times 2^{-k}$, $k = 0, \ldots, 8$.

The small Reynolds number case $Re = 1$. Fig. 1 presents the results for $Re = 1$. Considering the velocity error in $L^2(0,T;H^1(\Omega))$, it can be observed that for all schemes (except FS1) the velocity error reduces to the discretization error in space for $\Delta t = 0.1 \times 2^{-k}$, $k \geqslant 2$. The best results for the pressure error in $L^2(0,T;L^2(\Omega))$ are obtained with the stiffly accurate method ROS34PW2 and ROWDAIND2. The errors for all θ-schemes are much larger than for the Rosenbrock methods. The largest errors have FS1 and CN.

The higher Reynolds number case $Re = 1000$. The results for $Re = 1000$ are presented in Fig. 2. The velocity error in $L^2(0,T;H^1(\Omega))$ is reduced to the discretization error in space for most of the discretizations when $\Delta t = 0.1 \times 2^{-k}$, $k \geqslant 2$. BWE is by far the most inaccurate scheme for all time steps. The situation for the pressure error in $L^2(0,T;L^2(\Omega))$ is similar as in the case $Re = 1$.

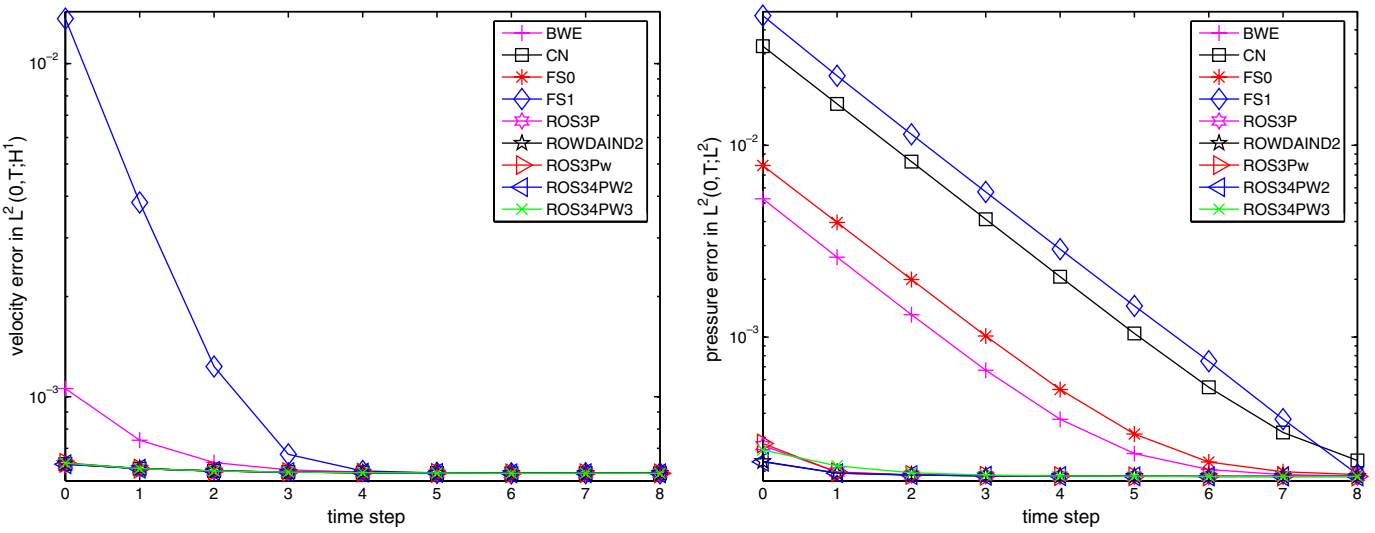

Fig. 1. Example 4.1, results for $Re = 1$.

