Analysis and Numerical Realisation of Discrete Projection Methods for Rotating Incompressible Flows

M. Sc. Andriy Sokolov

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Fakultät für Mathematik der

Technischen Universität Dortmund

vorgelegt am 13.11.2008

Summary

The first splitting schemes for solving the system of incompressible Navier-Stokes equations have been proposed already in the second half of the last century. Among them are continuous and discrete projection methods, ILU-based factorisation schemes, etc. The main purpose of splitting schemes is to divide the velocity-pressure coupled problem into smaller subproblems to be solved in an iterative way. Very often this leads to a drastic reduction of computational work and resources. The most popular splitting schemes are projection methods proposed by A. Chorin and R. Temam in the late 1960's. Since then, the projection methods were thoroughly analysed and constantly improved, particularly for new CFD configurations and applications.
In a broad class of industrial applications one has to deal with incompressible rotating flows. Very often it is helpful to perform coordinate transformation to a nonintertial frame of reference. On one hand, this technique facilitates the prescription of boundaries, since complex rotating objects become stationary, their mesh is constructed once and for all during the preprocessing phase. On the other hand, one has to consider the Navier-Stokes equations with rotational forces, which consist of Coriolis and centrifugal forces. In order to obtain an efficient and robust solver, it is important to come up with a proper handling of these extra terms. For coupled methods implementation of rotational forces is pretty much straightforward. For projection methods, however, some modifications in every step of the algorithm are required in order to guarantee fast (outer) convergence to a solution tuple $(\boldsymbol{u}, p)$. New numerical and mathematical problems arise in this context.

The work, presented in this thesis can be divided into two main steps:
First we propose a modified discrete projection method (DPM) for the incompressible Navier-Stokes equations with the Coriolis force term. The tuple $(\boldsymbol{u}, P)$ of velocity and new pressure variable[1] is computed in every time step by ILU-based methods in the context of a Pressure Schur Complement (PSC) approach with improved explicitly inverted preconditioners for both momentum and pressure Poisson-like equations. Proposed preconditioners are constructed in such a way that no additional computational efforts comparing to classical preconditioners are required. Numerical comparisons of nonmodified and modified projection schemes show that the latter scheme delivers better accuracy in time of velocity and pressure approximations for time-dependent problems, and faster convergence rates to the steady state for stationary problems. We prove that for the modified semi-discrete projection scheme the velocity is a weakly first-order approximation and the pressure is a weakly order $\frac{1}{2}$ approximation. Numerical experiments are performed for test models of unit square/cube geometries and for complex 3D models of stirred tank reactors.
In the second step, the proposed DPM-concept is extended to the general-purpose 'cross product operator' $\boldsymbol{w}(\boldsymbol{\omega}, \boldsymbol{u}, \cdot) \times \boldsymbol{u}$, which may include Coriolis force, convection and/or any other terms to be written in the cross-product form. It is shown that the non-diagonal dominance of the resulting matrix may lead to undesirable convergence and accuracy shortcomings of the algorithm. Nevertheless, implementations of additional techniques such as ∇div- and edge-oriented stabilizations make it possible to obtain sufficiently accurate solutions. Numerical tests are performed for the flow around cylinder and the lid-driven cavity benchmark problems.

The code of the algorithm is implemented into the open-source finite element software for the incompressible Navier-Stokes equations (`Featflow`).

Key words: Navier–Stokes equations, Coriolis force, discrete projection method, pressure Schur complement, rotational form of convection.

Andriy Sokolov, TU Dortmund

[1] New pressure variable contains pressure and centrifugal force

Acknowledgement

First of all, I would like to express my gratitude to my supervisor, Prof. Dr. Stefan Turek, who offered invaluable support and guidance throughout my PhD project. His constant optimism and encouragement in the non-trivial world of CFD has been inavaluable in the success of this project.

This research would also not have been possible without the support of my other supervisor, Prof. Dr. Maxim A. Olshanskii from the Moscow State university. I greatly appreciate his knowledge, constructive consultations and hospitality during my study and conference trips in Moscow.

Special thanks should go to Dr. Abderrahim Ouazzi for his key answers concerning algorithmic implementation and code details.

I would like to thank Prof. Dr. Volker Mehrmann and his team in the TU-Berlin for tight collaborations and fruitful discussions about stirred tank and backward facing step simulations, in particular Dr. Sonja Schmelter and Dr. Michael Schmidt.

I would like to thank Prof. Dr. Dmitri Kuzmin for valuable lectures and dialogues especially on the early stage of my research. Additionally I owe gratitude to my colleagues Dr. Otto Mierka, Dr. Christian Becker, Dr. Matthias Grajewski, Michael Köster, Jens Acker, Sven Buijssen, Dominik Göddeke and many others at the Chair for Applied Mathematics and Numerics at TU Dortmund for their openness and sharing of ideas in Mathematics and Computer Science.

Many thanks should also go to Jesse Blaine and Dr. Olga Gavrylyako for their valuable comments on the text of this thesis.

Finally, I would like to deeply thank my wife and my family for their understanding, patience and faith in me over the years of my study. Thank you very much!

A. S.

Contents

Part II Implementation and Future Extensions

1

Introduction

The design and numerical realisation 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 has been proposed and implemented into commercial and open-source codes. A detailed overview and a good mathematical foundation can be found, for instance, in [2, 19, 20, 31, 83].

In many physical and industrial applications there is the necessity of numerical simulations of models with rotating flows, for example, for the simulation of stirred tank reactor models in the chemical industry, analysis of the influence of the orbital motion of an inner cylinder on annular flow in the oil drilling industry [3, 93], shape optimization of mixers and ship propellers in food and shipbuilding industries, respectively, geophysical models, etc. These kinds of applications may possess complex 3D geometries and require large numerical simulations of fluid models with rotating boundary parts, those proper treatment can be often a very challenging task. Fortunately, there exists a large subclass of models, when complex treatment of rotating parts can be avoided by coordinate transformation and/or corresponding modification of PDEs for these models. In our work we will focus on this subclass of models and propose modified projection schemes for their numerical treatment.
As a representative of the subclass of ’rotating’ models, let us consider a Stirred Tank Reactor (STR) benchmark problem, which is shown in 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 appropriate boundary values and initial conditions are prescribed.

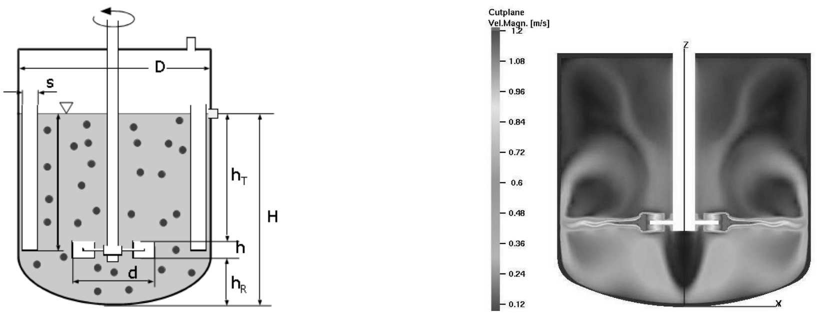

Fig. 1.1. (LEFT) STR geometry; (RIGHT) Numerical simulation (cutplane of velocity).

We assume that the STR model is satisfying the following conditions:

- There are no additional baffles attached to the outside wall of the stirred tank.
- Impeller rotates with constant angular speed $\boldsymbol{\omega} = (0, 0, \omega)^T$, where $\omega = const$ in the XY plane.
- The tank is filled with homogeneous liquid.

Later in our work we will show that the derived algorithmic framework remains valid and can be also applied to models which violate any of the requirements above.

It is natural to assume that the major changes in the velocity vector field in the stirred tank reactor are due to the blades of the rotating propeller. In order to accurately treat boundaries and flow field in a vicinity of the blades it is preferable to perform a coordinate transformation using a fixed mesh for the propeller around which the outside wall of the tank is rotating. When switching from an inertial to a noninertial frame of reference, we have to consider a new velocity vector $\boldsymbol{u} = \boldsymbol{v} + (\boldsymbol{\omega} \times \boldsymbol{r})$, where $\boldsymbol{\omega}$ is the angular speed and $\boldsymbol{r}$ is the radius vector from the center of coordinates. The new velocity $\boldsymbol{u}$ satisfies homogeneous Dirichlet boundary values on the blades of the propeller, while on the outside wall of the tank one has to satisfy $\boldsymbol{u} = \boldsymbol{\omega} \times \boldsymbol{r}$.

After some derivation (for details see § 10.1 and references therein), we obtain that the system of Navier-Stokes equations (1.1) in the new frame of reference can be written as follows

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

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.
Next, using the equality

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

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

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

We would like to note that the numerical analysis of (1.3) and its practical application is not new, see, e.g., [7, 21, 22, 23, 93]. Nevertheless, proper construction of projection methods, resp., operator splitting schemes for Coriolis dominating flows and its numerical analysis were not very well investigated until now. Therefore, in the first part of our work we will deal with the construction and numerical analysis of a discrete projection method for (1.3), modified in such a way as to effectively treat the Coriolis force term in every step of a projection scheme. Corresponding error estimates for velocity and pressure approximations will be derived. In the second part of our work we will show that the proposed modified DPM framework can be extended to a more general case - namely, effective treatment of any skew-symmetric term to be written in a cross-product form $\boldsymbol{w}(\boldsymbol{u}) \times \boldsymbol{u}$, for instance of rotational form of convection. The above transformation (1.3) possesses the following advantageous features for the numerical simulation of rotating flows:

- All calculations are performed on a static mesh with rotating/moving boundaries which can be adapted in a very precise way. At the same time if one uses Fictitious Domain [34], resp., Fictitious Boundary [106] methods, one can obtain only first-order approximations of the boundaries (due to the piecewise approximation of the interfaces). On the left hand side of Fig. 1.2 one can see that for the coordinate transformation strategy mesh is aligned along the boundaries of the rotating propeller. In this case the accuracy of prescription is fine even for a coarse level of refinement. At the same time from the right hand side of Fig. 1.2 one observes that for Fictitious Domain/Boundary approaches the given silhouette of the propeller is prescribed in a zigzag way. Therefore, the Fictitious Domain/Boundary approach is particularly suitable for the simulation of the qualitative flow behaviour of complex configurations only. If one likes to get quantitatively accurate results, then a good approximation of the ”real” boundaries and therefore a very fine mesh is needed, see Fig. 1.3. As another alternative, applying Arbitrary Lagrangian Eulerian [25] methods one can prescribe the mesh in an arbitrarily precise way, too, but additional complex reconstructions of the mesh in every time step, intermediate velocity updates, etc., are required anyway.

Precise approximations can be especially important if turbulence models with wall functions in a vicinity of the walls are used. Here the approach of coordinate transformation can be of a big help, too.

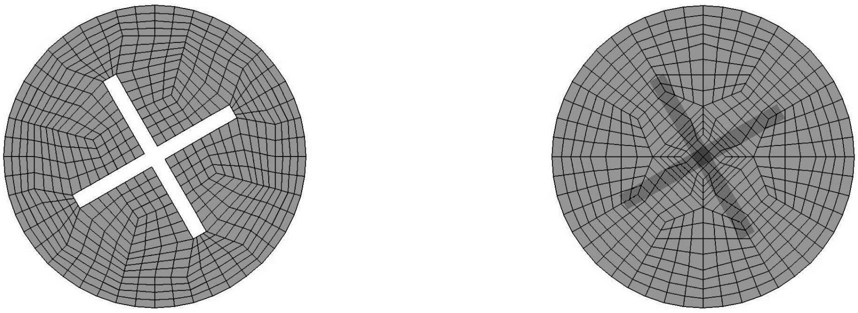

Fig. 1.2. Cross section, (LEFT) body-fitted mesh; (RIGHT) mesh for the fictitious boundary approach.

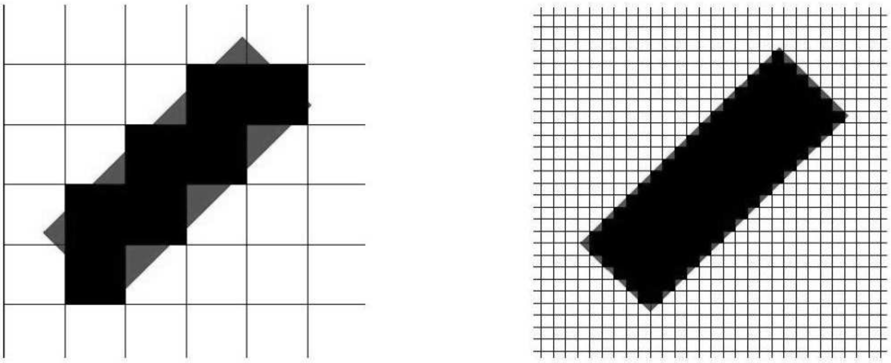

Fig. 1.3. Piecewise approximation of a moving blade. Gray: analytic description of the blade, black: approximation leading to a ”zig-zag” description of the boundary. (LEFT) very coarse mesh; (RIGHT) finer mesh [92].

- A fixed/’frozen’ mesh of the rotational model facilitates numerical calculations, when some optimization of geometry and/or flow field is required. In this case a mesh is constructed ’once and for all’ without any time consuming reassembly at every macro time step.
- The detailed analysis of (1.3) stimulates new Coriolis-oriented discrete projection schemes, which possess the same order of accuracy for velocity and pressure approximations as for the classical projection methods [18, 101] for the incompressible Navier-Stokes equations (1.1). 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

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

with the right-hand side

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

where $\boldsymbol{u}^*$ 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

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

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 $\boldsymbol{F}$ is a block matrix which is due to the discretized velocity operators in the momentum equation. The matrix $\boldsymbol{F}$ has the following block structure

$$\boldsymbol{F} = \begin{pmatrix} A & -\mathcal{M} & 0 \\ \mathcal{M} & A & 0 \\ 0 & 0 & A \end{pmatrix} , \tag{1.7}$$

where A is the block diagonal part of $\boldsymbol{F}$, which is due to the convective and diffusive terms, and $\mathcal{M}$ is the off-diagonal part of $\boldsymbol{F}$ due to the cross product operator $2\boldsymbol{\omega} \times \cdot$.
The implicit scheme (1.4)– (1.5) has excellent stability properties, see e.g. [105], 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, e.g. already mentioned projection methods. It has been observed by a number of authors, see, e.g. [82, 85], 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 $\boldsymbol{F}$ and for the pressure Schur complement matrix. Applying these principles, we will propose a modified projection scheme for a better treatment of the Coriolis force term $2\boldsymbol{\omega} \times \boldsymbol{u}$.

- The concept of projection schemes for the incompressible system of Navier-Stokes equations with Coriolis force (1.3) can be extended to the system of Navier-Stokes equations with a general-purpose cross-product term $\boldsymbol{w}(\boldsymbol{u}) \times \boldsymbol{u}$, which takes into account not only Coriolis, but also convection and/or any other term, which can be possible presented in the form of the cross-product operator. As an example, using the well-known inequality

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = (\nabla \times \boldsymbol{u}) \times \boldsymbol{u} + \nabla \left(\frac{\boldsymbol{u}^2}{2} \right) , \tag{1.8}$$

we can formulate our problem for a general case:

$$\begin{aligned} \boldsymbol{u}_t + \mathcal{W} \times \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla \mathcal{P} &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \quad \text{in } \Omega \times (0, T] \, , \tag{1.9}$$

where

– $\mathcal{W} = \nabla \times \boldsymbol{u}, \mathcal{P} = p + \frac{1}{2}\boldsymbol{u}^2$ for the system of the Navier-Stokes equations in the inertial frame of reference.

- $\mathcal{W} = \nabla \times \boldsymbol{u} + 2\boldsymbol{\omega}, \mathcal{P} = p + \dfrac{1}{2}\boldsymbol{u}^2 - \dfrac{1}{2}(\boldsymbol{\omega} \times \mathbf{r})^2$ for the system of Navier-Stokes equations in the noninertial frame of reference.

A variety of numerical problems may arise in the context of projection methods for the incompressible Navier-Stokes equations with the Coriolis force term (1.3) and a general-purpose formulation (1.9):

1. *Preconditioning*. How to construct effective Coriolis-oriented preconditioners in such a way that, on the one hand, they significantly improve the convergence behaviour of an iterative scheme (if compared to those with standard preconditioners) and, on the other hand, numerical costs of a scheme do not increase?
2. *Iterative behaviour*. To understand effectiveness of the constructed scheme, one has to examine both inner and outer iterative behaviour. For a right choice of preconditioners the outer convergence of the proposed scheme should be significantly faster and, at the same time, multigrid costs have to stay on the same level.
3. *Accuracy*. Before applying a new scheme one has to give an answer to the following question, namely, what order of accuracy can a user expect for velocity and pressure approximations?
4. *Stabilization*. The skew-symmetric nature of the Coriolis-operator $2\boldsymbol{\omega}$ and the general-purpose cross-product operator $\boldsymbol{w}(\boldsymbol{u}) \times \cdot$ may lead to undesired numerical properties and to bad convergence rates. Without proper stabilization techniques the corresponding numerical simulation is often not possible. So the question is, what stabilization methods to use and how will they influence the obtained solution and the solving process?
5. *Generalized framework*. The general-purpose cross-product operator $\boldsymbol{w}(\boldsymbol{u}) \times \boldsymbol{u}$, which contains rotational form of convection and Coriolis force term, requires additional techniques to stabilize the solution process and to guarantee sufficient accuracy of the obtained solution for medium and large Reynolds numbers. The obvious question is what techniques to choose and how to apply them to numerical simulations?

Further in the thesis we will discuss every topic in more detail, provide the corresponding derivations, examine numerical results and discuss possible implementations of the proposed schemes. The thesis is organized as follows: In chapter 2 we give a short overview on the Schur Complement approach for the classical Chorin-Temam projection methods. By applying the incomplete LU factorization we propose a modified projection method with improved explicitly inverted preconditioners for the system of incompressible Navier-Stokes equations with the Coriolis force term. Some auxiliary results concerning the proposed preconditioners are proved. Then, in chapter 3 we study the error estimate of a semi-discretized form of the modified projection method. We show that the velocity is a weakly first-order approximation and the pressure is weakly order $\frac{1}{2}$ approximation.
Proceeding further, in chapter 4 we introduce a general form of the cross-product operator $\boldsymbol{w}(\boldsymbol{u}) \times \cdot$, which includes not only the Coriolis force term, but also rotational convection. The modified projection method is extended to the general scheme. This extension makes it possible to treat any term to be written as $\boldsymbol{w}(\boldsymbol{u}, \cdot) \times \boldsymbol{u}$. In this chapter we also give a short theoretical outlook on Algebraic Flux Correction and Edge-oriented techniques for stabilization of standard and rotational forms of convection.
From theoretical derivations we move towards numerical experiments. Thus, in chapter 5 we present the numerical analysis of the modified projection method for the system of Stokes and Navier-Stokes equations with the Coriolis force term. We examine accuracy of the modified discrete projection method (DPM), multigrid behaviour for the pressure Poisson-like equation and convergence rates to the stationary solution for various Schur Complement preconditioners. Model problems in a unit square/cube are taken.
Next, we perform numerical experiments for the general-purpose modified projection scheme. In chapter 6 we test accuracy and iterative behaviour of the projection scheme for rotational form of convection. Taking the lid-driven cavity and the flow around cylinder benchmarks, we show that with the help of ∇div- and edge-oriented stabilization techniques one can obtain numerical solutions with sufficient accuracy. Pros and cons of numerical aspects of rotational convection are discussed.

Chapter 7 aims to give a short introduction into the non-commercial code `Featflow`. We discuss its applicability, main features, pre- and postprocessing steps, provide information concerning implementation of the coordinate transformation technique. Then, in chapter 8 we describe applications in the field of the stirred tank reactor simulations, where the modified projection method with its coordinate transformation strategy can be of a big importance. Complex 3D rotating flows for geometries of stirred tank reactors are simulated. For visualization purposes we take a particle tracing tool and observe moving behaviour of rotating particle in the calculated flow field. Here we also give an outlook on future investigations in the area of turbulent flows and population balance modeling.
Finally, chapter 9 summarizes the opportunities and drawbacks of the proposed scheme and Appendix A supplements the thesis by derivations of the incompressible system of the Navier-Stokes equations in the noninertial frame of reference.

The current work was supported by the Graduate School of Production Engineering and Logistics in TU Dortmund University, 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. Obtained scientific results were published in a series of papers. Thus, the content of chapter 2 including some numerical results from the chapter 5 was published as a research article in *Computer Methods in Applied Mechanics and Engineering* [98]. The theoretical derivation for the modified semi-discrete projection scheme was accepted to *Journal of Mathematical Fluid Mechanics* [75]. The content of the chapter 5 including some numerical results from the chapter 8 was published as a research article in *Electronic Transactions on Numerical Analysis*, [99].

Part I

Numerical Methods for Rotating Incompressible Flow Problems

2

Discrete projection methods for rotating incompressible flow with the Coriolis force

2.1 Discretization and solution aspects

In the current section we give an introductory insight onto the projection scheme with the Pressure Schur Complement concept. We also present space and time discretization to be used.

2.1.1 Discretization aspects

First, 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 $\Delta t = t_{n+1} - t_n$, then solve for $\boldsymbol{u} = \boldsymbol{u}^{n+1}$ and $p = p^{n+1}$ (for the convenience we denote $p = P$ in (1.3))

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

with the right-hand side

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

where $\boldsymbol{u}^*$ denotes a certain order extrapolation of velocity from n and $n-1$ time steps. Alternatively one may consider a fully implicit or explicit schemes by setting $\boldsymbol{u}^* = \boldsymbol{u}^{n+1}$ or $\boldsymbol{u}^* = \boldsymbol{u}^n$, respectively.

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 detailed description of these elements can be found in [88].

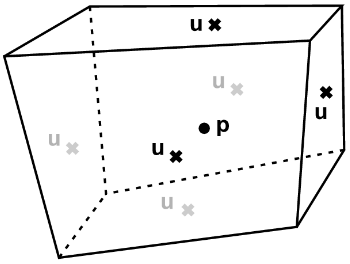

Fig. 2.1. Nodal points of the nonconforming finite element in 3D.

Applying the space discretization to (2.1), we obtain the following system of discrete 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 the continuous operators $(\boldsymbol{u} \cdot \nabla)\cdot$ and $(-\Delta)\cdot$, respectively; B is the *gradient* matrix. It is easy to check that the discrete divergence operator is equal to $(-B^T)$. In practical realization, $\Delta t B_i p$ is replaced by $B_i \widetilde{p}$ with $\widetilde{p} = \Delta t\, p$.

2.1.2 Pressure Schur Complement approach

If we denote by $\boldsymbol{g} = (g_1^{n+1}, g_2^{n+1}, g_3^{n+1})^T$ and set

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

we obtain the saddle point problem

$$\begin{pmatrix} \boldsymbol{F} & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ 0 \end{pmatrix} \tag{2.4}$$

with the block matrix $\boldsymbol{F}$ of the form (2.3). All the solution strategies for (2.4) can be divided into two categories: coupled and uncoupled, or co-called segregated methods. The representatives of coupled methods are, for instance, Vanka-like [111] approaches (see [105] for the details), or operator-splitting schemes (see [104, 105]). In our work we focus on the methods from the segregated class: namely, the Pressure Schur Complement reduction methods.

Assuming that the matix $\boldsymbol{F}$ is nonsingular (under some conditions [33]), we can perform the following factorization

$$\mathcal{A} = \begin{pmatrix} \boldsymbol{F} & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^T\boldsymbol{F}^{-1} & I \end{pmatrix} \begin{pmatrix} \boldsymbol{F} & 0 \\ 0 & \mathcal{S} \end{pmatrix} \begin{pmatrix} I & \boldsymbol{F}^{-1}B \\ 0 & I \end{pmatrix} \tag{2.5}$$

where $\mathcal{S} = -B^T\boldsymbol{F}^{-1}B$ is the Schur Complement of $\boldsymbol{F}$ in $\mathcal{A}$. By this we obtain in some sense the block LU factorization of the global matrix $\mathcal{A}$. Solvability is highly dependent on the diagonal dominance property of the 'middle' matrix on the right hand side of (2.5), which consists of the velocity matrix $\boldsymbol{F}$ and the Schur Complement operator $\mathcal{S}$. The only disadvantage is that $B^T\boldsymbol{F}^{-1}B$ is given only in implicit form, since $\boldsymbol{F}^{-1}$ is a dense matrix in general, and the computational effort for computing the inverse is too high. Therefore, appropriate iterative solution methods for (2.4) have to be constructed.

Some important properties can be derived from the factorization (2.5) (see [8]), in our analysis we will need the following:

- if $\boldsymbol{F}$ is symmetric positive definite and B has full column rank, then $\mathcal{S}$ is symmetric negative definite.
- $\mathcal{S}$, and hence $\mathcal{A}$, is invertible if and only if B has full column rank.

Constructed in $\widetilde{Q}_1/Q_0$ solution space, B does not have a full column rank (see Section 2.3.2 for the construction of B and corresponding discussions). Therefore the pressure is determined up to the constant and some extra condition on pressure is required, e.g. $\int_\Omega p(x)d\,x = 0$.

Let us consider the solvability of (2.4). In practice, we usually cannot know if the global matrix $\mathcal{A}$ is well-conditioned or not. The main criteria for the stability is to guarantee that the condition number of $\mathcal{A}$ remains bounded as size of the mesh $h \rightarrow 0$. This condition is named after Ladyzhenskaya, Babuska and Brezzi (LBB) [24, 33] and reads as follows:

$$\inf_{p_h \in Q_0} \sup_{\boldsymbol{u}_h \in \tilde{Q}_1} \frac{(p_h, \nabla \cdot \boldsymbol{u}_h)}{\|p_h\|_0 \|\nabla_h \boldsymbol{u}_h\|_0} \geq \gamma > 0 \tag{2.6}$$

Checking the inf-sup condition in its abstract form (2.6) is hard, therefore the following criteria can be useful in many cases.

Proposition 1 *Assume that $range(B)$ is closed and for any $\boldsymbol{u} \in V$, there exists $\boldsymbol{u}_h = \Pi_h \boldsymbol{u} \in V_h$ such that*

$$\begin{cases} b(\boldsymbol{u} - \Pi_h \boldsymbol{u}, q_h) = 0 \, \forall q_h \in Q_h \\ \|\Pi_h \boldsymbol{u}\|_V \leq c \|\boldsymbol{u}\|_V \end{cases} \tag{2.7}$$

with a constant $c > 0$ independent of h. Then, there exist k independent of h such that

$$\sup_{\boldsymbol{v}_h \in V_h} \frac{b(\boldsymbol{v}_h, q_h)}{\|\boldsymbol{v}_h\|_V} \geq k \|q_h\|_Q \, \forall q_h \in Q_h. \tag{2.8}$$

In our case, $b(q_h, v_h) = \langle Bq_h, v_h \rangle$. For our theoretical and practical purposes we choose velocity and pressure spaces to be as follows: $V_h = \tilde{Q}_1$, $Q_h = Q_0$. It is well known that the approximating pair $\tilde{Q}_1 \, / \, Q_0$ is numerically stable, i.e. the LBB condition is satisfied (for the description of the finite element spaces $\tilde{Q}_1 \, / \, Q_0$ and for the proof of their stability see [105]).

Now, let us assume that $\mathcal{A}$ and $\boldsymbol{F}$ are nonsingular matrices. Hence $\mathcal{S}$ is also nonsingluar. The first row of (2.4) reads

$$\boldsymbol{F}\boldsymbol{u} + Bp = \boldsymbol{g}. \tag{2.9}$$

Multiplying (2.9) by $B^T \boldsymbol{F}^{-1}$ and taking into account the incompressibility constraint $B^T \boldsymbol{u} = 0$ obtained from the second row of (2.4), we get the Pressure Poisson-like problem

$$B^T \boldsymbol{F}^{-1} Bp = B^T \boldsymbol{F}^{-1} \boldsymbol{g}. \tag{2.10}$$

Thus, the Pressure Schur Complement approach reads:

1. Solve for p the scalar equation (2.10).
2. Substitute the calculated p into (2.9) and solve it for $\boldsymbol{u}$.

Considering the time-dependent problem with fully implicit time-stepping schemes, we have to solve the sequence of saddle point problems (2.4) by the above Schur Complement approach (see [105], Chapter 2), i.e. to solve once (2.4) in every time step.

The above approach belongs to the class of *projection* methods [8, 33]. Indeed, from (2.4), we obtain (for details see [33]):

$$\begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} (I + \boldsymbol{F}^{-1} B \mathcal{S}^{-1} B^T) \boldsymbol{F}^{-1} \boldsymbol{g} \\ \mathcal{S}^{-1} B^T \boldsymbol{F}^{-1} \boldsymbol{g} \end{pmatrix} \tag{2.11}$$

We can see that the matrix

$$\mathit{\Pi} \equiv \boldsymbol{F}^{-1}B\mathcal{S}^{-1}B^T = \boldsymbol{F}^{-1}B(B^T\boldsymbol{F}^{-1}B)^{-1}B^T \tag{2.12}$$

is a projector ($\mathit{\Pi}^2 = \mathit{\Pi}$). It is an oblique projector onto $range(\boldsymbol{F}^{-1}B)$

$$\mathit{\Pi} v \in range(\boldsymbol{F}^{-1}B),\ \forall v \in \mathbb{R}^n$$

and is orthogonal to $range(B)$:

$$v - \mathit{\Pi} v \perp range(B),\ \forall v \in \mathbb{R}^n.$$

2.2 Discrete projection methods (DPM)

2.2.1 The classical Chorin-Temam projection method

Let us consider a well-known second order variant of the Chorin-Temam projection method [84, 19] 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):

1. Find intermediate velocity $\widetilde{\boldsymbol{u}}$ from

$$\frac{\widetilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\Delta t} + \frac{1}{2}((\boldsymbol{u}^* \cdot \nabla)\widetilde{\boldsymbol{u}} - \nu\Delta\widetilde{\boldsymbol{u}} + 2\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}) = \boldsymbol{g}^{n+1} - \nabla p^n, \tag{2.13}$$

   where $\boldsymbol{u}^*$ denotes a second order extrapolation of velocity from n and $n-1$ time steps.

