Comput. Methods Appl. Mech. Engrg. 197 (2008) 4512–4520

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A discrete projection method for incompressible viscous flow with coriolis force ☆

Andriy Sokolov [a,*], Maxim A. Olshanskii [b], Stefan Turek [a]

[a] Institut für Angewandte Mathematik, Dortmund University of Technology, Vogelpothsweg 87, 44227 Dortmund, NRW, Germany
[b] Department of Mechanics and Mathematics, Moscow State University, 119899 Moscow, Russia

ARTICLE INFO

Article history:
Received 26 October 2007
Received in revised form 18 April 2008
Accepted 20 May 2008
Available online 29 May 2008

Keywords:
Navier–Stokes equations
Coriolis force
Projection method
Pressure Schur complement
Multigrid

ABSTRACT

The paper presents a new discrete projection method for the numerical solution of the Navier–Stokes equations with Coriolis force term. On an algebraic level we interpret one time step of the projection method as an incomplete factorization of the linearized Navier–Stokes system and as the iteration of an Uzawa type algorithm with special preconditioning for the pressure block. This enables us to modify the well-known projection method in a way to account for possibly dominating Coriolis terms. We consider a special multigrid method for solving the velocity subproblems and a modified projection (pressure correction) step. Results of numerical tests are presented for a model problem as well as for 3D flow simulations in stirred tank reactors.

1. Introduction

In many physical and industrial applications there is the necessity of numerical simulations for CFD models with moving geometries. As an example, let us consider the numerical simulation of a Stirred Tank Reactor (STR) benchmark problem (Fig. 1.1). The fluid motion is modeled by the nonstationary incompressible Navier–Stokes equations

$$\boldsymbol{v}_t + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} - \nu\Delta\boldsymbol{v} + \nabla p = \boldsymbol{f}, \quad \nabla\cdot\boldsymbol{v} = 0 \quad \text{in } \Omega\times(0,T] \tag{1.1}$$

for given force $\boldsymbol{f}$ and kinematic viscosity $\nu > 0$. We also assume that some boundary values and an initial condition are prescribed. For constructing a mesh and performing numerical simulations we make the following simplifications:

- The propeller of the stirred tank reactor rotates around the Z-axis with constant angular velocity $\boldsymbol{\omega} = (0,0,\omega)^T$.
- We do not have any blades attached to the outside wall, i.e. the tank possesses a simple cylindrical geometry.
- The tank is filled with homogeneous liquid.

One way to treat the moving boundary parts of the propeller is to apply, say, a fictitious boundary method for modeling the moving parts [20,22]. Another approach for the simulation of the flow in a stirred tank reactor is the following one. We change the inertial frame of reference to the noninertial frame, rotating with the blades. Performing coordinate transformation we consider a new velocity $\boldsymbol{u} = \boldsymbol{v} + (\boldsymbol{\omega}\times\boldsymbol{r})$, where $\boldsymbol{\omega}$ is the angular velocity vector and $\boldsymbol{r}$ is the radius vector from the center of coordinates. The velocity $\boldsymbol{u}$ satisfies homogeneous Dirichlet boundary values on the blades of the propeller, while on the outside wall of the tank one obtains $\boldsymbol{u} = \boldsymbol{\omega}\times\boldsymbol{r}$. Thus, in the new reference frame the system (1.1) can be rewritten as

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}) + \nabla p &= \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} &= 0 \end{aligned} \quad \text{in } \Omega\times(0,T], \tag{1.2}$$

where $2\boldsymbol{\omega}\times\boldsymbol{u}$ and $\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r})$ are the so-called Coriolis and centrifugal forces, respectively. For a more detailed derivation of (1.2) see, e.g., [1] or [16]. Using the equality

$$\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}) = -\nabla\frac{1}{2}(\boldsymbol{\omega}\times\boldsymbol{r})^2$$

and setting $P = p - \frac{1}{2}(\boldsymbol{\omega}\times\boldsymbol{r})^2$ in (1.2), we get the following system of equations which will be treated in this paper:

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla P &= \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} &= 0 \end{aligned} \quad \text{in } \Omega\times(0,T]. \tag{1.3}$$

As a first step let us consider the semi-implicit second order time discretization: given $\boldsymbol{u}^n$ and the time step $\Delta t = t_{n+1} - t_n$, find $\boldsymbol{u}^{n+1}$ and p^{n+1} (for the convenience we denote $p = P$ in (1.3)) satisfying

☆ This research was supported by the German Research Foundation and the Russian Foundation for Basic Research through the Grant DFG-RFBR 06-01-04000/08-01-91957 and TU 102/21-1.

* Corresponding author. Tel.: +49 0231 755 3171; fax: +49 0231 755 5933.
E-mail addresses: asokolow@math.uni-dortmund.de, and_sokolovcamp@yahoo.com (A. Sokolov), maxim.olshanskii@mtu-net.ru (M.A. Olshanskii), ture@featflow.de (S. Turek).

0045-7825/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2008.05.014

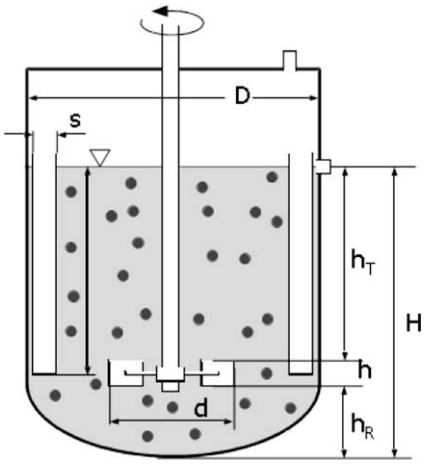

Fig. 1.1. STR geometry.

$$\begin{aligned}&\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\frac{1}{2}((\boldsymbol{u}^{*}\cdot\nabla)\boldsymbol{u}^{n+1}-\nu\Delta\boldsymbol{u}^{n+1}+2\boldsymbol{\omega}\times\boldsymbol{u}^{n+1})+\nabla p^{n+1}=\boldsymbol{g}^{n+1}\\&\nabla\cdot\boldsymbol{u}=0\end{aligned}\tag{1.4}$$

with the right-hand side

$$\boldsymbol{g}^{n+1}=\frac{1}{2}(\boldsymbol{f}^{\,n+1}+\boldsymbol{f}^{\,n})-\frac{1}{2}((\boldsymbol{u}^{*}\cdot\nabla)\boldsymbol{u}^{n}-\nu\Delta\boldsymbol{u}^{n}+2\boldsymbol{\omega}\times\boldsymbol{u}^{n}),\tag{1.5}$$

$\boldsymbol{u}^{*}$ denotes a second order extrapolation of velocity from n and $n-1$ time steps. Alternatively one may consider a fully implicit scheme by setting $\boldsymbol{u}^{*}=\boldsymbol{u}^{n+1}$. Discretization of (1.4), (1.5) in space leads to a saddle-point system to be solved in every time step. The system has the form

$$\begin{pmatrix}\boldsymbol{S} & B\\ B^{T} & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{u}\\ p\end{pmatrix}=\begin{pmatrix}\boldsymbol{g}\\ 0\end{pmatrix},\tag{1.6}$$

where $\boldsymbol{u}=(u_1,u_2,u_3)^{T}$ is the discrete velocity, p the discrete pressure; B and B^{T} are discrete gradient and divergence operators and $\boldsymbol{S}$ is a block matrix which is due to the discretized velocity operators in the momentum equation. The implicit scheme (1.4) and (1.5) has excellent stability properties, see e.g. [19], however solving the coupled system (1.6) in every time step is rather expensive. To avoid this, some splitting procedures are often used in practice such as the Chorin–Temam projection method. It has been observed by a number of authors, see, e.g. [12,21], that on the discrete level the projection method can be interpreted as a particular incomplete factorization of the matrix from (1.6), which involves preconditioners for $\boldsymbol{S}$ and for the pressure Schur complement matrix, see Section 2. Given this algebraic framework we design and analyze in the present paper more effective factorizations for the system (1.6) and, based on this, build a new discrete projection method for the time integration of (1.3). In particular, we construct several new preconditioners for pressure Schur complement of the system (1.6) by building on the previous work in [8]. This development leads to a modified pressure Poisson problem in every time step of the projection scheme. The modified projection step takes into account the influence of the Coriolis force and improves the performance of the scheme. The STR problem will serve us as an example of applying this technique. As basic software for our simulations we use the PP3D module of the open-source CFD package Featflow [18].