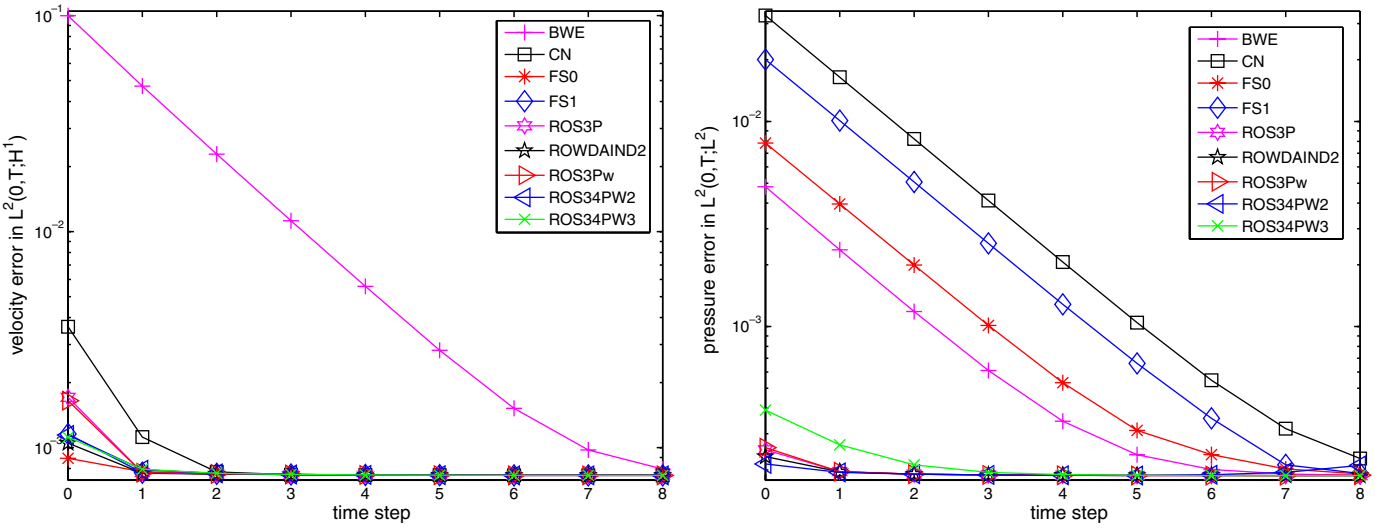

Fig. 2. Example 4.1, results for $Re = 1000$.

Summary of this example. If one is interested only in the velocity error in $L^2(0,T;H^1(\Omega))$, then CN, FS0 and all of the Rosenbrock schemes are comparable in accuracy for both Reynolds numbers. If the interest lies in the pressure error in $L^2(0,T;L^2(\Omega))$ or the error in the energy norm (square root of the sum of the velocity error squared and the pressure error squared), then the accuracy of the Rosenbrock methods for large time steps is achieved by the other methods only for much smaller time steps. If the pressure is of interest, the Rosenbrock methods are clearly to prefer to the other ones.

4.2. An example where only a discretization error in time occurs

Let $T = 1$ and $\Omega = (0,1)^2$. We consider (1) with $Re = 1$. The right-hand side $\mathbf{f}$, the initial condition $\mathbf{u}_0$ and the non-homogeneous Dirichlet boundary conditions are chosen such that

$$u_1(x,y) = t^3 y^2,$$
$$u_2(x,y) = t^2 x,$$
$$p(x,y) = tx + y - (t+1)/2$$

is the solution of (1). We used a mesh consisting of squares with edge length $h = 1/64$. Note that for any t the solution can be represented exactly by the finite element functions. Hence, all occurring errors will results from the temporal discretization. As time steps, we used $\triangle t = 0.1 \times 2^{-k}$, $k = 0, \ldots, 8$.

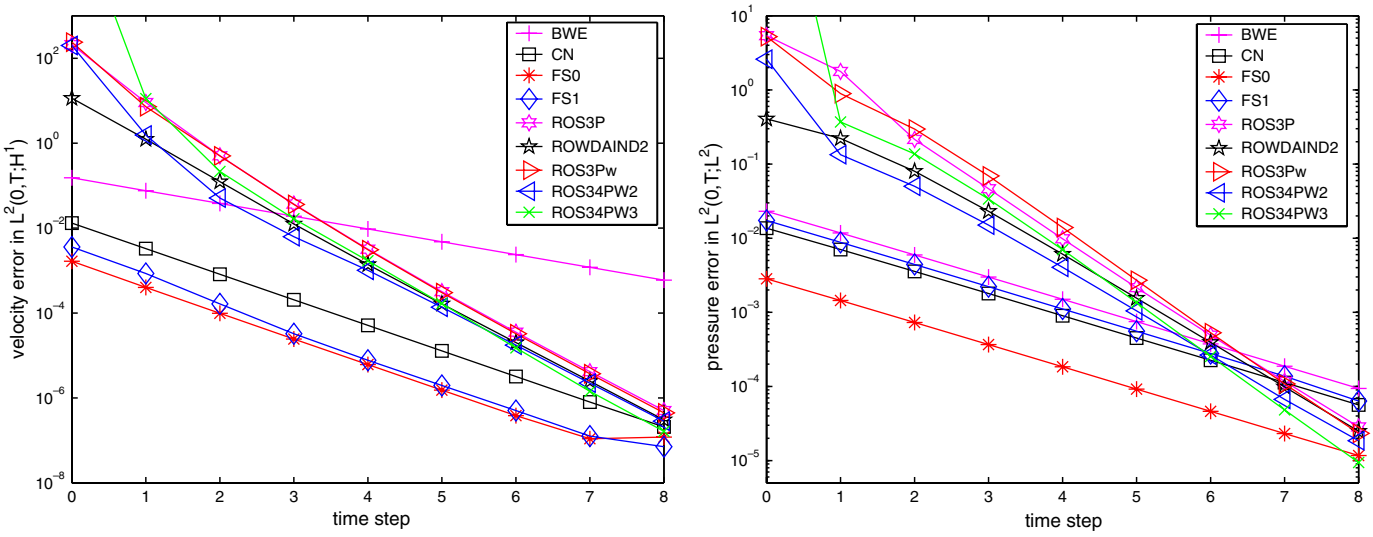

Fig. 3. Results for Example 4.2.

Fig. 3 shows the results of the calculations. The best velocity error in $L^2(0,1;H^1(\Omega))$ was obtained by the fractional-step θ-schemes FS0 and FS1 which differ only slightly. Also good results were produced by CN. All Rosenbrock methods behave almost the same. They have for large time steps larger errors than CN, FS0 and FS1. For smaller time steps the results of the Rosenbrock methods become better. BWE gave inaccurate results.