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

$$\begin{cases} \dfrac{\boldsymbol{u}^{n+1} - \widetilde{\boldsymbol{u}}}{\Delta t} + \dfrac{1}{2}(\nabla p^{n+1} - \nabla p^n) = 0 \\ \qquad\qquad\qquad\qquad \nabla \cdot \boldsymbol{u}^{n+1} = 0. \end{cases} \tag{2.14}$$

To avoid any possible misleading we assume to be working with discretized in space projection scheme (2.13)-(2.14). 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 (2.13)–(2.14) can be written in the following algebraic form:

$$\begin{pmatrix} \boldsymbol{F} & 0 \\ B^T & -B^T(\frac{1}{\Delta t}\boldsymbol{M})^{-1}B \end{pmatrix} \begin{pmatrix} I_u & (\frac{1}{\Delta t}\boldsymbol{M})^{-1}B \\ 0 & I_p \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{n+1} \\ q \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{g}} \\ 0 \end{pmatrix} \tag{2.15}$$

with $\widehat{\boldsymbol{g}} = \boldsymbol{g}^{n+1} + \frac{1}{\Delta t}\boldsymbol{u}^n - Bp^n$ and $q = \frac{1}{2}(p^{n+1} - p^n)$. The matrix product on the left-hand side of (2.15) 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

$$\begin{pmatrix} \boldsymbol{F} & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} & 0 \\ B^T & -B^T\boldsymbol{F}^{-1}B \end{pmatrix} \begin{pmatrix} I_u & \boldsymbol{F}^{-1}B \\ 0 & I_p \end{pmatrix}. \tag{2.16}$$

The velocity submatrix has the form

$$\boldsymbol{F} = \begin{pmatrix} A & -\omega M & 0 \\ \omega M & A & 0 \\ 0 & 0 & A \end{pmatrix}. \tag{2.17}$$

where $A = (\varDelta t)^{-1}M + \frac{1}{2}[\,N(\boldsymbol{u}^*) + \nu L\,]$ is the velocity stiffness matrix. Therefore, if the time step $\varDelta t$ is sufficiently small the scaled block diagonal mass-matrix $(\varDelta t)^{-1}\boldsymbol{M}$ is a reasonable approximation to $\boldsymbol{F}$ and the incomplete factorization in (2.15) is close to the exact factorization (1.6). This shows that in some sense the projection method (2.13)–(2.14) approximates the coupled implicit method (2.1) for small $\varDelta t$.

2.2.2 The modified projection method

From the above consideration one realizes that a better approximation to $\boldsymbol{F}^{-1}$, compared to $(\varDelta t)\boldsymbol{M}^{-1}$, may lead to more effective (possibly less restrictive w.r.t. size of $\varDelta t$) projection methods. Below, see (2.32) and (2.34), we consider an approximation $\boldsymbol{M}_{(\cdot)}^{-1}$ to $\boldsymbol{F}^{-1}$ which takes into account the Coriolis terms and to some extend the convection terms. Thus, we consider the system (2.15) 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 $t_n \rightarrow t_{n+1}$):

1. Given $p^n \simeq p(t_n)$, $u^n \simeq u(t_n)$, and $\widetilde{\boldsymbol{g}} = \boldsymbol{g}^{n+1} + \frac{1}{\Delta t}\boldsymbol{u}^n$, then solve for $\widetilde{\boldsymbol{u}}$ the equation
$$\boldsymbol{F}\widetilde{\boldsymbol{u}} = \widetilde{\boldsymbol{g}} - Bp^n\,. \tag{2.18}$$
2. Solve the modified discrete pressure problem
$$Pq = B^T\widetilde{\boldsymbol{u}} \qquad \text{with } P = B^T\boldsymbol{M}_{(\cdot)}^{-1}B\,. \tag{2.19}$$
3. Correct pressure and velocity
$$p^{n+1} = p^n + q\,, \tag{2.20}$$
$$\boldsymbol{u}^{n+1} = \widetilde{\boldsymbol{u}} - \boldsymbol{M}_{(\cdot)}^{-1}B\,q\,. \tag{2.21}$$

Although we perform all our calculations with the discrete projection method (2.18)–(2.21), it is instructive to write down its semi-discrete counterpart. This is easy to do for the case of $\boldsymbol{M}_{(\cdot)}$ defined in (2.32). Now the procedure for advancing from time step n to step $n+1$ reads (compare to (2.13)–(2.14)) for given $\boldsymbol{u}^n, p^n$ and $\boldsymbol{g}^{n+1}$ defined in (1.5):

1. Find intermediate velocity $\widetilde{\boldsymbol{u}}$ from
$$\frac{\widetilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\varDelta t} + \frac{1}{2}((\boldsymbol{u}^* \cdot \nabla)\widetilde{\boldsymbol{u}} - \nu\varDelta\widetilde{\boldsymbol{u}} + 2\boldsymbol{\omega} \times \widetilde{\boldsymbol{u}}) = \boldsymbol{g}^{n+1} - \nabla p^n. \tag{2.22}$$
2. Find new velocity and pressure as the result of the following projection into the divergence-free subspace
$$\left\{ \begin{aligned} \frac{\boldsymbol{u}^{n+1} - \widetilde{\boldsymbol{u}}}{\varDelta t} + \boldsymbol{\omega} \times (\boldsymbol{u}^{n+1} - \widetilde{\boldsymbol{u}}) + \frac{1}{2}(\nabla p^{n+1} - \nabla p^n) &= 0 \\ \nabla \cdot \boldsymbol{u}^{n+1} &= 0. \end{aligned} \right. \tag{2.23}$$

Remark 1. For efficient calculations with the original projection method (2.13)–(2.14) or with the modified one, we need an efficient solver for the velocity subproblem with the matrix $\boldsymbol{F}$ as well as for the (modified) pressure problem with matrix P. In section 2.3.2 we will show that the modified method leads to a *symmetric* pressure problem of the diffusion type.

Remark 2. If we compare the factorizations (2.15) and (2.16), 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 $-B^T\boldsymbol{F}^{-1}B$. Another way to realize this is the following, see, e.g., [104]. Eliminating $\widetilde{\boldsymbol{u}}$ we can rewrite (2.18)–(2.20) as

$$p^{n+1} = p^n + P^{-1}(B^T\boldsymbol{F}^{-1}Bp^n - g) \tag{2.24}$$

with $g = B^T\boldsymbol{F}^{-1}\widetilde{\boldsymbol{g}}$. 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 [29].

Remark 3. One possible variation of the projection method is to add a diffusion dependent term to the pressure correction step (2.20):

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

In [84] (for the case $\boldsymbol{\omega} = 0$) it was discussed that adding such term may reduce numerical boundary layers in projection methods.

Remark 4. Observing (2.23) or the choice of $\boldsymbol{M}_{(\cdot)}^{-1}$ in (2.32) and (2.34) one notes that the modified projection step essentially takes into account the Coriolis terms and only indirectly (in (2.34)) the convection terms. Therefore the proposed modification of the method is especially suitable for the case of moderate Rossby Ro numbers. See, however, propositions for the case of large Rossby numbers in the Chapter 4.1.

In the following of this section we mainly address the following two issues:

- Building an efficient multigrid solver for the velocity subproblem (2.18).
- Finding an appropriate matrix $\boldsymbol{M}_{(\cdot)}$ involved in steps (2.19) and (2.21).

2.3 Algorithmic details of the DPM

2.3.1 Velocity subproblem

Assuming a hierarchy of grids let us consider a multigrid method for solving equation (2.18). For smoothing iterations we take a linear iterative method of the form

$$\widetilde{\boldsymbol{u}}^{l+1} = \widetilde{\boldsymbol{u}}^l + \alpha\boldsymbol{C}^{-1}(\boldsymbol{g} - \Delta tBp^n - \boldsymbol{F}\widetilde{\boldsymbol{u}}^l)\ , \tag{2.25}$$

where α is a relaxation parameter and $\boldsymbol{C}$ is a suitable preconditioner of $\boldsymbol{F}$. 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 (2.17) have values equal or larger than those of the diagonal part. Note that in this case the skew-symmetric part of $\boldsymbol{F}$ 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 (2.17) to be $2\omega\Delta tM$. If this value is large enough, the Coriolis terms should be taken into account in $\boldsymbol{C}$. Following [74], we put

$$\boldsymbol{C}_{coriol} = \begin{pmatrix} diag(A) & -2\omega\Delta tM_L & 0 \\ 2\omega\Delta tM_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{pmatrix}\ , \tag{2.26}$$

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 [74] 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 $\boldsymbol{C}_{coriol}^{-1}$ by means of the proposition:

Proposition 2 *Consider a matrix of the following form:*

$$\boldsymbol{G} = \begin{pmatrix} a_1 & \dots & 0 & -l_1 & \dots & 0 & 0 & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \ddots & \\ 0 & \dots & a_n & 0 & \dots & -l_n & & & 0 \\ l_1 & \dots & 0 & b_1 & \dots & 0 & 0 & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \ddots & \\ 0 & \dots & l_n & 0 & \dots & b_n & & & 0 \\ 0 & & & 0 & & & c_1 & \dots & 0 \\ & \ddots & & & \ddots & & \vdots & \ddots & \vdots \\ & & 0 & & & 0 & 0 & \dots & c_n \end{pmatrix} \tag{2.27}$$

Then its inverse is

$$\boldsymbol{G}^{-1} = \begin{pmatrix} \frac{b_1}{a_1b_1+l_1^2} & \dots & 0 & \frac{l_1}{a_1b_1+l_1^2} & \dots & 0 & 0 & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \ddots & \\ 0 & \dots & \frac{b_n}{a_nb_n+l_n^2} & 0 & \dots & \frac{l_n}{a_nb_n+l_n^2} & & & 0 \\ \frac{-l_1}{a_1b_1+l_1^2} & \dots & 0 & \frac{a_1}{a_1b_1+l_1^2} & \dots & 0 & 0 & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \ddots & \\ 0 & \dots & \frac{-l_n}{a_nb_n+l_n^2} & 0 & \dots & \frac{a_n}{a_nb_n+l_n^2} & & & 0 \\ 0 & & & 0 & & & \frac{1}{c_1} & \dots & 0 \\ & \ddots & & & \ddots & & \vdots & \ddots & \vdots \\ & & 0 & & & 0 & 0 & \dots & \frac{1}{c_n} \end{pmatrix} \tag{2.28}$$

Proof. By direct calculation. □

In the corresponding section of our work we will present results of numerical experiments with the multigrid method using different smoothers. We will see that iterations (2.25) with the preconditioner $\boldsymbol{C}_{coriol}$ outperform such standard smoothers as Jacobi or SOR methods.

2.3.2 Modified pressure equation

The numerical solution of the pressure Schur complement problem (2.19) is typically done by applying the preconditioned Richardson iteration (2.24), where the choice of an optimal preconditioner P is most crucial.

If $\boldsymbol{F}\boldsymbol{u}$ 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 [105, 48]. If $\boldsymbol{F}\boldsymbol{u}$ 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 [71, 72].

Here we follow the approaches given in [72] and [105] to construct a preconditioner for the discrete counterpart of the Schur operator:

$$P_{\text{fact}} = -\nabla \cdot (\alpha I - \nu \Delta + \boldsymbol{w} \times)^{-1} \nabla \, . \tag{2.29}$$

To this end, let us consider the influences of mass, Coriolis and diffusion parts in (2.29) 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

$$P^{-1} = \alpha_R P_{\text{mass}}^{-1} + \alpha_D M_p^{-1} \tag{2.30}$$

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

$$P_{\text{mass+coriol}} = B^T \boldsymbol{M}_{(\text{mass+coriol})}^{-1} B \tag{2.31}$$

by choosing a 'Coriolis-oriented' mass matrix

$$\boldsymbol{M}_{(\text{mass+coriol})} = \begin{pmatrix} M_L & -2\omega \Delta t M_L & 0 \\ 2\omega \Delta t M_L & M_L & 0 \\ 0 & 0 & M_L \end{pmatrix} . \tag{2.32}$$

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

$$P_{\text{mass+coriol}} q = B^T \boldsymbol{M}_{(\text{mass+coriol})}^{-1} B \, q = \frac{1}{\Delta t} B^T \tilde{\boldsymbol{u}}. \tag{2.33}$$

We will see that (2.33) 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 $\boldsymbol{F}$ into the pressure diffusion operator. Namely, one can consider in (2.33)

$$\boldsymbol{M}_{(\text{diag+coriol})} = \begin{pmatrix} diag(A) & -2\omega \Delta t M_L & 0 \\ 2\omega \Delta t M_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{pmatrix} . \tag{2.34}$$

Below we discuss some important details of the modified projection step. First, note that the matrix $P_{\text{mass+coriol}}$ in (2.33) can be seen as a discretization of the following differential operator (see [72] 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 (use Proposition 2)

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

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 $\nabla \cdot (\nabla \times) \equiv 0$, this leads to $\nabla \cdot (\boldsymbol{w} \times \nabla q) = 0$. Therefore in the differential notation, equation (2.33) can be written as

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

Note that although the operator $\mathcal{M}$ is non-symmetric the resulting scalar 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 1. *For the discretization with the nonconforming Stokes finite element $\tilde{Q}_1/Q_0$ the matrix $P = B^T \boldsymbol{M}^{-1}_{(mass+coriol)} B$ is symmetric.*

Proof. Denote

$$P = \{p_{ij}\}, \ M_L = \{m_{ii}\}, \ B = (B_1, B_2, B_3)^T \text{ with } B_\mathcal{K} = \{b^\mathcal{K}_{ij}\}, \ s = 2\omega \Delta t \theta. \tag{2.35}$$

We need to prove that the matrix

$$P = \begin{pmatrix} B_1^T & B_2^T & B_3^T \end{pmatrix} \begin{pmatrix} M_L & -sM_L & 0 \\ sM_L & M_L & 0 \\ 0 & 0 & M_L \end{pmatrix}^{-1} \begin{pmatrix} B_1 \\ B_2 \\ B_3 \end{pmatrix} \tag{2.36}$$

is symmetric. Using notation (2.35) we get from (2.32)

$$p_{ij} = \sum_k \left(\frac{b^1_{ki} b^1_{kj}}{m_{kk}(1+s^2)} + \frac{b^1_{ki} b^2_{kj} s}{m_{kk}(1+s^2)} - \frac{b^2_{ki} b^1_{kj} s}{m_{kk}(1+s^2)} + \frac{b^2_{ki} b^2_{kj}}{m_{kk}(1+s^2)} + \frac{b^3_{ki} b^3_{kj}}{m_{kk}} \right). \tag{2.37}$$

It is obvious that equality

$$\sum_k \frac{b^1_{ki} b^2_{kj} s}{m_{kk}(1+s^2)} - \frac{b^2_{ki} b^1_{kj} s}{m_{kk}(1+s^2)} = \sum_k (b^1_{ki} b^2_{kj} - b^2_{ki} b^1_{kj}) \frac{s}{m_{kk}(1+s^2)} = 0 \tag{2.38}$$

would ensure that P is symmetric. Let us show that

$$b^1_{ki} b^2_{kj} - b^2_{ki} b^1_{kj} = 0, \qquad \forall i, j, k. \tag{2.39}$$

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., [33]). Denoting

$$\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3) \quad \text{with } \mathcal{B}_\mathcal{K} = \{d^\mathcal{K}_{ij}\} , \tag{2.40}$$

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

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

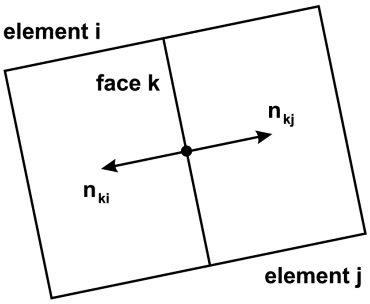

Fig. 2.2. Definition of entries $b^{\mathcal{K}}_{ki}$ and $b^{\mathcal{K}}_{kj}$.

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

$$\left(d^1_{ij},\, d^2_{ij},\, d^3_{ij}\right) = \int_{T_i} \nabla\cdot\boldsymbol{\phi}_j\psi_i dx = -\int_{T_i}\boldsymbol{\phi}_j\nabla\psi_i dx + \int_{\partial T_i}\boldsymbol{\phi}_j\cdot\boldsymbol{n}\psi_i d\sigma = \int_{\partial T_i}\boldsymbol{\phi}_j\cdot\boldsymbol{n}\psi_i d\sigma \tag{2.41}$$

and the entries of the gradient operator B

$$\left(b^1_{ij},\, b^2_{ij},\, b^3_{ij}\right)^T = \int_{\partial T_j}\boldsymbol{\phi}_i\cdot\boldsymbol{n}\psi_j d\sigma\ , \tag{2.42}$$

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 integral 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. 2.2). Then it holds $\boldsymbol{n}_{ki} = -\boldsymbol{n}_{kj}$ and (2.39) is satisfied. □

Remark 5. The proposition is true for any $P = B^T\boldsymbol{G}^{-1}B$, where $\boldsymbol{G}$ takes the form of (2.32) or (2.34). In particular it is valid for $\boldsymbol{G} = \boldsymbol{M}_{(\text{diag+coriol})}$ from (2.34).

2.3.3 Correction of velocity and pressure

Let us consider the last step of the DPM, i.e., equations (2.20), (2.21), and look for a necessary modification of velocity and pressure corrections. As an example we consider $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\text{mass+coriol})}$. Multiplying both sides of (2.21) by B^T and using (2.33) 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 (2.30) with $\alpha_R = 1$ and $\alpha_D = \Delta t\nu$, we obtain from (2.33) 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.

Remark 6. If $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{(\text{diag})}$, then the diffusion part is already included and one can set $\alpha_D = 0$.

2.3.4 Resulting algorithm

To conclude the discussions of this chapter we present the modified DPM algorithm (with p^n being the pressure from the previous time step), which reads as follows:

1. Solve for $\widetilde{\boldsymbol{u}}$ the equation
$$\boldsymbol{F}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^n \tag{2.43}$$
with a multigrid method with smoothing iterations involving the special preconditioner $\boldsymbol{C}$ described in (2.26).
2. Solve the discrete pressure problem
$$Pq = B^T \boldsymbol{M}_{(\cdot)}^{-1} B q = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}} \tag{2.44}$$
with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (2.32) or (2.34).
3. Calculate the pressure and the velocity approximations as
$$\begin{aligned} p &= p^n + q + \alpha M_p^{-1} B^T \widetilde{\boldsymbol{u}} \ , \\ \boldsymbol{u} &= \widetilde{\boldsymbol{u}} - \Delta t \boldsymbol{M}_{(\cdot)}^{-1} B q \end{aligned} \tag{2.45}$$
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}.

2.4 Summary of the modified DPM

In this chapter we constructed the modified discrete projection scheme (2.43)-(2.45) for the system of the incompressible Navier-Stokes equations with the Coriolis force term (1.3). Based on the incomplete ILU factorization for the matrix of the coupled linearized Navier-Stokes system and detailed evaluation of discrete operators, we proposed new explicitly inverted block-diagonal precondtioners for both momentum and projection steps. The modified DPM should guarantee better accuracy of velocity and pressure approximations and/or faster convergence to the steady state solution than the classical projection method especially for medium and large values of $\omega \ \Delta t$. Numerical experiments in chapter 5 will confirm this.
We would like to point that the constructed Pressure Schur Complement preconditioners can be used in a fully coupled way as well. In our work though we consider only the discrete projection approach.

3

Error analysis of the modified projection method

As it was proposed in the Chapter 2, to handle effectively the possibly dominating Coriolis force we modify the classical projection scheme [18, 101] in the following way: for given $\boldsymbol{u}^n \approx \boldsymbol{u}(t_n)$:
Step 1: Find intermediate velocity $\tilde{\boldsymbol{u}}^{n+1}$ from

$$\begin{cases} \dfrac{1}{k}(\tilde{\boldsymbol{u}}^{n+1} - \boldsymbol{u}^n) - \nu \Delta \tilde{\boldsymbol{u}}^{n+1} + (\boldsymbol{u}^n \cdot \nabla)\tilde{\boldsymbol{u}}^{n+1} + \boldsymbol{\omega} \times \tilde{\boldsymbol{u}}^{n+1} = \boldsymbol{f}(t_{n+1}) \\ \tilde{\boldsymbol{u}}^{n+1}|_{\Gamma} = 0 \end{cases} \tag{3.1}$$

Step 2: Find new velocity and pressure $\{\boldsymbol{u}^{n+1}, p^{n+1}\}$ as the result of the projection into the divergence-free subspace

$$\begin{cases} \dfrac{1}{k}(\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}) + \boldsymbol{\omega} \times (\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}) + \nabla p^{n+1} = 0 \\ \operatorname{div} \boldsymbol{u}^{n+1} = 0 \\ \boldsymbol{u}^{n+1} \cdot \boldsymbol{n}|_{\Gamma} = 0 \end{cases} \tag{3.2}$$

where k is the time step, $t_{n+1} = (n+1)k$, and $\boldsymbol{n}$ is the normal vector to Γ. One notes that the essential modification of the well-known Chorin-Temam method is introduced on the correction step 2, which is not an orthogonal projection anymore. In the present chapter we deal with the convergence analysis for the method (3.1)–(3.2).

A well established framework for numerical analysis of projection schemes is the following, see [84, 87]: one deduces an equivalent pseudo-compressibility or penalty method and further treats a projection scheme as the discretization of perturbed Navier-Stokes equations. However, applying this approach to (3.1)–(3.2) leads to a number of additional terms depending on $\boldsymbol{\omega}$, which are not easy to handle. Therefore we analyse the problem using the techniques developed by J. Shen in [94, 95] for the case of $\boldsymbol{\omega} = 0$. Although the arguments in [94, 95] essentially use the fact that the projection in step 2 is orthogonal, we show that similar convergence results can be proved for the modified method (3.1)–(3.2).

3.1 Preliminaries

Below we use the following notation:

$$|\cdot|^2 = \int_{\Omega} |\cdot|^2 dx, \quad \|\cdot\|^2 = \int_{\Omega} |\nabla \cdot|^2 dx, \quad \|\cdot\|_s \text{ – norm in } H^s(\Omega).$$

By $(\cdot,\cdot)$ we will denote the inner product in $L^2(\Omega)$ and by $\langle \cdot,\cdot \rangle$ – the duality between H^{-s} and H_0^{-s} for all $s > 0$.

We also define

$$H = \{\boldsymbol{u} \in \left(L^2(\Omega)\right)^d : \text{div } \boldsymbol{u} = 0, \, \boldsymbol{u} \cdot \boldsymbol{n}|_{\Gamma} = 0\},$$
$$V = \{\boldsymbol{v} \in \left(H_0^1(\Omega)\right)^d : \text{div } \boldsymbol{v} = 0\}.$$

In the following, we assume

$$\begin{cases} \boldsymbol{u}_0 \in (H^2(\Omega))^d \cap V, \\ \boldsymbol{f} \in L^\infty(0,T;(L^2(\Omega))^d) \cap L^2(0,T;(H^1(\Omega))^d), \\ \boldsymbol{f}_t \in L^2(0,T;H^{-1}), \\ \sup_{t\in[0,T]} \|\boldsymbol{u}(t)\| \le m_1. \end{cases} \tag{3.3}$$

We will use c or C as a generic positive constant which depends only on Ω, ν, T, and constants from various Sobolev inequalities. We will denote m or M as a generic positive constant which may additionally depend on $\boldsymbol{u}_0$, $\boldsymbol{f}$, $\boldsymbol{\omega}$ and the solution $\boldsymbol{u}$ through the constant m_1 in (3.3).

Under the assumption (3.3) one can prove the following inequalities

$$\sup_{t\in[0,T]} \{\|\boldsymbol{u}(t)\|_2 + |\boldsymbol{u}_t(t)| + |\nabla p(t)|\} \le M, \tag{3.4}$$

$$\int_0^T \|\boldsymbol{u}_t(t)\|^2 + t|\boldsymbol{u}_{tt}|^2 dt \le M, \tag{3.5}$$

which will be used in the sequel. Indeed, in [42] the estimates (3.4)–(3.5) were proved for the Navier-Stokes equations (1.1) without Coriolis term. However adding *linear skew-symmetric* term $\boldsymbol{\omega} \times \boldsymbol{u}$ to the momentum equation does not change arguments from [42],but leads to (3.4)–(3.5) with constant M depending, in general, on $\boldsymbol{\omega}$. Further we often use the following well-known [102] estimates for the bilinear form $b(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) = \int_\Omega (\boldsymbol{u}\cdot\nabla)\boldsymbol{v} \cdot \boldsymbol{w} \, \mathrm{d}\boldsymbol{x}$

$$b(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) \le \begin{cases} c\|\boldsymbol{u}\|\|\boldsymbol{v}\|^{\frac{1}{2}}|\boldsymbol{v}|^{\frac{1}{2}}\|\boldsymbol{w}\|, \\ c\|\boldsymbol{u}\|_2|\boldsymbol{v}|\|\boldsymbol{w}\|, \\ c\|\boldsymbol{u}\|\|\boldsymbol{v}\|_2|\boldsymbol{w}|. \end{cases} \tag{3.6}$$

and $b(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) = -b(\boldsymbol{u},\boldsymbol{w},\boldsymbol{v})$ for $\boldsymbol{u} \in H$.

Let P_H be the orthogonal projector in $\left(L^2(\Omega)\right)^d$ onto H and define the Stokes operator $\boldsymbol{A}\boldsymbol{u} = -P_H \Delta \boldsymbol{u}, \ \forall \boldsymbol{u} \in D(\boldsymbol{A}) = V \cap \left(H^2(\Omega)\right)^d$. We will use the following properties: $\boldsymbol{A}$ is an unbounded positive self-adjoint closed operator in H with domain $D(\boldsymbol{A})$, and its inverse $\boldsymbol{A}^{-1}$ is compact in H and satisfies the following relations [94, 95]:

$$\exists c, C > 0, \ \text{ such that } \ \forall \boldsymbol{u} \in H : \begin{cases} \|\boldsymbol{A}^{-1}\boldsymbol{u}\|_2 \le c|\boldsymbol{u}| \text{ and } \|\boldsymbol{A}^{-1}\boldsymbol{u}\| \le c\|\boldsymbol{u}\|_{V'}, \\ c\|\boldsymbol{u}\|_{V'}^2 \le (\boldsymbol{A}^{-1}\boldsymbol{u},\boldsymbol{u}) \le C\|\boldsymbol{u}\|_{V'}^2. \end{cases} \tag{3.7}$$

Further in this section we will prove several auxiliary lemmas. The first lemma shows that the projection (3.2) is uniformly (with respect to k) stable in H^1. Another two preliminary lemmas extend the results of lemma 2 from [94] and lemma A1 from [95] for the case of $\boldsymbol{\omega} \neq 0$ and non-orthogonal projection in (3.2). We also note that in [95] the result was proved only for the Stokes case (no nonlinear terms has been treated). Thus we include the nonlinear terms in the proof of lemma 4 and encounter additional assumption on the size of the time step.

Lemma 2. *The estimate*

$$\|\boldsymbol{u}^{n+1}\|_1 \le \tilde{m}\|\tilde{\boldsymbol{u}}^{n+1}\|_1$$

holds with some $\tilde{m}$ independent of $k \in (0,1]$.