2. Discrete projection method (DPM)

Let us start with a well-known second order variant of the Chorin–Temam projection method [13,2] applied to the problem (1.3). In its semi-discrete form it can be viewed as a two-step procedure for advancing from time step n to step $n+1$: with given $\boldsymbol{u}^{n}, p^{n}$ and $\boldsymbol{g}^{n+1}$ defined in (1.5):

Step 1: Find intermediate velocity $\tilde{\boldsymbol{u}}$ from

$$\frac{\tilde{\boldsymbol{u}}-\boldsymbol{u}^{n}}{\Delta t}+\frac{1}{2}((\boldsymbol{u}^{*}\cdot\nabla)\tilde{\boldsymbol{u}}-\nu\Delta\tilde{\boldsymbol{u}}+2\boldsymbol{\omega}\times\tilde{\boldsymbol{u}})=\boldsymbol{g}^{n+1}-\nabla p^{n}.\tag{2.1}$$

Step 2: Find new velocity and pressure as the result of the orthogonal projection into the divergence-free subspace

$$\begin{cases}\frac{\boldsymbol{u}^{n+1}-\tilde{\boldsymbol{u}}}{\Delta t}+\frac{1}{2}(\nabla p^{n+1}-\nabla p^{n}) &= 0\\ \nabla\cdot\boldsymbol{u}^{n+1} &= 0.\end{cases}\tag{2.2}$$

To motivate our modifications of the projection method, let us consider its algebraic counterpart. To this end, denote by $\boldsymbol{M}$ the velocity mass matrix and by I_u, I_p the identity matrices on discrete velocity and pressure spaces, respectively. It is easy to check that in the discrete setting the method (2.1), and (2.2) can be written in the following algebraic form:

$$\begin{pmatrix}\boldsymbol{S} & 0\\ B^{T} & -B^{T}(\frac{1}{\Delta t}\boldsymbol{M})^{-1}B\end{pmatrix}\begin{pmatrix}I_u & (\frac{1}{\Delta t}\boldsymbol{M})^{-1}B\\ 0 & I_p\end{pmatrix}\begin{pmatrix}\boldsymbol{u}^{n+1}\\ q\end{pmatrix}=\begin{pmatrix}\hat{\boldsymbol{g}}\\ 0\end{pmatrix}\tag{2.3}$$

with $\hat{\boldsymbol{g}}=\boldsymbol{g}^{n+1}+\frac{1}{\Delta t}\boldsymbol{u}^{n}-Bp^{n}$ and $q=\frac{1}{2}(p^{n+1}-p^{n})$. The matrix product on the left-hand side of (2.3) can be observed as the incomplete LU factorization for the matrix of the coupled linearized Navier–Stokes system (matrix from (1.6)). Indeed, it holds

$$\begin{pmatrix}\boldsymbol{S} & B\\ B^{T} & 0\end{pmatrix}=\begin{pmatrix}\boldsymbol{S} & 0\\ B^{T} & -B^{T}\boldsymbol{S}^{-1}B\end{pmatrix}\begin{pmatrix}I_u & \boldsymbol{S}^{-1}B\\ 0 & I_p\end{pmatrix}.\tag{2.4}$$

The velocity submatrix has the form

$$\boldsymbol{S}=\begin{pmatrix}A & -\omega M & 0\\ \omega M & A & 0\\ 0 & 0 & A\end{pmatrix},\tag{2.5}$$

where $A=(\Delta t)^{-1}M+\frac{1}{2}[N(\boldsymbol{u}^{*})+\nu L]$ is the velocity stiffness matrix, M is the mass matrix for a single velocity component, and the matrix operators $N(\boldsymbol{u}^{*})$ and L are the discrete analogues of $(\boldsymbol{u}\cdot\nabla)\cdot$ and $(-\Delta)\cdot$, respectively. Therefore, if the time step Δt is sufficiently small the scaled block diagonal mass-matrix $(\Delta t)^{-1}\boldsymbol{M}$ is a reasonable approximation to $\boldsymbol{S}$ and the incomplete factorization in (2.3) is close to the exact factorization (1.6). This shows that in some sense the projection method (2.1) and (2.2) approximates the coupled implicit method (1.4) for small Δt.

2.1. Modified projection method

From the above consideration one realizes that a better approximation to $\boldsymbol{S}^{-1}$, compared to $(\Delta t)\boldsymbol{M}^{-1}$, may lead to more effective (possibly less restrictive w.r.t. size of Δt) projection methods. Below, see (3.7) and (3.9), we consider an approximation $\boldsymbol{M}_{(\cdot)}^{-1}$ to $\boldsymbol{S}^{-1}$ which takes into account the Coriolis terms and to some extend the convection terms. Thus, we consider the system (2.3) with another velocity matrix approximation $\boldsymbol{M}_{(\cdot)}$ instead of $\frac{1}{\Delta t}\boldsymbol{M}$. In the algorithmic form one time step of the new discrete projection method reads (for $t_n\to t_{n+1}$):

1. Given $p^{n}\simeq p(t_n)$, $u^{n}\simeq u(t_n)$, and $\tilde{\boldsymbol{g}}=\boldsymbol{g}^{n+1}+\frac{1}{\Delta t}\boldsymbol{u}^{n}$, then solve for $\tilde{\boldsymbol{u}}$ the equation
$$\boldsymbol{S}\tilde{\boldsymbol{u}}=\tilde{\boldsymbol{g}}-Bp^{n}.\tag{2.6}$$
2. Solve the modified discrete pressure Poisson problem
$$Pq=B^{T}\tilde{\boldsymbol{u}}\quad\text{with}\quad P=B^{T}\boldsymbol{M}_{(\cdot)}^{-1}B.\tag{2.7}$$

3. Correct pressure and velocity

$$p^{n+1} = p^n + q, \tag{2.8}$$
$$\boldsymbol{u}^{n+1} = \tilde{\boldsymbol{u}} - \boldsymbol{M}_{(\cdot)}^{-1} B\, q. \tag{2.9}$$

Although we perform all our calculations with the discrete projection method (2.6)–(2.9), it is instructive to write down its semi-discrete counterpart. This is easy to do for the case of $\boldsymbol{M}_{(\cdot)}$ defined in (3.7). Now the procedure for advancing from time step n to step $n+1$ reads (compare to (2.1) and (2.2)): for given $\boldsymbol{u}^n, p^n$ and $\boldsymbol{g}^{n+1}$ defined in (1.5):

Step 1: Find intermediate velocity $\tilde{\boldsymbol{u}}$ from

$$\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\Delta t} + \frac{1}{2}((\boldsymbol{u}^* \cdot \nabla)\tilde{\boldsymbol{u}} - \nu\Delta\tilde{\boldsymbol{u}} + 2\boldsymbol{\omega} \times \tilde{\boldsymbol{u}}) = \boldsymbol{g}^{n+1} - \nabla p^n. \tag{2.10}$$

Step 2: Find new velocity and pressure as the result of the following projection into the divergence-free subspace

$$\begin{cases} \frac{\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}}{\Delta t} + \boldsymbol{\omega} \times (\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}) + \frac{1}{2}(\nabla p^{n+1} - \nabla p^n) &= 0 \\ \nabla \cdot \boldsymbol{u}^{n+1} &= 0. \end{cases} \tag{2.11}$$

Remark 2.1. For efficient calculations with the original projection method (2.1) and (2.2) or with the modified one, we need an efficient solver for the velocity subproblem with the matrix $\boldsymbol{S}$ as well as for the (modified) pressure problem with matrix P. In Section 3.3 we will show that the modified method leads to a *symmetric* pressure problem of the diffusion type.