If we look at the pressure error in $L^2(0,1;L^2(\Omega))$, we see that the difference between the results of the Rosenbrock method is quite small. For large time steps, their errors are larger than the errors for the implicit θ-schemes. The relation changes for smaller time steps. It is interesting to note that there are large differences between FS0 and FS1 where FS0 performed much better.

4.3. A very stiff problem

A very popular example for a stiff ODE is the initial value problem

$$\dot{u} = -\lambda u, \quad \lambda \gg 1, \quad u(t_0) = u_0.$$

Explicit ODE solvers like explicit Runge–Kutta methods may have the problem that they cannot approximate in general the solution for any step sizes. From the theory of ODEs it is known that the allowed step size is bounded by $\Delta t < \lambda^{-1}$.

In the following example, we include this problem into the second component of $\mathbf{u}$ with $\lambda = 50$. Hence, we consider the Navier–Stokes equations (1) with the solution

$$u_1 = t^3 y^2,$$
$$u_2 = \exp(-50t)x,$$
$$p = (10 + t)\exp(-t)(x + y - 1)$$

in $\Omega = (0,1)^2$. The computations were carried out with $Re = 1$, $T = 1$ and a spatial grid consisting of squares of edge length $h = 1/64$.

The computations were done for the time steps $\Delta t = 0.1 \times 2^{-k}$, $k = 0,\ldots,8$, see Fig. 4. Considering the velocity error in $L^2(0,T;H^1(\Omega))$, it can be observed that FS0 gives very good results for large step sizes and that ROS34PW2 and ROS34PW3 give very good results for small step lengths. A similar observation can be made for the pressure error in $L^2(0,T;L^2(\Omega))$. For large step sizes, FS0 and FS1 gave the best results whereas all Rosenbrock yield very good results for small step lengths. The most inaccurate results with respect to both, the velocity and the pressure, were computed with BWE.

To summarize the results obtained in this example, FS0 behaves best for large time steps and all Rosenbrock methods with the exception ROWDAIND2, for small time steps. This is due to the fact that ROWDAIND2 was not designed to solve semi-discretized PDEs. The most inaccurate results were obtained with BWE.

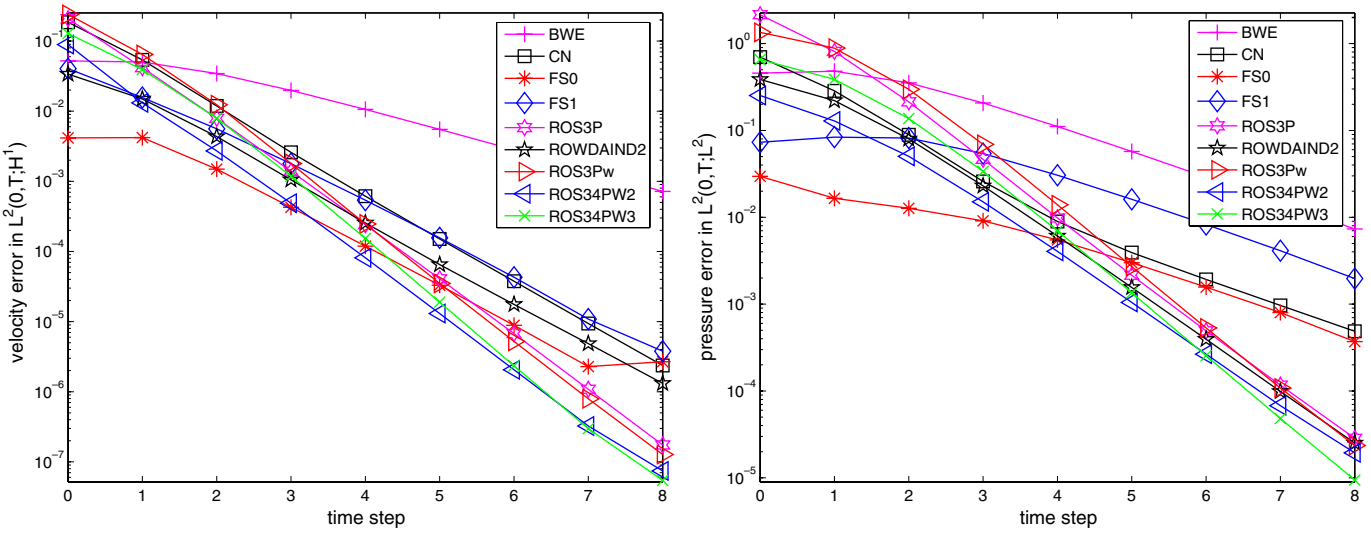

Fig. 4. Results for Example 4.3.

4.4. A vortex decay problem

The following problem, which can be found in [32], is defined in $\Omega = (0,1)^2$. The prescribed solution has the form

$$u_1 = -\cos(n\pi x)\sin(n\pi y)\exp(-2n^2\pi^2 t/\tau),$$
$$u_2 = \sin(n\pi x)\cos(n\pi y)\exp(-2n^2\pi^2 t/\tau),$$
$$p = -\frac{1}{4}(\cos(2n\pi x) + \cos(2n\pi y))\exp(-4n^2\pi^2 t/\tau).$$

For the relaxation time $\tau = Re$, this is a solution of the Navier–Stokes equations (1) with $\mathbf{f} = \mathbf{0}$ consisting of an array of oppositely signed vortices which decay exponentially as $t \to \infty$.