Proof. First note that the pressure p^{n+1} from (3.2) satisfies the following elliptic equation

$$\operatorname{div} \mathcal{M}^{-1}\nabla p^{n+1} = \frac{1}{k}\operatorname{div}\tilde{\boldsymbol{u}}^{n+1} \tag{3.8}$$

$$\left[\mathcal{M}^{-1}\nabla p^{n+1}\right]\cdot \boldsymbol{n}|_{\Gamma} = 0 \tag{3.9}$$

with $\mathcal{M} = [I + k\boldsymbol{\omega}\times]$. One can verify [72] that for $d = 3$ it holds

$$\mathcal{M}^{-1} = (1 + |\tilde{\boldsymbol{\omega}}|^2)^{-1}[I + \tilde{\boldsymbol{\omega}}\otimes\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\omega}}\times], \quad \tilde{\boldsymbol{\omega}} = k\,\boldsymbol{\omega}, \tag{3.10}$$

where $(\tilde{\boldsymbol{\omega}}\otimes\tilde{\boldsymbol{\omega}})_{ij} = \tilde{\boldsymbol{\omega}}_i\,\tilde{\boldsymbol{\omega}}_j$. (For the 2D case the identity (3.10) holds without $\tilde{\boldsymbol{\omega}}\otimes\tilde{\boldsymbol{\omega}}$ term.) Since $\tilde{\boldsymbol{\omega}}$ is a constant vector one has $\tilde{\boldsymbol{\omega}}\times\nabla q = \nabla\times(q\tilde{\boldsymbol{\omega}})$ for a scalar function q. Therefore $\operatorname{div}(\tilde{\boldsymbol{\omega}}\times\nabla q) = 0$ and the equation (3.8) can be written as

$$\operatorname{div}\mathcal{B}\,\nabla p^{n+1} = \frac{1}{k}\operatorname{div}\tilde{\boldsymbol{u}}^{n+1} \tag{3.11}$$

with the *symmetric* diffusion tensor $\mathcal{B} = \dfrac{1}{1+|\tilde{\boldsymbol{\omega}}|^2}[I + \tilde{\boldsymbol{\omega}}\otimes\tilde{\boldsymbol{\omega}}]$. One can easily see that the inequalities

$$m_1|\xi|^2 \le (\mathcal{B}\xi,\xi) \le m_2|\xi|^2 \tag{3.12}$$

hold with m_1 and m_2 independent on k, e.g. $m_1 = \dfrac{1}{1+|\tilde{\boldsymbol{\omega}}|^2}$, $m_2 = 1$. (For the 2D case $\mathcal{B}$ is the scaled identity matrix.) Furthermore, the boundary condition (3.9) can be rewritten as

$$\left.\frac{\partial p^{n+1}}{\partial \boldsymbol{l}}\right|_{\Gamma} = 0 \qquad \text{with } \boldsymbol{l} = \mathcal{M}^{-1}\boldsymbol{n}.$$

The angle $\phi(\boldsymbol{x})$ between the vector $\boldsymbol{l}(\boldsymbol{x})$ and tangential plane to Γ at $\boldsymbol{x}\in\Gamma$ is uniformly bounded from below. Indeed, it holds:

$$|\sin\phi| = \frac{|\boldsymbol{l}^T\cdot\boldsymbol{n}|}{\boldsymbol{l}^T\cdot\boldsymbol{l}} = \frac{|\boldsymbol{n}^T\mathcal{M}^{-1}\boldsymbol{n}|}{\boldsymbol{n}^T\mathcal{M}^{-T}\mathcal{M}^{-1}\boldsymbol{n}} \ge \frac{|\boldsymbol{n}^T\mathcal{B}\boldsymbol{n}|}{\|\mathcal{M}^{-1}\|^2} \ge \frac{m_1}{4}. \tag{3.13}$$

Here we used the identity $\mathcal{M}^{-T} + \mathcal{M}^{-1} = 2\mathcal{B}$, inequalities (3.12) and $\|\mathcal{M}^{-1}\| \le 2$. Thus the smoothness assumption on Ω, (3.12) and (3.13) imply the following H^2 estimate for the solution of (3.8)–(3.9) [59]:

$$\|p^{n+1}\|_2 \le m\,k^{-1}|\operatorname{div}\tilde{\boldsymbol{u}}^{n+1}| \le m\,k^{-1}\|\tilde{\boldsymbol{u}}^{n+1}\|_1$$

with some constant c independent of k. Finally, using this result we get from (3.2) and the triangle inequality

$$\|\boldsymbol{u}^{n+1}\|_1 \le \|\tilde{\boldsymbol{u}}^{n+1}\|_1 + k\|\mathcal{M}^{-1}\nabla p^{n+1}\| \le \|\tilde{\boldsymbol{u}}^{n+1}\|_1 + k\|\mathcal{M}^{-1}\|\|p^{n+1}\|_2 \le m\,\|\tilde{\boldsymbol{u}}^{n+1}\|_1.$$

□

It is straightforward to check that the solution to (3.8)–(3.9) satisfies the estimate

$$|\mathcal{M}^{-1}\nabla p^{n+1}| \le m\,k^{-1}|\tilde{\boldsymbol{u}}^{n+1}|.$$

Thus the projection (3.2) is also uniformly stable in L^2:

$$|\boldsymbol{u}^{n+1}| \le |\tilde{\boldsymbol{u}}^{n+1}| + k|\mathcal{M}^{-1}\nabla p^{n+1}| \le m\,|\tilde{\boldsymbol{u}}^{n+1}|. \tag{3.14}$$

Lemma 3. *Denote*

$$e^{n+1} = \boldsymbol{u}(t_{n+1}) - \boldsymbol{u}^{n+1} \quad \textit{and} \quad \tilde{e}^{n+1} = \boldsymbol{u}(t_{n+1}) - \tilde{\boldsymbol{u}}^{n+1}.$$

Assume (3.3), $2k|\boldsymbol{\omega}|^2 \leq \tilde{m}$ *and* $2k^2|\boldsymbol{\omega}|^2 \leq \frac{1}{2}$. *It holds:*

$$|e^{N+1}|^2 + |\tilde{e}^{N+1}|^2 + k\nu \sum_{n=0}^{N} \{\|\tilde{e}^{n+1}\|^2 + \|e^{n+1}\|^2\} \\ + \sum_{n=0}^{N} \{|e^{n+1} - \tilde{e}^{n+1}|^2 + |\tilde{e}^{n+1} - e^n|^2\} \leq m\, k \qquad \forall\, 0 \leq N \leq T/k - 1 \quad (3.15)$$

Proof. Let R^n be the truncation error defined by

$$\frac{1}{k}(\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n)) - \nu \Delta \boldsymbol{u}(t_{n+1}) + \boldsymbol{\omega} \times \boldsymbol{u}(t_{n+1}) \\ + (\boldsymbol{u}(t_{n+1}) \cdot \nabla)\boldsymbol{u}(t_{n+1}) + \nabla p(t_{n+1}) = \boldsymbol{f}(t_{n+1}) + R^n, \quad (3.16)$$

where R^n is the integral residual of the Taylor series, i.e.,

$$R^n = \frac{1}{k} \int_{t_n}^{t_{n+1}} (t - t_n) \boldsymbol{u}_{tt}(t) dt.$$

By subtracting (3.1) from (3.16), we obtain

$$\frac{1}{k}(\tilde{e}^{n+1} - e^n) - \nu \Delta \tilde{e}^{n+1} + \boldsymbol{\omega} \times \tilde{e}^{n+1} \\ = (\boldsymbol{u}^n \cdot \nabla)\tilde{\boldsymbol{u}}^{n+1} - (\boldsymbol{u}(t_{n+1}) \cdot \nabla)\boldsymbol{u}(t_{n+1}) + R^n - \nabla p(t_{n+1}) \quad (3.17)$$

Taking the inner product of (3.17) with $2k\tilde{e}^{n+1}$ and using the identity

$$(a - b, 2a) = |a|^2 - |b|^2 + |a - b|^2,$$

we derive

$$|\tilde{e}^{n+1}|^2 - |e^n|^2 + |\tilde{e}^{n+1} - e^n|^2 + 2k\nu\|\tilde{e}^{n+1}\|^2 + (\boldsymbol{\omega} \times \tilde{e}^{n+1}, 2k\tilde{e}^{n+1}) \\ = 2k\langle R^n, \tilde{e}^{n+1}\rangle - 2k(\nabla p(t_{n+1}), \tilde{e}^{n+1}) - 2kb(e^n, \tilde{\boldsymbol{u}}^{n+1}, \tilde{e}^{n+1}) \\ + 2kb(\boldsymbol{u}(t_n) - \boldsymbol{u}(t_{n+1}), \tilde{\boldsymbol{u}}^{n+1}, \tilde{e}^{n+1}) - 2kb(\boldsymbol{u}(t_{n+1}), \tilde{e}^{n+1}, \tilde{e}^{n+1}). \quad (3.18)$$

Since the Coriolis term vanishes: $(\boldsymbol{\omega} \times \tilde{e}^{n+1}, 2k\tilde{e}^{n+1}) = 0$, using the same arguments as in [94] for treating other terms, see pages 64–65, and applying inequality (3) from [95] one deduces from (3.18) the estimate

$$|\tilde{e}^{n+1}|^2 - |e^n|^2 + |\tilde{e}^{n+1} - e^n|^2 + 2k\nu\|\tilde{e}^{n+1}\|^2 \\ \leq \bar{m}\, k \left(\int_{t_n}^{t_{n+1}} t\|\boldsymbol{u}_{tt}\|_{-1}^2 \mathrm{d}t + k \int_{t_n}^{t_{n+1}} |\boldsymbol{u}_t|^2 \mathrm{d}t \right) + 2k^2|\nabla p(t_{n+1)}|^2 + \bar{m}\, k|e^n|^2. \quad (3.19)$$

From (3.17) we have

$$\frac{1}{k}(e^{n+1} - \tilde{e}^{n+1}) - \nabla p^{n+1} + \boldsymbol{\omega} \times (e^{n+1} - \tilde{e}^{n+1}) = 0. \quad (3.20)$$

Taking the inner product of (3.20) with $2ke^{n+1}$, we get

$$|e^{n+1}|^2 - |\tilde{e}^{n+1}|^2 + |e^{n+1} - \tilde{e}^{n+1}|^2 - 2k(\boldsymbol{\omega} \times \tilde{e}^{n+1}, e^{n+1} - \tilde{e}^{n+1}) = 0.$$

Then

$$|e^{n+1}|^2 - |\tilde{e}^{n+1}|^2 + |e^{n+1} - \tilde{e}^{n+1}|^2 - 2k^2|\boldsymbol{\omega}|^2|\tilde{e}^{n+1}|^2 - \frac{1}{2}|e^{n+1} - \tilde{e}^{n+1}|^2 \leq 0,$$

$$|e^{n+1}|^2 - (1 + k\tilde{m})|\tilde{e}^{n+1}|^2 + \frac{1}{2}|e^{n+1} - \tilde{e}^{n+1}|^2 \leq 0, \tag{3.21}$$

with $\tilde{m} = 2k|\boldsymbol{\omega}|^2$. Inequality (3.21) yields

$$(1 + k\tilde{m})^{-1}|e^{n+1}|^2 - |\tilde{e}^{n+1}|^2 + \frac{1}{2(1 + k\tilde{m})}|e^{n+1} - \tilde{e}^{n+1}|^2 \leq 0. \tag{3.22}$$

Since $(1 - b) \leq (1 + b)^{-1}$ for $b \geq 0$, from (3.22) we derive

$$|e^{n+1}|^2 - |\tilde{e}^{n+1}|^2 + \frac{1}{2(1 + k\tilde{m})}|e^{n+1} - \tilde{e}^{n+1}|^2 \leq k\tilde{m}|e^{n+1}|^2. \tag{3.23}$$

Taking the sum of (3.19) and (3.23) for $n = 0, \ldots, N$ ($0 \leq N \leq T/k - 1$), we obtain

$$\begin{aligned} &|e^{N+1}|^2 + \sum_{n=0}^{N} \left\{ \frac{1}{2(1 + k\tilde{m})}|e^{n+1} - \tilde{e}^{n+1}|^2 + \frac{1}{2}|\tilde{e}^{n+1} - e^n|^2 + k\nu\|\tilde{e}^{n+1}\|^2 \right\} \\ &\qquad \leq \bar{m}\,k \sum_{n=0}^{N} |e^n|^2 + \bar{m}\,k \left(\int_0^T t\|\boldsymbol{u}_{tt}\|_{-1}^2 dt + k\int_0^T |\boldsymbol{u}_t|^2 dt + \sup_{t\in[0,T]} |\nabla p(t)|^2 \right) + \sum_{n=0}^{N} k\tilde{m}|e^{n+1}|^2. \end{aligned}$$

Denoting $m = \max\{\bar{m}, \tilde{m}\}$, we can rewrite the previous inequality as

$$\begin{aligned} &|e^{N+1}|^2 + \sum_{n=0}^{N} \left\{ \frac{1}{2(1 + k\tilde{m})}|e^{n+1} - \tilde{e}^{n+1}|^2 + \frac{1}{2}|\tilde{e}^{n+1} - e^n|^2 + k\nu\|\tilde{e}^{n+1}\|^2 \right\} \\ &\qquad \leq m\,k \sum_{n=0}^{N} |e^n|^2 + m\,k \left(\int_0^T t\|\boldsymbol{u}_{tt}\|_{-1}^2 dt + k\int_0^T |\boldsymbol{u}_t|^2 dt + \sup_{t\in[0,T]} |\nabla p(t)|^2 \right) + k\tilde{m}|e^{N+1}|^2. \end{aligned}$$

Thanks to the condition $k\tilde{m} \leq \frac{1}{2}$ and (3.4)–(3.5) one can write

$$\begin{aligned} &|e^{N+1}|^2 + \sum_{n=0}^{N} \left\{ |e^{n+1} - \tilde{e}^{n+1}|^2 + \frac{1}{2}|\tilde{e}^{n+1} - e^n|^2 + k\nu\|\tilde{e}^{n+1}\|^2 \right\} \\ &\qquad \leq m\,k \sum_{n=0}^{N} |e^n|^2 + m\,k \left(\int_0^T t\|\boldsymbol{u}_{tt}\|_{-1}^2 dt + k\int_0^T |\boldsymbol{u}_t|^2 dt + \sup_{t\in[0,T]} |\nabla p(t)|^2 \right) \\ &\qquad \leq m\,k \sum_{n=0}^{N} |e^n|^2 + m\,k. \end{aligned}$$

Applying the discrete Gronwall lemma to the last inequality, we arrive at

$$|e^{N+1}|^2 + \sum_{n=0}^{N} \left\{ |e^{n+1} - \tilde{e}^{n+1}|^2 + |\tilde{e}^{n+1} - e^n|^2 + k\nu\|\tilde{e}^{n+1}\|^2 \right\} \leq m\,k \tag{3.24}$$

Further, lemma 2 provides the estimate

$$\|e^{n+1}\|_1 \leq \tilde{m}\|\tilde{e}^{n+1}\|_1 \tag{3.25}$$

Applying (3.25) and the triangle inequality $|\tilde{e}^{n+1}| \leq |e^{n+1}| + |e^{n+1} - \tilde{e}^{n+1}|$ and (3.24), we also obtain

$$|\tilde{e}^{N+1}|^2 + k\nu \sum_{n=0}^{N} \|e^{n+1}\|^2 \leq m\, k$$

This proves the lemma. □

Lemma 4. *Assume* (3.3) *and*

$$\int_0^T |\nabla p_t|^2 \leq m\,. \tag{3.26}$$

Moreover, assume that k *is sufficiently small, then it holds*

$$\sum_{n=0}^{N} |\tilde{e}^{n+1} - \tilde{e}^n|^2 + k\|\tilde{e}^{N+1}\|^2 \leq m\, k^2 \qquad \forall\, 0 \leq N \leq T/k - 1.$$

Proof. We shift the index $n+1 \to n$ in (3.20) and take the sum with (3.17). This brings us to

$$\begin{aligned}\frac{1}{k}(\tilde{e}^{n+1} - \tilde{e}^n) - \nu\Delta\tilde{e}^{n+1} + \boldsymbol{\omega} \times (\tilde{e}^{n+1} - \tilde{e}^n)& \\ = (\boldsymbol{u}^n \cdot \nabla)\tilde{\boldsymbol{u}}^{n+1} - (\boldsymbol{u}(t_{n+1}) \cdot \nabla)\boldsymbol{u}(t_{n+1}) + R^n - \nabla(p(t_{n+1}) - p^n) - \boldsymbol{\omega} \times e^n&\end{aligned} \tag{3.27}$$

We take the inner product of (3.27) with $k(\tilde{e}^{n+1} - \tilde{e}^n)$ and obtain

$$\begin{aligned}|\tilde{e}^{n+1} - \tilde{e}^n|^2 + \frac{k\nu}{2}\left(\|\tilde{e}^{n+1}\|^2 - \|\tilde{e}^n\|^2 + \|\tilde{e}^{n+1} - \tilde{e}^n\|^2\right)& \\ = -k(\boldsymbol{\omega} \times e^n, \tilde{e}^{n+1} - \tilde{e}^n) + k\langle R^n, \tilde{e}^{n+1} - \tilde{e}^n\rangle + k(p(t_{n+1}) - p^n, \mathrm{div}(\tilde{e}^{n+1} - \tilde{e}^n)\,)& \\ + kb(\boldsymbol{u}^n, \tilde{\boldsymbol{u}}^{n+1}, \tilde{e}^{n+1} - \tilde{e}^n) - kb(\boldsymbol{u}(t_{n+1}), \boldsymbol{u}(t_{n+1}), \tilde{e}^{n+1} - \tilde{e}^n).&\end{aligned} \tag{3.28}$$

Now we estimate the terms on the right-hand side of (3.28). Below δ is a positive constant to be determined later. Using (3.15) we get

$$-k(\boldsymbol{\omega} \times e^n, \tilde{e}^{n+1} - \tilde{e}^n) \leq \delta|\tilde{e}^{n+1} - \tilde{e}^n|^2 + m\, k^2|e^n|^2 \leq \delta|\tilde{e}^{n+1} - \tilde{e}^n|^2 + m\, k^3. \tag{3.29}$$

Thanks to the estimate $|R^n|^2 \leq c\, \int_{t_n}^{t_{n+1}} t|\boldsymbol{u}_{tt}|^2 \mathrm{d}t$ from [95] we have

$$k\langle R^n, \tilde{e}^{n+1} - \tilde{e}^n\rangle \leq \delta|\tilde{e}^{n+1} - \tilde{e}^n|^2 + c\, k^2 \int_{t_n}^{t_{n+1}} t|\boldsymbol{u}_{tt}|^2 \mathrm{d}t. \tag{3.30}$$

Let us estimate the pressure-dependent term. Denote $q^n = p(t_{n+1}) - p^n$, since div $e^{n+1} = 0$ and div $e^n = 0$, we obtain

$$k(p(t_{n+1}) - p^n, \mathrm{div}(\tilde{e}^{n+1} - \tilde{e}^n)\,) = k(\nabla q^n, \tilde{e}^n - \tilde{e}^{n+1} - e^n + e^{n+1}). \tag{3.31}$$

Then we deduce from (3.20) and (3.31):

$$\begin{aligned}&k(p(t_{n+1}) - p^n, \mathrm{div}(\tilde{e}^{n+1} - \tilde{e}^n)\,) \\ &\quad = k^2(\nabla q^n, \nabla(p^{n+1} - p^n)) + k^2(\nabla q^n, \boldsymbol{\omega} \times (\tilde{e}^{n+1} - e^{n+1} - \tilde{e}^n + e^n)) \\ &\quad \leq -k^2(\nabla q^n, \nabla(q^{n+1} - q^n)) + k^2(\nabla q^n, \nabla(p(t_{n+2}) - p(t_{n+1}))) \\ &\qquad + k^2(\nabla q^n, \boldsymbol{\omega} \times (\tilde{e}^{n+1} - e^{n+1}) - k^2(\nabla q^n, \boldsymbol{\omega} \times (\tilde{e}^n - e^n))\end{aligned} \tag{3.32}$$

We estimate the terms on the right-hand side of (3.32) separately:

$$-k^2(\nabla q^n, \nabla(q^{n+1}-q^n)) = \frac{k^2}{2}(\|q^n\|^2 - \|q^{n+1}\|^2 + \|q^{n+1}-q^n\|^2) \tag{3.33}$$

We obtain from (3.20) the following relation:

$$k\mathcal{M}^{-1}\nabla(q^{n+1}-q^n) = (\tilde{e}^{n+1}-e^{n+1}) - (\tilde{e}^n - e^n) + k\mathcal{M}^{-1}\nabla(p(t_{n+2})-p(t_{n+1})).$$

Multiplying by $\nabla(q^{n+1}-q^n)$ and using (3.12) and condition $k|\boldsymbol{\omega}| \le \frac{1}{2}$ we get

$$\begin{aligned}
&k^2\|q^{n+1}-q^n\|^2 \le \frac{5}{4}k^2(\mathcal{M}^{-1}\nabla(q^{n+1}-q^n), \nabla(q^{n+1}-q^n))\\
&\le \frac{5}{4}k(\tilde{e}^{n+1}-\tilde{e}^n, \nabla(q^{n+1}-q^n)) + \frac{5}{4}k^2(\mathcal{M}^{-1}\nabla(p(t_{n+2})-p(t_{n+1}), \nabla(q^{n+1}-q^n))\\
&\le \frac{1}{2}k^2\|q^{n+1}-q^n\|^2 + \frac{5}{4}(\frac{5}{8}+\sigma)|\tilde{e}^{n+1}-\tilde{e}^n|^2 + m\,k^2\int_{t_{n+1}}^{t_{n+2}}|\nabla p_t|^2\mathrm{d}t, \quad \forall\,\sigma > 0.
\end{aligned}$$

Thus, choosing sufficiently small σ we obtain:

$$\frac{k^2}{2}\|q^{n+1}-q^n\|^2 \le \frac{5}{6}|\tilde{e}^{n+1}-\tilde{e}^n|^2 + m\,k^2\int_{t_{n+1}}^{t_{n+2}}|\nabla p_t|^2\mathrm{d}t \tag{3.34}$$

The second term on the right-hand side of (3.32) we estimate as follows:

$$k^2(\nabla q^n, \nabla(p(t_{n+2})-p(t_{n+1}))) \le k^3\|q^n\|^2 + m\,k^2\int_{t_{n+1}}^{t_{n+2}}|\nabla p_t|^2\mathrm{d}t \tag{3.35}$$

For the third and the fourth terms on the right-hand side of (3.32) we have:

$$k^2(\nabla q^n, \boldsymbol{\omega}\times(\tilde{e}^{n+1}-e^{n+1})) - k^2(\nabla q^n, \boldsymbol{\omega}\times(\tilde{e}^n-e^n)) \le k^3\|q^n\|^2 + m\,k\sum_{i=0,1}|\tilde{e}^{n+i}-e^{n+i}|^2 \tag{3.36}$$

Now estimates (3.32)–(3.36) give

$$\begin{aligned}
k(p(t_{n+1})-p^n, \operatorname{div}(\tilde{e}^{n+1}-\tilde{e}^n)\,) &\le \frac{5}{6}|\tilde{e}^{n+1}-\tilde{e}^n|^2 + m\,k^3\|q^n\|^2\\
&+ k^2(\|q^n\|^2 - \|q^{n+1}\|^2) + m\,k^2\int_{t_{n+1}}^{t_{n+2}}|\nabla p_t|^2\mathrm{d}t + m\,k\sum_{i=0,1}|\tilde{e}^{n+i}-e^{n+i}|^2. \qquad (3.37)
\end{aligned}$$

Further, consider the following splitting:

$$\begin{aligned}
\boldsymbol{u}(t_{n+1})\cdot\nabla\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}^n\cdot\nabla\tilde{\boldsymbol{u}}^{n+1} &= \boldsymbol{u}(t_{n+1})\cdot\nabla\tilde{e}^{n+1}\\
&+ (\boldsymbol{u}(t_{n+1})-\boldsymbol{u}(t_n))\cdot\nabla\tilde{\boldsymbol{u}}^{n+1} + e^n\cdot\nabla\boldsymbol{u}(t_{n+1}) - e^n\cdot\nabla\tilde{e}^{n+1} \qquad (3.38)
\end{aligned}$$

Based on this splitting we estimate the last two terms on the right-hand side of (3.28). The first three resulting terms can be estimated in a straightforward manner with the help of (3.6) and a priori estimates (3.4), (3.5):

$$\begin{aligned}
kb(\boldsymbol{u}(t_{n+1}), \tilde{e}^{n+1}, \tilde{e}^{n+1}-\tilde{e}^n) &\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + k^2m\,\|\boldsymbol{u}(t_{n+1})\|_2^2\|\tilde{e}^{n+1}\|^2\\
&\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + k^2m\,\|\tilde{e}^{n+1}\|^2, \qquad (3.39)\\
kb(\boldsymbol{u}(t_{n+1})-\boldsymbol{u}(t_n), \tilde{\boldsymbol{u}}^{n+1}, \tilde{e}^{n+1}-\tilde{e}^n) &\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + m\,k^3\|\tilde{\boldsymbol{u}}^{n+1}\|\int_{t_n}^{t_{n+1}}\|u_t\|_2^2\\
&\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + k^3m\,, \qquad (3.40)\\
kb(e^n, \boldsymbol{u}(t_{n+1}), \tilde{e}^{n+1}-\tilde{e}^n) &\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + k^2m\,\|\boldsymbol{u}(t_{n+1})\|_2^2\|e^n\|^2\\
&\le \delta|\tilde{e}^{n+1}-\tilde{e}^n|^2 + k^2m\,\|e^n\|^2. \qquad (3.41)
\end{aligned}$$

Due to (3.6) the last term from the splitting (3.38) is treated as follows:

$$\begin{aligned} kb(e^n, \tilde{e}^{n+1}, \tilde{e}^{n+1} - \tilde{e}^n) &\leq m\, k\|e^n\|\|\tilde{e}^{n+1}\|\|\tilde{e}^{n+1} - \tilde{e}^n\|^{\frac{1}{2}}|\tilde{e}^{n+1} - \tilde{e}^n|^{\frac{1}{2}} \\ &\leq m\, k^{\frac{3}{2}}\|e^n\|^2\|\tilde{e}^{n+1}\|^2 + \sqrt{k\nu\delta}\|\tilde{e}^{n+1} - \tilde{e}^n\||\tilde{e}^{n+1} - \tilde{e}^n| \\ &\leq m\, k^{\frac{3}{2}}\|e^n\|^2\|\tilde{e}^{n+1}\|^2 + \frac{k\nu}{2}\|\tilde{e}^{n+1} - \tilde{e}^n\|^2 + \delta|\tilde{e}^{n+1} - \tilde{e}^n|^2 \end{aligned} \tag{3.42}$$

Finally (3.28) with (3.29)–(3.30) and (3.37)–(3.43) yield for sufficiently small $\delta > 0$:

$$\begin{aligned} &|\tilde{e}^{n+1} - \tilde{e}^n|^2 + \frac{k\nu}{2}(\|\tilde{e}^{n+1}\|^2 - \|\tilde{e}^n\|^2) + k^2(\|q^{n+1}\|^2 - \|q^n\|^2) \\ &\qquad \leq M\Big(k^3 + k^2\int_{t_{n+1}}^{t_{n+2}} |\nabla p_t|^2 \mathrm{d}t + k^2(\|\tilde{e}^{n+1}\|^2 + \|e^{n+1}\|^2) + k^{\frac{3}{2}}\|e^n\|^2\|\tilde{e}^{n+1}\|^2 \\ &\qquad\qquad + k^3\|q^n\|^2 + k\sum_{i=0,1}|\tilde{e}^{n+i} - e^{n+i}|^2\Big). \end{aligned} \tag{3.43}$$

We sum up the last inequalities for $n = 0, \dots, N$ and use the assumption (3.26) and the estimate (3.15). This gives

$$\sum_{n=0}^{N}|\tilde{e}^{n+1} - \tilde{e}^n|^2 + k^2\|q^{N+1}\|^2 + \frac{k\nu}{2}\|\tilde{e}^{N+1}\|^2 \leq M\Big(k^2 + \sum_{n=0}^{N}k^3\|q^n\|^2 + \sum_{n=0}^{N}k^{\frac{3}{2}}\|e^n\|^2\|\tilde{e}^{n+1}\|^2\Big).$$

This is equivalent to

$$\begin{aligned} &\sum_{n=0}^{N}|\tilde{e}^{n+1} - \tilde{e}^n|^2 + k^2\|q^{N+1}\|^2 + (\frac{k\nu}{2} - Ck^{\frac{3}{2}}\|e^N\|^2)\|\tilde{e}^{N+1}\|^2 \\ &\qquad\qquad \leq M\Big(k^2 + \sum_{n=0}^{N}k^3\|q^n\|^2 + \sum_{n=1}^{N}k^{\frac{3}{2}}\|e^{n-1}\|^2\|\tilde{e}^n\|^2\Big). \end{aligned}$$

To apply the Gronwall lemma we need $(\frac{k\nu}{2} - Ck^{\frac{3}{2}}\|e^N\|^2) \geq ck\nu$. Thus we have to assume that k is sufficiently small such that $\sqrt{k}\|e^N\|^2\nu^{-1} < C$ holds. This is true for a sufficiently small k since $\|e^N\|$ is uniformly bounded due to lemma 3. We believe that the implied restriction $\sqrt{k}\nu^{-1} < c$ can be pessimistic in practice. In particular estimate (3.15) suggests $\|e^N\|^2 \ll 1$. Now the discrete Gronwall inequality and the bound for $\sum_{n=0}^{N}\|e^n\|^2$ from (3.15) yields

$$\sum_{n=0}^{N}|\tilde{e}^{n+1} - \tilde{e}^n|^2 + \frac{k\nu}{2}\|\tilde{e}^{N+1}\|^2 \leq m\, k^2\, \exp(\sqrt{k}\sum_{n=0}^{N}\|e^n\|^2) \leq m\, k^2\, \exp(\sqrt{k}M)$$

□

Thanks to the embedding $H^{-1} \hookrightarrow L^2$ and the L^2 stability of projection, see (3.14), we conclude:

$$\|e^{n+1} - e^n\|_{-1} \leq m\, |e^{n+1} - e^n| \leq m\, |\tilde{e}^{n+1} - \tilde{e}^n|.$$

Therefore the lemma 4 yields

$$\sum_{n=0}^{N}\|e^{n+1} - e^n\|_{-1}^2 \leq m\, k^2 \qquad \forall\, 0 \leq N \leq T/k - 1. \tag{3.44}$$

3.2 Error estimate

In this section we show that the scheme (3.1)–(3.2) for the Navier-Stokes equations with the Coriolis force (1.2) has the same order of accuracy as the classical projection scheme [18, 101] for the Navier-Stokes equations (1.1). The following theorem is the main result of this chapter.