Remark 2.2. If we compare the factorizations (2.3) and (2.4), it is easy to notice that the matrix $-B^T(\frac{1}{\Delta t}\boldsymbol{M})^{-1}B$, corresponding to the discrete pressure Poisson problem, can be considered as a preconditioner for the Schur complement matrix $-B^T\boldsymbol{S}^{-1}B$. Another way to realize this is the following, see, e.g., [17]. Eliminating $\tilde{\boldsymbol{u}}$ we can rewrite (2.6)–(2.8) as

$$p^{n+1} = p^n + P^{-1}(B^T\boldsymbol{S}^{-1}Bp^n - g)$$

with $g = B^T\boldsymbol{S}^{-1}\tilde{\boldsymbol{g}}$. Thus with respect to the pressure variable one step of the projection method can be seen as one iteration of the preconditioned Uzawa algorithm. This relates the efficiency of the projection methods with the issue of pressure Schur complement preconditioning, see also [3].

Remark 2.3. One possible variation of the projection method is to add a diffusion dependent term to the pressure correction step (2.8):

$$p^{n+1} = p^n + q + \nu M_p^{-1} B^T \tilde{\boldsymbol{u}}.$$

In [13] (for the case $\omega = 0$) it was discussed that adding such term may reduce numerical boundary layers in projection methods.

Remark 2.4. Observing (2.11) or the choice of $\boldsymbol{M}_{(\cdot)}^{-1}$ in (3.7) and (3.9) one notes that the modified projection step essentially takes into account the Coriolis terms and only indirectly (in (3.9)) the convection terms. Therefore the proposed modification of the method is especially suitable for the case of moderate Rossby numbers. See, however, the comments for the case of large Rossby numbers in Section 3.4.

In the next section we mainly address the following two issues:

- Building an efficient multigrid solver for the velocity subproblem (2.6).
- Finding an appropriate matrix $\boldsymbol{M}_{(\cdot)}$ involved in steps (2.7) and (2.9).

3. Algorithmic details of the DPM

3.1. Space discretization

For the space discretization we use the mixed finite element method (nonconforming Rannacher–Turek elements $\widetilde{Q}_1$ for velocity vector field $\boldsymbol{u}$ and piecewise constant elements Q_0 for pressure p, see Fig. 3.1). The analysis of these elements can be found in [14].

The spaces $\widetilde{Q}_1$ and Q_0 lead to numerically stable approximations as $h \to 0$, i.e. they satisfy the *Babuska–Brezzi condition* with a mesh independent constant γ:

$$\inf_{p_h \in Q_0} \sup_{\boldsymbol{u}_h \in \widetilde{Q}_1} \frac{(p_h, \nabla \cdot \boldsymbol{u}_h)}{\|p_h\|_0 \|\nabla_h \boldsymbol{u}_h\|_0} \geqslant \gamma > 0.$$

Further in this section we describe an efficient solver for the velocity subproblem (2.6) and we build a matrix $\boldsymbol{M}_{(\cdot)}^{-1}$ that approximates $\boldsymbol{S}^{-1}$. The latter is also related to a pressure Schur complement preconditioning or, in the terms of projection methods, to a new "pressure Poisson" equation.

As a first step we neglect convective terms and consider the DPM applied to the system of Stokes equations with the Coriolis force term:

$$\begin{aligned} \boldsymbol{u}_t - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega} \times \boldsymbol{u} + \nabla p &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \quad \text{in } \Omega \times (0, T]. \tag{3.1}$$

Both the discretized velocity subproblem and the scalar pressure equation will be solved by multigrid methods with special smoothers and coarse grid solvers to be explained below.

3.2. Velocity subproblem

Assuming a hierarchy of grids let us consider a multigrid method for solving Eq. (2.6). For smoothing iterations we take a linear iterative method of the form

$$\tilde{\boldsymbol{u}}^{l+1} = \tilde{\boldsymbol{u}}^l + \alpha\boldsymbol{C}^{-1}(\tilde{\boldsymbol{g}} - Bp^n - \boldsymbol{S}\tilde{\boldsymbol{u}}^l), \tag{3.2}$$

where α is a relaxation parameter and $\boldsymbol{C}$ is a suitable preconditioner of $\boldsymbol{S}$. We are interested in an efficient smoother for the case of large values of the Coriolis force term, i.e. when the off-diagonal parts in the matrix (2.5) have values equal or larger than those of the diagonal part. Note that in this case the skew-symmetric part of $\boldsymbol{S}$ is dominant. Thus standard smoothing iterations like Jacobi or Gauss–Seidel may not lead to a robust multigrid solver.

The off diagonal values in (2.5) should be taken into account in $\boldsymbol{C}$, if the Coriolis terms are large enough. Following [10], we put

$$\boldsymbol{C} := \boldsymbol{C}_{\text{coriol}} = \begin{pmatrix} \operatorname{diag}(A) & -\omega M_L & 0 \\ \omega M_L & \operatorname{diag}(A) & 0 \\ 0 & 0 & \operatorname{diag}(A) \end{pmatrix}, \tag{3.3}$$

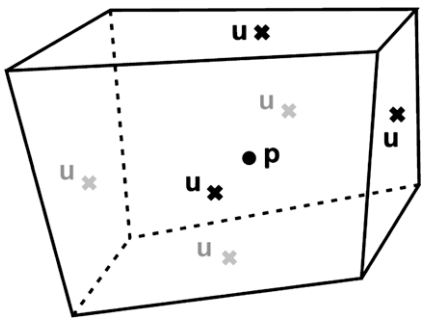

Fig. 3.1. Nodal points of the nonconforming finite element in 3D.

where M_L is the lumped mass matrix. The lumped mass matrix is a diagonal matrix with diagonal elements defined as $m_i = \sum_j m_{ij}$, where m_{ij} are the entries of M. M_L is often taken as an approximation for the consistent mass matrix. For the two-dimensional velocity problem discretized by a conforming finite element method on a regular grid it was proved in [10] that a standard geometric multigrid method with such smoothing is robust with respect to all relevant problem parameters. We will see that the multigrid method stays very efficient in more practical settings, too.

Taking into account the fact that all blocks of $\boldsymbol{C}_{\text{coriol}}$ are diagonal matrices, one can explicitly find its inverse $\boldsymbol{C}^{-1}_{\text{coriol}}$ [15]. In Section 4.1 we will present results of numerical experiments with the multigrid method using different smoothers. We will see that iterations (3.2) with the preconditioner $\boldsymbol{C}_{\text{coriol}}$ outperform such standard smoothers as Jacobi or SOR methods.

3.3. Modified pressure equation

The choice of the matrix $\boldsymbol{M}^{-1}_{(\cdot)}$ in (3.8) and (2.9) defines the discrete projection method. In particular this choice defines the matrix $P = B^{\mathrm{T}}\boldsymbol{M}^{-1}_{(\cdot)}B$ of the discrete pressure equation. Matrix P can be considered as a preconditioner for the pressure Schur complement of the original system (1.6), cf. Remark 2.2. Only few results can be found in the literature related to the preconditioning of the pressure Schur complement operator for fluid equations with Coriolis terms, see for instance [7–9]. Here we follow the approaches given in [8,9] and [19] to construct a preconditioner for the discrete counterpart of the Schur operator:

$$P_{\text{cont}} = -\nabla\cdot\left(\frac{1}{\Delta t}I - \nu\Delta + \boldsymbol{w}\times\right)^{-1}\nabla. \tag{3.4}$$

To this end, let us consider the influences of mass, Coriolis and diffusion parts in (3.4) separately. From $A = (\Delta t)^{-1}M + \nu L$ we get that if the time step or the kinematic viscosity is small enough, then we can assume that $A \approx (\Delta t)^{-1}M$ and therefore $P^{-1} = (\Delta t)^{-1}(B^{\mathrm{T}}\boldsymbol{M}^{-1}B)^{-1}$. If the time step or the kinematic viscosity is sufficiently large, then we assume that $A \approx \nu L$ and hence $P^{-1} = \nu M_p^{-1}$, where M_p is the pressure mass matrix. Then, as preconditioner for the general Stokes case, we can define the matrix P^{-1} as linear interpolation of the above extreme cases, namely

$$P^{-1} = (\Delta t)^{-1}(B^{\mathrm{T}}\boldsymbol{M}^{-1}B)^{-1} + \nu M_p^{-1}. \tag{3.5}$$