The right-hand side $\mathbf{f}$, the initial condition $\mathbf{u}_0$ and the non-homogeneous Dirichlet boundary conditions are chosen such that $(u_1, u_2, p)^{\mathrm{T}}$ is the closed form solution of (1) for a given set of parameters. We will present computations for the relaxation time $\tau = 1$, the vortex configuration $n = 4$ and the final time $T = 1$ with different Reynolds numbers on a fixed spatial grid consisting of squares with edge length $h = 1/64$.

The small Reynolds number case $Re = 1$. The computations were done for the time steps $\Delta t = 0.05 \times 2^{-k}$, $k = 0, \ldots, 6$, see Fig. 5 for the results. Concerning the velocity error in $L^2(0,T;H^1(\Omega))$, one can observe that both fractional-step θ-schemes FS0 and FS1 give very good results. Also the results obtained with ROS34PW2 and ROWDAIND2 are among the best

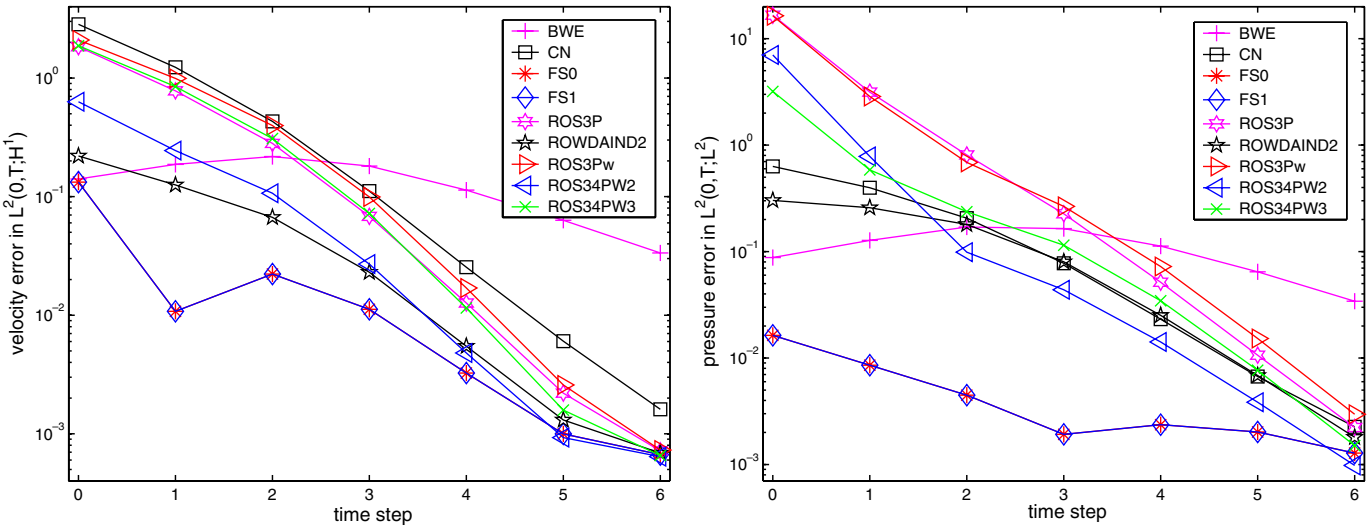

Fig. 5. Example 4.4, results for $Re = 1$.

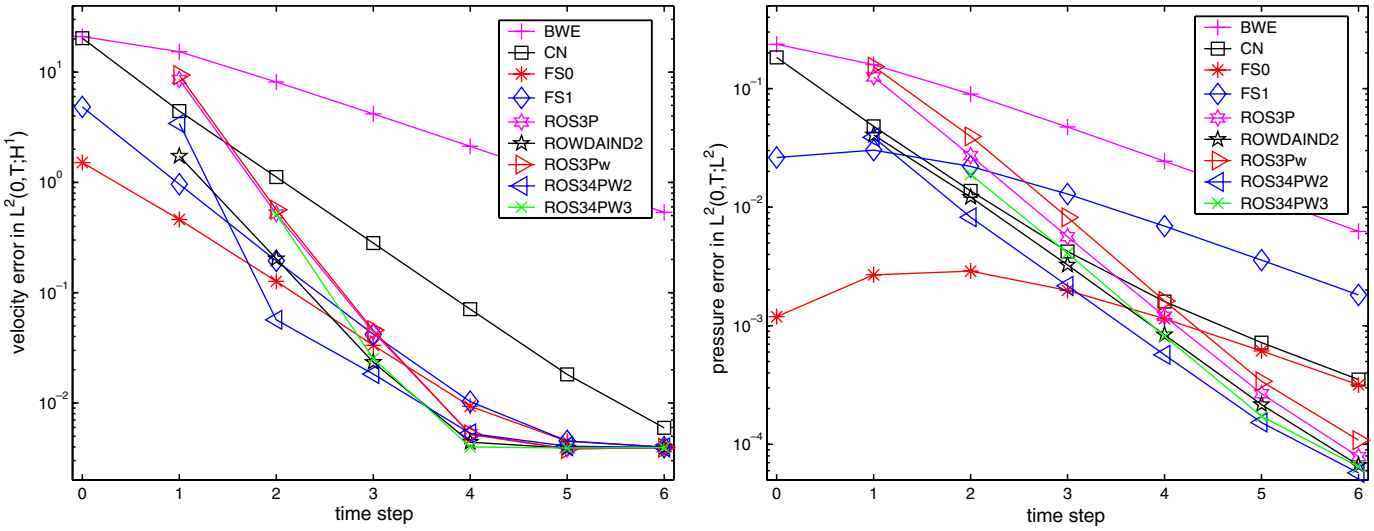

Fig. 6. Example 4.4, results for $Re = 10^6$.

