A DISCRETE PROJECTION METHOD FOR INCOMPRESSIBLE VISCOUS FLOW WITH CORIOLIS FORCE *

ANDRIY SOKOLOV*, MAXIM A. OLSHANSKII†, AND STEFAN TUREK*

Abstract. The paper presents a new discrete projection method (DPM) for the numerical solution of the Navier-Stokes equations with Coriolis force term. We treat one time step of the projection method 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 multi-grid 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.

Key words Navier–Stokes equations, Coriolis force, discrete projection method, pressure Schur complement

1. Introduction. The construction of an efficient solver for the incompressible Navier-Stokes equations is a long-term purpose of CFD researchers. Since decades an evident progress is observed: a large variety of methods and algorithms were proposed and implemented into commercial and open-source codes. A detailed overview and a good mathematical foundation can be found, for instance, in [2, 4, 5, 7, 18].

In many physical and industrial applications there is the necessity of numerical simulations for models with moving geometries. In the literature one can find several techniques for handling such type of problems. Among them are Fictitious Domain [9], resp., Fictitious Boundary [25] and Arbitrary Lagrangian Eulerian [3] methods. Although being quite popular these methods require often a large amount of CPU time to simulate even 2D benchmark models if high accuracy is desired. Moreover, their handling of geometry and meshes serves as a source of additional errors in velocity and pressure fields. For example, the Fictitious Boundary approach often uses a fixed mesh and therefore may capture boundaries of a moving object not sufficiently accurate unless the mesh is very fine. At the same time, there is a large class of "rotating" models, when the application of the above methods can be avoided by some modifications of the underlying PDEs and/or by special transformations of the model that allow considering a static computational domain. However, different numerical problems may arise in this context.

As an example, let us consider the numerical simulation of a Stirred Tank Reactor (STR) benchmark problem (Fig. 1.1). The fluid motion is modelled by the nonstationary incompressible Navier-Stokes equations

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

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:

*Institut für Angewandte Mathematik, Universität Dortmund, Email addresses: asokolow@math.uni-dortmund.de, ture@featflow.de.

†Department of Mechanics and Mathematics, Moscow State University, 119899 Moscow, Email address: maxim.olshanskii(at)mtu-net.ru.

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

- 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.

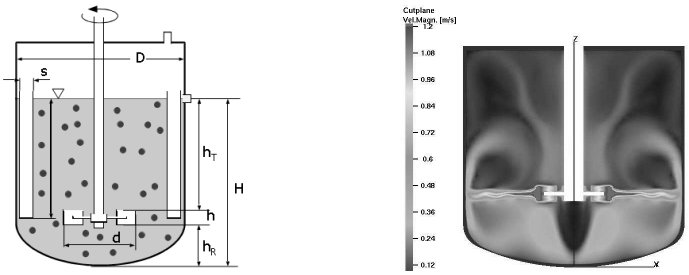

FIG. 1.1. *(LEFT) STR geometry; (RIGHT) Numerical simulation (cutplane of velocity).*

An obvious question is how to treat the moving boundary parts of the propeller. One way is to apply a Fictitious Boundary method for modeling the moving parts. This method has been implemented in the Featflow package [23]. The corresponding solver is based on iterative filtering techniques in combination with the fictitious boundary conditions for the moving tread patterns. All the calculations can be performed on a static mesh. From a practical point of view, this approach seems to be appropriate if the grid is sufficiently fine such that the realistic movement of the time-dependent tread patterns could be captured. More detailed description and analysis of the method can be found in [25, 27]. However, this approach leads only to a piecewise constant approximation of the moving parts. Therefore to prescribe properly the moving boundary parts of the stirred tank reactor we need a very fine mesh in that part of the domain where the rotating blades of the propeller are located. Furthermore, faces of the mesh elements are often not aligned with the boundaries of the blades and therefore can cause undesirably large perturbations in the velocity and pressure fields. These perturbations decrease the simulation accuracy in important parts of the domain and make it hardly possible to evaluate and optimize the shape of the blades. Experiencing such perturbations may be disadvantageous for the whole simulation, for example in the $k-\varepsilon$ turbulence modelling [11], when a very accurate prescription of the boundary layers is required.

Another approach for the simulation of the flow in a stirred tank reactor is the following one. We change the noninertial frame of reference to the inertial 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

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

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 [21]. Using the equality

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

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

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

The implicit discretization of (1.3) in time and in space leads to a saddle-point system to be solved in every time step. The system has the form (Δt is the time step)

$$\begin{pmatrix} \boldsymbol{S} & \Delta t B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ 0 \end{pmatrix}\,, \tag{1.4}$$

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 matrix $\boldsymbol{S}$ has the following block structure

$$\boldsymbol{S} = \begin{pmatrix} A & -\mathcal{M} & 0 \\ \mathcal{M} & A & 0 \\ 0 & 0 & A \end{pmatrix}\,, \tag{1.5}$$

where A is the block diagonal part of $\boldsymbol{S}$, which is due to the convective and diffusive terms, and $\mathcal{M}$ is the off-diagonal part of $\boldsymbol{S}$ due to the discretized Coriolis force term $2\boldsymbol{\omega} \times \boldsymbol{u}$. More details on the structure of the matrices A and $\mathcal{M}$ will be given in the next section.

The main objective of the paper is the design and analysis of efficient iterative solution methods for the system (1.4) and, based on this, building a discrete projection method for the time integration of (1.3). The STR problem will serve us as an example of applying this technique. Further in the paper, we construct several new preconditioners for pressure Schur complement type for the system (1.4) by building on the previous work in [14]. This development leads to a modified pressure Poisson problem in every time step of the discrete projection scheme. The modified projection step takes into account the influence of the Coriolis force and improves the performance of the scheme. As basic software for our simulations we use the PP3D module of the open-source CFD package Featflow [23].