When the time step is small, the diffusion-oriented part of the preconditioner νM_p^{-1} is often neglected, leading to the standard projection step as in (2.3). In the case of the Coriolis force term involved, we use instead of $P = \Delta t(B^{\mathrm{T}}\boldsymbol{M}^{-1}B)$ the modified preconditioner

$$P_{\text{mass+coriol}} = B^{\mathrm{T}}\boldsymbol{M}^{-1}_{(\text{mass+coriol})}B \tag{3.6}$$

by choosing the 'Coriolis-oriented' mass matrix

$$\boldsymbol{M}_{(\text{mass+coriol})} = \begin{pmatrix} \frac{1}{\Delta t}M_L & -\omega M_L & 0 \\ \omega M_L & \frac{1}{\Delta t}M_L & 0 \\ 0 & 0 & \frac{1}{\Delta t}M_L \end{pmatrix}. \tag{3.7}$$

Here, the off-diagonal parts represent the contribution of the $\boldsymbol{w}\times$ operator. Thus, the modified pressure Poisson equation reads

$$P_{\text{mass+coriol}}\ q = B^{\mathrm{T}}\boldsymbol{M}^{-1}_{(\text{mass+coriol})}B\ q = B^{\mathrm{T}}\tilde{\boldsymbol{u}}. \tag{3.8}$$

We will see that (3.8) can be interpreted as the discrete counterpart of a modified pressure Poisson problem with *symmetric* diffusion tensor.

To take into account the influence of the viscous terms, the matrix νM_p^{-1} can be also included in the pressure correction step, cf. Remark 2.3. Alternatively one can include the diagonal part of $\boldsymbol{S}$ into the pressure diffusion operator. Namely, one can consider in (3.8)

$$\boldsymbol{M}_{(\text{diag+coriol})} = \begin{pmatrix} \text{diag}(A) & -\omega M_L & 0 \\ \omega M_L & \text{diag}(A) & 0 \\ 0 & 0 & \text{diag}(A) \end{pmatrix}. \tag{3.9}$$

Below we discuss some important details of the modified projection step. First, note that the matrix $P_{\text{mass+coriol}}$ in (3.8) can be seen as a discretization of the following differential operator (see [8] p. 365 for more details):

$$\mathscr{L} = -\nabla\cdot\mathscr{M}^{-1}\nabla \qquad \text{with} \quad \mathscr{M} = [(\Delta t)^{-1}I + \boldsymbol{w}\times], \quad \boldsymbol{w} = (0,0,\omega)^{\mathrm{T}}.$$

One finds

$$\mathscr{M}^{-1} = \Delta t\ (1+|\tilde{\boldsymbol{w}}|^2)^{-1}[I + \tilde{\boldsymbol{w}}\otimes\tilde{\boldsymbol{w}} - \tilde{\boldsymbol{w}}\times], \qquad \tilde{\boldsymbol{w}} = \Delta t\boldsymbol{w}$$

where $(\tilde{\boldsymbol{w}}\otimes\tilde{\boldsymbol{w}})_{ij} = \tilde{\boldsymbol{w}}_i\tilde{\boldsymbol{w}}_j$. Since $\tilde{\boldsymbol{w}}$ is a constant vector one has $\tilde{\boldsymbol{w}}\times\nabla q = \nabla\times q\tilde{\boldsymbol{w}}$ for a scalar function q. Since $\nabla\cdot(\nabla\times) \equiv 0$, one gets $\nabla\cdot(\tilde{\boldsymbol{w}}\times\nabla q) = 0$. Therefore in the differential notations Eq. (3.8) can be written as

$$-(1+|\tilde{\boldsymbol{w}}|^2)^{-1}\nabla\cdot[I + \tilde{\boldsymbol{w}}\otimes\tilde{\boldsymbol{w}}]\nabla q = -(\Delta t)^{-1}\nabla\cdot\tilde{\boldsymbol{u}}.$$

Note that although the operator $\mathscr{M}$ is non-symmetric the resulting diffusion type problem for the pressure update q is symmetric. The important property of symmetry-preserving on the discrete level is verified in the following lemma.

Lemma 3.1. *For the discretization with the nonconforming Stokes finite element $\tilde{Q}_1/Q_0$ the matrix $P = B^T\boldsymbol{M}^{-1}_{(\text{mass+coriol})}B$ is symmetric.*

Proof. Let N_p be the number of elements and N_u the number of internal faces. Denote $P = \{p_{ij}\}$, $i,j = 1,\dots,N_p$,

$$M_L = \{m_{ii}\}, \quad B = (B_1,B_2,B_3)^T \quad \text{with} \quad B_{\mathscr{K}} = \{b_{ij}^{\mathscr{K}}\},$$
$$i = 1,\dots,N_u, \quad j = 1,\dots,N_p. \tag{3.10}$$

Using (3.7) and notation (3.10) we compute

$$p_{ij} = \Delta t\sum_{k=1}^{N_u} \times\left(\frac{b^1_{ki}b^1_{kj}}{m_{kk}(1+s^2)} + \frac{b^1_{ki}b^2_{kj}s}{m_{kk}(1+s^2)} - \frac{b^2_{ki}b^1_{kj}s}{m_{kk}(1+s^2)} + \frac{b^2_{ki}b^2_{kj}}{m_{kk}(1+s^2)} + \frac{b^3_{ki}b^3_{kj}}{m_{kk}}\right)$$

with $s = \Delta t\omega$. It is sufficient to show that

$$b^1_{ki}b^2_{kj} - b^2_{ki}b^1_{kj} = 0, \qquad \forall i,j = 1,\dots,N_p, \quad k = 1,\dots,N_u. \tag{3.11}$$

By the definition of B we have

$$b^k_{ij} = \int_{T_j}\nabla\cdot\phi^k_i\psi_j\,\mathrm{d}x = -\int_{T_j}\phi^k_i\nabla\psi_j\,\mathrm{d}x + \int_{\partial T_j}\phi^k_i\cdot\boldsymbol{n}_j\psi_j\,\mathrm{d}\sigma$$
$$= \int_{\partial T_j}\phi^k_i\cdot\boldsymbol{n}_j\psi_j\,\mathrm{d}\sigma, \tag{3.12}$$

where $\psi_j \in Q_0$, $\phi^k_i \in \tilde{Q}_1$ are basis functions for the pressure and the kth velocity component that corresponds to the jth element and ith face, $\boldsymbol{n}_j$ is a unit outward normal for T_j. Denote by $\boldsymbol{n}_{ij} = (n^1_{ij}, n^2_{ij}, n^3_{ij})^T$ a unit outward normal to the ith face of the jth element. Then (3.12) implies $\boldsymbol{n}_{ki} = -\boldsymbol{n}_{kj}$. Thus (3.11) follows. □

Remark 3.2. The proposition is true for any $P = B^{\mathrm{T}}\boldsymbol{A}^{-1}B$, where $\boldsymbol{A}$ takes the form of (3.7) or (3.9). In particular it is valid for $A = \boldsymbol{M}_{(\text{diag+coriol})}$ from (3.9).

Remark 3.3. If the angular velocity ω increases, then the preconditioning matrix (3.6) becomes close to the degenerate case of a tridiagonal matrix (see [15]). The situation is somewhat less critical

for the preconditioner based on (3.9) thanks to the contribution of the discrete stabilized convective term in the diag(A) entries.

3.4. Treatment of the convective term

In the previous section we considered the system of Stokes equations with the Coriolis force term. However, performing numerical calculations for medium and high Reynolds number flows, one has to take into account the convective term as well. To prevent numerical oscillations we used the algebraic flux correction scheme of TVD type [6] for the discretization of convection terms. Moreover, adding such stabilization makes multigrid solver for the velocity subproblem more effective. Another relevant question is how to include the terms due to convection in the projection step. As we have seen, cf. Remark 2.2, this issue can be related to the question of building effective pressure Schur complement preconditioners for the case of dominating convection terms. This though question attracted a lot of considerations during the last decade, see an overview in [4,11]. However, we are not aware of any successful attempt to adopt these recently suggested preconditioners in a projection type scheme. The presence of the Coriolis force, makes the question even more difficult to address. Hence the modifications proposed in this paper are expected to improve performance of the projection scheme mostly for the case of moderate Rossby numbers.