ones. The velocity error with CN was always larger than the velocity error of all Rosenbrock methods. Bad results are obtained with BWE (with small step lengths). Similar observations can be made with respect to the pressure error in $L^2(0,T;L^2(\Omega))$.

The high Reynolds number case $Re = 10^6$. For $Re = 10^6$, we carried out computations with the time steps $\Delta t = 0.01 \times 2^{-k}$, $k = 0,\ldots,6$. The results are presented in Fig. 6. For the largest time step, $\Delta t = 0.01$, the Rosenbrock methods did not converge, ROS34PW3 even not for $\Delta t = 0.005$. Considering first the velocity error in $L^2(0,T;H^1(\Omega))$, one can see that FS0, FS1 and all Rosenbrock schemes are similar accurate for finer time steps ($k \geqslant 3$). These schemes reached the level of the discretization error in space for $k = 4$ (Rosenbrock schemes) or for $k = 5$ (FS0, FS1), respectively. The worst results were obtained with BWE. Also CN was in comparison to the other methods rather inaccurate. With respect to the pressure error in $L^2(0,T;L^2(\Omega))$, we obtained the best results for coarse time steps with FS0. For small time steps, all Rosenbrock schemes were more accurate than the θ-schemes. In particular, ROS34PW2 was for all time steps slightly more accurate than the other Rosenbrock methods. Again, FS0 computed more accurate results for the pressure than FS1. The most inaccurate results were obtained with BWE.

Summary of this example. For large time steps, FS0 gave the best results for both Reynolds numbers. It is still a good method for small time steps. The methods ROS34PW2 computed very accurate results for small time steps. Bad results, in particular in the high Reynolds number case, are obtained with BWE, CN (velocity) and FS1 (pressure).

4.5. The flow around a cylinder

The flow around a cylinder which will be considered was defined as a benchmark problem in [4] and studied numerically in detail in [5].

Fig. 7 presents the flow domain. The right-hand side of the Navier–Stokes equations (1) is $\mathbf{f} = \mathbf{0}$, the final time is $T = 8$ and the inflow and outflow boundary conditions are given by

$$\mathbf{u}(t;0,y) = \mathbf{u}(t;2.2,y) = 0.41^{-2}\sin(\pi t/8)(6y(0.41-y),0)\ \mathrm{m\,s^{-1}}, \quad 0 \leqslant y \leqslant 0.41.$$

On all other boundaries, the no-slip condition $\mathbf{u} = \mathbf{0}$ is prescribed. The Reynolds number of the flow, based on the mean inflow, the diameter of the cylinder and the prescribed viscosity $\nu = 10^{-3}\ \mathrm{m^2\,s^{-1}}$ is $0 \leqslant Re(t) \leqslant 100$.

The coarsest grid (level 0) is presented in Fig. 8. All computations have been carried out on level 3 of the spatial grid refinement resulting in 107,712 velocity d.o.f. and 39,936 pressure d.o.f. The time step was chosen to be $\Delta t = 0.01$.

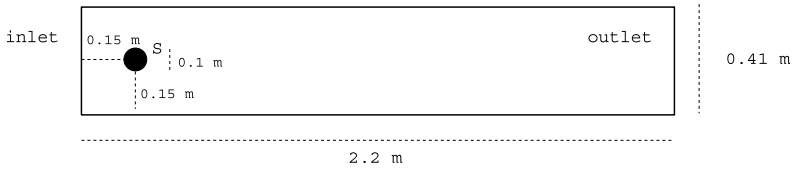

Fig. 7. Example 4.5, the channel with the cylinder.

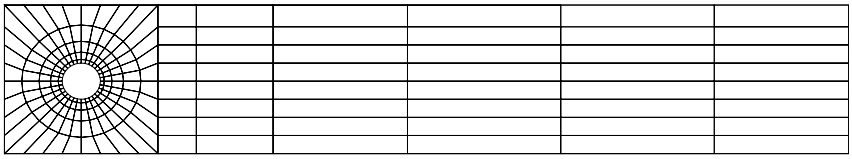

Fig. 8. Example 4.5, the coarsest grid (level 0).