2. Discrete projection method.

2.1. Discretization. We discretize the time derivative in the Navier-Stokes equations (1.3) by the one-step θ-scheme method. Given $\boldsymbol{u}^n$ and the time step $\Delta t = t_{n+1} - t_n$, find $\boldsymbol{u} = \boldsymbol{u}^{n+1}$ and $p = p^{n+1}$ (for the convenience we denote $p = P$ in (1.3)) satisfying

$$\begin{aligned} \frac{\boldsymbol{u} - \boldsymbol{u}^n}{\Delta t} + \theta((\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega} \times \boldsymbol{u}) + \nabla p &= \boldsymbol{g}^{n+1} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \quad \text{in } \Omega \times (0, T] \tag{2.1}$$

with the right-hand side

$$\boldsymbol{g}^{n+1} = \theta\boldsymbol{f}^{n+1} + (1-\theta)\boldsymbol{f}^n - (1-\theta)((\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \nu\Delta\boldsymbol{u}^n + 2\boldsymbol{\omega} \times \boldsymbol{u}^n).$$

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. 2.1). The analysis of these elements can be found in [19].

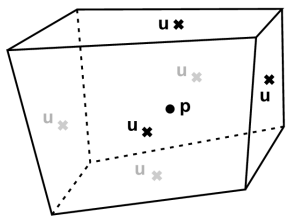

FIG. 2.1. *Nodal points of the nonconforming finite element in 3D.*

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:

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

Discretizing (2.1) in space, we obtain the following system of algebraic equations

$$\begin{pmatrix} A & -2\Delta t\theta\omega M & 0 & \Delta t B_1 \\ 2\Delta t\theta\omega M & A & 0 & \Delta t B_2 \\ 0 & 0 & A & \Delta t B_3 \\ B_1^T & B_2^T & B_3^T & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ p \end{pmatrix} = \begin{pmatrix} g_1^{n+1} \\ g_2^{n+1} \\ g_3^{n+1} \\ 0 \end{pmatrix} \, , \tag{2.2}$$

where $A = M + \Delta t\theta[N(\boldsymbol{u}) + \nu L]$ is the velocity stiffness matrix, M is the mass matrix and the matrix operators $N(\boldsymbol{u})$ and L are the discrete analogues of $(\boldsymbol{u} \cdot \nabla)\cdot$ and $(-\Delta)\cdot$, respectively; B is the gradient matrix. In practical realization, $\Delta t B_i p$ is replaced by $B_i \widetilde{p}$ with $\widetilde{p} = \Delta t \, p$. To stabilize the discretization of convection terms in the case of high Reynolds numbers we use the algebraic flux correction scheme of TVD type [12]. If a fixed point or Newton-like method is applied to (2.2), then on every nonlinear step one has to solve linear problems of similar of even the same block structure as (2.2).

2.2. Projection method. The system (2.2) can be rewritten in the form (1.4) by denoting $\boldsymbol{g} = (g_1^{n+1}, g_2^{n+1}, g_3^{n+1})^T$ and

$$\boldsymbol{S} = \begin{pmatrix} A & -2\omega\Delta t\theta M & 0 \\ 2\omega\Delta t\theta M & A & 0 \\ 0 & 0 & A \end{pmatrix} . \tag{2.3}$$

Several approaches for solving such systems are known from the literature. One is a fully coupled Vanka-type [26] approach (see [24] for the details). Alternatively one can consider operator-splitting schemes (see [22], [24]) of pressure Schur complement type. The latter is the approach taken in this paper.

First we linearize the nonlinear convective term in (2.3) by taking $A = M + \Delta t\theta[N(\boldsymbol{u}^*) + \nu L]$, where $\boldsymbol{u}^*$ is an approximation to the velocity from the previous time steps. Assuming that the resulting matrix $\boldsymbol{S}$ is nonsingular, we obtain from the first row of (1.4):

$$\boldsymbol{u} = \boldsymbol{S}^{-1}(\boldsymbol{g} - \Delta t B p) \, . \tag{2.4}$$

After substituting into the second row of (1.4), we get the Schur complement equation for p:

$$B^T \boldsymbol{S}^{-1} B p = \frac{1}{\Delta t} B^T \boldsymbol{S}^{-1} \boldsymbol{g} \, . \tag{2.5}$$

Using the decomposition $p = p^n + q$ with given p^n, we rewrite (2.5) in the form

$$B^T \boldsymbol{S}^{-1} B q = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}} \tag{2.6}$$

with $\widetilde{\boldsymbol{u}}$ being the solution of the auxiliary problem

$$\boldsymbol{S}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^n \,. \tag{2.7}$$

Thus, the coupled system (1.4) can be handled as follows:

1. Solve the momentum equation (2.7) for given p^n.
2. Solve the pressure Schur complement (PSC) equation (2.6) for q and set $p = p^n + q$.
3. Substitute p into relation (2.4) and compute the velocity $\boldsymbol{u}$.

However, the matrix $P_{\text{fact}} := B^T \boldsymbol{S}^{-1} B$ is never generated in practice. Doing so would be prohibitively expensive in terms of CPU time and memory requirements. Instead, we perform L iterations of a preconditioned Richardson method to solve the pressure Schur complement equation (2.5) approximately

$$p^{l+1} = p^l + \beta P^{-1}(\frac{1}{\Delta t} B^T \boldsymbol{S}^{-1} \boldsymbol{g} - P_{\text{fact}} p^l), \qquad l = 0, \ldots, L-1, \tag{2.8}$$

where β is a relaxation parameter and P is a suitable preconditioner to P_{fact} which is supposed to be a reasonable approximation to P_{fact} but being easier to 'invert'. Iterations (2.8) can be rewritten in the following way:

1. Find $\widetilde{\boldsymbol{u}}$ from $\boldsymbol{S}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^l$.
2. Solve $Pq = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}}$ for q.
3. Correct the pressure $p^{l+1} = p^l + \beta q$
4. Iterate 1-3 until convergence or $l = L - 1$.

The number of cycles L can be fixed or chosen adaptively so as to achieve a prescribed tolerance for the residual. The choice of $L = 1$ with appropriate preconditioner P is equivalent to what is known in the literature as the Discrete Projection Method (DPM), see [22].
Assuming that the pressure Schur complement preconditioner P has the form $P = B^T M_{(\cdot)}^{-1} B$, an approximation for the velocity $\boldsymbol{u}$ can be found by computing

$$\boldsymbol{u} = \widetilde{\boldsymbol{u}} - \Delta t M_{(\cdot)}^{-1} B q \tag{2.9}$$

instead of solving (2.4). Note that the velocity $\boldsymbol{u}$ in (2.9) satisfies the incompressibility constraint $B^T \boldsymbol{u} = 0$. We will discuss the choice of the auxiliary matrix $M_{(\cdot)}^{-1}$ later in the paper.