A promising approach for the case of large Rossby numbers is the following. Using the well-known inequality

$$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = (\nabla\times\boldsymbol{u})\times\boldsymbol{u} + \nabla\left(\frac{\boldsymbol{u}^2}{2}\right)$$

and introducing a new pressure variable (Bernoulli pressure), we can replace the convective operator by the cross product one:

$$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla p = \boldsymbol{w}(\boldsymbol{u})\times\boldsymbol{u} + \nabla P \tag{3.13}$$

with $\boldsymbol{w}(\boldsymbol{u}) = \nabla\times\boldsymbol{u} + 2\boldsymbol{\omega}$ and $P = p + \frac{\boldsymbol{u}^2}{2}$. Thus, the Coriolis force term and the convective operator can be handled on the second step of the projection method simultaneously in the same way as described above in this paper, see [8] for the analysis of similar approach in the context of the Schur complement preconditioners for the linearized Navier–Stokes problem.

Thus, in the 'rotating' system of the Navier–Stokes equations we can treat convection and rotating forces either as the right or the left part of (3.13). While both treatments are equal on the continuous level, they may lead to discrete systems with *quite different* properties. In particular, many reliable methods for the stabilization of convection dominated flows have been developed by the CFD community. Among them are streamline-diffusion and upwinding schemes, edge-oriented stabilization, algebraic flux correction, etc. At the same time, not so much is known about stabilization techniques available for the term $(\nabla\times\boldsymbol{u})\times\boldsymbol{u}$. The approach based on the rotation form of convection term from (3.13) is a topic of our current research and will be addressed in more detail elsewhere.

4. Numerical experiments

In this chapter we analyze the numerical properties of the suggested algorithms for the system of the Stokes and Navier–Stokes equations with the Coriolis force term. We will compare the preconditioners, evaluate convergence rates, examine stabilization techniques and present numerical results for a model problem posed in the unit cube. In every case we assume that the Coriolis term corresponds to a rotation around the Z-axis. The unit cube geometry $[-1,1]\times[-1,1]\times[-1,1]$ was taken as the simplest configuration to test the algorithm. In all experiments we set $\nu = 1$, $\Delta t = 10^{-3}$, the value of ω may vary. For a discretization we consider a uniform Cartesian mesh. In the geometric multigrid solver we use several grid levels. In Table 4.1 we adopt the following notation: NEL is the number of elements, NAT is the number of faces, NVT and NEQ are the number of vertices and the total number of unknowns on different grid levels.

Table 4.1
Mesh characteristics of a unit cube with equidistant meshing

Level	NEL	NAT	NVT	NEQ
1st level	8	36	27	116
2nd level	64	125	240	439
3rd level	512	1,728	729	5,696
4th level	4,096	13,056	4,913	43,264
5th level	32,768	101,376	35,973	336,896

4.1. Multigrid method for velocity problems

Step 1 of the projection method involves a solution of the velocity subproblem with matrix $\boldsymbol{S}$ given in (2.5). Here we test a geometric multigrid method (V-cycle) with smoothing iterations defined in Section 3.2. We compare it with the multigrid involving more standard pointwise SOR type smoothing iterations. This smoothing iterations can be defined as (3.2) with

$$\boldsymbol{C} := \boldsymbol{C}_{\mathrm{SOR}} = \begin{pmatrix} \mathrm{lower\_part}(A) & 0 & 0 \\ 0 & \mathrm{lower\_part}(A) & 0 \\ 0 & 0 & \mathrm{lower\_part}(A) \end{pmatrix} \quad \text{or}$$

$$\boldsymbol{C} := \boldsymbol{C}_{\mathrm{SORcoriol}} = \begin{pmatrix} \mathrm{lower\_part}(A) & 0 & 0 \\ \omega M_L & \mathrm{lower\_part}(A) & 0 \\ 0 & 0 & \mathrm{lower\_part}(A) \end{pmatrix}$$

Both $\boldsymbol{C}_{\mathrm{SORcoriol}}$ and $\boldsymbol{C}_{\mathrm{coriol}}$ matrices take into account convective and Coriolis force terms. However, only $\boldsymbol{C}_{\mathrm{coriol}}$ from (3.3) uses the full Coriolis force terms and, at the same time, we can explicitly construct its inverse matrix. In Table 4.2 we present the number of multigrid iterations to gain 3 digits of defect improvement for several problem parameters and various smoothers.

For larger values of $\omega\Delta t$ the multigrid method with $\boldsymbol{C}_{\mathrm{coriol}}$-based smoother outperforms the SOR-type smoothers. Moreover, the block diagonal structure of $\boldsymbol{C}_{\mathrm{coriol}}$ makes it possible to find the inverse matrix explicitly. This makes the calculation of $\boldsymbol{C}^{-1}_{\mathrm{coriol}}$ for a given vector q very fast and easily done in parallel.

4.2. Multigrid solver for the modified pressure Poisson problem

We solve both the velocity problem in step 1 of the DPM and the modified pressure equation in step 2 by multigrid methods.

Table 4.2
Number of multigrid iterations of the momentum equation

Preconditioner	$\omega\Delta t$	Meshing level		
		3	4	5
$\boldsymbol{C}_{\mathrm{SOR}}$	0.6	2	2	2
$\boldsymbol{C}_{\mathrm{SORcoriol}}$	0.6	2	2	2
$\boldsymbol{C}_{\mathrm{coriol}}$	0.6	2	2	2
$\boldsymbol{C}_{\mathrm{SOR}}$	6	2	2	2
$\boldsymbol{C}_{\mathrm{SORcoriol}}$	6	2	2	2
$\boldsymbol{C}_{\mathrm{coriol}}$	6	2	2	2
$\boldsymbol{C}_{\mathrm{SOR}}$	60	Div	Div	Div
$\boldsymbol{C}_{\mathrm{SORcoriol}}$	60	3	3	3
$\boldsymbol{C}_{\mathrm{coriol}}$	60	2	2	2
$\boldsymbol{C}_{\mathrm{SOR}}$	600	Div	Div	Div
$\boldsymbol{C}_{\mathrm{SORcoriol}}$	600	10	16	12
$\boldsymbol{C}_{\mathrm{coriol}}$	600	2	2	2

Numerical results of Section 4.1 show that the geometric multigrid method with special smoothings is very effective for solving the velocity problem. However the overall efficiency of the DPM also depends on whether a fast solver is available for (3.8). Lemma 3.1 and the analysis of Section 3.3 ensure that the matrix $P = B^{\mathrm{T}}\boldsymbol{M}_{(\cdot)}^{-1}B$ with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (3.7) to (3.9) is sparse, symmetric, positive definite and corresponds to a mixed discretization of an elliptic problem with symmetric diffusion tensor. Thus one expects that standard multigrid techniques work well in this case. Numerical tests however show that the standard geometric multigrid method with SOR smoother does not provide a satisfactory solver for this problem in all practical cases. Therefore, we also test'stronger' smoothers such as ILU(k) and BiCGStab(ILU(k)).

The procedure to measure the multigrid convergence rates was chosen as follows: for given ω we apply several DPM iterations until some prescribed stopping criteria are satisfied. The obtained steady state solution $(\tilde{\boldsymbol{u}}, \tilde{p})$ is used as an initial solution so that $\mathrm{diag}(A) = \mathrm{diag}(A(\tilde{\boldsymbol{u}}))$. Further we solve the pressure diffusion equation by the multigrid method with two different smoothers and various values of $\omega\Delta t$. In Table 4.3 convergence rates are given for the V-cycle with four post-smoothing steps (no pre-smoothing) by ILU(1) iterations or two post-smoothing steps by BiCGStab with ILU(1) preconditioning. Thus in either case the computational complexity of the multigrid was approximately the same. Summarizing our numerical results for the pressure problem, we conclude:

- The convergence rates are almost level independent.
- For large values of $\omega\Delta t$ the matrix $P = B^{\mathrm{T}}\boldsymbol{M}_{(\mathrm{mass+coriol})}^{-1}B$ tends towards a tridiagonal matrix. This explains the excellent convergence rates with the ILU(1) and BiCGStab(ILU(1)) smoother since they are exact solvers for tridiagonal matrices. However, although the pressure diffusion equation with these matrices is easy to solve, the global behaviour of the outer DPM may get worse as the following section illustrates.