Theorem 1. *Assume* (3.3) *and* $2k|\boldsymbol{\omega}|^2 \leq 1$, *then both* $\tilde{\boldsymbol{u}}^{n+1}$ *and* $\boldsymbol{u}^{n+1}$ *are weakly first-order approximations to* $\boldsymbol{u}(t_{n+1})$ *in* $L^2(\Omega)^d$*:*

$$k\nu \sum_{n=0}^{T/k-1} \left\{|e^{n+1}|^2 + |\tilde{e}^{n+1}|^2\right\} \leq m\, k^2 \tag{3.45}$$

Additionally assume that k *is sufficiently small and* $\int_0^T |\nabla p_t|^2 \leq m$, *then* p^{n+1} *as well as* $(I - k\,\nu\Delta)p^{n+1}$ *are weakly order* $\frac{1}{2}$ *approximations to* $p(t_{n+1})$ *in* $L^2(\Omega)/R$*:*

$$k \sum_{n=0}^{T/k-1} \left\{|p^{n+1} - p(t_{n+1})|^2_{L^2(\Omega)/R} + |(I - k\,\nu\Delta)p^{n+1} - p(t_{n+1})|^2_{L^2(\Omega)/R}\right\} \leq m\, k \tag{3.46}$$

Proof. (i) *Error estimate for the velocity.*
Taking the sum of (3.1) and (3.2), we obtain

$$\frac{1}{k}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) - \nu\Delta\tilde{\boldsymbol{u}}^{n+1} + (\boldsymbol{u}^n \cdot \nabla)\tilde{\boldsymbol{u}}^{n+1} + \boldsymbol{\omega} \times \boldsymbol{u}^{n+1} + \nabla p^{n+1} = \boldsymbol{f}(t_{n+1}). \tag{3.47}$$

Let us denote

$$\tilde{q}^{n+1} = p(t_{n+1}) - p^{n+1}.$$

Subtracting (3.47) from (3.16), we obtain

$$\begin{aligned}\frac{1}{k}(e^{n+1} - e^n) - \nu\Delta\tilde{e}^{n+1} + \boldsymbol{\omega} \times e^{n+1} + \nabla\tilde{q}^{n+1}&\\ = (\boldsymbol{u}^n \cdot \nabla)\tilde{\boldsymbol{u}}^{n+1} - (\boldsymbol{u}(t_{n+1}) \cdot \nabla)\boldsymbol{u}(t_{n+1}) + R^n.&\end{aligned} \tag{3.48}$$

Taking the inner product of (3.48) with $2\,k\,\boldsymbol{A}^{-1}e^{n+1}$, splitting the nonlinear term into three parts, using (3.7) and noticing that

$$(\boldsymbol{A}^{-1}\boldsymbol{u}, \nabla p) = 0, \quad \forall \boldsymbol{u} \in H,$$

we derive (for details see [94] p. 67)

$$\begin{aligned}\|e^{n+1}\|^2_{V'} - \|e^n\|^2_{V'} + \|e^{n+1} - e^n\|^2_{V'} + \frac{15k\nu}{8}|e^{n+1}|^2&\\ \leq -2k(\boldsymbol{\omega} \times e^{n+1}, \boldsymbol{A}^{-1}e^{n+1}) + 2k\langle R^n, \boldsymbol{A}^{-1}e^{n+1}\rangle - 2kb(e^n, \tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{A}^{-1}e^{n+1})&\\ - 2kb(\boldsymbol{u}(t_{n+1}), \tilde{e}^{n+1}, \boldsymbol{A}^{-1}e^{n+1}) + 2kb(\boldsymbol{u}(t_n) - \boldsymbol{u}(t_{n+1}), \tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{A}^{-1}e^{n+1})&\\ + m\,k|e^{n+1} - \tilde{e}^{n+1}|^2.&\end{aligned} \tag{3.49}$$

The Coriolis term is estimated as follows

$$\begin{aligned}|2k(\boldsymbol{\omega} \times e^{n+1}, \boldsymbol{A}^{-1}e^{n+1})| &\leq m\,k\|\boldsymbol{A}^{-1}e^{n+1}\||e^{n+1}|\\ &\leq m\,k\|e^{n+1}\|_{V'}|e^{n+1}| \leq \frac{\nu k}{8}|e^{n+1}|^2 + m\,k\|e^{n+1}\|^2_{V'}.\end{aligned} \tag{3.50}$$

Applying the same arguments as in [94] we deduce from (3.49) and (3.50) the estimate

$$\begin{aligned}\|e^{n+1}\|_{V'}^2 - \|e^n\|_{V'}^2 + \nu k|e^{n+1}|^2 + \|e^{n+1}-e^n\|_{V'}^2 \\ \le m\,k\|e^{n+1}\|_{V'}^2 + m\,(k^2+k^3)\|\tilde{e}^{n+1}\|^2 + m\,k|\tilde{e}^{n+1}-e^n| \\ + m\,k|e^{n+1}-\tilde{e}^{n+1}|^2 + m\,k\left(\int_{t_n}^{t_{n+1}} t\|\boldsymbol{u}_{tt}\|_{-1}^2\mathrm{d}t + k\int_{t_n}^{t_{n+1}} |\boldsymbol{u}_t|^2\mathrm{d}t\right).\end{aligned} \tag{3.51}$$

The only modification of the arguments from [94] here is that instead of identity

$$|\tilde{e}^{n+1}|^2 = |e^{n+1}|^2 + |e^{n+1}-\tilde{e}^{n+1}|^2,$$

which is no longer true we use the triangle inequality

$$|\tilde{e}^{n+1}|^2 \le |e^{n+1}|^2 + |e^{n+1}-\tilde{e}^{n+1}|^2, \tag{3.52}$$

Taking the sum of (3.51) for $n = 0,\ldots,N$, $N \in [0, T/k-1]$, we derive from lemma 3 that

$$\|e^{N+1}\|_{V'}^2 + \sum_{n=0}^{N}\{\|e^{n+1}-e^n\|_{V'}^2 + k\,\nu|e^{n+1}|^2\} \le m\,k^2 + m\,k\sum_{n=0}^{N+1}\|e^n\|_{V'}^2.$$

By applying the discrete Gronwall lemma to the last inequality, we obtain

$$\|e^{N+1}\|_{V'}^2 + \sum_{n=0}^{N}\{\|e^{n+1}-e^n\|_{V'}^2 + k\,\nu|e^{n+1}|^2\} \le m\,k^2 \quad \forall 0 \le N \le T/k-1.$$

Then, from (3.52) and lemma 3 we arrive at

$$k\sum_{n=0}^{N}|\tilde{e}^{n+1}|^2 \le k\sum_{n=0}^{N}\{|e^{n+1}|^2 + |\tilde{e}^{n+1}-e^{n+1}|^2\} \le m\,k^2 \quad \forall 0 \le N \le T/k-1. \tag{3.53}$$

(ii) *Error estimate for the pressure.*
The skeleton of our derivations for the pressure estimate remains the same as in [94]. Remarks from [95] are applied through lemma 4.

We start from rearranging (3.48) to

$$\begin{aligned}\nabla q_*^{n+1} = \frac{1}{k}(e^{n+1}-e^n) - \nu\Delta e_*^{n+1} + \boldsymbol{\omega}\times e^{n+1} \\ + (\boldsymbol{u}(t_{n+1})\cdot\nabla)\boldsymbol{u}(t_{n+1}) - (\boldsymbol{u}(t_n)\cdot\nabla)\tilde{\boldsymbol{u}}(t_{n+1}) - R^n,\end{aligned} \tag{3.54}$$

where $\{e_*^{n+1}, q_*^{n+1}\} = \{\tilde{e}^{n+1}, \tilde{q}^{n+1}\}$. The same relation holds for $\{e_*^{n+1}, q_*^{n+1}\} = \{e^{n+1}, q^{n+1}\}$, where $q^{n+1} = p(t_{n+1}) - (I - k\nu\nabla)p^{n+1}$. Hence we can consider simultaneously the two pressure approximations.
We split the nonlinear term on the right hand side of (3.54) as

$$\begin{aligned}(\boldsymbol{u}(t_{n+1})\cdot\nabla)\boldsymbol{u}(t_{n+1}) - (\boldsymbol{u}^n\cdot\nabla)\tilde{\boldsymbol{u}}^{n+1} \\ = ((\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n))\cdot\nabla)\boldsymbol{u}(t_{n+1}) + (e^n\cdot\nabla)\boldsymbol{u}(t_{n+1}) + (\boldsymbol{u}^n\cdot\nabla)\tilde{e}^{n+1}.\end{aligned}$$

From lemma 3 we derive that

$$\|\boldsymbol{u}^n\| \le \|e^n\| + \|\boldsymbol{u}(t_n)\| \le m \quad \forall n.$$

By using (3.6) we obtain that, for all $\boldsymbol{v} \in H_0^1(\Omega)^d$,

$$\begin{aligned}((\boldsymbol{u}(t_{n+1})\cdot\nabla)\boldsymbol{u}(t_{n+1}) - (\boldsymbol{u}^n\cdot\nabla)\tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{v})&\\ \le c\,|\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n)|\|\boldsymbol{u}(t_{n+1})\|_2\|\boldsymbol{v}\|&\\ + c\|e^n\|\|\boldsymbol{u}(t_{n+1})\|\|v\| + c\|\boldsymbol{u}^n\|\|\tilde{e}^{n+1}\|\|\boldsymbol{v}\|&\\ \le \bar{c}\{\|\tilde{e}^{n+1}\| + \|e^n\| + |\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n)|\}\|\boldsymbol{v}\|.&\end{aligned} \tag{3.55}$$

Using the Schwarz inequality we have also, for all $\boldsymbol{v}\in H_0^1(\Omega)^d$,

$$\begin{aligned}&\left(\frac{1}{k}(e^{n+1} - e^n) - \nu\Delta e_*^{n+1} + \boldsymbol{\omega}\times e^{n+1} - R^n, \boldsymbol{v}\right) \le\\ &\left(\frac{1}{k}\|e^{n+1} - e^n\|_{-1} + \nu\|e_*^{n+1}\| + \tilde{m}\,\|e^{n+1}\| + \|R^n\|_{-1}\right)\|\boldsymbol{v}\|.\end{aligned} \tag{3.56}$$

From the inequalities (3.54), (3.55), (3.56) and

$$|p|_{L_2(\Omega)/R} \le \hat{c}\sup_{\boldsymbol{v}\in H_0^1(\Omega)^d}\frac{(\nabla p, \boldsymbol{v})}{\|\boldsymbol{v}\|},$$

we obtain that

$$\begin{aligned}|q_*^{n+1}|_{L^2(\Omega)/R} \le \hat{c}\sup_{v\in H_0^1(\Omega)^d}\frac{(\nabla q_*^{n+1}, v)}{\|v\|} \le \frac{m}{k}\|e^{n+1} - e^n\|_{-1}&\\ + m\,(\|R^n\|_{-1} + \|\tilde{e}^{n+1}\| + \|e^n\| + (1+\tilde{m})\|e^{n+1}\| + |\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n)|).&\end{aligned}$$

Therefore, applying lemmas 3 and 4, and the inequality (3.53), we derive

$$\begin{aligned}k\sum_{n=0}^{T/k-1}|q_*^{n+1}|^2_{L^2(\Omega)/R} \le m\,k\sum_{n=0}^{T/k-1}\{\,\|\tilde{e}^{n+1}\|^2 + (1+\tilde{m})\|e^{n+1}\|^2&\\ + \|R^n\|^2_{-1} + |\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n)|^2\}&\\ + \frac{1}{k}\sum_{n=0}^{T/k-1}\|e^{n+1} - e^n\|^2_{-1} \le m\,k.&\end{aligned}$$

The proof of theorem 1 is complete. □

Remark 7. It was discussed in [42] that the assumption $\int_0^T|\nabla p_t|^2 \le m$, which we need to prove pressure error estimate does not hold for general flows, but requires a compatibility condition on given data, cf. [42]. The sufficient assumption for this condition to be valid is $\boldsymbol{f}(\boldsymbol{x},t)|_{t=0} = \boldsymbol{0}$.

4

Treatment of convective term

4.1 DPM framework for rotational form of convection

In the chapters 2 and 3 we considered the system of Stokes equations with the Coriolis force term. However, while performing numerical calculations for the system of Navier-Stokes equations with the Coriolis force especially for medium and high Reynolds numbers, one has to take into account the convective term as well. To prevent numerical oscillations we use the algebraic flux correction scheme of TVD type [56] or the edge-oriented stabilization [107, 81] for the discretization of convection terms. Moreover, adding such stabilization techniques makes multigrid solvers 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 in $\S$ 2.2.2, this issue can be related to the question of building effective pressure Schur complement preconditioners for the case of dominating convection terms. This tough question attracted a lot of considerations during the last decade, see an overview in [30] and [76]. 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 the previous chapters 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:

$$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla p = \boldsymbol{w}(\boldsymbol{u})\times\boldsymbol{u} + \nabla P \tag{4.1}$$

with $\boldsymbol{w}(\boldsymbol{u}) = \nabla\times\boldsymbol{u} + 2\boldsymbol{\omega}$ and $P = p + \frac{\boldsymbol{u}^2}{2}$.
$(\nabla\times\boldsymbol{u})$ is a vorticity function. For two dimensions we define

$$\nabla\times\boldsymbol{u} = -\frac{\partial u_1}{x_2} + \frac{\partial u_2}{x_1}$$

and

$$\alpha\times\boldsymbol{u} = -\boldsymbol{u}\times\alpha = \begin{cases} -\alpha u_2 \\ \alpha u_1 \end{cases} .$$

Therefore, 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 the Chapter 2, see [72] for the analysis of similar approach in the context of the Schur complement preconditioners for the linearized Navier-Stokes problem.

For ’rotating’ flows in the system of Navier-Stokes equations we can treat convection and rotating forces either as the right or the left part of (4.1). While both treatments are equal on the continuous level and conservative (see [72]), 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 $(\nabla \times \boldsymbol{u}) \times \boldsymbol{u}$. Although very popular in fluid mechanics, the rotational form did not find much attention among numerical analysts until the papers of Olshanskii and coauthors [65, 72, 73, 74], see also recent papers [62, 89]. Nowadays rotational form of convection still deserves more careful analysis.

Many algebraic splitting methods can be interpreted in terms of approximate block factorizations [17, 63, 85, 90, 112, 113]. These splittings naturally lead to preconditioners for Krylov methods both for steady and unsteady problems ([30], Chapter 8.3.5). Though these preconditioners were designed for the standard form of convection, some of them can be extended to the rotation form, e.g. Elman’s so-called $B\,F\,B^T$ preconditioner [28]. Benzi and Lia also proposed an approach for treating rotation form of convection - *Hermitian/skew-Hermitian splitting* (HSS) [9, 64]. HSS preconditioners guarantee robust behaviour of the algorithm, resp. quality of approximation, as the viscosity ν approaches zero.

The positive sides of the rotation form of the convective operator are:

- Treatment of a zero-order operator (rotational convection) instead of a first-order operator (standard convection) in linearized equations.
- Natural inclusion of the Coriolis force or any other reactive term which can be written in the cross-product form.
- Construction of discrete projection methods with explicitly invertible matrices of block-diagonal preconditioners.

The discrete projection approach based on the rotation form of convection term from (4.1) is a topic of the next section.

4.1.1 General-purpose version of the modified DPM

Let us take a closer look at the operator $\boldsymbol{w}(\boldsymbol{u})\times$:

- 2D case

$$\nabla \times \boldsymbol{u}\times = \boldsymbol{w}(\boldsymbol{u})\times = w\times = \begin{pmatrix} 0 & -w \\ w & 0 \end{pmatrix}, \tag{4.2}$$

- 3D case

$$\nabla \times \boldsymbol{u}\times = \boldsymbol{w}(\boldsymbol{u})\times = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \times = \begin{pmatrix} \dfrac{\partial u^3}{x_2} - \dfrac{\partial u^2}{x_3} \\ \dfrac{\partial u^1}{x_3} - \dfrac{\partial u^3}{x_1} \\ \dfrac{\partial u^2}{x_1} - \dfrac{\partial u^1}{x_2} \end{pmatrix} \times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}. \tag{4.3}$$

As we will see below, the main properties of the two-dimensional operator $\boldsymbol{w}(\boldsymbol{u})\times$ are the same as those of the three-dimensional one. All formulas below are presented for the 3D case. Following the same idea, the interested reader can easily derive them for the two dimensional case.

From (4.3) we obtain the following properties:

1. $\boldsymbol{w}(\boldsymbol{u})\times$ is skew-symmetric.
2. $\boldsymbol{w}(\boldsymbol{u})\times$ is off-diagonal dominant.
3. $det|\boldsymbol{w}(\boldsymbol{u})\times| = 0$, i.e. it is singular.
4. $(\boldsymbol{w}(\boldsymbol{u})\times \boldsymbol{u}, \boldsymbol{u})_{L_2} = 0$, i.e. $Im(\boldsymbol{w}(\boldsymbol{u})\times) \perp \boldsymbol{u}$

The resulting matrix $\boldsymbol{F}$ is of the form:

$$\begin{aligned}\boldsymbol{F} = (M + \nu L + \boldsymbol{w}(\boldsymbol{u})\times) &= \begin{pmatrix} M+\nu L & 0 & 0 \\ 0 & M+\nu L & 0 \\ 0 & 0 & M+\nu L \end{pmatrix} + \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} M+\nu L & -w_3 & w_2 \\ w_3 & M+\nu L & -w_1 \\ -w_2 & w_1 & M+\nu L \end{pmatrix}, \end{aligned} \tag{4.4}$$

where M is a consistent mass matrix and L is a discrete Laplacian operator. When flow is laminar, i.e. $M + \nu L$ is large w.r.t. w_i, operator $\boldsymbol{F}$ is diagonal dominant and therefore is 'easy' for the solver. When Re increases, values of w_i also increase (rotational motion of vortices becomes more influential on the whole flow), whereas $M + \nu L$ stays unchangeable. Therefore, as soon as $\boldsymbol{F}$ becomes more off-diagonal dominant/oriented, its conditional number increases and $det|\boldsymbol{F}|$ approaches zero (again, if measured with respect to values of $M + \nu L$ and w_i).

Approximating w_i, for $i = 1, 2, 3$, on the discrete level by their diagonal counterparts (as we did for the Coriolis force term in the previous chapters), we obtain a 3 by 3 operator every entry of which is occupied by a diagonal matrix. Thus, the obtained global matrix is easy to invert and therefore is easy to use as a preconditioner (see [72] for a numerical analysis). We will utilize this idea to construct a discrete projection method for the system of Navier-Stokes equations with convection to be written in the rotational form. Namely, proceeding from (4.4) we introduce the Schur Complement operator $\mathcal{S}$ in the following form:

$$\mathcal{S} = B^T \boldsymbol{M}^{-1}_{\text{(laplace+rotation)}} B \tag{4.5}$$

with

$$\boldsymbol{M}_{\text{(laplace+rotation)}} = \begin{pmatrix} M+\nu L & -w_3 & w_2 \\ w_3 & M+\nu L & -w_1 \\ -w_2 & w_1 & M+\nu L \end{pmatrix}. \tag{4.6}$$

The preconditioner P of $\mathcal{S}$ can be constructed by 'lumping' or taking only the main diagonal of diagonal $M+\nu L \rightarrow diag(M + \nu L)$ and off-diagonal $w_i \rightarrow \widetilde{w}_i$ block-matrices to obtain the following (3×3) block-diagonal matrix:

$$\widetilde{\boldsymbol{M}}_{\text{(laplace+rotation)}} = \begin{pmatrix} diag(M+\nu L) & -\widetilde{w}_3 & \widetilde{w}_2 \\ \widetilde{w}_3 & diag(M+\nu L) & -\widetilde{w}_1 \\ -\widetilde{w}_2 & \widetilde{w}_1 & diag(M+\nu L) \end{pmatrix}. \tag{4.7}$$

Thus, we have $P = B^T \widetilde{\boldsymbol{M}}^{-1}_{\text{(laplace+rotation)}} B$. The matrix $\widetilde{\boldsymbol{M}}_{\text{(laplace+rotation)}}$ is easy to invert. Its inverse can be used as a preconditioner for $\boldsymbol{F}$ while solving the velocity equation or constructing the modified Schur Complement operator

$P_{\text{(laplace+rotation)}} = B^T \widetilde{\boldsymbol{M}}^{-1}_{\text{(laplace+rotation)}} B.$

Now we are ready to write the discrete projection method for the incompressible system of the unsteady Navier-Stokes equations with convection written in the rotational form:

1. Solve for $\widetilde{\boldsymbol{u}}$ the ”viscous Burgers” equation
$$\boldsymbol{F}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^n \tag{4.8}$$
with $\boldsymbol{C} = \widetilde{\boldsymbol{M}}_{\text{(laplace+rotation)}}$ as a special preconditioner of $\boldsymbol{F}$.
2. Solve the discrete ”Pressure Poisson-like” problem
$$\mathcal{S}q = B^T \boldsymbol{M}^{-1}_{\text{(laplace+rotation)}} Bq = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}} \tag{4.9}$$
with $P = B^T \widetilde{\boldsymbol{M}}^{-1}_{\text{(laplace+rotation)}} B$ as a special preconditioner of $\mathcal{S}$.
3. Calculate the pressure and the velocity approximations as
$$\begin{aligned} p &= p^n + q + \frac{\alpha_D}{\Delta t} M_p^{-1} B^T \widetilde{\boldsymbol{u}} \\ \boldsymbol{u} &= \widetilde{\boldsymbol{u}} - \Delta t \boldsymbol{M}^{-1}_{\text{(laplace+rotation)}} Bq \end{aligned} \tag{4.10}$$
with $\alpha_D = 0$ or $\alpha_D = \nu \Delta t$. 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}.

Anticipated drawbacks of the above DPM in comparison to those with the standard convection are following:

- The memory costs will increase due to the full coupling of equations in the velocity block, see (4.4).
- The stabilization technique for the term $(\nabla \times \boldsymbol{u}) \times \boldsymbol{u}$ is not clear.

To overcome the second drawback, which may occur for numerical simulations at medium and large Reynolds numbers, one has to think about ’recovering’ the diagonal dominance of the resulting matrix $\boldsymbol{F}$. This can be done, e.g., by scaling the main diagonal matrix M by some parameter κ, where M is a consistent mass matrix. Since
$$M \sim h^2$$
we obtain
$$\Delta t w_h(\boldsymbol{u}_h) \sim \Delta t(|\nabla \times u| + |\boldsymbol{\omega}|)h^2.$$
Therefore, we can choose
$$\kappa \sim \Delta t(|\nabla \times u| + |\boldsymbol{\omega}|) + 1.$$

In practice we will not do it this way. Instead, we note that the edge-oriented stabilization is built in some sense on the same principle of increasing the main diagonal. Therefore we can apply it to stabilize the rotational term of convection. The edge-oriented technique will be described later in this chapter. Its influences on the solving process (4.8)–(4.10) and on the numerical solution will be discussed in chapter 6.

4.1.2 Summary of the DMP framework for rotation form of convection

In this section we introduced the generalized cross-product operator $\boldsymbol{w}(\boldsymbol{\omega}, \boldsymbol{u}, \cdot) \times \boldsymbol{u}$, which takes into account convection (in its rotational skew-symmetric form), Coriolis force and/or any cross-product-like terms. The operator

$\boldsymbol{w}(\boldsymbol{\omega}, \boldsymbol{u}, \cdot) \times \boldsymbol{u}$ made it possible to extend the modified discrete scheme (2.43)-(2.45) to a more general case. In the following we will show that the non-diagonal dominance nature of $\boldsymbol{w}(\boldsymbol{\omega}, \boldsymbol{u}, \cdot) \times \boldsymbol{u}$ may lead to poor convergence and bad accuracy. Nevertheless, by applying additional techniques one can obtain sufficiently good solutions. In chapter 6 we will present numerical experiments for the lid-driven cavity and flow around cyclinder benchmark problems and discuss the obtained results with respect to their accuracy and efficiency.

4.2 Algebraic Flux Correction

While treating convection dominated flows, one has to take into account the possible growth of nonphysical oscillations due to the domination of the convective term. Until nowadays this problem is not completely solved. Nevertheless, numerous efforts of CFD specialists and numerical analysists gave rise to many robust and efficient methods, which allow to calculate sufficient solutions. As we have already mentioned in the section § 4.1, among these methods are upwind and streamline-diffusion schemes, edge-oriented stabilization, algebraic flux correction, etc. In our realisation we utilize two of them, namely: edge-oriented stabilization and Algebraic Flux Correction of TVD type techniques. In this section we will give a brief theoretical overview on the Algebraic Flux Correction. The general idea beyond the algorithm is to modify a discrete system of the Navier-Stokes equations in some specific way as to satisfy conditions, which preclude appearance of non-physical oscillations and at the same time to recover the high-order approximation of the velocity vector field. The algebraic flux correction methodology was motivated by the pioneering article of Boris and Book [11] in 1971. Since then the theory was extended onto the multidimensional FEM discretizations for the convection-dominated transport equations.

4.2.1 One dimensional case

We start our description from the simple 1D cases for the finite difference schemes and will evolve to the nonconforming FEM discretization of the Navier-Stokes equations for the multidimensional case (2D and 3D).

Let us consider a scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0. \tag{4.11}$$

Setting $f = vu$, we obtain a 1D convection transport equation

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0. \tag{4.12}$$

In order to prevent oscillations of u we require the discrete scheme of (4.11) to satisfy the following properties of *monotonicity*:

- No new local extremum should be created.
- Existing local maximum should not increase and local minimum should not decrease.

In other words it means that a total variation

$$TV(u_h) = \sum_i |u_h(x_{i+1}) - u_h(x_i)|$$

as a function of the approximate velocity u_h is nondecreasing, i.e.

$$TV(u_h^{n+1}) \leq TV(u_h^n). \tag{4.13}$$

Next, denoting L to be the discretization operator:

$$u_h^{n+1} = L \cdot u_h^n, \tag{4.14}$$

we say that a finite difference scheme (4.14) is *total variation nonincreasing* (TVNI) if L satisfies (4.13).

In 1982 A. Harten, see [40], proved the following results:

- a TVNI scheme is monotonicity preserving; i.e. if u_h^n is monotone, so is $L \cdot u_h^n$
- a semi-discrete scheme of (4.11)
  $$\frac{du_i}{dt} + \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} = 0 \tag{4.15}$$
  is total variation nonincreasing if it can be rewritten in a form
  $$\frac{du_i}{dt} = c_{i-1/2}(u_{i-1} - u_i) + c_{i+1/2}(u_{i+1} - u_i), \tag{4.16}$$
  where $c_{i-1/2} \geq 0$, $c_{i+1/2} \geq 0$ and $c_{i-1/2} + c_{i+1/2} \leq 1$.

So constructed scheme will definitely guarantee a nonoscillating profile of u. On the other hand, according to a well-known theorem of S. Godunov [35] a finite-linear scheme

$$u_h^{n+1}(x_j) = \sum_{l=-k}^{k} c_l u_h^n(x_{j+l}), \qquad c_l = const$$

is monotonicity preserving if and only if

$$c_l \geq 0, \qquad -k \leq l \leq k.$$

As a result, any monotonicity preserving scheme and, hence, any TVNI scheme is first order accurate. However, in the regions of the domain, where the flux f changes not rapidly, one can recover high orders of accuracy, see [40]. Accumulating both ideas (construction of monotonicity preserving schemes with local recovery of high orders of accuracy), we introduce an adaptive correction factor $\Phi_{i\pm1/2}$, the purpose of which is to "diffuse" the change of flux in domains with steep changes of gradients and to reconstruct high-order approximation in those domains with smooth changes of flux:

$$f_{i\pm1/2} = f_{i\pm1/2}^L + \Phi_{i\pm1/2}[f_{i\pm1/2}^H - f_{i\pm1/2}^L], \tag{4.17}$$

where $f_{i+1/2}^L$ is a discrete value of a low order flux in the i-th mesh point and $f_{i+1/2}^H$ is a discrete value of a high order flux.

Numerical experiments showed that forward and central difference methods for a simple 1D convection equation (4.12) with $v > 0$ give rise to the oscillating behaviour of u for some values of the Peclet number (see e.g. [31]) and, therefore, are not stable. At the same time a backward difference scheme is stable, but it adds some extra diffusion into the solution. Following the idea (4.17) for the construction of a difference scheme, we take a backward difference method as a low (first) order non-oscillatory scheme:

$$f_{i+1/2}^L = vu_i. \tag{4.18}$$

After substituting (4.18) into (4.15) and rewriting the obtained expression in the form of (4.16), we end up with the coefficients $c_{i-1/2} = v/\varDelta x$ and $c_{i-1/2} = 0$, which satisfy the Harten's theorem. Therefore, the obtained difference scheme is total variation nonincreasing. To recover the second order of accuracy we choose a central difference as a higher order scheme:

$$f^H_{i+1/2} = v\frac{u_{i+1} + u_i}{2}. \tag{4.19}$$

Bringing (4.18) and (4.19) together we end up with

$$f_{i+1/2} = vu_i + \frac{v}{2}\varPhi_{i+1/2}(u_{i+1} - u_i). \tag{4.20}$$

To complete the derivation of the 'non-oscillatory' scheme we left to specify the flux limiter $\varPhi_{i+1/2}$ at every mesh point. It is natural to require $\varPhi_{i+1/2}$ to decrease, as soon as the changes in the velocity gradient increase at the i-th point. Therefore, we construct our flux limiter in the form $\varPhi_{i+1/2} = \varPhi(r_i)$, where

$$r_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_i} \tag{4.21}$$

is the so-called slope ratio. It approaches unity for smooth domains, is negative for the local extremum and is large in domains with the rapid change of gradients.

Generalizing flux limiting, A. Jameson in his article [45] introduced the operator $L : \mathbb{R}^2 \rightarrow \mathbb{R}$ as a limited average of its two variables u and v with the following properties:

P1. $L(u, v) = L(v, u)$
P2. $L(\alpha u, \alpha v) = \alpha L(v, u)$
P3. $L(u, u) = u$
P4. $L(u, v) = 0$ if $uv < 0$, otherwise $sign\ L(u, v) = sign\ u$.

(P1)-(P3) are natural properties of an average and (P4) is needed for the construction of a TVD (in multi dimensions LED) schemes. Setting

$$\varPhi(r) = L(1, r) = L(r, 1) \tag{4.22}$$

and keeping in mind (P4), we obtain $\varPhi(r) = 0$, if $r \leq 0$ and $\varPhi(r) > 0$ otherwise. At the same time properties (P1) and (P2) give us

$$\varPhi(r) = L(1, r) = rL(1/r, 1) = r\varPhi(1/r). \tag{4.23}$$

Applying (4.23) to (4.20) we derive with

$$\varPhi(r_i)(u_{i+1} - u_i) = L(u_{i+1} - u_i, u_i - u_{i-1}) = \varPhi(1/r_i)(u_i - u_{i-1}). \tag{4.24}$$

The equality (4.24) means that an 'antidiffusive' flux from the $i+1$-node into the i-th node is considered to be 'diffusive' (stabilizing) flux, received by the i-th node from the $i-1$ if $\varPhi(1/r_i) > 0$, and vice versa. Substituting $\varPhi(r_i)$ into (4.20) we obtain a discrete scheme of the type (4.16) with the following coefficients

$$c_{i-1/2} = \frac{v}{2\varDelta x}[2 + \frac{\varPhi(r_i)}{r_i} - \varPhi(r_{i-i})], \qquad c_{i+1/2} = 0, \tag{4.25}$$

where $\varPhi(\cdot)$ is chosen as to satisfy the Harten's theorem. There was proposed a large number of flux limiters in the literature, the most popular are (for test results and comparative analysis see, i.g. [49, 56])

$$
\begin{array}{cl}
\text{minmod} & \Phi(r_i) = min\{1, r_i\} \\
\text{VanLeer} & \Phi(r_i) = \dfrac{2r_i}{(1 + r_i)} \\
\text{MC} & \Phi(r_i) = \dfrac{(1 + r_i)}{2} \\
\text{Koren} & \Phi(r_i) = \dfrac{(1 + r_i)}{3} \\
\text{superbee} & \Phi(r_i) = max\{1, r_i\}
\end{array}
$$

The main drawback of almost of them is the fact that they were derived for the specific benchmark models and are based on certain discretizations, i.e. they lack is generalization. In the next sections we will present the extension, proposed by D. Kuzmin [51], of the flux-limiting methodology onto arbitrary finite element discretizations on unstructured grids.