Thus, one step of the DPM reads for $t_n \to t_{n+1}$:

1. Given $p^n \simeq p(t_n)$ solve for $\widetilde{\boldsymbol{u}}$ equation (2.7)
2. Solve the modified discrete pressure Poisson problem

$$Pq = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}} \qquad \text{with } \; P = B^T M_{(\cdot)}^{-1} B \,. \tag{2.10}$$

3. Correct pressure and velocity

$$p^{n+1} = p^n + \beta q \,, \tag{2.11}$$
$$\boldsymbol{u}^{n+1} = \widetilde{\boldsymbol{u}} - \Delta t M_{(\cdot)}^{-1} B \, q \,. \tag{2.12}$$

In the next section we address the following two issues:

- Building an efficient multigrid solver for the velocity subproblem (2.7).
- Finding an appropriate matrix $M_{(\cdot)}$ involved in steps (2.10) and (2.12).

Note that taking $M_{(\cdot)}$ equals a pressure mass matrix leads to a discrete counterpart of the well-known Chorin projection method (see, e.g., [4]). Our choice for the matrix $M_{(\cdot)}$ below results from considering P as a preconditioner for the pressure Schur complement of the problem and accounts for the influence of the Coriolis terms in the equations (1.3).

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

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

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.

3.1. Velocity subproblem. Assuming a hierarchy of grids let us consider a multigrid method for solving equation (2.7). For smoothing iterations we take a linear iterative method of the form

$$\widetilde{\boldsymbol{u}}^{l+1} = \widetilde{\boldsymbol{u}}^{l} + \alpha\boldsymbol{C}^{-1}(\boldsymbol{g} - \Delta t B p^n - \boldsymbol{S}\widetilde{\boldsymbol{u}}^{l})\,, \tag{3.2}$$

where α is a relaxation parameter and $\boldsymbol{C}$ is a suitable preconditioner of $\boldsymbol{S}$. We are interested in a smoother efficient for the case of large values of the Coriolis force term, i.e. when the off-diagonal parts in the matrix (2.3) 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.

Taking an implicit θ-scheme, for instance $\theta = 1$ (Backward/Implicit Euler), we obtain the off diagonal values in (2.3) to be $2\omega\Delta t M$. If this value is large enough, the Coriolis terms should be taken into account in $\boldsymbol{C}$. Following [16], we put

$$\boldsymbol{C} := \boldsymbol{C}_{coriol} = \begin{pmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{pmatrix}\,, \tag{3.3}$$

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 [16] 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 $\boldsymbol{C}_{coriol}$ are diagonal matrices, one can explicitly find its inverse $\boldsymbol{C}_{coriol}^{-1}$ [20]. 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 $\boldsymbol{C}_{coriol}$ outperform such standard smoothers as Jacobi or SOR methods.

3.2. Modified pressure equation. The numerical solution of the pressure Schur complement problem (2.6) is typically done by applying the preconditioned Richardson iteration (2.8), where the choice of an optimal preconditioner P is most crucial. If $\boldsymbol{Su}$ corresponds to

$$\alpha \boldsymbol{u} - \nu \Delta \boldsymbol{u} \, ,$$

then an effective preconditioner for S is known and its detailed construction can be found, for instance, in [24, 10]. If $\boldsymbol{Su}$ corresponds to

$$\alpha \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \boldsymbol{w} \times \boldsymbol{u} \, ,$$

then an effective preconditioner is harder to develop. Furthermore, in the more general case this operator contains not only the Coriolis force, but also the convective term, and therefore having effective preconditioners is of great practical importance especially for the case of higher Reynolds numbers. 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 [13, 14, 15].
Here we follow the approaches given in [14, 15] and [24] to construct a preconditioner for the discrete counterpart of the Schur operator:

$$P_{\text{fact}} = -\nabla \cdot (\alpha I - \nu \Delta + \boldsymbol{w}\times)^{-1} \nabla \, . \tag{3.4}$$

To this end, let us consider the influences of mass, coriolis and diffusion parts in (3.4) separately. From $A = M + \Delta t \nu L$ we get that if the time step or the kinematic viscosity is small enough, then we can assume that $A \approx M$ and therefore $P^{-1} = P_{\text{mass}}^{-1}$, where $P_{\text{mass}} = B^T M_L^{-1} B$. If the time step or the kinematic viscosity is sufficiently large, then we assume that $A \approx \Delta t \nu L$, with $B^T L^{-1} B \sim I$, and hence $P^{-1} = M_p^{-1}$, 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

$$P^{-1} = \alpha_R P_{\text{mass}}^{-1} + \alpha_D M_p^{-1} \tag{3.5}$$

with appropriate coefficients, for instance $\alpha_R = 1$, $\alpha_D = \Delta t \nu$. When the time step is small the diffusion-oriented part of the preconditioner $\alpha_D M_p^{-1}$ is often neglected (i.e. $\alpha_D = 0$), leading to a standard projection step as in the well-known Chorin scheme. In the case of the Coriolis force term involved, we use instead of P_{mass} the modified preconditioner

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

by choosing a 'Coriolis-oriented' mass matrix

$$\boldsymbol{M}_{(\text{mass+coriol})} = \begin{pmatrix} M_L & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & M_L & 0 \\ 0 & 0 & M_L \end{pmatrix} . \tag{3.7}$$

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

$$P_{\text{mass+coriol}} q = B^T \boldsymbol{M}_{(\text{mass+coriol})}^{-1} B \, q = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}}. \tag{3.8}$$

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 $\alpha_D M_p^{-1}$ can be also included in the definition of P. Alternatively one can include the diagonal part of $\boldsymbol{S}$ into the pressure diffusion operator. Namely, one can consider in (3.8)

$$\boldsymbol{M}_{(\text{diag+coriol})} = \begin{pmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{pmatrix}. \tag{3.9}$$