4.3. Numerical results for the DPM

We start numerical experiments with finding a stationary limit of unsteady solutions to the Stokes and the Navier–Stokes problem. This is done by performing a pseudo-time-stepping with the DPM until the steady state is achieved. To monitor the convergence to a steady solution we compute the values of $\|\boldsymbol{u}_t\|_{l_2}/\|\boldsymbol{u}\|_{l_2}$ and $\|p_t\|_{l_2}/\|p\|_{l_2}$. In the next section the DPM is used to compute the fully unsteady solution of the STR problem. Important characteristics for incompressible viscous flows with Coriolis force are given by the dimensionless Reynolds $(Re = \frac{UL}{\nu})$ and Rossby $(Ro = \frac{U}{2\omega L})$ numbers, where L and U are reference length and velocity, respectively. For the numerical experiments with the Stokes problem in Section 4.3.1 these numbers are not relevant, however. For the Navier–Stokes case in Section 4.3.2 we compute these numbers based on the reference values $L = 1$ and U equals the maximum velocity prescribed on the boundary $\partial\Omega$. This leads to $Ro = \frac{1}{\sqrt{2}}$; the Reynolds number varies.

Table 4.3
Multigrid convergence rates for different preconditioners $P = B^{\mathrm{T}}\boldsymbol{M}_{(\cdot)}^{-1}B$ with 4 smoothing steps, respectively, 2 smoothing steps for BiCGStab

Level	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
$\boldsymbol{M}_{(\mathrm{mass+coriol})}$					
Level 3	ILU(1)	0.17−02	0.14−02	0.35−05	0.57−07
Level 4	ILU(1)	0.19−02	0.19−02	0.77−03	0.12−06
Level 5	ILU(1)	0.50−02	0.52−02	0.47−02	0.24−06
Level 3	BiCGStab(ILU(1))	0.95−03	0.70−03	0.73−07	0.56−07
Level 4	BiCGStab(ILU(1))	0.39−03	0.35−03	0.12−03	0.12−06
Level 5	BiCGStab(ILU(1))	0.53−03	0.58−03	0.70−03	0.24−06
$\boldsymbol{M}_{(\mathrm{diag})}$					
Level 3	ILU(1)	0.31−01	0.14+00	0.23+00	0.25+00
Level 4	ILU(1)	0.28−01	0.20+00	0.34+00	0.35+00
Level 5	ILU(1)	0.13+00	0.38+00	0.44+00	0.45+00
Level 3	BiCGStab(ILU(1))	0.37−02	0.51−02	0.75−02	0.13−01
Level 4	BiCGStab(ILU(1))	0.95−02	0.45−01	0.79−01	0.78−01
Level 5	BiCGStab(ILU(1))	0.78−01	0.16+00	0.19+00	0.19+00
$\boldsymbol{M}_{(\mathrm{diag+coriol})}$					
Level 3	ILU(1)	0.31−01	0.10+00	0.13+00	0.25+00
Level 4	ILU(1)	0.28−01	0.20+00	0.32+00	0.35+00
Level 5	ILU(1)	0.10+00	0.31+00	0.36+00	0.45+00
Level 3	BiCGStab(ILU(1))	0.37−02	0.51−02	0.05−02	0.18−01
Level 4	BiCGStab(ILU(1))	0.89−02	0.29−01	0.71−01	0.78−01
Level 5	BiCGStab(ILU(1))	0.70−01	0.02+00	0.16+00	0.18+00

4.3.1. Results for the Stokes equations with Coriolis force

First we find a steady limit for the solution of (3.1) by the DPM with homogeneous force term $\boldsymbol{f} = 0$. The velocity equation in step 1 of the DPM is solved (almost) exactly. For the projection and correction steps 2 and 3 we examine two options for choosing $\boldsymbol{M}_{(\cdot)}$. One is $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\mathrm{mass})}$ leading to a standard projection method, another choice is $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\mathrm{mass+coriol})}$ defined in (3.7).

It is natural to expect that as soon as the value of $\omega\Delta t$ increases, the off-diagonal block of the matrix $\boldsymbol{M}_{(\mathrm{mass+coriol})}$, which is due to the Coriolis force, plays a more important role and the solution converges to a steady state in a smaller number of time steps. And vice versa, if $\omega\Delta t$ is small there is no big difference in the behavior of the standard and modified DPM. We illustrate both phenomena in Fig. 4.1.

4.3.2. Results for the Navier–Stokes equations

Similar to the Stokes case for the Navier–Stokes Eq. (1.3) one can expect to gain a substantial improvement by applying the modified DPM with the matrix $\boldsymbol{M}_{(\mathrm{mass+coriol})}$. However one may also take care about the contribution of convective terms to the matrix P in (3.8). As it was proposed in the previous section, the convective terms are taken into account by defining $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\mathrm{diag+coriol})}$ as in (3.9). The simple choice is given by

$$\boldsymbol{M}_{(\mathrm{diag})} = \begin{pmatrix} \mathrm{diag}(A) & 0 & 0 \\ 0 & \mathrm{diag}(A) & 0 \\ 0 & 0 & \mathrm{diag}(A) \end{pmatrix}$$

Fig. 4.2 compares the performance of the DPM with $\boldsymbol{M}_{(\cdot)}$ equal to $\boldsymbol{M}_{(\mathrm{mass+coriol})}$, $\boldsymbol{M}_{(\mathrm{diag+coriol})}$ and $\boldsymbol{M}_{(\mathrm{diag})}$. We note that although the use of $\boldsymbol{M}_{(\mathrm{mass+coriol})}$ leads for large $\omega\Delta t$ to almost tridiagonal matrix and therefore extremely fast multigrid convergence for pressure diffusion problem (see Table 4.3), the overall convergence behavior of the DPM is better with $\boldsymbol{M}_{(\mathrm{diag+coriol})}$.

For the last test case from these series, we perform computations with the linearized convective term of the form $\boldsymbol{U}\cdot\nabla\boldsymbol{u}$. To choose an appropriate $\boldsymbol{U}$, we first perform the numerical simulation for the Navier–Stokes equations until steady state. Then we set $\boldsymbol{U} = \boldsymbol{u}$ and solve this linear problem with the DPM which allows now much higher values of $\omega\Delta t$, since the convection part becomes linear. For the higher values of $\omega\Delta t$ the matrix $\boldsymbol{M}_{(\mathrm{diag+coriol})}$ in P ensures significantly better convergence to a steady solution than $\boldsymbol{M}_{(\mathrm{diag})}$ or other choice. Results are shown in Fig. 4.3.