4.2.2 Multidimensional case. Extension of the 1D approach.

Let us consider a nonsteady convective-diffusion equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u - \varepsilon \nabla u) = 0 \quad \text{in } \Omega, \tag{4.26}$$

where $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ is a bounded, connected domain with a piecewise smooth Lipschiz boundary Γ, $\boldsymbol{v} : \Omega \times T \rightarrow \mathbb{R}^d$ is an advection velocity, assumed to be known analytically or computed numerically, and $u : \Omega \times T \rightarrow \mathbb{R}$ is either a scalar density ρ or a single component of velocity vector field $\boldsymbol{u}$ with prescribed initial $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$ and Dirichlet and Neumann boundary conditions on corresponding boundary part of Γ.

Denoting $\boldsymbol{w}$ to be an element from the space of velocity test function, the weak form of (4.26) reads

$$\int_\Omega \boldsymbol{w} \left[\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{v}u - \varepsilon \nabla u) \right] = 0 \quad \forall \boldsymbol{w} \tag{4.27}$$

Next, we use the Galerkin method to convert partial differential equations (4.27) to a problem of linear algebra by approximating $\boldsymbol{u}$ by

$$\boldsymbol{u}_h = \sum_{i=1}^{N} u_i \phi_i, \tag{4.28}$$

where N is the number of degrees of freedom. As usual, we assume the test function to depend on the space coordinates ($\phi_i = \phi_i(\boldsymbol{x})$) and the 'velocity' component to depend on the time variable t ($u_i = u_i(t)$). Substitution of (4.28) into (4.27) and integration by parts yield the following system of equations (we consider divergence-free advective velocity field $\boldsymbol{v}$, which comes naturally from the continuity equation for the incompressible flows):

$$\sum_j \left[\int_\Omega \phi_i \phi_j dx \right] \frac{du_j}{dt} + \sum_j \left[\int_\Omega \phi_i \boldsymbol{v}_j \cdot \nabla \phi_j dx + \varepsilon \nabla \phi_i \cdot \nabla \phi_j dx \right] u_i = 0. \tag{4.29}$$

Let us rewrite (4.29) in a matrix form. We denote

$$m_{ij} = \int_\Omega \phi_i \phi_j dx, \qquad \boldsymbol{c}_{ij} = \int_\Omega \phi_i \nabla \phi_j dx, \qquad s_{ij} = \int_\Omega \nabla \phi_i \cdot \nabla \phi_j dx, \tag{4.30}$$

where m_{ij} are entries of a consistent mass matrix M_C, c_{ij} are due to the convective and s_{ij} are due to the diffusive parts. All coefficients in (4.30) depend on the basis functions ϕ_i only, hence, they can be assembled once during the initializing process, by this saving CPU time in future.

Next, we assemble the discrete operator $K = \{k_{ij}\}$:

$$k_{ij} = -\boldsymbol{v}_j \cdot \boldsymbol{c}_{ij} - \varepsilon s_{ij}.$$

Replacing the consistent mass matrix $M_C = \{m_{ij}\}$ by its diagonal counterpart $M_L = \{m_i\}$, the so-called lumped mass matrix, we end up with the system of ODEs:

$$M_L \frac{du}{dt} = Ku. \tag{4.31}$$

For convection dominated flows the nonstabilized convective part of the discrete transport operator K is not stable, causing the growth of non-physical oscillations in velocity vector field if some high-order time discretization technique is used. To conquer this problem we appeal to the already approbated idea of using high-order schemes in domains with small changes of the velocity vector field and of using low-order diffusive schemes in domains, where velocity gradients changes abruptly.

We say that a scheme is local extremum diminishing (LED) if a local maximum does not increase and the local minimum does not decrease. We modify the transport operator K in such a way that our discrete scheme satisfies the LED conditions. Using linear basis functions (see section 2.1.1), we note that $\sum_j c_{ij} = 0$, $\sum_j s_{ij} = 0$ and hence $\sum_j k_{ij} = 0$. Thus, (4.31) can be rewritten as

$$m_i \frac{du_i}{dt} = \sum_{j \neq i} k_{ij}(u_j - u_i) + r_i u_i, \qquad \text{where} \qquad r_i = \sum_j k_{ij}. \tag{4.32}$$

$r_i u_i$ is a discrete counterpart of $u\nabla \cdot \boldsymbol{v}$, which vanishes for the divergence-free velocity fields. A semi-discrete system (4.32) reduces to

$$\frac{du_i}{dt} = \sum_{j \neq i} b_{ij}(u_j - u_i), \qquad \text{where} \qquad b_{ij} = \frac{k_{ij}}{m_i}. \tag{4.33}$$

The discrete scheme (4.33) is stable, i.e. it satisfies the LED conditions, if all off-diagonal entries b_{ij} are nonnegative. Indeed, if u_i is a local maximum and hence $(u_j - u_i) \leq 0$, we get $\frac{du_i}{dt} \leq 0$ and u_i cannot increase. Similarly, if u_i is a local minimum and hence $(u_j - u_i) \geq 0$, we get $\frac{du_i}{dt} \geq 0$ and u_i cannot decrease. After discretization in time, such schemes remain positivity-preserving: the solution update $u^{n+1} \to u^n$ is equivalent to the solution of the following system

$$Au^{n+1} = Bu^n, \tag{4.34}$$

where $A = \{a_{ij}\}$ in an M-matrix and $B = \{b_{ij}\}$ has no negative entries, see, e.g., [54]. For the system (4.34) one can show that

$$\text{if} \quad u^n \geq 0 \Rightarrow u^{n+1} = A^{-1}Bu^n \geq 0.$$

Let us remind, that A is an *M-matrix* iff A is a Z-matrix (all off-diagonal entries are less than or equal to zero) and it satisfies the following condition

$$\text{if} \quad Au \geq 0 \Rightarrow u \geq 0.$$

To provide K with the LED condition we cannot just simply go through its nondiagonal entries and remove those, which are negative, since the conserved property requires row and column sums of the operator K to stay unchangeable. For this reason, we define a *generalized diffusion operator* $D = \{d_{ij}\}$:

1. $d_{ij} = d_{ji} = max\{0, -k_{ij}, -k_{ji}\}$. Hence, a low-order operator $L = K + D = \{l_{ij}\}$ has no negative off-diagonal entries and, therefore, is of the LED type.

2. $\sum_i d_{ik} = \sum_k d_{ik} = 0$ and the mass conservation property is satisfied.

From 1. we obtain the symmetry of the operator D. From 2. we see that, in order to guarantee the conservation property, for every i-th node we have to modify the $\{l_{ii}, l_{ij}, l_{ji}, l_{jj}\}$ matrix entries as well:

$$\begin{aligned} l_{ii} &= l_{ii} - d_{ij}, \qquad l_{ij} = l_{ij} + d_{ij}, \\ l_{ji} &= l_{ji} + d_{ij}, \qquad l_{jj} = l_{jj} - d_{ij}. \end{aligned} \tag{4.35}$$

If $d_{ij} \neq 0$, then it nullifies either l_{ij} or l_{ji} matrix entry. Basing on this fact, we can introduce orientation of edges: we say that the node i is located *upwind*, if $l_{ij} = 0$ and $l_{ji} = |k_{ji} - k_{ij}|$.

On the physical level the process of summing D to K can be considered as the addition of some extra diffusion into the system. In order to make our scheme more accurate, we have to eliminate the superfluous diffusion where it is possible. Now we can define a multidimensional variant of the flux limiter $\Phi(r_i)$ in the mesh point i, for every $i = \overline{1, N}$. The purpose of $\Phi(r_i)$ and its nature are similar to those from 1D case - the estimation of the steepness of the velocity vector field in the i-th point. Gathering everything together we construct a discrete scheme with a modified operator $\widetilde{K}$

$$\frac{du_i}{dt} = \widetilde{K}\, u = (K + D - D_\Phi)\, u, \tag{4.36}$$

where D_Φ is the corrected antidiffusive flux, which is built as the diffusive flux scaled by the limiter Φ.
The last question we left to answer is how to construct the flux limiter $\Phi(r_i)$ in every i-th mesh point for the multidimensional case. This topic we address in the section 4.2.3.

4.2.3 Flux limiter in multidimensions.

There are different approaches proposed in the literature how to construct the antidiffusive flux. One of them is a slope limiting method. In the slope-limiter FEM-TVD scheme one constructs a limiter $\Phi(r_i)$ for every edge $\overline{ij}$ by analyzing the flux contribution into every surrounding node in the element ”opposite” to the $\overline{ij}$-direction. Then, using interpolation technique one finds and allocates some virtual node, say k, to calculate the necessary antidiffusion to be added into the system. Description of this algorithm can be found in papers by Lyra *et. al.* [66, 67]. The algorithm was implemented and exhaustively tested by Kuzmin from the `Featflow` group [55]. He showed that with the help of the algorithm it is possible to obtain acceptable simulation results for a wide range of CFD applications. Nevertheless, it was also observed that the issues of the method includes the low convergence rate of nonconforming FEM on unstructured meshes. This was mainly the reason of choosing a slightly different way - the so-called generalized approach of the node-based strategy, first proposed by Zalesak in [115]. The generalized approach allows the construction of pure multidimensional flux limiters for any types of meshes.
In Fig. 4.1 one can see that the i-th node receives fluxes from various directions/neighbouring node. These fluxes can be either diffusive (those with positive coefficients), which do not give rise to wiggles and therefore harmless, or antidiffusive (those with negative coefficients), which have to be bounded to avoid non-physical oscillations of the velocity field. Splitting the transport operator K into the sum of positive (diffusive) fluxes

$$Q_i = Q_i^+ + Q_i^-. \qquad Q_i^\pm = \sum_{j \neq i} max\{0, k_{ij}\} \begin{matrix} max \\ min \end{matrix} \{0, u_j - u_i\} \tag{4.37}$$

and negative (antidiffusive) fluxes

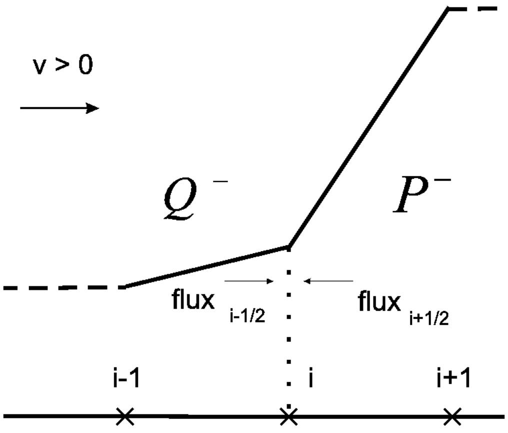

Fig. 4.1. Definition of the limiter

$$P_i = P_i^+ + P_i^-. \qquad P_i^{\pm} = \sum_{j \neq i} min\{0, k_{ij}\} \begin{matrix} min \\ max \end{matrix} \{0, u_j - u_i\} \tag{4.38}$$

we obtain

$$\sum_{j \neq i} k_{ij}(u_j - u_i) = P_i + Q_i.$$

After calculating diffusive and antidiffusive fluxes received by the node i, we can define our limiter Φ as the ratio of these fluxes. Then, applying Φ to the operator K, we limit the influence of Φ from above to avoid excessive overdiffusion. Mathematically it reads as follows

$$R_i^{\pm} = \Phi(Q_i^{\pm}/P_i^{\pm}). \tag{4.39}$$

Due to the property (P4) there is no need to evaluate either R_i or Q_i, if $sign\,(P_i\,Q_i) = -1$. Moreover, for zero antidiffusion P_i we do not calculate R_i as well. Recalling the orientation of edges, which was introduced at the end of § 4.2.2, we define the antidiffusive flux f_{ij}^{α} from the downwind node j into the upwind node i in the following way

$$f_{ij}^{\alpha} = \begin{cases} min\{R_i^+ d_{ij}, l_{ji}\}(u_i - u_j), & u_i \geq u_j \\ min\{R_i^- d_{ij}, l_{ij}\}(u_i - u_j), & u_i < u_j \end{cases} \tag{4.40}$$

So modified transport operator $\widetilde{K}$ satisfies the LED condition and, hence, is safe from the non-physical oscillations in the velocity field. We make slight changes in (4.39) and (4.40). Namely, we allow flexibility in choice of the bound for $Q_i^{\pm}$:

$$Q_i^+ = \sum_{j \neq i} q_{ij} max\{0, u_j - u_i\}, \qquad Q_i^- = \sum_{j \neq i} q_{ij} min\{0, u_j - u_i\}, \tag{4.41}$$

where q_{ij} is either $q_{ij} = max\{0, k_{ij}\}$ or $q_{ij} = max\{0, l_{ij}\}$. In this case the antidiffusive flux (those to be subtracted from the operator $K + D$) is calculated as follows

$$f_{ij}^{\alpha} = \begin{cases} R_i^+ f_{ij}, & f_{ij} > 0 \\ R_i^- f_{ij}, & f_{ij} \leq 0 \end{cases} \qquad f_{ji}^{\alpha} = -f_{ij}^{\alpha}, \tag{4.42}$$

where $f_{ij} = k_{ij}(u_j - u_i)$. This approach is known as the *Flux-limiter FEM-TVD scheme*.

Example:
Let us consider a central difference discretization of (4.12):

$$\frac{du_i}{dt} + \frac{vu_{i+1/2} - vu_{i-1/2}}{\Delta x} = 0, \qquad vu_{i+1/2} = vu_i - f_{ij}^{\alpha} \tag{4.43}$$

with r_i from (4.21) and antidiffusive flux of the TVD type being

$$f_{ij}^{\alpha} = max\{0, min\{2, \phi_i, 2r_i\}\} d_{ij}(u_i - u_j), \tag{4.44}$$

where $\phi_i = \xi + (1 - \xi) r_i, \ \ 0 \le \xi \le 1$. Then, the antidiffusive flux in the Flux-limiter FEM-TVD scheme reduces to

$$f_{ij}^{\alpha} = max\{0, min\{1, 2r_i\}\} d_{ij}(u_i - u_j). \qquad \Box$$

Taking into account (4.42) we rewrite the process of construction of flux limiters [50]:

1. Compute sums for positive and negative antidiffusive fluxes
$$P_i^+ = P_i^+ + max\{0, f_{ij}\}, \qquad P_i^- = P_i^- + min\{0, f_{ij}\}. \tag{4.45}$$
2. Compute the upper/lower bounds for $Q_i^{\pm}$
$$\begin{array}{ll} Q_i^+ = Q_i^+ + max\{0, -f_{ij}\}, & Q_j^+ = Q_j^+ + max\{0, f_{ij}\}, \\ Q_i^- = Q_i^- + min\{0, -f_{ij}\}, & Q_j^- = Q_j^- + min\{0, f_{ij}\}. \end{array} \tag{4.46}$$
3. Construct the nodal correction factor $R_i^{\pm}$ for every 'upwind' (according to the introduces orientation of edges) node i
$$R_i^{\pm} = min\{1, Q_i^{\pm}/P_i^{\pm}\}. \tag{4.47}$$

4.2.4 Resulting algorithm

To summarize section 4.2 we present a sketch of the stabilization technique for the convection dominated flows by the flux correction algorithm in multidimensions for the nonconforming FEM on unstructured meshes. For the detailed description of the Flux-limiter FEM-TVD scheme strategy see [56]:

1. Perform a FEM Galerkin discretization in space to obtain a high-order semi-discrete linear system
$$M_C \frac{du}{dt} = Ku. \tag{4.48}$$
Transport matrix K is not 'stabilized'. In the sense of LED criteria it means that $\exists j \neq i \ : \ k_{ij} < 0$.
2. Satisfy the LED condition by eliminating negative off-diagonal entries and preserve mass conservation property. This can be done by applying a generalized diffusion operator D according to (4.35). Replace the conservative mass matrix M_C by its lumped counterpart M_L. The low-order scheme reads
$$M_L \frac{du}{dt} = Lu, \tag{4.49}$$
where $L = K + D$ is such that $l_{ij} \geq 0, \, \forall j \neq i$.

3. Remove excessive diffusion where possible by adding limited antidiffusion operator

$$M_L \frac{du}{dt} = K^* u, \tag{4.50}$$

where $K^* = K + D - D_\Phi$. Though $\exists j \neq i \ : \ k^*_{ij} < 0$, the whole matrix K^* is LED-equivalent to L^*, i.e. $L^* u = K^* u$, where $l^*_{ij} \geq 0$, $\forall j \neq i$.

The discussed above method shows good and robust behaviour for complex 2D and 3D problems. It takes into account both mathematical and physical interpretations of the problem and is based on the careful step-by-step modification of the flux in the 'crucial' regions.

Remark 8. In the content of the modified discrete projection method, convective oriented Pressure Schur complement preconditioners might also act as sources of nonphysical oscillations in pressure and velocity. This fact leads to destabilizing behaviour of the algorithm and, as a result, the failure of a solver. Therefore, numerical treatment of such preconditioners requires additional stabilization or relaxation, which will be introduced in § 5.3.

4.3 Edge-oriented stabilization

The main idea of the edge-oriented stabilization is to augment the original finite element discretization by an interior penalty term involving the jump of the function values or of the gradient of the approximate FEM solution. In the literature, several jump terms were introduced for different situations:

1) Jump terms including function values [46, 109]

$$j_1(\boldsymbol{u}, \boldsymbol{v}) = \sum_{\text{edge E}} \gamma \nu \frac{1}{|E|} \int_E [\boldsymbol{u}][\boldsymbol{v}] d\sigma. \tag{4.51}$$

2) Jump terms including the gradient [27, 15, 13, 16]

$$\begin{aligned} j_{2,\alpha}(\boldsymbol{u}, \boldsymbol{v}) &= \sum_{\text{edge E}} \gamma |E|^\alpha \int_E [\nabla \boldsymbol{u}][\nabla \boldsymbol{v}] d\sigma, \\ j_{3,\alpha}(\boldsymbol{u}, \boldsymbol{v}) &= \sum_{\text{edge E}} \gamma |E|^\alpha \int_E [\boldsymbol{n} \cdot \nabla \boldsymbol{u}][\boldsymbol{n} \cdot \nabla \boldsymbol{v}] d\sigma, \\ j_{4,\alpha}(\boldsymbol{u}, \boldsymbol{v}) &= \sum_{\text{edge E}} \gamma |E|^\alpha \int_E [\boldsymbol{t} \cdot \nabla \boldsymbol{u}][\boldsymbol{t} \cdot \nabla \boldsymbol{v}] d\sigma, \\ j_{5,\alpha}(\boldsymbol{u}, \boldsymbol{v}) &= \sum_{\text{edge E}} \gamma |E|^\alpha \int_E [(\boldsymbol{t} \cdot \nabla \boldsymbol{u}) \cdot \boldsymbol{n}][(\boldsymbol{t} \cdot \nabla \boldsymbol{v}) \cdot \boldsymbol{n}] d\sigma. \end{aligned} \tag{4.52}$$

3) Jump terms including the divergence [13]

$$j(\boldsymbol{u}, \boldsymbol{v}) = \sum_{\text{edge E}} \gamma |E|^2 \int_E [\nabla \cdot \boldsymbol{u}][\nabla \cdot \boldsymbol{v}] d\sigma. \tag{4.53}$$

4) Jump terms including the normal component of function values [14]

$$j(\boldsymbol{u},\boldsymbol{v}) = \sum_{\text{edge E}} \gamma\nu \frac{1}{|E|} \int_E [\boldsymbol{n}\cdot\boldsymbol{u}][\boldsymbol{n}\cdot\boldsymbol{v}]d\sigma. \tag{4.54}$$

For our purposes we chose a new variant of the jump term, proposed in works by Ouazzi and Turek [79, 81, 107]

$$\langle \mathbf{J}\boldsymbol{u}_h, \boldsymbol{v}_h\rangle = \sum_{\text{edge E}} \max(\gamma^*\nu h_E, \gamma h_E^2) \int_E [\nabla \boldsymbol{u}_h][\nabla \boldsymbol{v}_h]d\sigma, \tag{4.55}$$

where $h_E = |E|$. The jump term (4.55) is added to the original bilinear form, resp., discretized stiffness matrices, and it uses only the gradient of the approximate solution. Numerical experiments showed that the parameters γ, γ^* can be chosen more or less arbitrarily in the interval $[0.0001, 1]$, with no significant influence on the resulting accuracy, robustness and efficiency.

4.4 Summary for the treatment of convective term

In this chapter we extended the modified projection method (proposed in chapter 2) to the general-purpose 'cross product operator' $\boldsymbol{w}(\boldsymbol{\omega}, \boldsymbol{u}, \cdot) \times \boldsymbol{u}$, which may include Coriolis force, convection and/or any other terms to be written in the cross-product form. We also gave a brief overview of the algebraic flux correction and the edge-oriented stabilization techniques for the stabilization of the convective term. Our aim was not to propose any improvements to these techniques, but to thoroughly test them for standard and rotational forms of convection in the framework of our modified discrete projection method. For the corresponding numerical results the reader is referred to chapters 5, 6 and 8.

5

Numerical results for the modified DPM

In this chapter we present numerical studies of the modified discrete projection method for systems of incompressible Stokes and Navier-Stokes equations with the Coriolis force term for model problems of unit square/cube geometries (numerical simulations for complex 3D geometries of the Stirred Tank Reactor model will be shown and discussed in $\S$ 8.1). We will compare accuracy of the modified and nonmodified schemes, examine the multigrid behaviour for the arising momentum and pressure Poisson-like subproblems for different values of time step, angular velocity, etc., and observe the (outer) convergence behaviour of the modified DPM scheme for various preconditioners. At the end of this chapter we will discuss some aspects concerning the convective part in the pressure Schur Complement preconditioner.

5.1 Numerical results for the unit square

In this section we examine the accuracy in time of the pressure and velocity for the modified projection scheme and compare results with those of the nonmodified scheme (with the $\boldsymbol{\omega}$-independent orthogonal projection step (3.2)). We take a test model of a unit square domain $[-1,1]\times[-1,1]$ and solve the system of the incompressible Stokes equations with the Coriolis force term

$$\begin{gathered}\boldsymbol{u}_t - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla p = \boldsymbol{f}\\ \nabla\cdot\boldsymbol{u} = 0\end{gathered} \tag{5.1}$$

with homogeneous Dirichlet boundary conditions for the velocity. The exact solution $(\boldsymbol{u},p)$ of (5.1) is chosen as in [37, 38]:

$$u_1(x,y) = \pi sin(t) sin(2\pi y) sin^2(\pi x), \tag{5.2}$$

$$u_2(x,y) = -\pi sin(t) sin(2\pi x) sin^2(\pi y), \tag{5.3}$$

$$p(x,y) = sin(t) cos(\pi x) sin(\pi y). \tag{5.4}$$

It is easy to calculate the right hand side of (5.1) for the analytical solution $(\boldsymbol{u},p)$ from (5.2)-(5.3):

$$\begin{aligned} f_1 = &\quad \cos(t)\sin(2\pi y)\sin^2(\pi x) - 2\pi^3\sin(t)\sin(2\pi y)\cos^2(\pi x) + 6\pi^3\sin(t)\sin(2\pi y)\sin^2(\pi x)\\ &- \pi\sin(t)\sin(\pi x)\sin(\pi y) + 2\,\omega\,\pi\sin(t)\sin(2\pi x)\sin^2(\pi y),\\ f_2 = &- \pi\cos(t)\sin(2\pi x)\sin^2(\pi y) + 2\pi^3\sin(t)\sin(2\pi x)\cos^2(\pi y) - 6\pi^3\sin(t)\sin(2\pi x)\sin^2(\pi y)\\ &+ \pi\sin(t)\cos(\pi x)\cos(\pi y) + 2\,\omega\pi\sin(t)\sin(2\pi y)\sin^2(\pi x). \end{aligned}$$

We denote

$$v_{mean}(t) = \frac{1}{NDF} \sum_{k=1}^{NDF} |v_{analyt}(t,k) - v_{numer}(t,k)|,$$

where v_{analyt} is the velocity magnitude $|\boldsymbol{u}|$ or the pressure p from (5.2)-(5.4) and v_{numer} is a corresponding numerical value, NDF is a number of degrees of freedom. To compute the averaged error until some time T_{mes} we use the formula:

$$v_{err}(\Delta t) = \frac{1}{N} \sum_{\gamma=1}^{N} v_{mean}^2(\gamma\, \Delta t), \quad N = \frac{T_{mes}}{\Delta t}.$$

The following setting is chosen: $\nu = 1$, $T_{mes} = 1.8$. Then, we perform simulations for various time steps $\Delta t \in \{0.025,\ 0.05,\ 0.1,\ 0.15\}$, angular velocity values $|\boldsymbol{\omega}| \in \{1,\ 5,\ 10\}$ and mesh sizes $h \in \{1/32,\ 1/128\}$. A uniform cartesian mesh is used.

In Figures 5.1 and 5.2 we show graphics for $\boldsymbol{u}_{err}(\Delta t)$ and $p_{err}(\Delta t)$ with the angular velocities $|\boldsymbol{\omega}| \in \{1, 5\}$ and $|\boldsymbol{\omega}| = 10$, respectively.

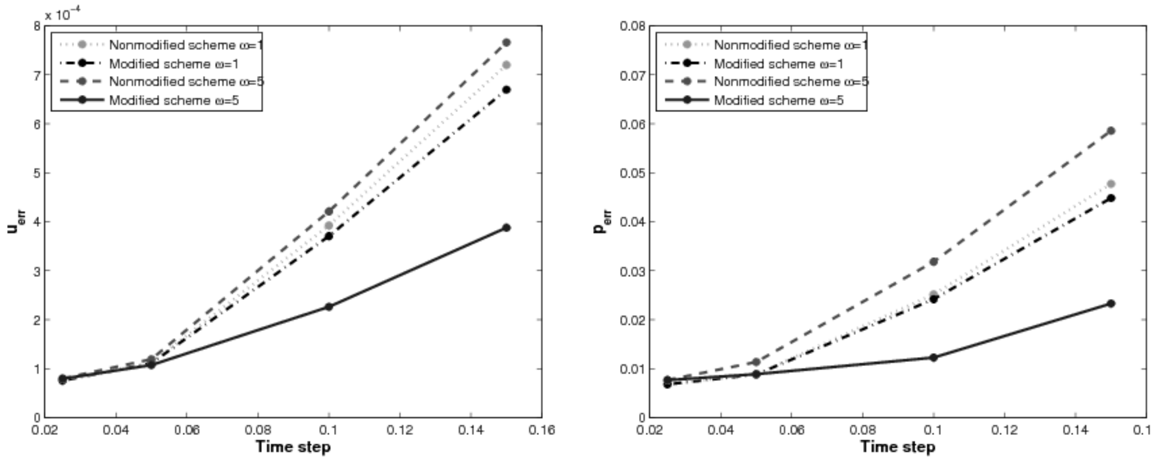

Fig. 5.1. Accuracy in time as a function on Δt, $|\boldsymbol{\omega}| \in \{1,\ 5\}$, $h = 1/32$,(LEFT) $\boldsymbol{u}_{err}(\Delta t)$, (RIGHT) $p_{err}(\Delta t)$.

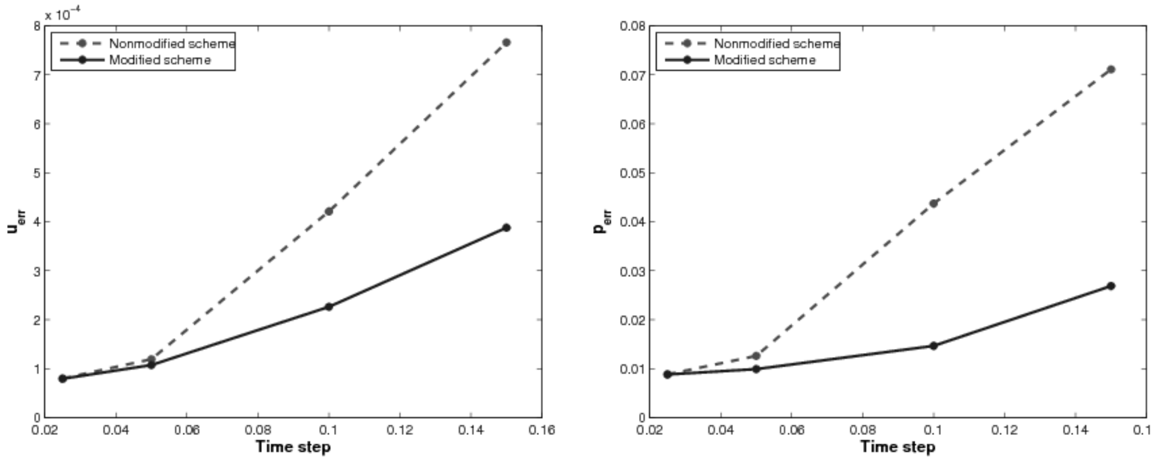

Fig. 5.2. Accuracy in time as a function on Δt, $|\boldsymbol{\omega}| = 10$, $h = 1/32$, (LEFT) $\boldsymbol{u}_{err}(\Delta t)$, (RIGHT) $p_{err}(\Delta t)$.

Exemplary graphics for $\boldsymbol{u}_{mean}(t)$, $u_{1mean}(t)$, $u_{2mean}(t)$ and $p_{mean}(t)$ for the setting $\Delta t = 0.1$, $|\boldsymbol{\omega}| = 10$ and $h = 1/32$ are shown in Figures 5.3 and 5.4.

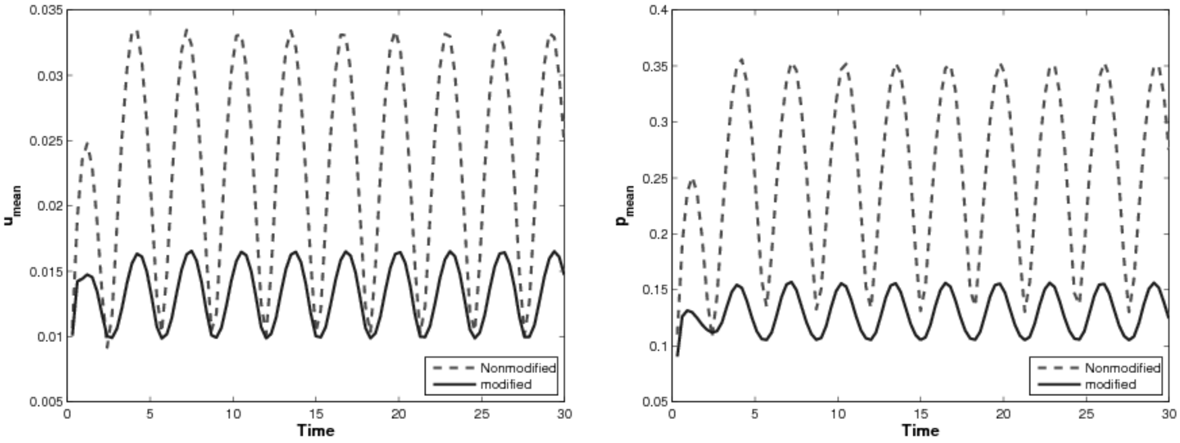

Fig. 5.3. Error distribution, $\Delta t = 0.1$, $|\boldsymbol{\omega}| = 10$, $h = 1/32$, (LEFT) $\boldsymbol{u}_{mean}$, (RIGHT) p_{mean}.