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 [14] p. 365 for more details):

$$\mathcal{L} = -\nabla \cdot \mathcal{M}^{-1}\nabla \qquad \text{with } \mathcal{M} = [I + \boldsymbol{w}\times\,], \quad \boldsymbol{w} = (0, 0, 2\omega\Delta t)^T .$$

One finds

$$\mathcal{M}^{-1} = (1 + |\boldsymbol{w}|^2)^{-1} \left[I + \boldsymbol{w} \otimes \boldsymbol{w} - \boldsymbol{w}\times\right] ,$$

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

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

Note that although the operator $\mathcal{M}$ 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. Denote

$$P = \{p_{ij}\},\ M_L = \{m_{ii}\},\ B = (B_1, B_2, B_3)^T \text{ with } B_{\mathcal{K}} = \{b_{ij}^{\mathcal{K}}\},\ s = 2\omega\Delta t\theta. \tag{3.10}$$

We need to prove that the matrix

$$P = \begin{pmatrix} B_1^T & B_2^T & B_3^T \end{pmatrix} \begin{pmatrix} M_L & -sM_L & 0 \\ sM_L & M_L & 0 \\ 0 & 0 & M_L \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} \tag{3.11}$$

is symmetric. Using notation (3.10) we get from (3.7)

$$p_{ij} = \sum_k \left(\frac{b_{ki}^1 b_{kj}^1}{m_{kk}(1+s^2)} + \frac{b_{ki}^1 b_{kj}^2 s}{m_{kk}(1+s^2)} - \frac{b_{ki}^2 b_{kj}^1 s}{m_{kk}(1+s^2)} + \frac{b_{ki}^2 b_{kj}^2}{m_{kk}(1+s^2)} + \frac{b_{ki}^3 b_{kj}^3}{m_{kk}} \right). \tag{3.12}$$

It is obvious that equality

$$\sum_k \frac{b_{ki}^1 b_{kj}^2 s}{m_{kk}(1+s^2)} - \frac{b_{ki}^2 b_{kj}^1 s}{m_{kk}(1+s^2)} = \sum_k (b_{ki}^1 b_{kj}^2 - b_{ki}^2 b_{kj}^1) \frac{s}{m_{kk}(1+s^2)} = 0 \tag{3.13}$$

would ensure that P is symmetric. Let us show that

$$b_{ki}^1 b_{kj}^2 - b_{ki}^2 b_{kj}^1 = 0, \qquad \forall i, j, k. \tag{3.14}$$

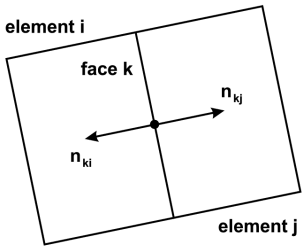

FIG. 3.1. *Definition of entries $b^{\mathcal{K}}_{ki}$ and $b^{\mathcal{K}}_{kj}$.*

To construct a discrete gradient operator B we assemble a discrete divergence operator $\mathcal{B}$ and use the equality $B = \mathcal{B}^T$ (see e.g., [8]). Denoting

$$\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3) \quad \text{with } \mathcal{B}_{\mathcal{K}} = \{d^{\mathcal{K}}_{ij}\}\ , \tag{3.15}$$

from the incompressibility constraint we get for a sum of integrals over all quadrilaterals T_k

$$\sum_k \langle \mathcal{B}\boldsymbol{u}, q\rangle_{T_k} = 0 \quad \forall q \in Q_0\ .$$

This is

$$\sum_k \langle \mathcal{B}_1 u_1 + \mathcal{B}_2 u_2 + \mathcal{B}_3 u_3, q\rangle_{T_k} = 0 \quad \forall q \in Q_0\ .$$

Performing integration by parts and taking into account that the pressure is piecewise constant, we construct the entries of the divergence operator $\mathcal{B}$

$$\left(d^1_{ij},\, d^2_{ij},\, d^3_{ij}\right) = \int_{T_i} \nabla\cdot\boldsymbol{\phi}_j\psi_i dx = -\int_{T_i} \boldsymbol{\phi}_j \nabla\psi_i dx + \int_{\partial T_i} \boldsymbol{\phi}_j\cdot\boldsymbol{n}\psi_i d\sigma = \int_{\partial T_i} \boldsymbol{\phi}_j\cdot\boldsymbol{n}\psi_i d\sigma \tag{3.16}$$

and the entries of the gradient operator B

$$\left(b^1_{ij},\, b^2_{ij},\, b^3_{ij}\right)^T = \int_{\partial T_j} \boldsymbol{\phi}_i\cdot\boldsymbol{n}\psi_j d\sigma\ , \tag{3.17}$$

where $\psi_j \in Q_0$, $\boldsymbol{\phi}_i \in \tilde{Q}^3_1$ such that the degrees of freedom of its components are defined through the surface inegral along the i-th face; $\boldsymbol{n} = \boldsymbol{n}_{ij} = (n^1_{ij}, n^2_{ij}, n^3_{ij})^T$ is a unit normal to the i-th face of the j-th element. In other words we obtain

$$B_1 = \{b^1_{ij} = n^1_{ij}\},\quad B_2 = \{b^2_{ij} = n^2_{ij}\},\quad B_3 = \{b^3_{ij} = n^3_{ij}\}.$$

Thus, for entries $b^{\mathcal{K}}_{ki}$ we use a vector $\boldsymbol{n}_{ki}$ and for entries $b^{\mathcal{K}}_{kj}$ we use $\boldsymbol{n}_{kj}$ (see Fig. 3.1). Then it holds $\boldsymbol{n}_{ki} = -\boldsymbol{n}_{kj}$ and (3.14) is satisfied. □

Remark 1: The proposition is true for any $P = B^T\boldsymbol{A}^{-1}B$, 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 2: If the angular velocity $\boldsymbol{\omega}$ increases, then the preconditioning matrix (3.6) becomes close to the degenerate case of a tridiagonal matrix (see [20]). 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.