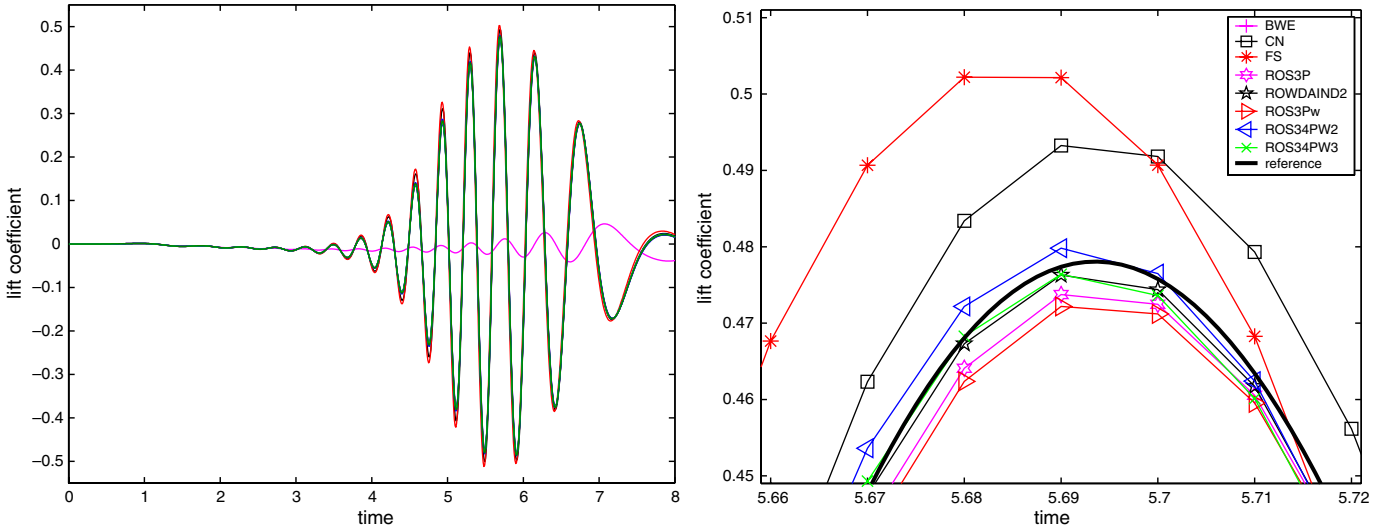

Fig. 9. Example 4.5, lift coefficient and zoom around the maximal lift value, the completely wrong curve (left picture) was produced by BWE.

The characteristic values of the flow are the drag coefficient $c_d(t)$ and the lift coefficient $c_l(t)$ at the cylinder. These coefficients can be computed by

$$c_d(t) = -20[(\mathbf{u}_t, \mathbf{v}_d) + (\nu\nabla\mathbf{u}, \nabla\mathbf{v}_d) + ((\mathbf{u}\cdot\nabla)\mathbf{u}, \mathbf{v}_d) - (p, \nabla\cdot\mathbf{v}_d)],$$
$$c_l(t) = -20[(\mathbf{u}_t, \mathbf{v}_l) + (\nu\nabla\mathbf{u}, \nabla\mathbf{v}_l) + ((\mathbf{u}\cdot\nabla)\mathbf{u}, \mathbf{v}_l) - (p, \nabla\cdot\mathbf{v}_l)]$$

for any function $\mathbf{v}_d \in (H^1(\Omega))^2$ with $(\mathbf{v}_d)|_S = (1,0)^T$ and $\mathbf{v}_d$ vanishes on all other boundaries and for any test function $\mathbf{v}_l \in (H^1(\Omega))^2$ with $(\mathbf{v}_l)|_S = (0,1)^T$ and $\mathbf{v}_l$ vanishes on all other boundaries, respectively. Note, there is a misprint in these formulas in [5] missing the term with $\mathbf{u}_t$. We will concentrate in the evaluation of the results on the lift coefficient because this coefficient is much more sensitive than the drag coefficient. Another benchmark value in [4] is the difference of the pressure between the front and the back at the cylinder at the final time $p(8;0.15,0.2) - p(8;0.25,0.2)$. Reference values for this difference and the maximal values of the drag and the lift coefficient are given in [5].

Since the right-hand side of the Navier–Stokes equations vanishes, the schemes FS0 and FS1 are identical for this problem. We mark the results obtained with these schemes by FS.

Fig. 9 shows the lift coefficient as functions of time. In both graphs, also the reference curve from [5] is given. We see that BWE produced the most inaccurate results. This is the only method which is, for this length of the time step, unable to generate the correct oscillations in the lift coefficient. From the zoom of lift coefficient curve (right picture in Fig. 9) it becomes obvious that all Rosenbrock methods produced results which differ only slightly and which are close to the reference curve. The best results were obtained by ROS34PW2, ROS34PW3 and ROWDAIND2.

Table 3
Pressure difference at $t = 8$, $\Delta p_{ref} = -0.1116$ from [5]

Methods	Δp	$\Delta p - \Delta p_{ref}$	$\left\lvert\frac{\Delta p - \Delta p_{ref}}{\Delta p_{ref}}\right\rvert \times 100\%$
BWE	−1.17553e−01	−5.9531e−03	5.33e+00
CN	−1.10304e−01	1.2957e−03	1.16e+00
FS	−1.10170e−01	1.4301e−03	1.28e+00
ROS3P	−1.11683e−01	−8.3245e−05	7.46e−02
ROWDAIND2	−1.11750e−01	−1.4972e−04	1.34e−01
ROS3Pw	−1.11653e−01	−5.2525e−05	4.71e−02
ROS34PW2	−1.11570e−01	3.0263e−05	2.71e−02
ROS34PW3	−1.11572e−01	2.7619e−05	2.47e−02

In Table 3, the pressure difference at time $t = 8$ is given. We present the value itself, its deviation from the reference value given in [5] and the relative error. The best results were obtained with ROS34PW2 and ROW34PW3. All other Rosenbrock methods produced quite accurate results which are much better than the results obtained with the implicit θ-schemes. Also for this value, the result from BWE is the most inaccurate one. One reason is probably the damping which is introduced by BWE.