5. Numerical experiments with the STR configuration

Finally, we demonstrate the behavior of the new DPM scheme for a more realistic configuration, namely the stirred tank reactor geometry which is shown in Fig. 5.1 (left). We solve the time-

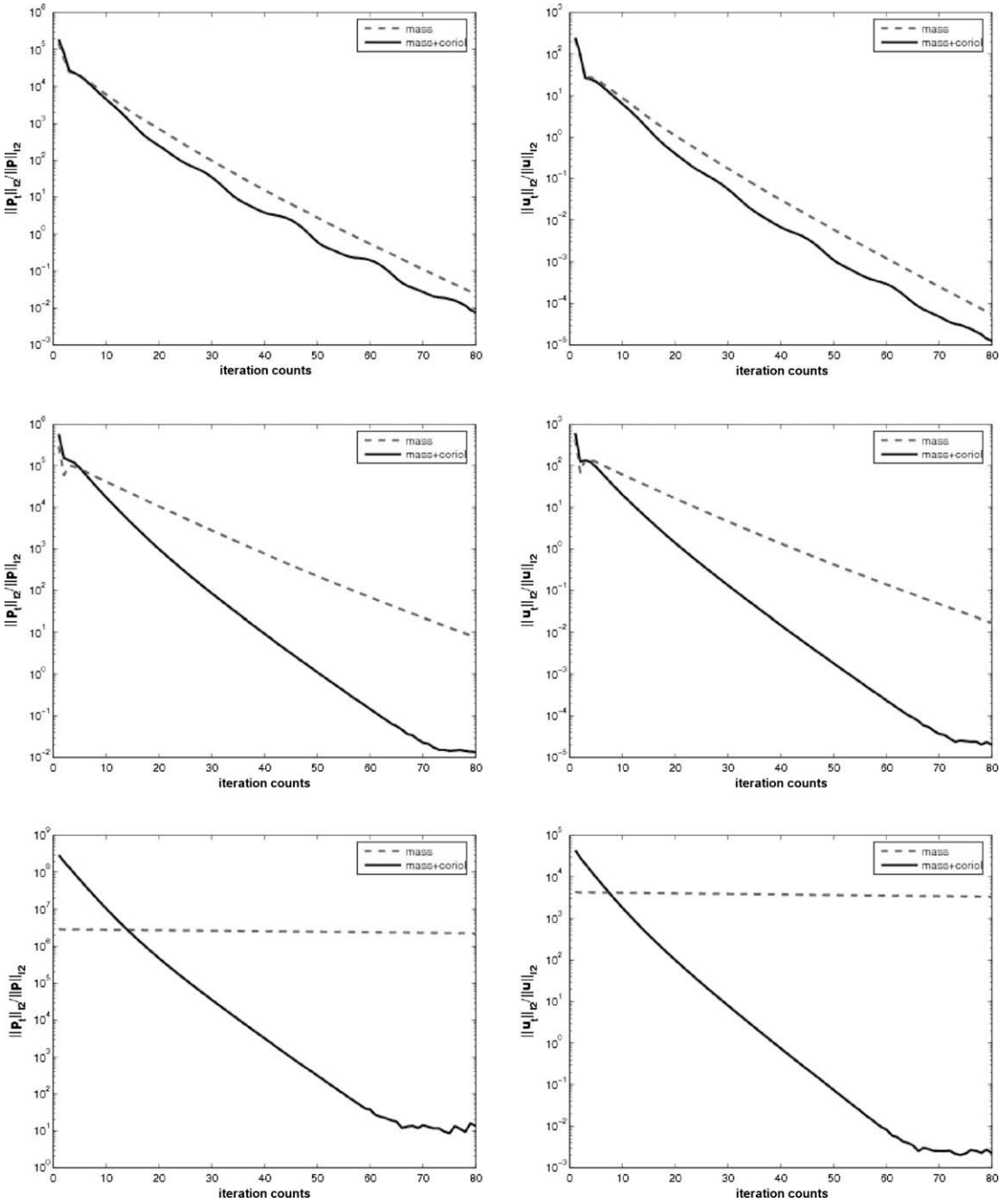

Fig. 4.1. Stokes equations (upper) $2\omega\Delta t = 0.5$; (middle) $2\omega\Delta t = 1.0$; (bottom) $2\omega\Delta t = 10.0$.

dependent Navier–Stokes equations with the following given data: the height of the tank $H = 4$, the radius of the tank $R = 10$, the length of the propeller $L_{\text{prop}} = 6$, the angular velocity of the propeller (ω) equals 2π. Fluid enters the tank through an inlet on the right side with a constant velocity equals 40, then it is 'mixed' by the rotating propeller and leaves the stirred tank through an outlet located on the left side. Thus for the Reynolds and the Rossby numbers we have $Re \approx 630$ and $Ro = 0.5$, where R is taken as a reference length and the velocity of the outer boundary is taken as a reference velocity. The coarsest mesh contains 22,528 quadrilaterals, 25,074 vertices and 70,144 faces. This mesh is presented in Fig. 5.1 (left). The finest mesh used in the STR simulation is two levels higher and possesses 884,736 quadrilaterals, 908,802 vertices and 2,678,272 faces leading to approximately 9 million unknowns.

The coordinate transformation made it possible to preserve the mesh aligned with the boundaries of the propeller such that even the small-scale flow features are resolved. At the end of the simulation, in the postprocessing phase, the backward coordinate transformation (from the noninertial to the inertial one) is performed and the velocity field is changed respectively to provide the user with the 'standard' motion of the propeller in the stirred tank reactor. The iso-surfaces and isolines of $|\boldsymbol{u}|$ are shown in Figs. 5.1 (right) and 5.2 (left). Next we compare the performance of the standard scheme (2.1) and (2.2) and the modified scheme. The Fig. 5.2 (right) shows the kinetic energy plots obtained for the for STR using both schemes with relatively large time step with the 'reference' kinetic energy computed with a smaller time step. One can note that the solution computed with the new projection method is somewhat more accurate.

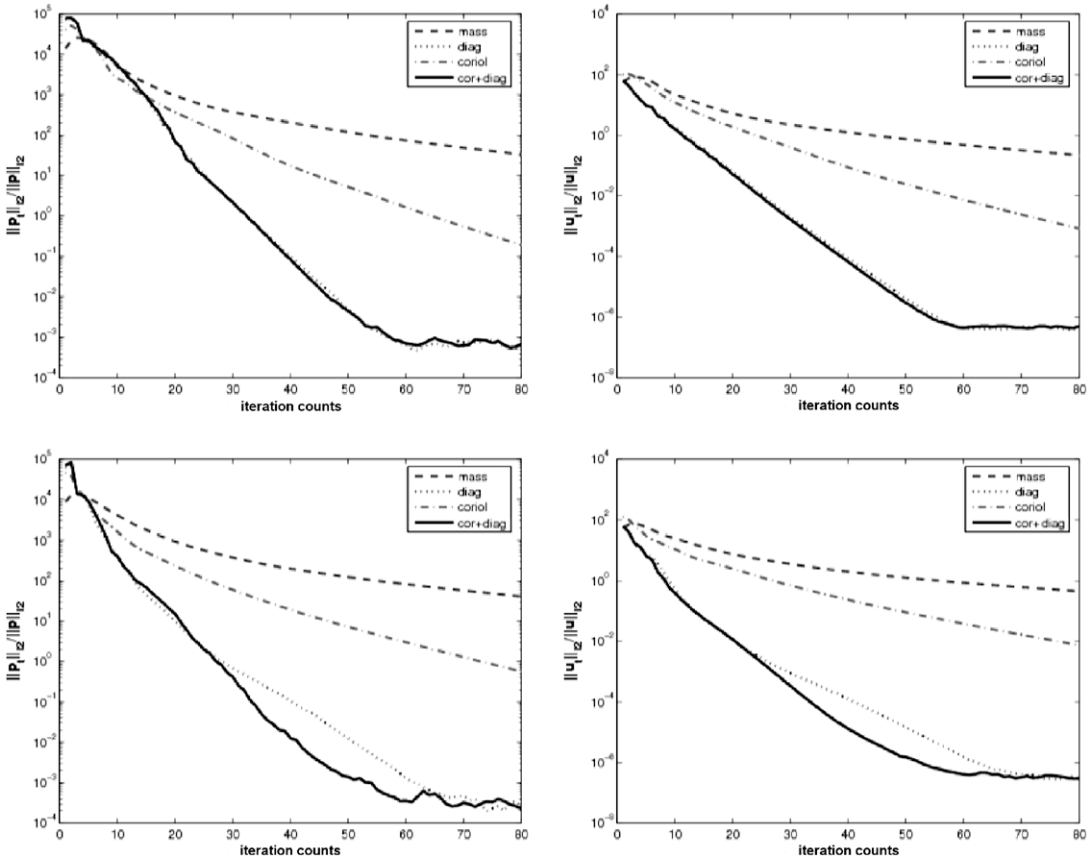

Fig. 4.2. Navier–Stokes equations (TOP) $2\omega\Delta t = 1.5$; (bottom) $2\omega\Delta t = 2.5$.