3.3. Correction of velocity and pressure. Let us consider the last step of the DPM, i.e., equations (2.11), (2.12), and look for a necessary modification of velocity and pressure corrections. As an example consider $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\text{mass+coriol})}$. Multiplying both sides of (2.12) by B^T and using (3.8) we get

$$B^T\boldsymbol{u} = B^T\widetilde{\boldsymbol{u}} - \Delta t B^T \boldsymbol{M}^{-1}_{(\text{mass+coriol})} Bq = \Delta t(\frac{1}{\Delta t}B^T\widetilde{\boldsymbol{u}} - P_{\text{mass+coriol}}q) = 0.$$

Thus the discrete incompressibility constraint is satisfied for $\boldsymbol{u}$.

The equation for the pressure correction undergoes some modifications as well. Applying (3.5) with $\alpha_R = 1$ and $\alpha_D = \Delta t\nu$, we obtain from (3.8) the final equation for the pressure correction

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

where M_p is the pressure mass matrix.

3.4. Treatment of the convective term. In the previous section we considered a pressure Schur complement approach for 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. It is well known that in this case resulting numerical oscillations can cause the simulation to become unstable. Therefore some stabilization methods have to be implemented (see, for example, [7]). Another relevant question is how to include the terms due to convection and stabilization into appropriate pressure Schur complement preconditioners. This though question attracted a lot of considerations during last decade, see an overview in [6] and [17]. The off-diagonal nature and skew-symmetry of the $\boldsymbol{\omega}\times\cdot$ operator, which represents the Coriolis force, makes the question even more difficult to address. Note, however, that the convective and $\boldsymbol{\omega}\times\cdot$ operators can be written in a similar form. Using the well-known inequality

$$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = (\nabla\times\boldsymbol{u})\times\boldsymbol{u} + \nabla\left(\frac{\boldsymbol{u}^2}{2}\right) \tag{3.18}$$

and introducing a new pressure variable (Bernoulli pressure), we can replace the convective operator by the cross product one. Thus, the Coriolis force term and the convective operator can be treated simultaneously by $\boldsymbol{w}\times\cdot$, viz.

$$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla p = \boldsymbol{w}(\boldsymbol{u})\times\boldsymbol{u} + \nabla P \tag{3.19}$$

with $\boldsymbol{w}(\boldsymbol{u}) = \nabla\times\boldsymbol{u}+2\boldsymbol{\omega}$ and $P = p+\frac{\boldsymbol{u}^2}{2}$. 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 (3.19). While they are equal on the continuous level, their treatment and particularly the corresponding stabilization/implementation techniques may differ significantly on the discrete level.

Many reliable methods for the stabilization of convection dominated flows have been developed by the CFD community and are implemented in the Featflow code. Among them are streamline-diffusion and upwinding schemes, edge-oriented stabilization, algebraic flux correction, etc. However, at the same time not so much is known about stabilization techniques available for the term $(\nabla\times\boldsymbol{u})\times\cdot$. On the other hand, if we use (3.19) to transform both convection and Coriolis forces into the discrete cross product operator of a general form $\boldsymbol{w}(\boldsymbol{u})\times\cdot$, where $\boldsymbol{w}(\boldsymbol{u}) = (w_1, w_2, w_3)^T$, then in the 3-dimensional case one obtains

$$\boldsymbol{w}(\boldsymbol{u})\times\boldsymbol{v} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}. \tag{3.20}$$

Approximating w_i by diagonal counterparts (as we did for the Coriolis force term in the previous chapter), we obtain an operator which occurs to be nothing else but a 3×3 matrix, every entry of which forms a diagonal matrix. Thus, the obtained global matrix is easy to invert and therefore easy to use as a preconditioner (see [14] for a numerical analysis). However, due to the lack of stabilization techniques, we do not explore this approach at the moment and skip numerical tests for future.

3.5. Resulting algorithm. Finally we can write the modified DPM algorithm as follows (with p^n being the pressure from the previous time step):

1. Solve for $\widetilde{\boldsymbol{u}}$ the equation

$$\boldsymbol{S}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^n$$

with a multigrid method with smoothing iterations involving the special preconditioner $\boldsymbol{C}$ described in (3.3).

2. Solve the discrete pressure problem

$$Pq = B^T \boldsymbol{M}_{(\cdot)}^{-1} Bq = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}} \tag{3.21}$$

with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (3.7) or (3.9).

3. Calculate the pressure and the velocity approximations as

$$\begin{aligned} p &= p^n + q + \alpha M_p^{-1} B^T \widetilde{\boldsymbol{u}}\,, \\ \boldsymbol{u} &= \widetilde{\boldsymbol{u}} - \Delta t \boldsymbol{M}_{(\cdot)}^{-1} Bq \end{aligned} \tag{3.22}$$

with $\alpha = 0$ or $\alpha = \nu$. In the case of DPM set $p^{n+1} = p$, $\boldsymbol{u}^{n+1} = \boldsymbol{u}$ or perform several loops of these steps to get the fully coupled solution at time t_{n+1}.

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. For a discretization we consider a uniform Cartesian mesh. In the geometric multigrid solver we use several grid levels. In Table 4 we adopt the following

TABLE 4.1
Mesh characteristics of a unit cube with equidistant meshing.

level	NEL	NAT	NVT	NEQ
1st level	8	36	27	116
2d level	64	125	240	439
3d 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

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.

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 (2.3). Here we test a geometric multigrid method (V-cycle) with smoothing iterations defined in section 3.1. We compare it with the multigrid involving more standard pointwise SOR type smoothing iterations. This smoothing iterations can be defined as (3.2) with

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

$$\boldsymbol{C} := \boldsymbol{C}_{SORcoriol} = \begin{pmatrix} \text{lower\_part}(A) & 0 & 0 \\ 2\omega\Delta t M_L & \text{lower\_part}(A) & 0 \\ 0 & 0 & \text{lower\_part}(A) \end{pmatrix}$$

Both $\boldsymbol{C}_{SORcoriol}$ and $\boldsymbol{C}_{coriol}$ matrices take into account convective and Coriolis force terms. However, only $\boldsymbol{C}_{coriol}$ from (3.3) uses the full Coriolis force terms and, at the same time, we can explicitly construct its inverse matrix. In Table 4.1 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 $\boldsymbol{C}_{coriol}$-based smoother outperforms the

TABLE 4.2
Number of multigrid iterations of the momentum equation.