4.6. Some general remarks to the computational tests

Besides the accuracy of the results, the computing times are a second important measure of the efficiency of a method.

We have presented computations on fixed grids in time and space. As mentioned above, an s-stage Rosenbrock method requires in each discrete time the solution of s saddle point problems. Example 4.5 shows that for accurate results, these saddle point problems have to be solved accurately. In contrast, for the θ-schemes, the number of saddle point problems which must be solved in each discrete time is not known a priori and this number can be different for different discrete times. This number depends on the difficulty of the given problem, the length of the time step and the stopping criterion for the fixed point iteration.

In the examples presented in this paper, we could generally observe that BWE and CN were the fastest schemes. FS0 and FS1 needed roughly twice as much computing time. A mutual comparison in the class of these θ-schemes shows that the ratio of error and computing time is often similar for CN and FS0. The Rosenbrock methods needed in general 3–4 times the computing times of BWE and CN.

A very important tool for increasing the efficiency of time stepping schemes is an adaptive step length control. For the θ-schemes, Turek [18] proposes to use extrapolation techniques. However, he states that these techniques might be rather expensive since additional time steps with the θ-schemes have to be computed. For Rosenbrock methods, the use of embedded methods for the step length control is possible, which is a very efficient approach. Thus, the Rosenbrock methods have a good potential of increasing their efficiency if this type of step length control is applied.

A comprehensive study of the costs of the presented methods, including a step length control, for achieving a given accuracy will be subject to a forthcoming study.

5. Summary

The paper presented a numerical study of two approaches for the temporal discretization and the linearization of the incompressible Navier–Stokes equations. One approach consists in using implicit θ-schemes which are combined with a fixed point iteration in each discrete time. The second approach uses Rosenbrock methods with s stages which require the solution of s linear systems in each discrete time. The numerical tests were carried out at two-dimensional examples on a fixed mesh in time and space.

An evaluation of the numerical results is presented in Table 4. Among the implicit θ-schemes, the fractional-step θ-scheme type 0 (FS0) gave clearly the most accurate results. For the Crank–Nicolson scheme (CN), often the ratio of the error and the computing time is comparable. If the computation of the right-hand side is not extraordinarily expensive, FS0 should be preferred to FS1. In several examples, some θ-schemes were more accurate than the Rosenbrock methods for large time steps. But in general, in particular for small time steps, all Rosenbrock methods computed more accurate results on the same grid in time and space (at the expense of larger computing times than the θ-schemes). There is no clear ranking among the Rosenbrock methods. For the considered examples, ROS34PW2 was always among the best performing ones.

Table 4
Evaluation of the computational studies: (++) best, (+) good schemes, (∘) satisfactory, (−) bad, (− −) very bad

Methods	Example 4.1		Example 4.2	Example 4.3	Example 4.4		Example 4.5
	$Re = 1$	$Re = 1000$			$Re = 1$	$Re = 10^6$	
BWE	−	− −	− −	− −	− −	− −	− −
CN	−	− −	+	∘	∘	−	+
FS0	∘	−	++	+	++	+	+
FS1	− −	−	+	−	++	−	+
ROS3P	++	++	∘	++	∘	+	++
ROWDAIND2	++	++	∘	+	+	++	++
ROS3Pw	++	++	∘	++	∘	+	++
ROS34PW2	++	++	∘	++	+	++	++
ROS34PW3	++	++	∘	++	∘	+	++

The best performing θ-schemes (CN, FS0) and the Rosenbrock methods will be studied further. One point of interest is to consider not only their accuracy but also their computational costs, in particular, when adaptive time-step control is used. A second forthcoming study will explore the behavior of these methods for the three-dimensional incompressible Navier–Stokes equations.