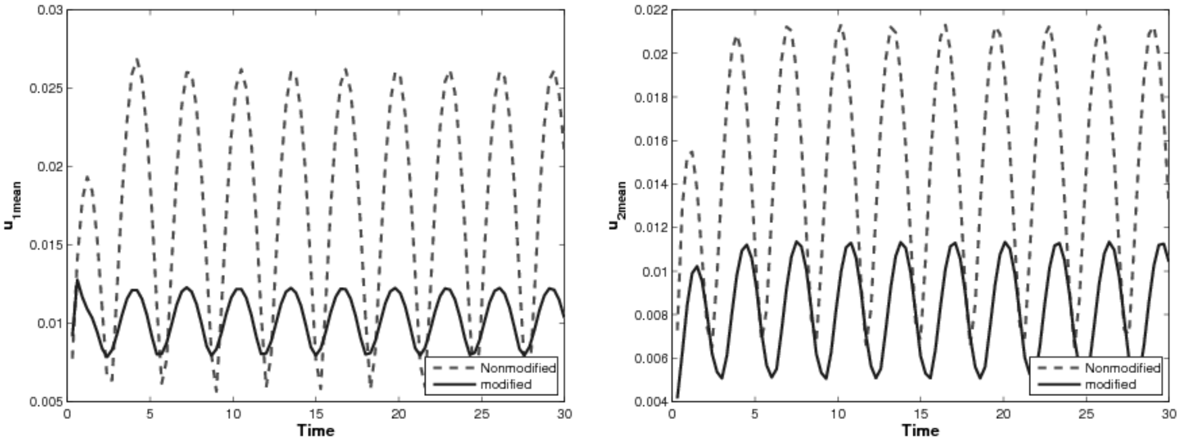

Fig. 5.4. Error distribution, $\Delta t = 0.1$, $|\boldsymbol{\omega}| = 10$, $h = 1/32$, (LEFT) u_{1mean}, (RIGHT) u_{2mean}.

As the next step we refine our mesh and evaluate velocity and pressure errors. Numerical results for the case $h = 1/128$ and $|\boldsymbol{\omega}| = 10$ we show in Fig. 5.5.

From the presented numerical results one observes that the modified projection scheme for the system of incompressible Stokes equations with the Coriolis force term is in general more accurate than the standard one. Improvement in accuracy is proportional to $\Delta t\,|\boldsymbol{\omega}|$ value. Naturally, the difference in accuracy between modified and nonmodified schemes becomes less noticeable if Δt goes to 0, then only spatial error is visible.
Though we use very course Q_0-pressure approximation, for higher levels of refinement it is possible to reach convergence state for the modified projection method, when one is able to observe layers with large pressure errors. This presence of the large pressure errors at the corners of the square domain was reported in [38, 96] and was

conjectured by the lack of smoothness of the domain. In Fig. 5.6 we show pressure error at $t = T_{mes} = 1.8$ for a uniform cartesian mesh with the mesh-size $1/128$.

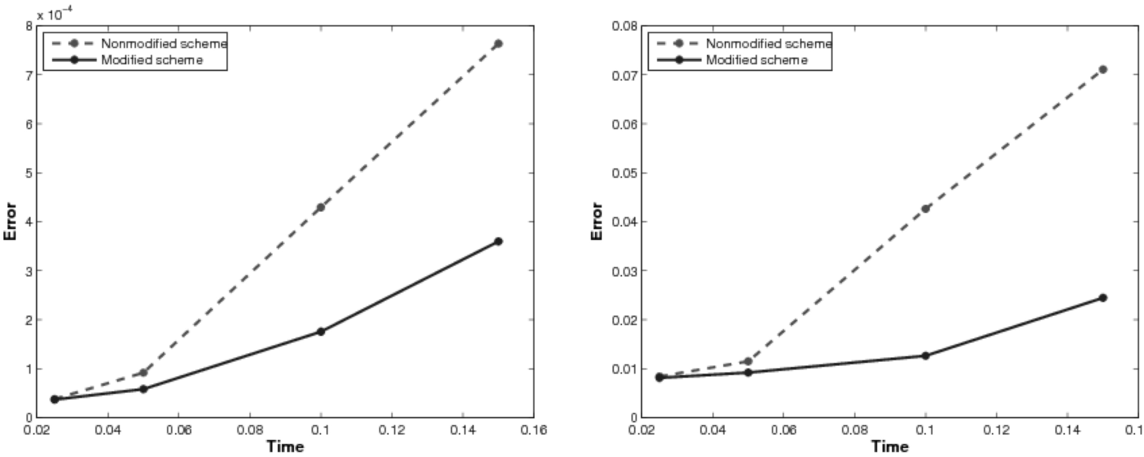

Fig. 5.5. Accuracy in time as a function on Δt, $|\boldsymbol{\omega}| = 10$, $h = 1/128$,(LEFT) $\boldsymbol{u}_{err}(\Delta t)$, (RIGHT) $p_{err}(\Delta t)$.

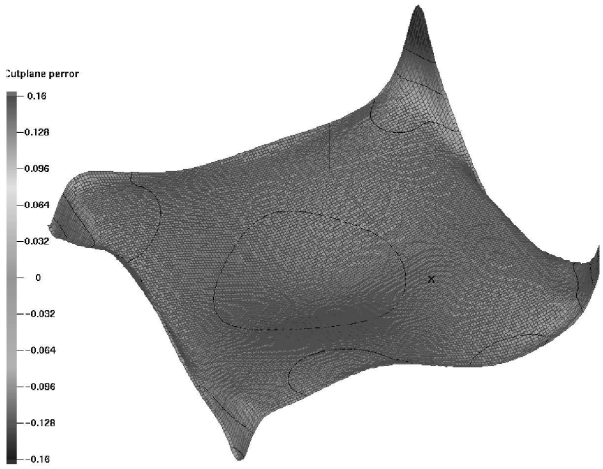

Fig. 5.6. Pressure error, $\Delta t = 0.1$, $t = T_{mes} = 1.8$, $h = 1/128$.

We would like to note that due to the block-diagonal structure of preconditioning matrices, computational resources required by standard/nonmodified and modified discrete projection schemes remain on the same level.

5.2 Numerical results for the unit cube

Since our end goal is the application of the modified projection scheme for the complex 3D problems of the STR type, we proceed with the 3D case and examine the behaviour of the modified schemes. So, let us take a unit cube model, which is shown in Fig 5.7. This choice is motivated by the relevant simplicity of the geometry and the transparency of all matrix-operator constructions, which substantially facilitates the test of the proposed algorithm and makes the observation of its numerical behaviour easier. Thus, in this section our aim will be to analyze the numerical properties of the modified discrete projection method for the system of the Stokes and Navier-Stokes equations with the Coriolis force term, to compare the efficiency of preconditioners, to evaluate the convergence rates of the multigrid for velocity and pressure subproblems, to examine the convergence of the outer DPM scheme depending on preconditioners and to present numerical results for a model problem in the unit cube. As before we assume that the Coriolis force term corresponds to a rotation around the Z-axis, i.e. the angular velocity $\boldsymbol{\omega}$ is parallel to OZ. For space discretization of a unit cube $[-1,1]\times[-1,1]\times[-1,1]$ we consider a uniform Cartesian mesh. In the geometric multigrid solver we use several grid levels. In Table 5.1 we adopt the following notation:

Table 5.1. Mesh characteristics of a unit cube with equidistant meshing.

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

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.

5.2.1 Multigrid method for velocity problems

Step 1 of the projection method involves a solution of the velocity subproblem with matrix S given in (2.17). Here we test a geometric multigrid method (V-cycle) with smoothing iterations defined in section 2.3.1. We compare it with the multigrid involving more standard pointwise SOR type smoothing iterations. This smoothing iteration can be defined as (2.25) with

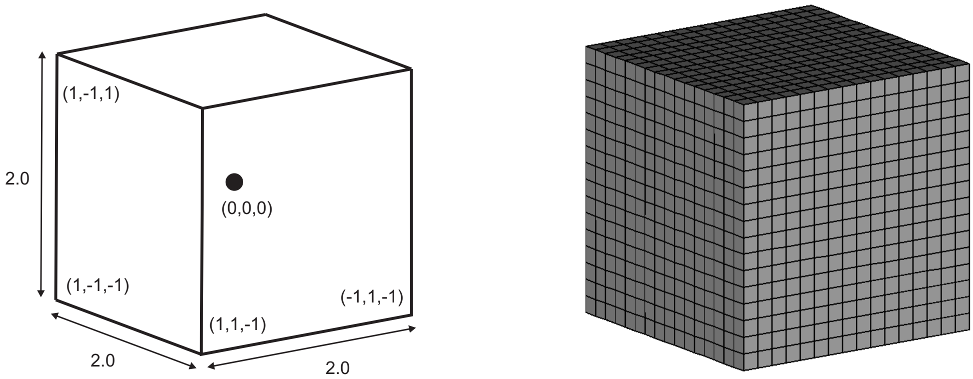

Fig. 5.7. Unit cube, (LEFT) Sketch, (RIGHT) Mesh on the 4th level.

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

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

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

Both $\boldsymbol{C}_{SORcoriol}$ and $\boldsymbol{C}_{coriol}$ matrices take into account convective and Coriolis force terms. However, only $\boldsymbol{C}_{coriol}$ from (2.26) uses the full Coriolis force terms. In Table 5.2 we present the number of multigrid iterations to gain 3 digits of defect improvement for several problem parameters and various smoothers.

Table 5.2. 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

For larger values of $\omega\Delta t$ the multigrid method with $\boldsymbol{C}_{coriol}$-based smoother outperforms the SOR-type smoothers. Moreover, the block diagonal structure of $\boldsymbol{C}_{coriol}$ makes it possible to find the inverse matrix explicitly (use Proposition 2 from § 2.3.1). This makes the calculation of $\boldsymbol{C}_{coriol}^{-1}$ for a given vector q very fast and easily done in parallel.

5.2.2 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 § 5.2.1 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 (2.44). Lemma 1 and the analysis of § 2.3.2 ensure that the matrix $P = B^T \boldsymbol{M}_{(\cdot)}^{-1} B$ with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (2.32) or (2.34) 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)), k=1,3, see [43].

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 equation by the multigrid method with two different smoothers and various values of $\omega\Delta t$. In Table 5.3 convergence rates are given for the V-cycle with four post-smoothing steps (no pre-smoothing) by SOR, ILU(k) iterations, or two post-smoothing steps by BiCGStab with ILU(k) preconditioning, where k=1,3. Level of the mesh refinement is 3 (on every upper level there are 8 times more elements than on the previous one). In Tables 5.4, 5.5 and 5.6 we present the comparison of multigrid rates on the 4th and 5th levels for pressure Schur preconditioners $B^T\boldsymbol{M}^{-1}_{\text{(mass+coriolis)}}B$, $B^T\boldsymbol{M}^{-1}_{\text{(diag)}}B$ and $B^T\boldsymbol{M}^{-1}_{\text{(diag+coriolis)}}B$, respectively. 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.
- Numerical results show that for large values of $\omega\Delta t$ the matrix $P = B^T\boldsymbol{M}^{-1}_{\text{(mass+coriol)}}B$ tends towards a tridiagonal matrix. One can check this by substituting $B = \{b_{ij}\}$ and $\boldsymbol{M}_{\text{(mass+coriol)}} = \{m_{ij}\}$ values into $P = B^T\boldsymbol{M}^{-1}_{\text{(mass+coriol)}}B$ for large $\omega\Delta t$. This explains the excellent convergence rates with the ILU(k) and BiCGStab(ILU(k)) smoother since they are exact solvers for tridiagonal matrices. However, although the pressure equation with these matrices is easy to solve, the global behaviour of the outer DPM may get worse as the following section illustrates.

5.2.3 Numerical analysis of the new DPM

For the numerical analysis of the computational performance of the new DPM we consider two different cases. We start testing the algorithm by solving a quasi-stationary problem and calculate until the steady state is achieved by pseudo-time-stepping with DPM. In the chapter 8 we will use the modified DPM to compute the fully unsteady case for the Stirred Tank Reactor (STR) problem. To monitor the convergence to the steady solution we compute values of $\|\boldsymbol{u}_t\|_{l_2}/\|\boldsymbol{u}\|_{l_2}$. Values of $\|p_t\|_{l_2}/\|p\|_{l_2}$ behave in a very similar way. In the ideal case (when the preconditioner is exact) we could expect that the convergence of the solution to the steady case would be very fast. However, the inversion of the exact pressure Schur Complement as preconditioner is prohibitively expensive and therefore it cannot be used in practice. The constructed approximating preconditioners $\boldsymbol{M}_{\text{(mass+coriolis)}}$, $\boldsymbol{M}_{\text{(diag)}}$ and $\boldsymbol{M}_{\text{(diag+coriol)}}$ might loose in the convergence speedup if compared with the exact one, but should definitely deliver better convergence behaviour if compared with the original nonmodified preconditioner $\boldsymbol{M}_{\text{(mass)}}$. Moreover, the speedup in the convergence rate should grow bigger, when larger values $\omega\Delta t$ are used. In the following, we perform the tests for every of the discussed choices for the unit cube geometry. For the STR configurations the convergence of the DPM has the same tendency, though due to the higher mesh complexity of the STR the relevant upper bound of the $\omega\Delta t$ value is smaller.

Schur Complement preconditioning for the Stokes equation with Coriolis force

Let us consider the system of the Stokes equation:

$$\begin{aligned} \boldsymbol{u}_t - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla P = \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} = 0 \end{aligned} \qquad \text{in } \Omega\times(0,T] \tag{5.5}$$

Table 5.3. Multigrid, Level3, NSMP=4 (number of pre/postsmoothing steps for pressure).

PSC	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
$M_{(\text{mass})}$	SOR	0.21-01	0.21-01	0.21-01	0.21-01
	ILU(0)	0.77-02	0.77-02	0.77-02	0.77-02
	ILU(1)	0.17-02	0.17-02	0.17-02	0.17-02
	ILU(3)	0.19-03	0.19-03	0.19-03	0.19-03
	BiCGStab(ILU(1)), NSMP=2	0.95-03	0.95-03	0.95-03	0.95-03
	BiCGStab(ILU(1)), NSMP=4	0.27-05	0.27-05	0.27-05	0.27-05
	BiCGStab(ILU(3)), NSMP=2	0.38-04	0.38-04	0.38-04	0.38-04
	BiCGStab(ILU(3)), NSMP=4	0.77-07	0.77-07	0.77-07	0.77-07
$M_{(\text{diag})}$	SOR	0.67-01	0.25+00	0.39+00	0.42+00
	ILU(0)	0.46-01	0.20+00	0.33+00	0.36+00
	ILU(1)	0.31-01	0.14+00	0.23+00	0.25+00
	ILU(3)	0.72-02	0.37-01	0.76-01	0.87-01
	BiCGStab(ILU(1)), NSMP=2	0.37-02	0.51-02	0.75-02	0.13-01
	BiCGStab(ILU(1)), NSMP=4	0.20-05	0.64-06	0.78-06	0.20-05
	BiCGStab(ILU(3)), NSMP=2	0.64-04	0.35-04	0.74-04	0.21-03
	BiCGStab(ILU(3)), NSMP=4	0.79-07	0.83-07	0.1-06	0.82-07
$M_{(\text{mass+coriol})}$	SOR	0.21-01	0.22-01	0.26+00	0.60+00
	ILU(0)	0.77-02	0.80-02	0.25-02	0.57-07
	ILU(1)	0.17-02	0.14-02	0.35-05	0.57-07
	ILU(3)	0.19-03	0.13-03	0.57-07	0.57-07
	BiCGStab(ILU(1)), NSMP=2	0.95-03	0.70-03	0.73-07	0.56-07
	BiCGStab(ILU(1)), NSMP=4	0.26-05	0.33-05	0.61-07	0.57-07
	BiCGStab(ILU(3)), NSMP=2	0.37-04	0.66-05	0.62-07	0.58-07
	BiCGStab(ILU(3)), NSMP=4	0.77-07	0.64-07	0.63-07	0.57-07
$M_{(\text{diagXY})}$	SOR	0.46-01	0.13+00	0.34+00	0.58+00
	ILU(0)	0.17-01	0.26-01	0.28-01	0.61-07
	ILU(1)	0.18-02	0.36-02	0.14-02	0.58-07
	ILU(3)	0.19-03	0.24-03	0.24-05	0.59-07
	BiCGStab(ILU(1)), NSMP=2	0.79-03	0.11-02	0.85-05	0.56-07
	BiCGStab(ILU(1)), NSMP=4	0.80-06	0.35-05	0.60-07	0.65-07
	BiCGStab(ILU(3)), NSMP=2	0.44-04	0.64-05	0.66-07	0.62-07
	BiCGStab(ILU(3)), NSMP=4	0.74-07	0.63-07	0.60-07	0.63-07
$M_{(\text{diag+coriol})}$	SOR	0.67-01	0.25+00	0.38+00	0.43+00
	ILU(0)	0.46-01	0.18+00	0.17+00	0.38+00
	ILU(1)	0.31-01	0.10+00	0.13+00	0.25+00
	ILU(3)	0.72-02	0.32-01	0.96-02	0.92-01
	BiCGStab(ILU(1)), NSMP=2	0.37-02	0.51-02	0.05-02	0.18-01
	BiCGStab(ILU(1)), NSMP=4	0.19-05	0.52-06	0.76-06	0.20-05
	BiCGStab(ILU(3)), NSMP=2	0.64-04	0.35-04	0.74-04	0.23-03
	BiCGStab(ILU(3)), NSMP=4	0.77-07	0.83-07	0.07-06	0.82-07

Table 5.4. Multigrid for $M_{\text{(mass+coriol)}}$, NSMP=4.

Meshing level	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
4	ILU(1)	0.19-02	0.19-02	0.77-03	0.12-06
5	ILU(1)	0.50-02	0.52-02	0.47-02	0.24-06
4	ILU(3)	0.20-03	0.19-03	0.64-05	0.12-06
5	ILU(3)	0.52-03	0.58-03	0.49-03	0.24-06
4	BiCGStab(ILU(1))	0.39-03	0.35-03	0.12-03	0.12-06
5	BiCGStab(ILU(1))	0.53-03	0.58-03	0.70-03	0.24-06
4	BiCGStab(ILU(3))	0.12-03	0.11-03	0.15-05	0.12-06
5	BiCGStab(ILU(3))	0.13-03	0.10-03	0.64-04	0.24-06

Table 5.5. Multigrid for $M_{\text{(diag)}}$, NSMP=4.

Meshing level	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
4	ILU(1)	0.28-01	0.20+00	0.34+00	0.35+00
5	ILU(1)	0.13+00	0.38+00	0.44+00	0.45+00
4	ILU(3)	0.87-02	0.13+00	0.21+00	0.21+00
5	ILU(3)	0.83-01	0.30+00	0.38+00	0.39+00
4	BiCGStab(ILU(1))	0.95-02	0.45-01	0.79-01	0.78-01
5	BiCGStab(ILU(1))	0.78-01	0.16+00	0.19+00	0.19+00
4	BiCGStab(ILU(3))	0.23-02	0.54-02	0.53-02	0.39-02
5	BiCGStab(ILU(3))	0.25-01	0.29-01	0.39-01	0.42-01

Table 5.6. Multigrid for $M_{\text{(diag+coriol)}}$, NSMP=4.

Meshing level	Smoother	$2\omega\Delta t$			
		0.05	0.5	5.0	50.0
4	ILU(1)	0.28-01	0.20+00	0.32+00	0.35+00
5	ILU(1)	0.10+00	0.31+00	0.36+00	0.45+00
4	ILU(3)	0.87-02	0.07+00	0.10+00	0.21+00
5	ILU(3)	0.50-01	0.08+00	0.33+00	0.38+00
4	BiCGStab(ILU(1))	0.89-02	0.29-01	0.71-01	0.78-01
5	BiCGStab(ILU(1))	0.70-01	0.02+00	0.16+00	0.18+00
4	BiCGStab(ILU(3))	0.19-02	0.43-02	0.52-02	0.39-02
5	BiCGStab(ILU(3))	0.23-01	0.24-01	0.25-01	0.40-01

First we find a steady limit for the solution of (5.5) 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 $\boldsymbol{M}_{(\cdot)} = \boldsymbol{M}_{\text{(mass+coriol)}}$:

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

It is natural to expect that as soon as the parameter $\omega\Delta t$ increases (either Δt gets larger, ω or both), 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 $\omega\Delta t$ decreases, the iterative behaviour of the solver with the preconditioner $B^T\boldsymbol{M}^{-1}_{\text{(mass+coriol)}}B$ approaches that obtained with the standard preconditioner $B^T\boldsymbol{M}^{-1}_{\text{(mass)}}B$. We illustrate this in Fig. 5.8.

Schur Complement preconditioners for the Navier-Stokes case

While considering the system of the Navier-Stokes equations

$$\begin{aligned} \boldsymbol{u}_t + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla P = \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} = 0 \end{aligned} \qquad \text{in } \Omega\times(0,T] \tag{5.6}$$

we can expect to gain a substantial improvement in the convergence rates by applying the Schur Complement preconditioner with the matrix $\boldsymbol{M}_{\text{(mass+coriol)}}$. But in this case we also have to care about the convective term in the Schur Complement preconditioner. As it was proposed in the previous section, the convective term will be treated by means of the preconditioning matrix $P = B^T M^{-1}_{(\cdot)}B$ with $M_{(\cdot)}$ to be of the following choice:

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

In Fig. 5.9 we see that that the convection-oriented Schur Complement preconditioner $\boldsymbol{M}_{\text{(diag)}}$ provides a substantial improvement in the outer-iteration convergence rate, i.e. decrease of $\|\boldsymbol{u}_t\|_{l_2}/\|\boldsymbol{u}\|_{l_2} \to 0$ and $\|p_t\|_{l_2}/\|p\|_{l_2} \to 0$. Some difficulties may arise because of the oscillatory behaviour of the convective part in $\boldsymbol{M}_{\text{(diag)}}$. For the discussion of this question we refer the reader to § 5.3.

Now we are ready to perform the corresponding tests for the Navier-Stokes equation with the full Schur complement preconditioner inside of the DPM, where both parts of convection and the Coriolis force terms are included:

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

This preconditioner covers both the Stokes and Navier-Stokes cases and under the right choice of parameters guarantees the fastest decrease of considered $\|\boldsymbol{u}_t\|_{l_2}/\|\boldsymbol{u}\|_{l_2} \to 0$ and $\|p_t\|_{l_2}/\|p\|_{l_2} \to 0$ values on the outer DPM loop, see Fig. 5.9.

As the last test case on the unit cube geometry 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. 5.10.

All the numerical tests presented above show that the modification of the Schur Complement operator is necessary to guarantee fast outer convergence of the DPM. Observing the iterative process for large values of $\Delta t\omega$, we notice

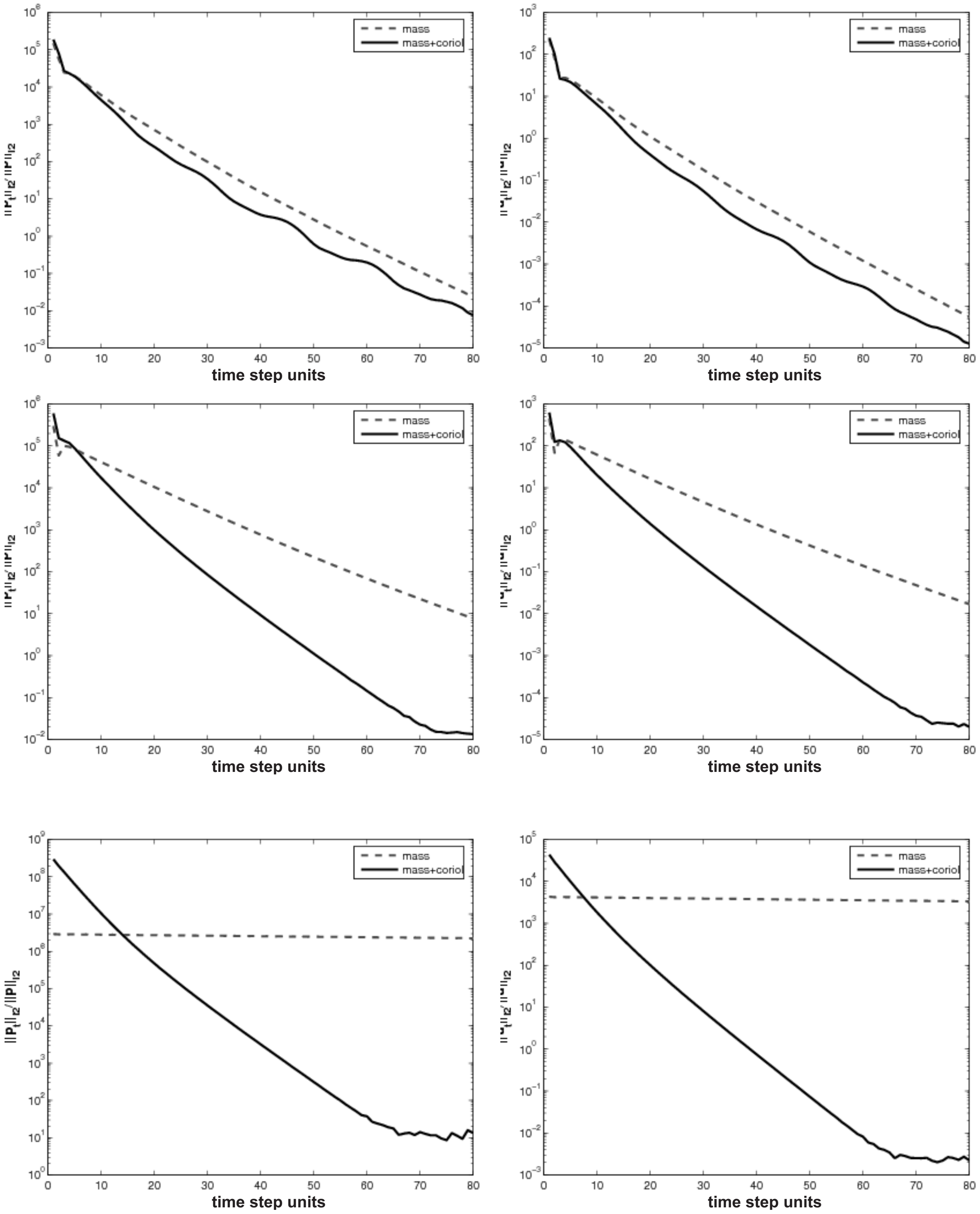

Fig. 5.8. Stokes equations (UPPER) $2\omega\varDelta t = 0.5$; (MIDDLE) $2\omega\varDelta t = 1.0$; (BOTTOM) $2\omega\varDelta t = 10.0$.

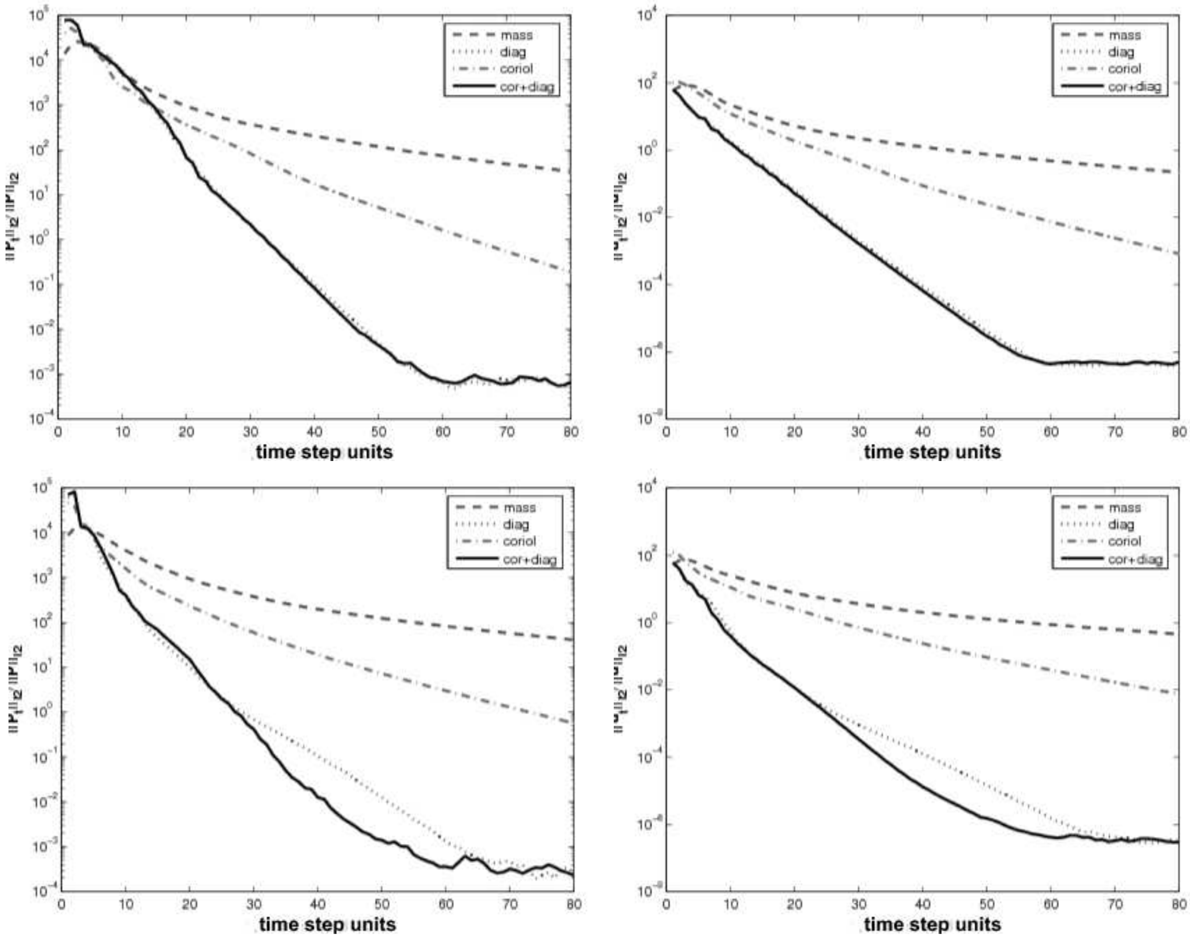

Fig. 5.9. Navier-Stokes equations (TOP) $2\omega\Delta t = 1.5$; (BOTTOM) $2\omega\Delta t = 2.5$.

that improvement in the convergence to the stationary solution can be seen even visually. To demonstrate this effect we present the pressure distribution of the nonmodified DPM with $P = B^T M^{-1}_{(\text{mass})} B$ and of the modified DPM with $P = B^T M^{-1}_{(\text{ diag+coriol })} B$. In Fig. 5.11 one can see that the pressure for the modified Schur Complement preconditioner has more accurate profile, closer to the stationary state of the pressure, than those for the nonmodified Schur Complement preconditioner. The same effect could be observed for already presented simulations of the unit square geometries from § 5.1. Smaller errors of the modified discrete projection scheme are due to the capability of the algorithm to 'catch faster' changes in velocity and pressure fields by considering the Coriolis force and convection in every step of the algorithm. At the same time the nonmodified scheme neglects treatment of these terms in the second projection step. That is why for some values of $\omega\Delta t$ the standard scheme cannot respond properly to the changes in $(\boldsymbol{u}(t), p(t))$ and therefore 'generates' larger errors.