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

SOR-type smoothers. Moreover, the block diagonal structure of $\boldsymbol{C}_{coriol}$ makes it possible to find the inverse matrix explicitly. This makes the calculation of $\boldsymbol{C}_{coriol}^{-1}$ 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. Numerical results of § 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 (3.21). Lemma 3.1 and the analysis of § 3.2 ensure that the matrix $P = B^T \boldsymbol{M}_{(\cdot)}^{-1} B$ with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (3.7) or (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 $(\widetilde{\boldsymbol{u}}, \widetilde{p})$ is used as an initial solution so that $diag(A) = diag(A(\widetilde{\boldsymbol{u}}))$. Further we solve the pressure diffusion equation by the multigrid method with two different smoothers and various values of $\omega\Delta t$. In Table 4.2 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 $P = B^T M_{(\text{mass+coriol})} B$ 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.

TABLE 4.3

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

level	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
$M_{(\text{mass+coriol})}$					
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
$M_{(\text{diag})}$					
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
$M_{(\text{diag+coriol})}$					
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. Numerical results for the DPM. We start numerical experiments with finding a stationary limit of unsteady solution 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.

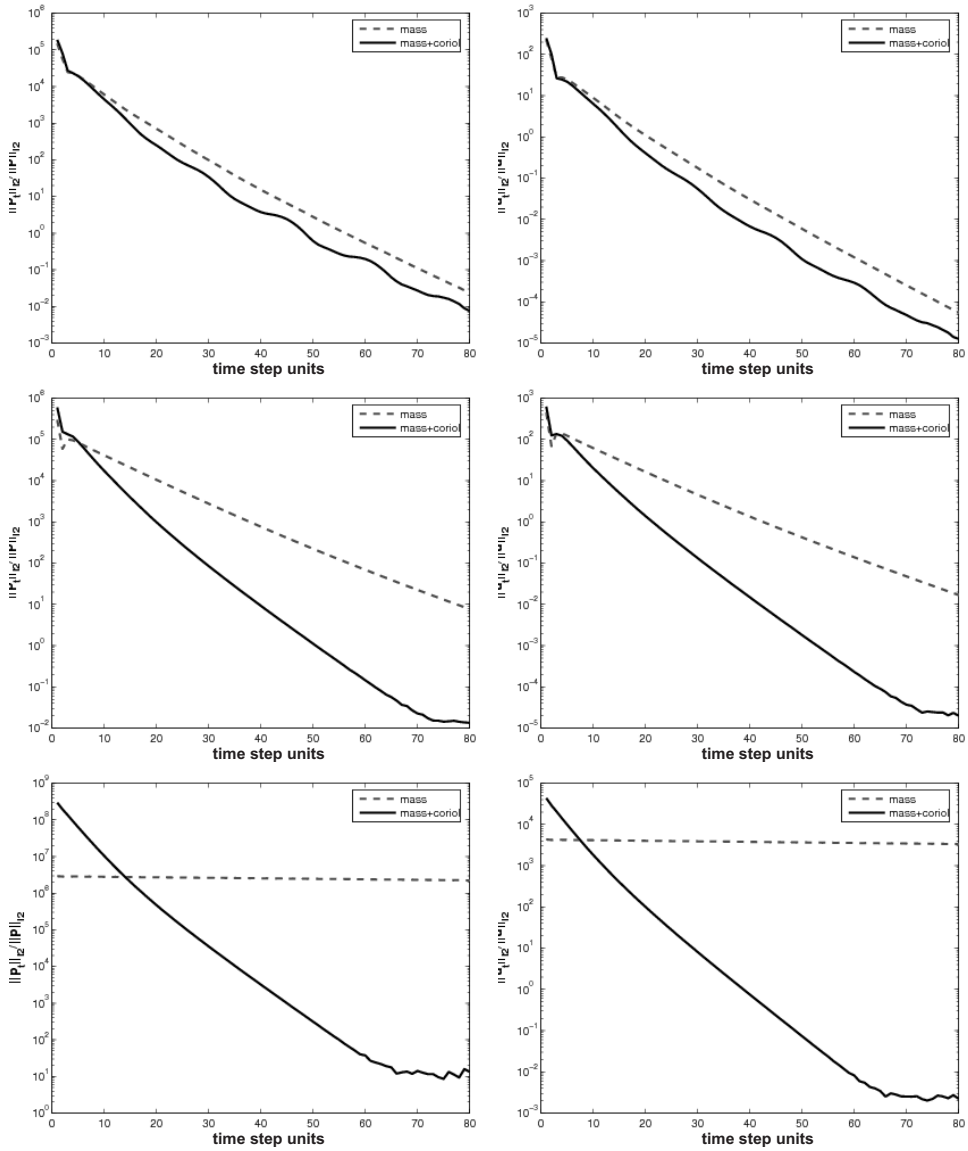

FIG. 4.1. *Stokes equations (UPPER)* $2\omega\Delta t = 0.5$*; (MIDDLE)* $2\omega\Delta t = 1.0$*; (BOTTOM)* $2\omega\Delta t = 10.0$.