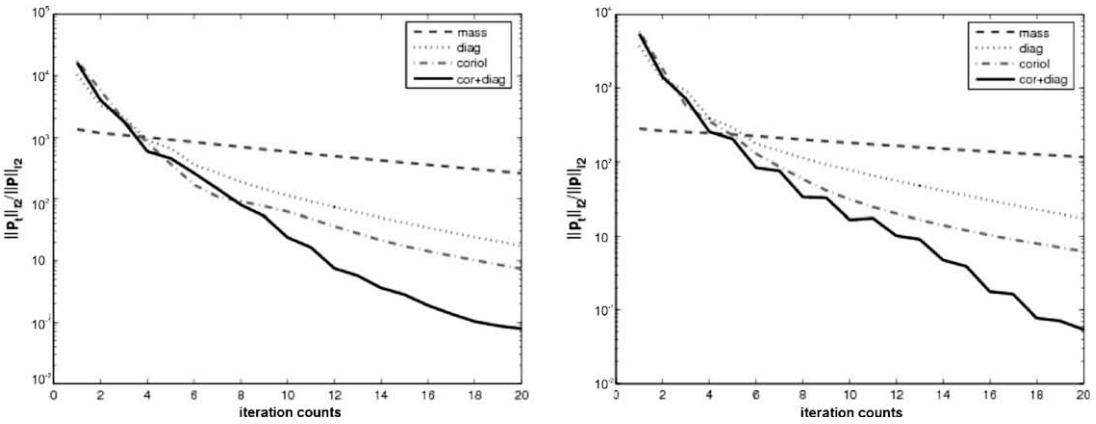

Fig. 4.3. Linearized Navier–Stokes equations (LEFT) $2\omega\Delta t = 5.0$; (right) $2\omega\Delta t = 10.0$.

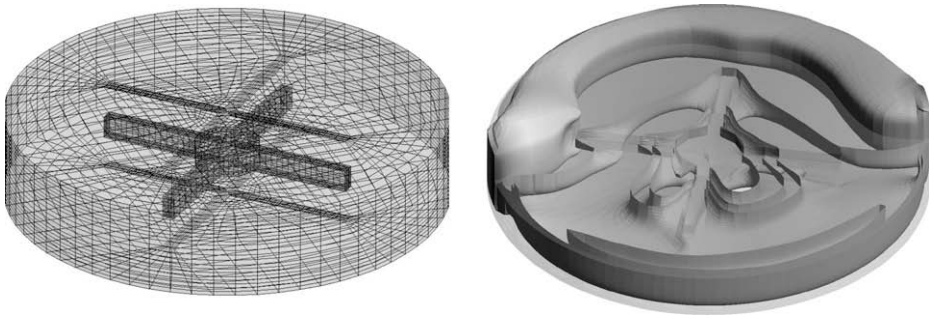

Fig. 5.1. (left) Coarse mesh; (right) iso-surface of $|\boldsymbol{u}|$.

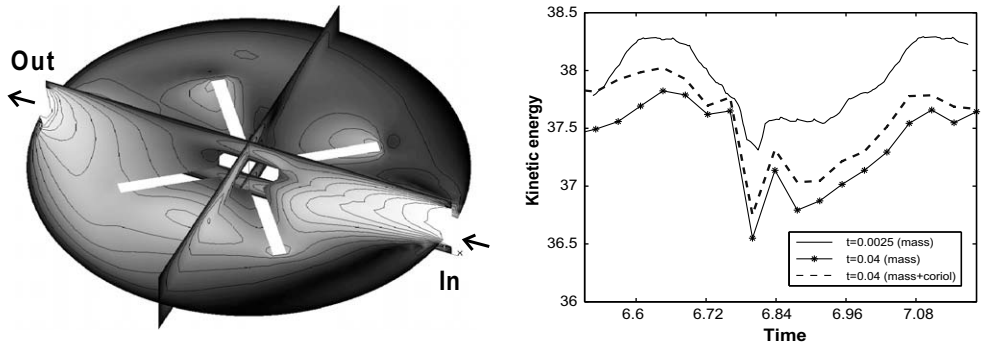

Fig. 5.2. (left) iso-lines of $|\boldsymbol{u}|$; (right) kinetic energy.

6. Conclusions

We proposed a new discrete projection method for the incompressible Navier–Stokes equations with Coriolis force which includes new multigrid and preconditioning techniques for the arising subproblems for pressure and velocity. In particular, the constructed multigrid method for the velocity matrix shows a robust convergence behavior for a wide range of $\omega\Delta t$ values. Moreover, its explicit inversion does not require any additional memory or computational resources. The modified discrete pressure Poisson operator in a projection step was deduced using the pressure Schur complement preconditioning technique. It appears to be much more efficient than the standard one since convective as well as rotational parts were taken into account. The numerical results show that the modified DPM is more efficient and robust with respect to the variation in problem parameters than standard projection schemes.

References

[1] J. Aharoni, Lectures on Mechanics, Oxford University Press, 1972.
[2] A.J. Chorin, J.E. Marsden, A Mathematical Introduction to Fluid Mechanics, third ed., Springer, 1992.
[3] H.C. Elman, Preconditioning strategies for models of incompressible flow, J. Sci. Comput. 25 (2005) 347–366.
[4] H.C. Elman, D. Silvester, A.J. Wathen, Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics, in: Numer. Math. Sci. Comput., Oxford University Press, Oxford, UK, 2005.
[6] D. Kuzmin, S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter, J. Comput. Phys. 198 (2004) 131–158.
[7] A.C. de Niet, F.W. Wubs, Two preconditioners for saddle point problems in fluid flows, Int. J. Numer. Methods Fluids 54 (2007) 355–377.
[8] M.A. Olshanskii, An iterative solver for the Oseen problem and numerical solution of incompressible Navier–Stokes equations, Numer. Linear Algebra Appl. 6 (1999) 353–378.
[9] M.A. Olshanskii, A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: A stabilization issue and iterative methods, Comput. Methods Appl. Mech. Engrg. 191 (2002) 5515–5536.
[10] M.A. Olshanskii, A. Reusken, Navier–Stokes equations in rotation form: A robust multigrid solver for the velocity problem, SIAM J. Sci. Comput. 23 (2002) 1683–1706.
[11] M.A. Olshanskii, Yu.V. Vassilevski, Pressure Schur complement preconditioners for the discrete Oseen problem, SIAM J. Sci. Comput. 29 (2007) 2686–2704.
[12] B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[13] A. Prohl, Projection and quasi-compressibility methods for solving the incompressible Navier–Stokes equations, BG Teubner, 1997.
[14] R. Rannacher, S. Turek, A simple nonconforming quadrilateral stokes element, Numer. Methods Partial Diff. Eq. 8 (1992) 97–111.
[15] A. Sokolov, Analysis and numerical implementation of discrete projection methods for rotating incompressible flows. PhD thesis, Universität Dortmund, in press.
[16] I.E. Tarapov, Mehanika sploshnoy sredi, Kharkov National University Press, 2002.
[17] S. Turek, On discrete projection methods for the incompressible Navier–Stokes equations: An algorithmical approach, Comput. Methods Appl. Mech. Engrg. 143 (1997) 271–288.
[18] S. Turek et al., FEATFLOW Finite element software for the incompressible Navier–Stokes equations: User Manual, Release 1.2 (<www.featflow.de>), Technical report, Universität Dortmund, 2000.
[19] S. Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, Springer, Berlin, 1999.
[20] S. Turek, W. Decheng and L. Rivkind, The fictitious boundary method for the implicit treatment of dirichlet boundary conditions with applications to incompressible flow simulation, in: Challenges in Scientific Computing CISC 2002, LNCSE, Springer, Berlin, 2002, pp. 37–68.
[21] A. Quarteroni, F. Saleri, A. Veneziani, Factorization methods for the numerical approximation of Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 188 (2000) 505–526.
[22] D. Wan, S. Turek, L. Rivkind, An efficient multigrid FEM solution technique for incompressible flow with moving rigid bodies, in: Numerical Mathematics and Advanced Applications, Springer, Berlin, 2003, pp. 844–853. Enumath 2003 Prague; ISBN-Nr. 3-540-21460-7.