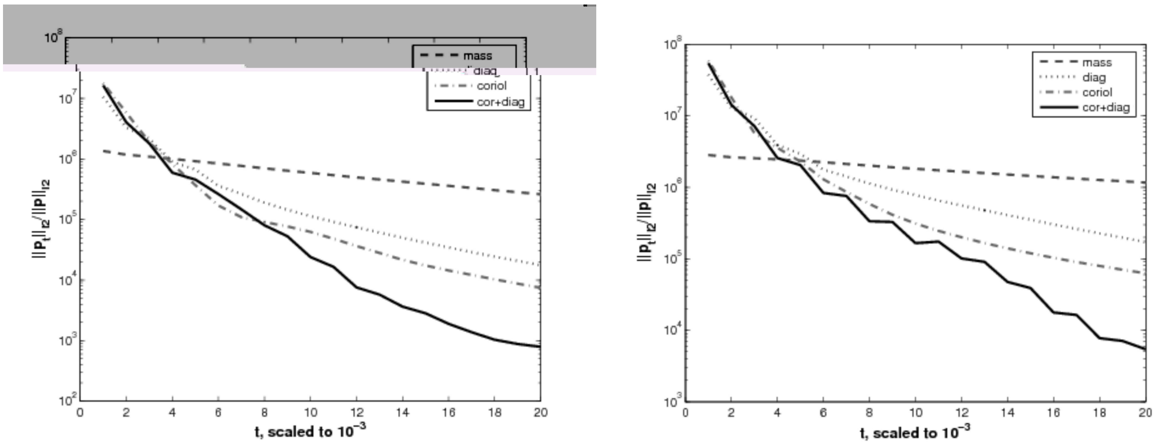

Fig. 5.10. Navier-Stokes equations with $\boldsymbol{U} \cdot \nabla \boldsymbol{u}$, (LEFT) $2\omega \Delta t = 5.0$, (RIGHT) $2\omega \Delta t = 10.0$.

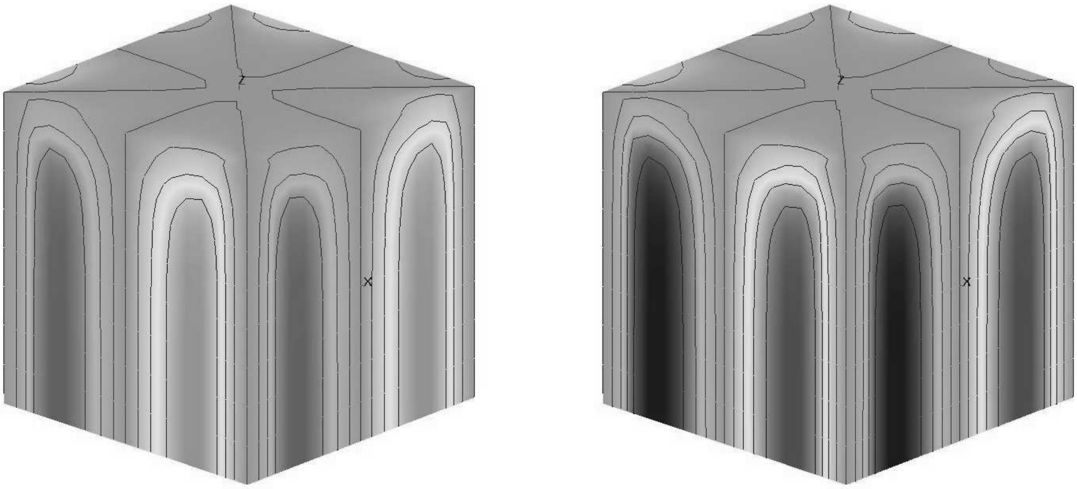

Fig. 5.11. Pressure, $2\omega \Delta t = 0.6$, (LEFT) $P = B^T \boldsymbol{M}_{(\text{mass})}^{-1} B$, (RIGHT) $P = B^T \boldsymbol{M}_{(\text{diag+coriol})}^{-1} B$.

5.3 Stabilization of convection in the Schur Complement operator

In the previous sections we considered the influence of modified Schur Complement preconditioners on the inner (multigrid convergence rate) and the outer (convergence of $\|\boldsymbol{u}_t\|_{l_2}/\|\boldsymbol{u}\|_{l_2}$ and $\|p_t\|_{l_2}/\|p\|_{l_2}$ values) numerical behaviour of the proposed discrete projection method. Nevertheless, the evaluation of one important fact we left undisclosed until now. Namely, the oscillatory behaviour of the convective part in the preconditioners

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

An exemplary test in Fig. 5.12 shows that the convective part in $\boldsymbol{M}_{(\text{diag})}$ gives birth to wiggles first in the pressure and then in the velocity field. If not 'stabilized', they may cause divergence of the algorithm for some setting of $2\omega\varDelta t$. The same effect one can also observe while using $\boldsymbol{M}_{(\text{diag+coriol})}$.

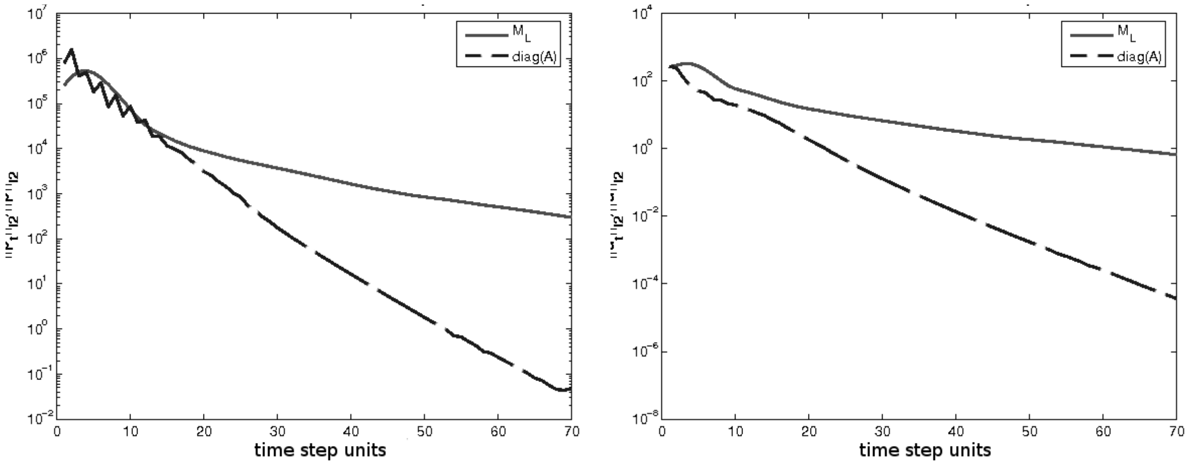

Fig. 5.12. Unit cube, $2\omega\varDelta t = 2.0$, (LEFT) pressure, (RIGHT) velocity.

Numerical experiments allow us to assume that

$$\kappa(\boldsymbol{M}_{(\text{mass})}) \leq \kappa(\boldsymbol{M}_{(\text{diag})}), \tag{5.7}$$

where $\kappa(\boldsymbol{A}) = \dfrac{\sigma_{max}(\boldsymbol{A})}{\sigma_{min}(\boldsymbol{A})}$ is the condition number and $\sigma_{max}(\boldsymbol{A})$ and $\sigma_{min}(\boldsymbol{A})$ are maximal and minimal singular values, resp., of some matrix $\boldsymbol{A}$. Indeed, comparing the magnitude of entries in $\boldsymbol{M}_{(\text{mass})}$ and $\boldsymbol{M}_{(\text{diag})}$ and taking into account the diagonal nature of both matrices, we see that for some large enough $|\boldsymbol{u}|$ (where $|\boldsymbol{u}| \sim |\boldsymbol{\omega} \times \boldsymbol{r}|$) and some matrix entry i the absolute value of $|m_{ii}^{\text{diag}}|$ is larger than those of $|m_{ii}^{\text{mass}}|$, where $\boldsymbol{M}_{(\text{mass})} = \{m_{ii}^{\text{mass}}\}$ and $\boldsymbol{M}_{(\text{diag})} = \{m_{ii}^{\text{diag}}\}$. On the other hand, since again $|\boldsymbol{u}| \sim |\boldsymbol{\omega} \times \boldsymbol{r}|$, there exists an entry k, which corresponds to the k-th degree of freedom in the mesh geometry, such that the absolute value of $|m_{kk}^{\text{diag}}|$ is smaller or equal to $|m_{kk}^{\text{mass}}|$. Next, since B (and, hence, B^T) does not depend neither on $\omega\varDelta t$ nor on $\boldsymbol{u}$, the following inequality takes place:

$$\kappa(P_{(\text{mass})}) \leq \kappa(P_{(\text{diag})}), \tag{5.8}$$

where $P_{(\cdot)} = B^T \boldsymbol{M}_{(\cdot)} B$ is the Schur Complement preconditioner. For the large $|\boldsymbol{u}|$ condition number $\kappa(\boldsymbol{M}_{(\text{diag})})$ and therefore $\kappa(P_{(\text{diag})})$ increase and make the pressure Poisson-like problem closer to the ill-conditioned case. As a result, some stabilizing techniques have to be applied. The following methods were suggested and tested for validity:

1. We have seen that the DPM's convergence with the generalized preconditioner $P_{(\text{diag+coriol})} = B^T \boldsymbol{M}^{-1}_{(\text{diag+coriol})} B$ is much faster than with the standard one $P_{(\text{mass})} = B^T \boldsymbol{M}^{-1}_{(\text{mass})} B$. For large $\omega\varDelta t$ the standard preconditioner shows extremely slow descent to the steady state. At the same time the modified one must deliver much faster convergence, but it suffers from the destabilizing behaviour of the convective part. To stabilize convective part in the Schur Complement preconditioners we will restrict "additional" parts in such a way that, on one hand, no destabilizing jumps appear and, on the other hand, the outer convergence of the DMP is the fastest possible. Mathematically it reads:

$$M_{(\text{stab})} = \begin{pmatrix} M_L & 0 & 0 \\ 0 & M_L & 0 \\ 0 & 0 & M_L \end{pmatrix} + \alpha_1 \begin{pmatrix} diag(A) - M_L & 0 & 0 \\ 0 & diag(A) - M_L & 0 \\ 0 & 0 & diag(A) - M_L \end{pmatrix} + \\ + \alpha_2 \begin{pmatrix} 0 & -2\omega \Delta t M_L & 0 \\ 2\omega \Delta t M_L & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{5.9}$$

where $\alpha_1,\ \alpha_2 \in [0, 1]$ are relaxation parameters for convective and Coriolis parts, respectively. Then the Schur Complement preconditioner is constructed in the following way:

$$P_{(\text{stab})} = B^T \boldsymbol{M}^{-1}_{(\text{stab})} B. \tag{5.10}$$

Let us note that if

- $\alpha_1 = 0,\ \alpha_2 = 0 \Longrightarrow \boldsymbol{M}_{(\text{mass})}$
- $\alpha_1 = 0,\ \alpha_2 = 1 \Longrightarrow \boldsymbol{M}_{(\text{mass+coriol})}$
- $\alpha_1 = 1,\ \alpha_2 = 0 \Longrightarrow \boldsymbol{M}_{(\text{diag})}$
- $\alpha_1 = 1,\ \alpha_2 = 1 \Longrightarrow \boldsymbol{M}_{(\text{diag+coriol})}$.

It is easy to see that so constructed matrix $\boldsymbol{M}_{(\text{stab})}$ can be inverted explicitly (apply Proposition 2 from chapter 2).

2. The other choice of stabilization for the modified pressure subproblem is based on the fact that we use an iterative solver:

$$p^n = p^n + \alpha \Delta p, \tag{5.11}$$

where $\alpha \in [0, 1]$ is a relaxation parameter. If no stabilization is required then we choose $\alpha = 1$. In any other choice α is taken smaller if stronger suppression of wiggles is required.

The proposed schemes 1 and 2 are in some sense equivalent: both approaches restrict the range of the pressure after the regular iterative step and thus relax pressure in implicit (the first choice) or explicit (the second choice) way. That is why numerical appearance of these schemes look pretty much similar.

Now let us have a look at Figures 5.13-5.15. In Fig 5.13 we show the first 16 iterations in time for the parameter setting $\omega \Delta t = 1.0$. In the case of $\boldsymbol{M}_{(\text{diag})}$ one can clearly see zigzags, which appear first in the pressure and then in some reduced form in the velocity. Relaxation of the convective part in the Schur Complement preconditioner $P_{(\text{diag})} = B^T \boldsymbol{M}^{-1}_{(\text{diag})} B$ makes the $\|p_t\|$-curve smoother if the lower value of the relaxation parameter α_1 from (5.9) is used. Moreover, one can also note that the $\|\boldsymbol{u}_t\|$-curve is smoothed out as well.

In Fig. 5.14 one can see that the stabilized curve ($diag\ with\ relaxation$) is smoother than those of $diag$, steeper than those of $mass$ and, though delivers slightly worse convergence rate in the beginning, reaches the steady state almost at the same number of macro time steps as $diag$.

From the graphics presented in Fig 5.15 the reader can convince him- or herself that the proposed stabilization techniques help to significantly increase the convergence to the steady state solution even if the addition of the convective part in its pure form into the Schur Complement preconditioner is not applicable because of arising wiggles with the subsequent divergence of the solving process. For the given setting of $\omega \Delta t$ one can also observe that a $diag$ choice of the Schur Complement preconditioner is destabilizing, whereas standard $mass$

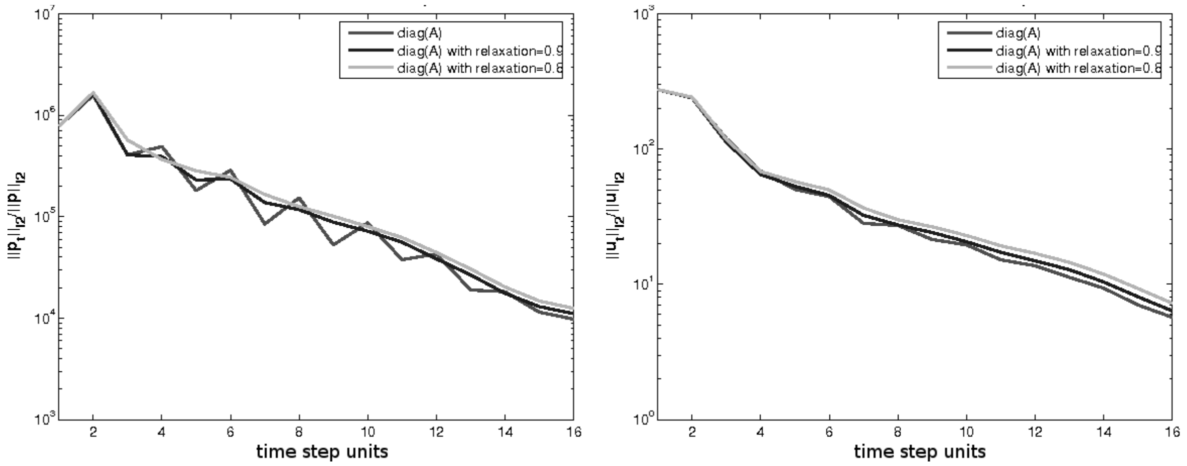

Fig. 5.13. Unit cube, $2\omega\Delta t = 1.0$, (LEFT) pressure, (RIGHT) velocity.

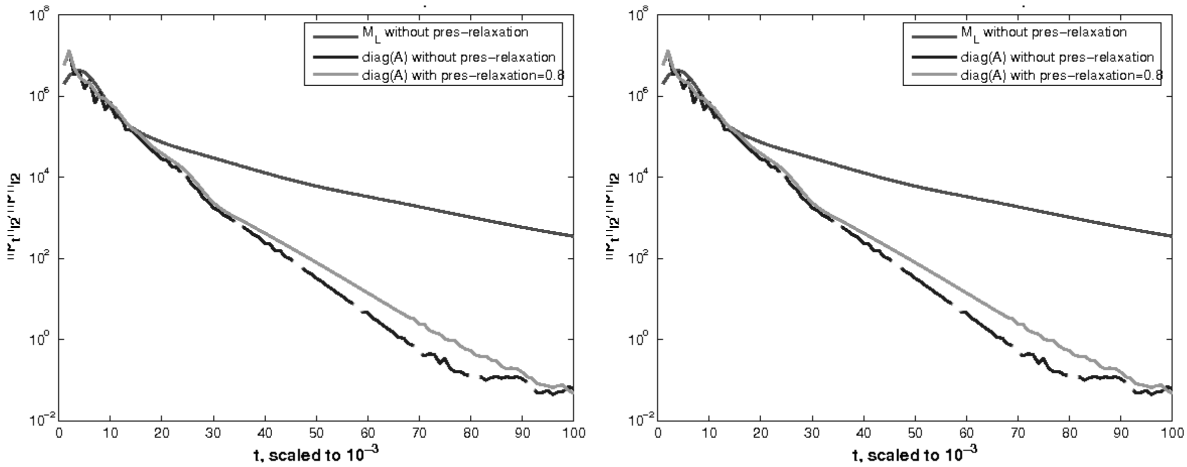

Fig. 5.14. Unit cube, $2\omega\Delta t = 2.0$, (LEFT) pressure, (RIGHT) velocity.

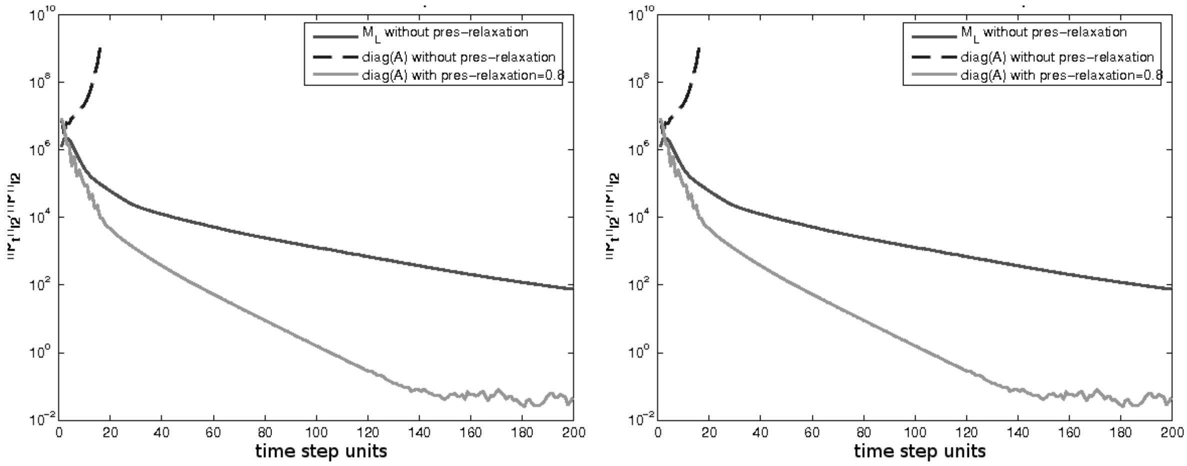

Fig. 5.15. Unit cube, $2\omega\Delta t = 3.0$, (LEFT) pressure, (RIGHT) velocity.

preconditioner converges with the rate of convergence being much slower than those of a relaxed $diag$ preconditioner.

Basing on the results presented above we can make the following concluding remarks concerning the proposed relaxation in the pressure subproblem:

- Relaxation helps to fasten the convergence to the steady state solution in comparison to the standard choice of the Schur Complement preconditioner even if the usage of $diag$ in its complete form is not possible.
- If oscillatory jumps in pressure do not lead to the divergence of the solver, then $diag$ is better than any other relaxed choice.

At the end of this section we would like to mention an observation, which was obtained during numerical tests of this kind. Namely, if velocities in Z-direction are small with respect to those in XY-plane, one can substitute matrices $\boldsymbol{M}_{\text{(diag)}}$ and $\boldsymbol{M}_{\text{(diag+coriol)}}$ by the following ones:

$$\boldsymbol{M}_{\text{(diagXY)}} = \begin{pmatrix} diag(A) & 0 & 0 \\ 0 & diag(A) & 0 \\ 0 & 0 & M_L \end{pmatrix} \quad \text{and} \quad \boldsymbol{M}_{\text{(diagXY+coriol)}} = \begin{pmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & M_L \end{pmatrix}. \tag{5.12}$$

The lumped mass matrix in the entry $(3,3)$ is independent on $\boldsymbol{u}$ and Δt and therefore preserves the whole system from the rapid growth of condition number and the subsequent difficulty with the iterative solving process. One one hand, it plays the stabilizing role of the standard preconditioner and, on the other hand, the upper two-by-two block matrix with Coriolis and convective parts produces the improved outer behaviour of the DPM.

Drawbacks of $P_{\text{(diagXY)}} = B^T \boldsymbol{M}^{-1}_{\text{(diagXY)}} B$ and $P_{\text{(diagXY+coriol)}} = B^T \boldsymbol{M}^{-1}_{\text{(diagXY+coriol)}} B$ are obvioius:

- Inability to use (5.12) in an arbitrarily oriented system of coordinates or when the velocity flow field is large in the Z-direction.
- Inability to control the smoothing effect of wiggles for $\|p_t\|$- and $\|\boldsymbol{u}_t\|$-curves, i.e. in general the convergence to the steady state is very far from being optimal.

Convergence behaviour of the iterative process with the preconditioner $\boldsymbol{M}_{\text{(diagXY)}}$ is shown in Fig. 5.16 in comparison to those with preconditioners$\boldsymbol{M}_{\text{(mass)}}$ and$\boldsymbol{M}_{\text{(diag)}}$.

5.4 Summary of the numerical results for the modified DPM

Summarizing these tests, we have shown that:

- The modified discrete projection method delivers better accuracy in time for pressure and velocity for medium and large values of $\omega\Delta t$ than the classical/nonmodified discrete projection method.
- The proposed explicitly inverted preconditioners both for momentum and pressure Poisson-like equations show much better convergence rate than corresponding standard preconditioners with the lumped mass matrix.
- Convergence to the steady state solution of the modified DPM is much faster. At the same time multigrid costs for the proposed Schur Complement preconditioners remain on almost the same level as those for the standard preconditioner.
- $\boldsymbol{M}_{\text{(diag+coriol)}}$ is the most efficient preconditioner among all considered, since it includes parts due to convective and Coriolis terms. For large values of $\omega\Delta t$ the convective part in $\boldsymbol{M}_{\text{(diag+coriol)}}$ may lead to the oscillatory

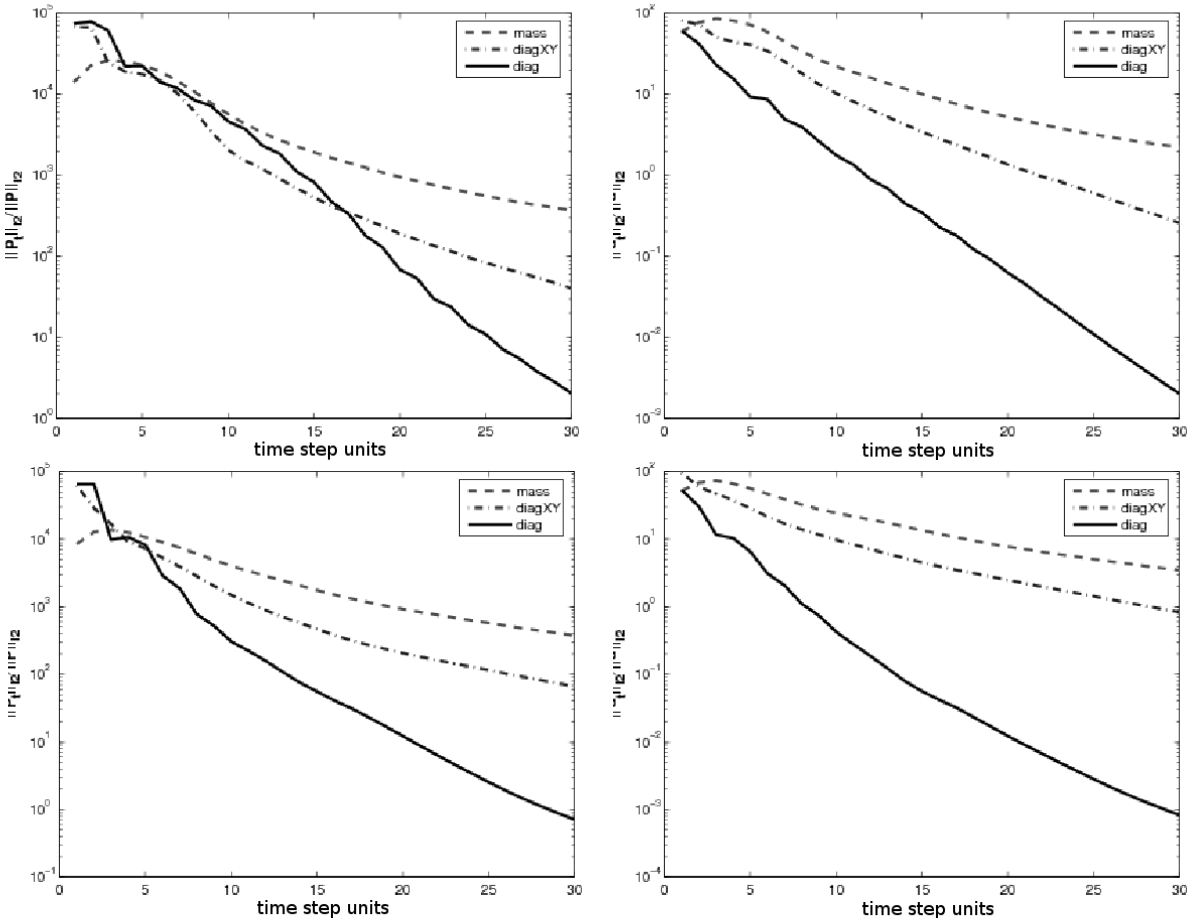

Fig. 5.16. Navier-Stokes equations (TOP) $2\omega\Delta t = 1.5$; (BOTTOM) $2\omega\Delta t = 2.5$.

behaviour of the iterative solver and, therefore, has to be stabilized. Proposed relaxation of the convective part on the stage of preconditioner's assembly or, later, during pressure correction step helps to avoid destabilizing behaviour. Nevertheless, further study of automatic choice of the relaxation parameter is required.

6

Numerical results for the rotation form of convection

This chapter addresses numerical aspects of the theoretical results described in the chapter 4.1. We are mainly interested in the behaviour of the rotation form of convection and the influence of the modified pressure on the iterative process and numerical solution of the incompressible system of the Navier-Stokes equations

$$\begin{aligned} \boldsymbol{u}_t + \boldsymbol{w}(\boldsymbol{u}) \times \boldsymbol{u} - \nu \Delta \boldsymbol{v} + \nabla P = \boldsymbol{f}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{aligned} \tag{6.1}$$

with $\boldsymbol{w}(\boldsymbol{u}) = \nabla \times \boldsymbol{u}$ and $P = p + \frac{\boldsymbol{u}^2}{2}$. As it was shown before, in the case of the coordinate transformation the Coriolis force can be naturally added into $\boldsymbol{w}(\boldsymbol{u})$ and the centrifugal force into P. We choose driven cavity and flow around cylinder benchmark problems to compare obtained numerical solutions for various Reynolds numbers with highly-accurate solutions available in literature [12, 62, 73, 108, 105]. Then, we examine the iterative behaviour of the algorithm, discuss some issues and suggest techniques to resolve them.

6.1 Driven cavity benchmark problem

We solve (6.1) in $\Omega = [0,1] \times [0,1]$ with Dirichlet boundary condition $\boldsymbol{u}_{\text{flow}} = (1,0)^T|_{y=1}$. For a space discretization we take a uniform Cartesian mesh. In geometric multigrid solver several grid levels are used, the finest level of which possesses 131584 degrees of freedom (edges), 66048 vertices and 65536 elements. We define the Reynolds number as $Re = \dfrac{U\,L}{\nu}$, where L is a length of the upper lid, U is a velocity value of the upper lid and ν is a parameter of viscosity. Reynolds numbers were taken in the following range: Re $\in \{100, 500, 1000\}$.