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 $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\text{mass})}$ leading to a standard projection method, another choice is $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\text{mass+coriol})}$ 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 $\boldsymbol{M}_{(\text{mass+coriol})}$, 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 equations (1.3) one can expect to gain a substantial improvement by applying the modified DPM with the matrix $\boldsymbol{M}_{(\text{mass+coriol})}$. However one may also take care about the contribution of convective terms to the matrix P in (3.21). As it was proposed in the previous section, the convective terms are taken into account by defining $M_{(\cdot)} = \boldsymbol{M}_{(\text{diag+coriol})}$ as in (3.9). The simple choice is given by

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

The Fig. 4.2 compares the performance of the DPM with $M_{(\cdot)}$ equal to $\boldsymbol{M}_{(\text{mass+coriol})}$,

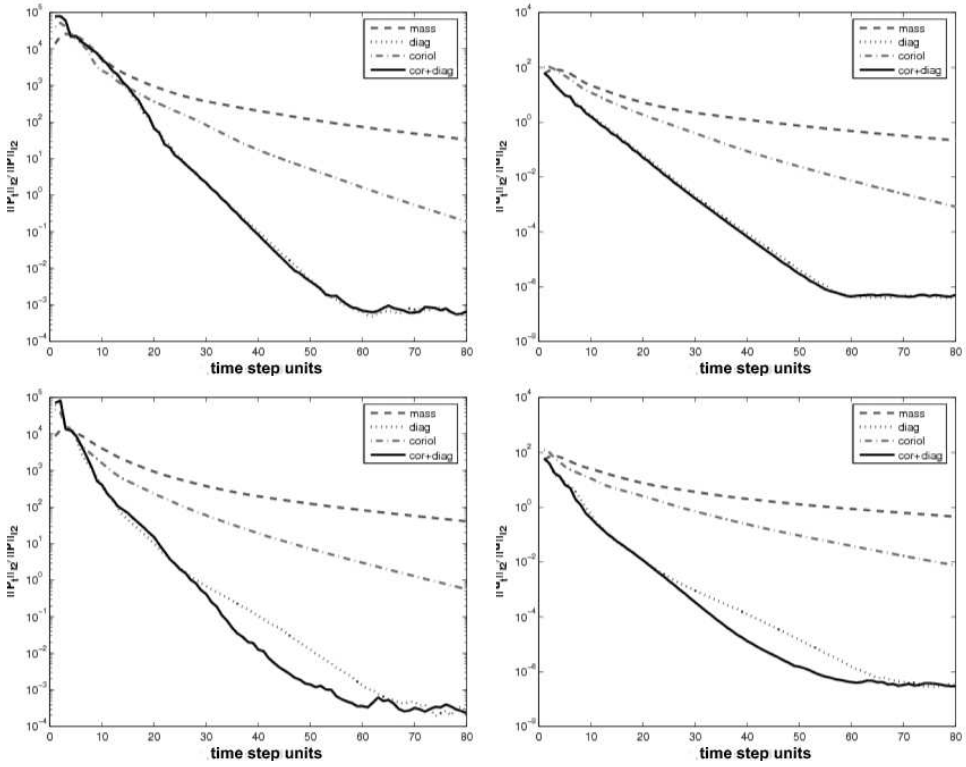

FIG. 4.2. *Navier-Stokes equations (TOP)* $2\omega\Delta t = 1.5$*; (BOTTOM)* $2\omega\Delta t = 2.5$.

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

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 $\boldsymbol{M}_{(\text{diag+coriol})}$ in P ensures significantly better convergence to a steady solution than $\boldsymbol{M}_{(\text{diag})}$ or other choice. Results are shown in Fig. 4.3.

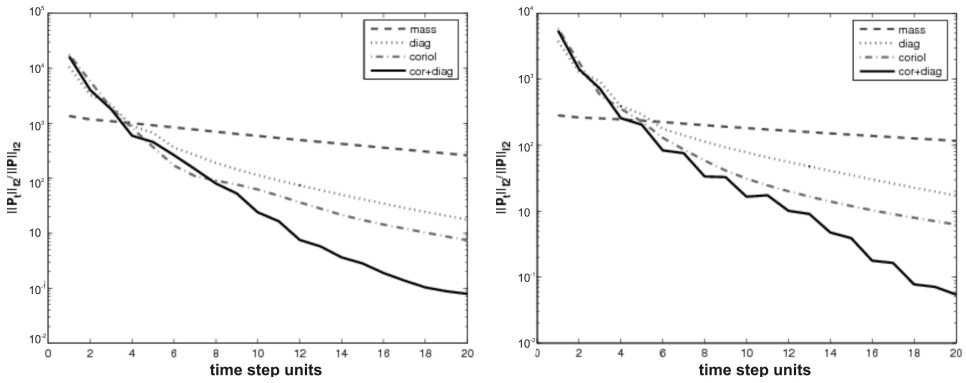

FIG. 4.3. *Navier-Stokes equations with* $\boldsymbol{U} \cdot \nabla \boldsymbol{u}$ *(LEFT)* $2\omega \Delta t = 5.0$*; (RIGHT)* $2\omega \Delta t = 10.0$.

5. Numerical experiments with the STR configuration. Finally, we demonstrate the behaviour of the new DPM scheme for a more realistic configuration, namely the stirred tank reactor geometry which is shown in Fig. 5.1 (left). The configuration is as follows: height of the tank $H = 4$, radius of the tank $R = 10$, length of the propeller $L_{prop} = 6$. 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.

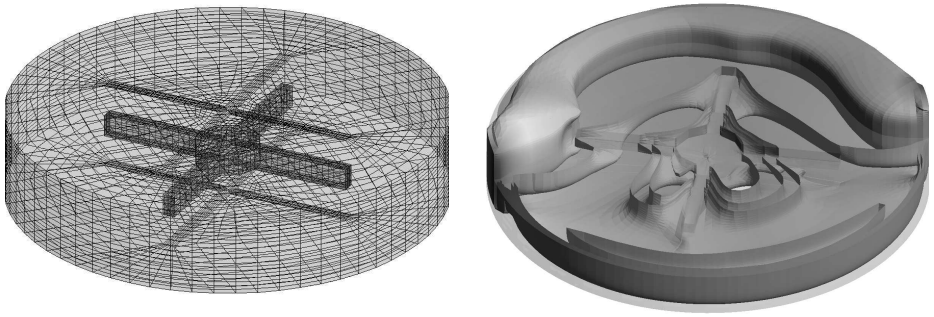

FIG. 5.1. *(LEFT) Coarse mesh; (RIGHT) Numerical simulation: velocity.*

Fluid enters the tank through an inlet on the right side, then it is 'mixed' by the rotating propeller and leaves the stirred tank through an outlet located on the left side. 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