References

[1] P. Knobloch, Solvability and finite element discretization of a mathematical model related to Czochralski crystal growth, Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, 1996.
[2] U. Krewer, Y. Song, K. Sundmacher, V. John, R. Lübke, G. Matthies, L. Tobiska, Direct methanol fuel cell (DMFC): Analysis of residence time behaviour of anodic flow bed, Chem. Eng. Sci. 59 (2004) 119–130.
[3] P. Gresho, R. Sani, Incompressible Flow and the Finite Element Method, Wiley, Chichester, 2000.
[4] M. Schäfer, S. Turek, The benchmark problem "Flow around a cylinder", in: E. Hirschel (Ed.), Flow Simulation with High-Performance Computers II, Notes on Numerical Fluid Mechanics, vol. 52, Vieweg, 1996, pp. 547–566.
[5] V. John, Reference values for drag and lift of a two-dimensional time dependent flow around a cylinder, Int. J. Numer. Methods Fluids 44 (2004) 777–788.
[6] V. John, G. Matthies, Higher order finite element discretizations in a benchmark problem for incompressible flows, Int. J. Numer. Methods Fluids 37 (2001) 885–903.
[7] V. John, Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations, Int. J. Numer. Methods Fluids 40 (2002) 775–798.
[8] V. John, On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3d flow around a cylinder, Int. J. Numer. Methods Fluids, in press.
[9] W. Dettmer, D. Perić, An analysis of the time integration algorithms for the finite element solutions of incompressible Navier–Stokes equations based on a stabilised formulation, Comput. Methods Appl. Mech. Engrg. 192 (2003) 1177–1226.
[10] M. Bristeau, R. Glowinski, J. Periaux, Numerical methods for the Navier–Stokes equations: applications to the simulation of compressible and incompressible viscous flows, Comput. Phys. Rep. 6 (1987) 73–187.
[11] R. Glowinski, Finite element methods for incompressible viscous flow, in: P.G. Ciarlet et al. (Eds.), Numerical Methods for Fluids (Part 3), Handbook of Numerical Analysis, vol. 9, North-Holland, Amsterdam, 2003, pp. 3–1176.
[12] S. Müller-Urbaniak, Eine Analyse des Zwischenschritt-θ-Verfahrens zur Lösung der instationären Navier–Stokes–Gleichungen, Preprint 94-01, Universität Heidelberg, Interdisziplinäres Zentrum für wissenschaftliches Rechnen, 1994.
[13] E. Emmrich, Analysis von Zeitdiskretisierungen des inkompressiblen Navier–Stokes-Problems, Ph.D. thesis, Technische Universität Berlin, appeared also as book from Cuvillier-Verlag Göttingen, 2001.
[14] R. Temam, Navier–Stokes equations, Theory and Numerical Analysis, Studies in Mathematics and Its Applications, vol. 2, North-Holland Publishing Company, Amsterdam, 1977.
[15] J. Heywood, R. Rannacher, Finite element approximation of the nonstationary Navier–Stokes problem IV: error analysis for second order time discretizations, SIAM J. Numer. Anal. 27 (1990) 353–384.
[16] M. Bause, Optimale Konvergenzraten für voll diskretisierte Navier–Stokes-Approximationen höherer Ordnung in Gebieten mit Lipischitz-Rand, Ph.D. thesis, Universität-Gesamthochschule Paderborn, 1997.
[17] P. Klouček, F. Rys, Stability of the fractional step θ-scheme for the nonstationary Navier–Stokes equations, SIAM Numer. Anal. 31 (1994) 1312–1335.
[18] S. Turek, Efficient solvers for incompressible flow problems: an algorithmic and computational approachLecture Notes in Computational Science and Engineering, vol. 6, Springer, Berlin, 1999.
[19] A. Ostermann, Über die Wahl geeigneter Approximationen an die Jacobimatrix bei linear-impliziten Runge–Kutta Verfahren., Ph.D. thesis, Universität Innsbruck, Innsbruck, 1988.
[20] E. Hairer, G. Wanner, Solving ordinary differential equations II: stiff and differential-algebraic problems, second ed., Springer Series in Computational Mathematics, vol. 14, Springer-Verlag, Berlin, 1996.
[21] K. Strehmel, R. Weiner, Linear-implizite Runge–Kutta-Methoden und ihre Anwendung, Teubner-Texte zur Mathematik, vol. 127, Teubner, Stuttgart, 1992.
[22] K. Brenan, S. Campbell, L. Petzold, Numerical solution of initial-value problems in differential-algebraic equations, Classics in Applied Mathematics, vol. 14, SIAM, Philadelphia, 1996.
[23] J. Rang, Stability estimates and numerical methods for degenerate parabolic differential equations, Ph.D. thesis, Technische Universität Clausthal, appeared also as book from Papierflieger-Verlag Clausthal, 2005 (2004).
[24] J. Lang, J. Verwer, ROS3P—an accurate third-order Rosenbrock Solver designed for parabolic problems, BIT 41 (2001) 730–737.
[25] J. Lang, Adaptive multilevel solution of nonlinear parabolic PDE systems, Lecture Notes in Computational Science and Engineering, vol. 16, Springer-Verlag, Berlin, 2001.
[26] C. Lubich, M. Roche, Rosenbrock methods for differential-algebraic systems with solution-dependent singular matrix multiplying the derivative, Computing 43 (1990) 325–342.
[27] J. Rang, L. Angermann, New Rosenbrock methods of order 3 for PDAEs of Index 1, BIT, (2006), to appear.
[28] G. Matthies, L. Tobiska, The inf–sup condition for the mapped $Q_k/P_{k-1}^{\mathrm{disc}}$ element in arbitrary space dimensions, Computing 69 (2002) 119–139.
[29] Y. Saad, A flexible inner–outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (1993) 461–469.
[30] V. John, Large Eddy simulation of turbulent incompressible flows, Analytical and Numerical Results for a Class of LES Models, Lecture Notes in Computational Science and Engineering, vol. 34, Springer-Verlag, Berlin, Heidelberg, New York, 2004.
[31] V. John, G. Matthies, MooNMD—program package based on mapped finite element methods, Comput. Visual. Sci. 6 (2004) 163–170.
[32] A. Chorin, Numerical solution for the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.