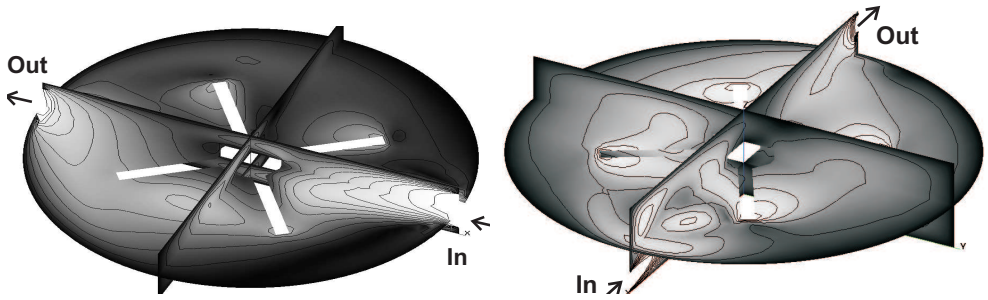

FIG. 5.2. *(LEFT) Example of the simulation 1: inlet and outlet are in the middle of the tank; (RIGHT) Example of the simulation 2: inlet is near the bottom and outlet is near the top of the tank.*

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 discrete projection method, considered in this paper, shows a very robust and accurate behaviour for this complex unsteady problem. The developed code exploits such advanced CFD techniques as stable non-conforming finite elements, robust high-resolution stabilization of the convective term, multigrid solvers, etc. Furthermore, the approach can be extended to population balance models or turbulent flows ($k - \varepsilon$ turbulence model) which is our current research.

6. Conclusions. We proposed a new Discrete Projection Method for the incompressible Navier-Stokes equations with Coriolis force due to a rotating system 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 behaviour 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] J. D. ANDERSON, *Computational Fluid Dynamics, the Basics with Applications*. McGraw-Hill Inc, 1995.
[3] C. DAHONG, *Numerische Simulation von Strömungsvorgängen mit der "Arbitrary Lagrangian-Eulerian method" (ALE-Methode)*. Aachen: Mainz, 1997. XVII, ISBN 3-89653-610-9.
[4] A. J. CHORIN AND J. E. MARSDEN, *A Mathematical Introduction to Fluid Mechanics, Third edition*. Springer, 1992.
[5] T. J. CHUNG, *Computational Fluid Dynamics*. Cambridge University Press, 2001.
[6] H. C. ELMAN, D. SILVESTER, AND A.J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Numer. Math. Sci. Comput., Oxford University Press, Oxford, UK, 2005.
[7] J. H. FERZIGER AND M. PERIC, *Computational methods for fluid dynamics*. Springer, 1996.
[8] V. GIRAULT AND P. A. RAVIART, *Finite Element Methods for Navier–Stokes Equations*. Springer, Berlin, 1972.

[9] R. GLOWINSKI, T. W. PAN, T. I. HELSA, D. D. JOSEPH, J. PERIAUX, *A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow*, J. Comput. Phys., Vol. 169, pp. 363–426, 2001.

[10] G. M. KOBELKOV AND M. A. OLSHANSKII, *Effective preconditioning of Uzawa type schemes for a generalized Stokes problem*, Numerische Mathematik, Vol. 86, pp. 443–470, 2000.

[11] KUZMIN, D. AND MEHRMANN, V. AND SCHLAUCH, S. AND SOKOLOV, A. AND TUREK, S., *Population Balances Coupled with the CFD-Code FeatFlow*, Ergebnisberichte des Instituts für Angewandte Mathematik, Nr. 324, FB Mathematik, Universität Dortmund, June 2006, (available on http://www.mathematik.uni-dortmund.de/lsiii).

[12] D. KUZMIN AND S. TUREK, *High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter*, J. Comput. Phys., Vol. 198, pp. 131–158, 2004.

[13] A. C. DE NIET AND F. W. WUBS, *Two preconditioners for saddle point problems in fluid flows*, Int. J. Numer. Methods Fluids, Vol. 54, pp. 355–377, 2007.

[14] M. A. OLSHANSKII, *An Iterative Solver for the Oseen Problem and Numerical solution of Incompressible Navier-Stokes Equations*, Numerical Linear Algebra and Applications, Vol. 6, pp. 353–378, 1999.

[15] 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*, Comp. Meth. Appl. Mech. Eng., Vol. 191, pp. 5515–5536, 2002.

[16] M. A. OLSHANSKII AND A. REUSKEN, *Navier-Stokes equations in rotation form: A robust multigrid solver for the velocity problem*, *SIAM Journal on Scientific Computing*, Vol. 23(5), pp. 1683–1706, 2002.

[17] M. A. OLSHANSKII AND YU. V. VASSILEVSKI , *Pressure Schur complement preconditioners for the discrete Oseen problem*, to appear in SIAM J. Sci. Comp. 2007.

[18] O. PIRONNEAU, *Finite Element Methods for Fluids*. Wiley, Masson, Paris, 1989.

[19] R. RANNACHER AND S. TUREK, *A Simple Nonconforming Quadrilateral Stokes Element*, Numerical Methods for Partial Differential Equations, Vol. 8, pp. 97–111, 1992.

[20] A. SOKOLOV, *Analysis and numerical implementation of discrete projection methods for rotating incompressible flows*. PhD thesis, Universität Dortmund, to appear in 2008.

[21] I. E. TARAPOV, *Mehanika sploshnoy sredi*. Kharkov National University Press, 2002.

[22] S. TUREK, *On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach*, Comp. Meth. Appl. Mech. Eng., Vol. 143, pp. 271–288, 1997.

[23] 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.

[24] S. TUREK, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.

[25] 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, pp. 37–68, Springer, Berlin, 2002.

[26] S. P. VANKA, *Implicit Multigrid Solutions of Navier-Stokes Equations in Primitive Variables*, J. Comput. Phys., Vol. 65, pp. 138–158, 1985.

[27] D. WAN, S. TUREK AND L. RIVKIND, *An Efficient Multigrid FEM Solution Technique for Incompressible Flow with Moving Rigid Bodies*, In Numerical Mathematics and Advanced Applications, pp. 844–853, Springer, Berlin, 2003. Enumath 2003 Prague; ISBN-Nr. 3-540-21460-7.