Numerical tests showed that with the increase of Reynolds numbers the iterative solver suffers in the nonlinear convergence. For example, for $Re = \{1000\}$ a fixed-point method with the linearized term $\boldsymbol{w}(\boldsymbol{u}^n) \times \boldsymbol{u}^{n+1}$ does not converge. To overcome this problem we did as follows:

- First, we implement a Newton-like method, when the contribution of the Fréchet derivative $\boldsymbol{w}(\boldsymbol{u}^{n+1}) \times \boldsymbol{u}^n$ is added to the linearized term $\boldsymbol{w}(\boldsymbol{u}^n) \times \boldsymbol{u}^{n+1}$.
- Second, we use the edge-oriented stabilization of the convective term, which was described in § 4.3. By varying the edge-oriented parameter γ_{edge} we can control in some sense the diagonal/nondiagonal ratio.
- Third, it was shown in [62, 73] that large norms of the Bernoulli pressure gradient $\nabla P = \nabla P(\boldsymbol{u})$ compared to the velocity H^2 norms may lead to a poor convergence of the finite element velocity if one does not include ∇div stabilization into the momentum equation of (6.1):

$$\nu(\nabla \boldsymbol{u}_h, \nabla \boldsymbol{v}_h) + N(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{v}_h) + \gamma_{div}(div\, \boldsymbol{u}_h, div\, \boldsymbol{v}_h) - (P_h, div\, \boldsymbol{v}_h) = (\boldsymbol{f}, \boldsymbol{v}_h) \quad \forall \boldsymbol{v}_h \in \boldsymbol{U}_h, q_h \in Q_h$$

where $N(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{v}_h)$ is a nonlinear term due to convection, $\boldsymbol{U}_h$ and Q_h are spaces of test functions for the velocity and the pressure, respectively. Otherwise, the dependence of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_1$ on ν is much milder. On the other hand, too large values of γ_{div} “overstabilize“ the problem and make corresponding linear algebraic system poor conditioned.

The proposed techniques help us to obtain convergence of nonlinear iterations for all Reynolds numbers (with different accuracy, of course). The comparison of profiles for u_1 and u_2 along cutlines $x = 0.5$ and $y = 0.5$, respectively, with those of reference solutions is presented in Figures 6.1-6.3.

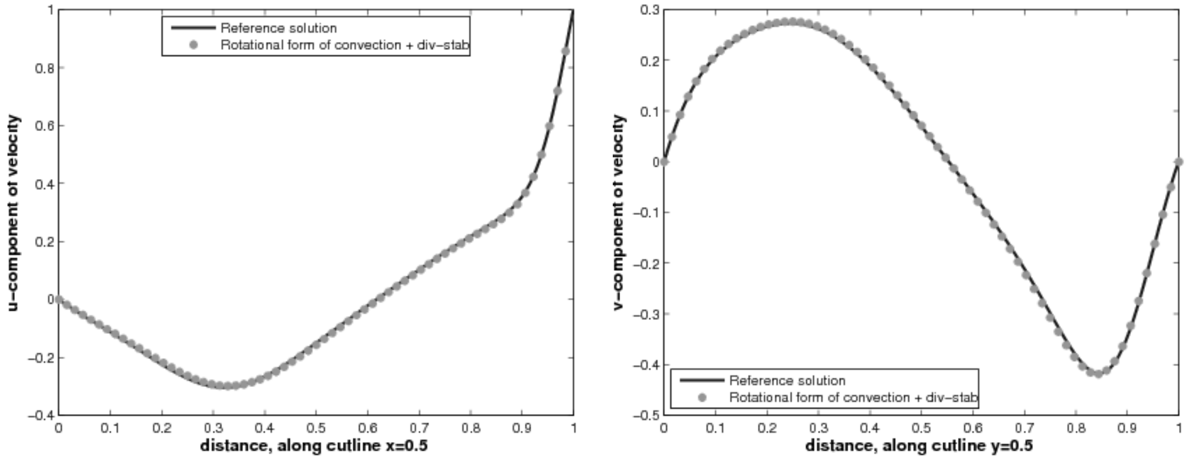

Fig. 6.1. Re=300 (TOP) $u_1|_{x=0.5}$; (BOTTOM) $u_2|_{y=0.5}$.

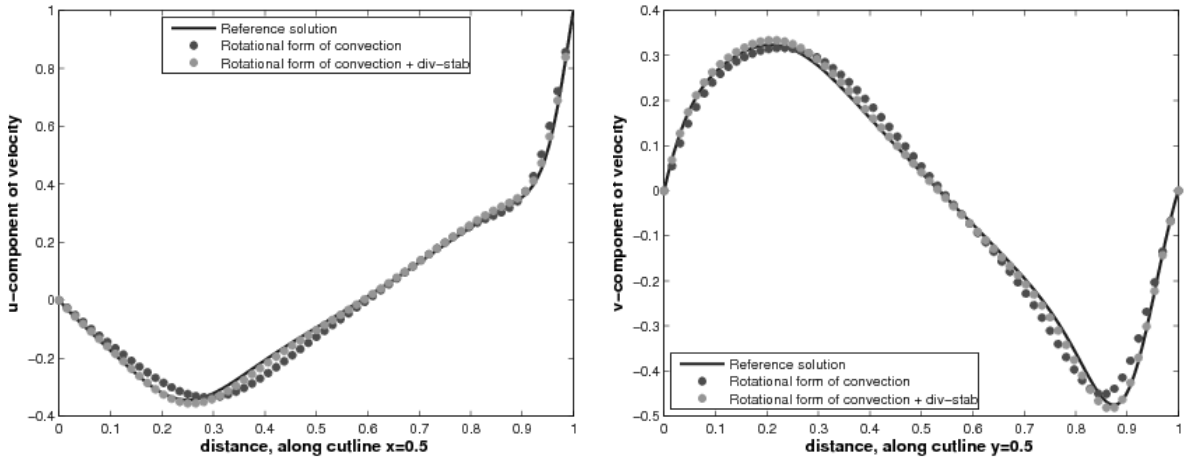

Fig. 6.2. Re=500 (TOP) $u_1|_{x=0.5}$; (BOTTOM) $u_2|_{y=0.5}$.

We notice that the discrepancy between numerical and references solution get bigger, if larger Re is used. Edge-oriented stabilization and Newton-like methods help to obtain the nonlinear convergence. At the same time the ∇div stabilization helps to improve accuracy, see Figures 6.2 and 6.3. Moreover, while using ∇div-stabilization one can perform simulation for smaller values of γ_{edge}, which are not possible otherwise.

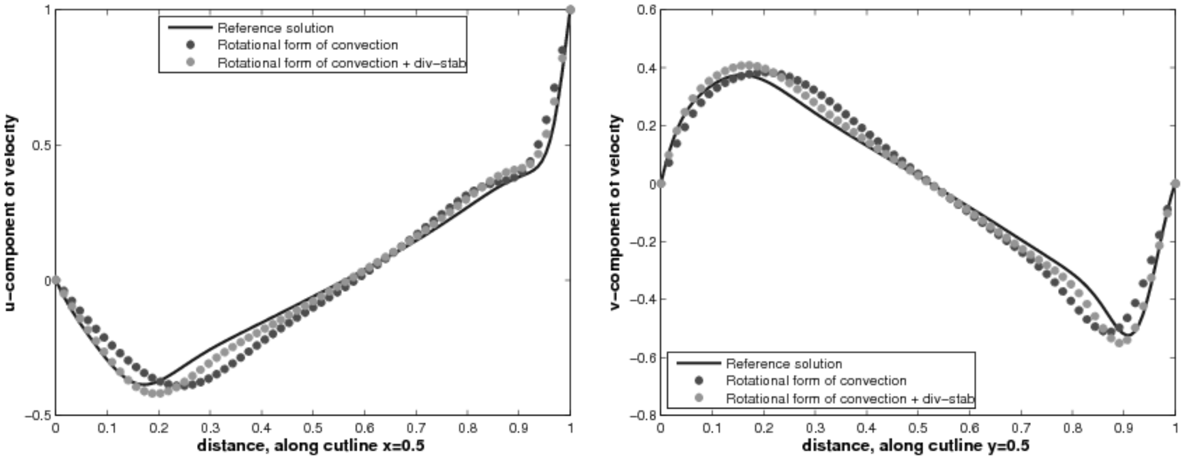

Fig. 6.3. Re=1000 (TOP) $u_1|_{x=0.5}$; (BOTTOM) $u_2|_{y=0.5}$.

Table 6.1. Kinetic energy. Comparison of standard and rotation form of convection.

structured mesh		Energy	Energy
Level	cells	without pressure separation	with pressure separation
		Standard form of convection, γ_{edge}=0.05	
4	1024	5.007440861022897E-002	5.211682419890467E-002
5	4096	4.726577915032740E-002	4.825706662364793E-002
6	16384	4.504907900502248E-002	4.548087349148145E-002
7	65536	4.451536462306269E-002	4.470287712512187E-002
8	262144	4.447346732722830E-002	4.455086811892013E-002
9	1048576	4.449511194952627E-002	4.452439567412509E-002
		Rotational form of convection, γ_{edge}=0.05 γ_{div}=0.6	
4	1024	5.917218667382527E-002	6.034126831523878E-002
5	4096	5.160284164224632E-002	5.198083422220890E-002
6	16384	4.651826977387501E-002	4.661805861205227E-002
7	65536	4.493929188531996E-002	4.496463428798708E-002
8	262144	4.460509057916964E-002	4.460508688884178E-002
9	1048576	4.453347774356769E-002	4.453346723495192E-002
		Standard form of convection, γ_{edge}=0.2	
4	1024	4.697275081040202E-002	4.751183897739284E-002
5	4096	5.060866534334785E-002	5.098802583646743E-002
6	16384	4.670217951963666E-002	4.686025705766404E-002
7	65536	4.495614575196347E-002	4.502254198984097E-002
8	262144	4.457381775562132E-002	4.460223628885603E-002
9	1048576	4.451922086434779E-002	4.451922885437246E-002
		Rotational form of convection, γ_{edge}=0.2	
4	1024	4.534717050856769E-002	4.708663256505627E-002
5	4096	4.900355289002138E-002	5.064787120518587E-002
6	16384	4.583846271124172E-002	4.672960606432790E-002
7	65536	4.455734529223019E-002	4.498479522355114E-002
8	262144	4.439685360860982E-002	4.459296577837202E-002
9	1048576	4.444239174269192E-002	4.452908673732455E-002

Aiming for the better accuracy we also examine the use of the pressure separation technique (PSepA) [108]. Its purpose is to improve the incompressible flow simulations for problems with large pressure gradients, exactly what we need in the case of large Bernoulli pressure gradient $\nabla P = \nabla P(\boldsymbol{u})$. In Table 6.1 we present the kinetic energy for various mesh levels, calculated with and without pressure separation technique, and compare its values with those obtained for the standard convection. For the numerical results in the case of the standard convective term and the corresponding discussions the reader is referred to [108].

From graphics in Figures 6.1, 6.2, 6.3 and values of kinetic energy in Table 6.1 we see that:

- Numerical solution for the Navier-Stokes equations with the rotational convection is more accurate if smaller Reynolds number is used. For $Re = 300$ the reference solution and the numerical solution of the 'rotational convection' almost coincide. For larger Reynolds numbers discrepancies in velocity profiles are observable.
- For larger Reynolds one observes a poor convergence rate of nonlinear iterations for the Navier-Stokes equations with the rotational convection. For example, for $Re = 1000$ the solver does not converge if no additional techniques are implemented. The Newton-like method the and edge-oriented stabilization help to improve nonlinear convergence and accuracy of numerical solutions.
- Accuracy of the numerical solution, which we estimate by values of kinetic energy in Table 6.1, is better if finer mesh is used.

6.2 Flow around cylinder benchmark problem

As the second test case we take the flow around cylinder benchmark problem. The coarse mesh is shown in Fig. 6.4. The finest mesh level of the geometric multigrid solver possesses 267072 d.o.f., 133952 vertices and 133120 elements.

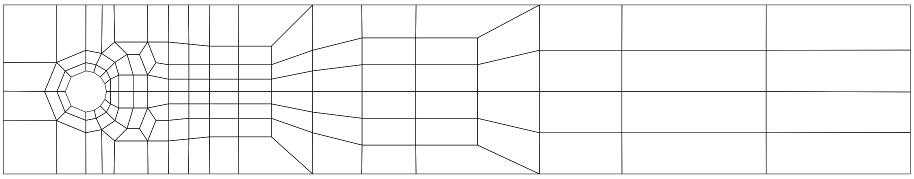

Fig. 6.4. The coarse mesh for flow around cylinder benchmark.

In this case we again define the Reynolds number as $Re = \dfrac{U\,L}{\nu}$, where U is an average velocity of inflow, L is a diameter of a cylinder and ν is a parameter of viscosity. For small Reynolds numbers (i.e. Re < 20) discrepancies between the obtained velocity field for rotation convection and those for standard convection are very small and iterative behaviour of our solver shows almost no difficulties. For large Reynolds numbers (Re ≥ 20) one has to resort to the help of stabilization techniques described in the previous section. We perform simulations for Re $= 20$, measure drag and lift forces on the boundaries of the inner cylinder and compare their values with those obtained for the case of standard convection [108]. Results are presented in Table 6.2.

Table 6.2. Darg and Lift. Comparison of standard and rotation form of convection.

structured mesh		Without pres. separation	With pres. separation
Level	cells	Drag/Lift	Drag/Lift
		Standard form of convection, γ_{edge}=0.3	
3	2080	0.77197D+01 / 0.21923D-01	0.79200D+01 / 0.23073D-01
4	8320	0.58216D+01 / 0.11291D-01	0.58517D+01 / 0.11433D-01
5	33280	0.56091D+01 / 0.10591D-01	0.56096D+01 / 0.10640D-01
6	133120	0.55822D+01 / 0.10513D-01	0.55784D+01 / 0.10529D-01
		Rotational form of convection, γ_{edge}=0.3	
3	2080	0.77898D+01 / 0.23215D-01	0.79742D+01 / 0.24085D-01
4	8320	0.58423D+01 / 0.11326D-01	0.58537D+01 / 0.11402D-01
5	33280	0.56142D+01 / 0.10612D-01	0.56154D+01 / 0.10609D-01
6	133120	0.55835D+01 / 0.10526D-01	0.55799D+01 / 0.10519D-01

In general, for the flow around cylinder test case one encounters the same problems of diagonal nondominance, slow nonlinear convergence rate and large norm of the Bernoulli pressure gradient as in the case of the lid-driven cavity from § 6.1. From the results of Fig. 6.2 we conclude that with the help of edge-oriented stabilization for convective term, ∇div-stabilization for the pressure gradient and Newton-like method for 'improved' nonlinear convergence one can tend to the sufficient level of accuracy.

6.3 Summary of the numerical results for the rotation form of convection

In numerical simulations of the Navier-Stokes equations with convective term written in the rotation form (6.1) for medium and large Reynolds numbers one has to take care of two major problems. The first one is a slow nonlinear convergence. Numerical experiments showed that in order to improve nonlinear convergence one may add a contribution of the Fréchet derivative $\boldsymbol{w}(\boldsymbol{u}^{n+1}) \times \boldsymbol{u}^n$ to the linearized term $\boldsymbol{w}(\boldsymbol{u}^n) \times \boldsymbol{u}^{n+1}$. The second problem is a large gradient of the Bernoulli pressure $P = p + \nabla \frac{\boldsymbol{u}^2}{2}$. Large gradients of the Bernoulli pressure have to be resolved on a very fine mesh, otherwise they lead to the loss of accuracy for the velocity field $\boldsymbol{u}$. Without some extra techniques to be implemented the simulation is not possible. We showed that the edge-oriented stabilization and the $\nabla\, div$-method help to guarantee convergence and to obtain sufficiently accurate solution.

Part II

Implementation and Future Extensions

7

Implementation

For the simulation of the Navier-Stokes equations a lot of CFD software are available. One can choose either between commercial codes such as `CFX` [44], `StarCD` [100], `Fluent` [32] or non-commercial codes, among which are `OpenFoam` [77], `Deal.II` [26], `MooNMD` [70] and many others. While choosing any of them, one should take into account many aspects, e.g. price of a license, sufficient accuracy of the obtained results, capability to modify desired subroutines and parts of a code, etc. As the 1995 DFG benchmark showed [110], even commercial CFD packages can fail already for laminar flow calculations. For the numerical calculations presented in this work we chose the home-developed free-source CFD code `Featflow`, which fulfils all our requirements. In this chapter we will give a short introduction into `Featflow`, describe its structure and aspects of pre- and post-processing steps, show specific details concerning the realisation of our modified projection scheme (2.43)-(2.45) and mention direction of further development and modernization of the software. Detailed information about `Featflow` can be found in [6], [105] and references therein or at www.featflow.de.

7.1 About Featflow

`Featflow` is ”Finite element software for the incompressible Navier-Stokes equations”. It was developed at the University of Heidelberg within the working group of R. Rannacher and S. Turek (the list of involved persons can be found in Appendix B of [5]). `Featflow` is a set of programs/solvers based on the finite element libraries `Feat2D` [10] and `Feat3D` [39], which were originally written in `Fortran 77`. The most advantageous characteristics of `Featflow` are following:

1. powerful geometric multigrid solver [80, 105],
2. mixed Finite Element method with stable nonconforming Rannacher-Turek elements [88],
3. fast projection schemes for the system of the Navier-Stokes equations [104, 105].

In 2007 Michael Köster from TU Dortmund completely reorganized old packages into carefully documented data-encapsulated code written in `Fortran 90`, by this giving birth to the second release of `Featflow 2` available at www.featflow.de.

The structure of `Featflow` can be divided into two parts: an outer ‘user-setting’ part, to be described in § 7.2.2, and an inner solution part, which provides an approximate solution of discretized system for the given configuration. The inner part can be considered as a ”solver engine”, which has to be modified only if some changes in the algorithmic flow of a scheme is supposed, e.g. realisation of the modified projection scheme.

By treating the nonlinear problems `Featflow` solvers can be divided into two approaches:

1. In the first one the nonlinearity is treated by outer nonlinear iterations of fixed-point or quasi-Newton type or by a linearization technique through extrapolation in time. Then the obtained linear indefinite subproblems of Oseen type are solved by a coupled or a splitting approach (`cc2d/cc3d` solvers).
2. In the second one, the coupled system is first split to obtain definite subproblems in $\boldsymbol{u}$ (Burgers-equations) as well as in p (linear pressure-Poisson problems). Then the nonlinear subproblems in $\boldsymbol{u}$ are treated by an appropriate iteration or a linearization technique (`pp2d/pp3d` solvers).

These approaches form the Navier-Stokes tree of `Featflow` solvers. Numerical tests [103] have shown that the solvers `pp2d/pp3d` are superior to `cc2d/cc3d`, especially for highly nonstationary flows. Moreover, one can easily verify that the `pp2d/pp3d` approach fits into the concept of the modified projection scheme (2.43)-(2.45). This was the reason why we took its code as a basis for the programming realisation of the proposed modified projection scheme (2.43)-(2.45).

At the end of this section we would like to add couple words about potentials for a `Featflow` user. Namely, in a `Featflow` code one can easily include any extrenal mathematical library, such as `BLAS`, `LAPACK`, `UMFPACK`, etc. A user can couple it with some external mathematical or engineering tools, e.g for a `Featflow-Matlab` coupling see [41], for a `Featflow-Parsival` interaction see [52]. The open source nature helps to modify any part of pre-, post- or solving steps. A `Featflow` user can choose from large variety of stabilization techniques, preconditioners, viscous stress tensors, building-blocks for convection-diffusion transport equations and other helpful already implemented features. At the moment a large scientific group of Stefan Turek in TU Dortmund keeps working on further extensions of the `Featflow` code in the directions of population balance and turbulence models, visco-elastic flows, optimization techniques, particulate flows, parallelization, etc. Detailed information can be found at www.mathematik.uni-dortmund.de/lsiii.

7.2 Preprocessing

The preprocessing phase in `Featflow` consists of construction of an appropriate mesh, which has to be written according to the `Featflow` format, setting of required data parameters, prescription of boundary conditions for the velocity, assignment of the right hand side and parametrization/smoothing of a mesh for finer levels. Mesh construction will be described in § 7.2.1. Setting of data parameters, boundary conditions, etc., will be provided in § 7.2.2.

7.2.1 Grid generation

As it was mentioned in the chapter 2, a grid for the `Featflow` simulation must be constructed of quadrilateral elements for the 2D case and of hexahedrons for the 3D case. The main restrictions on the mesh are following:

1. All elements have to be convex and contain no ‘hanging’ nodes, see Fig. 7.1.
2. Every element is described by a series of vertices. In 2D a series consists of 4 vertices, orders of which must correspond to anticlockwise direction. In 3D a series consists of 2×4 vertices, where the first and the last 4 vertices should form quadrilaterals in anticlockwise direction.

3. No element can have vertices, which belong to different boundaries of a given geometry.
4. Elements should be of moderate aspect ratios, i.e. too long and thin elements should be avoided, if possible.

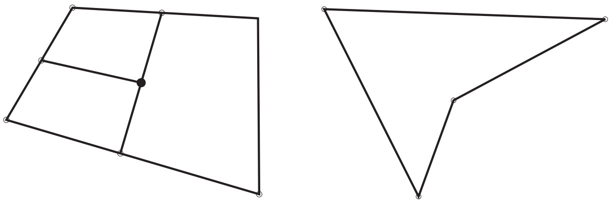

Fig. 7.1. Forbidden elements, (LEFT) hanging node, (RIGHT) non-convex element.

In general, it is much more complex to cover a domain with quadrilateral elements than with triangular ones. Nevertheless, the recently developed `DeViSoRGrid3` [68] offers a user broad opportunities concerning mesh construction and visualization. This java-written program with a friendly interface can provide a user with helpful information about nodes, edges, elements and boundaries of a mesh. It can automatically fulfil a domain with quadrilaterals of a given radius/mesh size, check the constructed mesh for validity, etc. Having a 2D mesh (e.g. again to construct it in `DeViSoRGrid3`), one can easily prolong it into the three-dimensional space by a layer-to-layer algorithm, the so-called ‘sandwich’ mesh generation. If direct ‘sandwich’ style of construction is not possible, e.g. in the case of considered stirred tank reactor geometries from Figures 8.1 and 8.2, one can resort to the help of special scripts, which rewrite boundaries of a domain in the required format and automatically fulfil a domain with quadrilaterals. For the latest release of `DeViSoRGrid3` check www.featflow.de.

7.2.2 User settings

We would like to note, that `Featflow` programs solve the Navier-Stokes equations in their dimensionless form. To characterizes the flow field one can think of the Reynolds number. Thus, if one works with experimental data, all required parameters and boundary conditions for velocity have to be scaled in such a way as to correspond to a certain Reynolds number.
User settings in Featflow are divided into two categories:

1. *solver parameters*,
2. *code prescribed settings*.

Solver parameters are the corresponding parameters for the differential subproblems. They are located in the separate `pp2d.dat` or `pp3d.dat` files and can be changed in their values without recompiling the program. A list of some solver parameters together with their short descriptions for the code of the modified discrete projection method is shown in Table 7.1.

Code prescribed settings include boundary conditions for the velocity, assignment of the right hand side and parameterisation/smoothing of a mesh for finer levels. All of them are to be written in the user-oriented files in `Fortran` programming language. After changes are made, the program has to be recompiled. An advanced `Featflow` user can modify not only these files, but any block of `Featflow` solver or `Feat2D`, resp., `Feat3D` libraries, as long as he or she clearly understands interaction of added blocks with specifics of the original code, which can be either read in documentation or found out from developers.

Table 7.1. Short list of selected parameters for the modified projection code.

CPARM	name of parameterisation file
CMESH	name of coarse mesh file
CFILE	name of protocol file
ISTART	input of start vector =0: start with homogeneous vector (only boundary conditions) $\neq$0: read start vector
CSTART	name of start vector file
ISOL	output of solution vector =0: no output; $\neq$0: output
CSOL	name of solution vector file
NLMIN / NLMAX	minimal / maximal multigrid level
IMASS	parameter for mass matrix type (lumped of real mass matrix)
IMASSL	parameter for element type of lumped mass matrix
IUPW	parameter for convective terms (streamline diffusion, upwinding or edge-oriented stabilization)
INLMIN / INLMAX	minimal / maximal number of nonlinear iterations used for the solution of the nonlinear transport-diffusion equations INLMIN = INLMAX = 1: linear extrapolation in time INLMIN = INLMAX = -1: constant extrapolation in time
ICYCU / ICYCP	parameter for multi-cycle for velocity / pressure (F-, V-, or W-cycle)
ILMINU / ILMINP	minimal number of multigrid steps for velocity / pressure
ILMAXU / ILMAXP	maximal number of multigrid steps for velocity / pressure
ISMU / ISMP	parameter for multigrid-smoother for velocity / pressure (Jacobi, SOR, SSOR or ILU(k))
ISLU / ISLP	parameter for multigrid-solver for velocity / pressure (SOR, BiCGSTAB, CG, ILU(k), or BiCGSTAB+ILU(k) / CG+ILU(k))
IJUMP	kind of the jump term
DJUMP	relaxation parameter for the jump
INEWTON	Newton method: 1=in, 0=out
DNEWTON	relaxation parameter of the Newton method
BCORLS	Coriolis force term: TRUE=in, FALSE=out
BMSCMP	projection scheme: TRUE=modified, FALSE=nonmodified
BRCONV	type of convection: TRUE=rotation, FALSE=standard
DHDIV	parameter γ for ∇div stabilization
ANGVEL	angular velocity
RE	parameter for viscosity (Reynolds number)
NITNS	maximum number of macro time steps
TIMENS	absolute start time
THETA	parameter for time-stepping value (Implicit Euler or Crank-Nicolson, only used if IFRSTP=0)
TSTEP	starting time step
IFRSTP	parameter for time-stepping scheme (one-step or fractional step scheme)
DTGMV	time difference for gmv-output
IGMV	level for gmv-output
TIMEMX	maximum absolute time
IADTIM	parameter for adaptive time-step control =0: no control, fixed time step TSTEP is used $\neq$0: time step control is used
DTMIN / DTMAX	minimal / maximal time step during adaptive control
DTFACT	factor for largest possible time step changes

7.3 Postprocessing

For visualization of obtained numerical results we use a freely available visualization `GMV` program (short for `General Mesh Viewer`), see [78]. `GMV` is a quite powerful postprocessing tool, at least if to compare among freely available software. Knowing `GMV` format, one can prescribe the output of any wanted parameters: i.e. to calculate and to output required vector/scalar fields, to set and to colour materials and boundaries, to prescribe cutlines, cutplanes, iso-lines/surfaces, to set various visual attributes and so on.

The proposed idea of the coordinate transformation concept brings us to the necessity of calculating and setting boundary conditions for the transformed velocity and pressure, but outputting non-transformed user-accustomed values of the velocity and pressure during the postprocessing phase. Thus, prescription of Dirichlet boundary conditions for the velocity $\boldsymbol{v} = -\boldsymbol{\omega} \times \boldsymbol{r}$ for problems, presented in Figures 8.1 and 8.2, will look as follows:

```
!******************************
...
angvel = -some_positive_value
...
!   v_x component
fdatin = y*angvel
...
!   v_y component
fdatin =-x*angvel
!******************************
```

where we assume that blades of a propeller rotate clockwise direction with the angular velocity $\boldsymbol{\omega} = (0, 0, -\text{some\_positive\_value})$.
After calculating in `FeatFlow` the approximation tuple $(\boldsymbol{v}, P)$, but before outputting its values into the `gmv`-formatted file, we have to perform a 'backward'-coordinate transformation to output a user-acquainted values $(\boldsymbol{u}, p)$, see Fig 7.2.

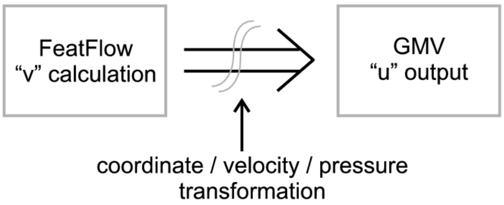

Fig. 7.2. Transformation for the GMV output

A 'backward'-coordinate transformation of geometry is done according to the formula $\boldsymbol{x} = \boldsymbol{U}^T\boldsymbol{\xi}$. In the `Featflow`-code it reads as follows:

```
!******************************
alpha = angvel*timens
...
x_trans_coord = real(dcorvg(1,ivt))
y_trans_coord = real(dcorvg(2,ivt))
```

```
x_coord = cos(alpha)*x_transformed_coord -
          sin(alpha)*y_transformed_coord
...
y_coord = sin(alpha)*x_transformed_coord +
          cos(alpha)*y_transformed_coord
!*****************************
```

A ‘backward’-transformation for the velocity $\boldsymbol{v} = -\boldsymbol{\omega} \times \boldsymbol{r}$ can be written in the code as

```
!*****************************
! v_x component
write(munit,1000) (v_x(ivt)-real(dcorvg(2,ivt))*angvel)
...
! v_y component
write(munit,1000) (v_y(ivt)+real(dcorvg(1,ivt))*angvel)
!*****************************
```

For a ‘backward’-pressure transformation $P = p - \frac{1}{2}(\boldsymbol{\omega} \times \boldsymbol{r})^2$ one can write

```
!*****************************
write(munit,1000) (p(ivt) +              &
          0.5*angvel*angvel*             &
        ( dcorvg(1,ivt)*dcorvg(1,ivt) +  &
          dcorvg(2,ivt)*dcorvg(2,ivt) )  )
!*****************************
```

8

Future Extensions

The proposed approach of coordinate transformation was motivated by extensive work lead in the numerical simulation of population balance models for turbulent flow fields in stirred tank reactor geometries [41, 52, 91, 92]. Here, the moving boundaries of a propeller have to be approximated in a very precise way not only because they are sources of main ‘perturbations’ in the velocity flow field to be used further in the population balance modeling, but also because in many turbulence models (e.g. $k - \varepsilon$, $k - \omega$, etc.) the prescription of boundary layers is of supreme importance, see e.g. [47, 60, 114]. The main purpose of the current chapter is to show that the modified projection scheme can be used not only for simple test problems presented in the cahpter 5, but also for complex 3D simulations of real-life models. As a representative of such models we chose the stirred tank reactor. In the subsequent text we will apply the modified DPM to calculate the flow field in the stirred tank. Afterwards, we will examine convergence of the iterative solver for various preconditioners, among which are newly proposed and classical ones. Then, we will use a particle tracing tool to obtain a visual understanding of the flow field in a tank. And finally, we will give a brief insight into the future CFD extensions, where the modified projection method can be of big importance. This is, namely, the population balance modeling for turbulent flows in the stirred tank reactor.

8.1 Numerical results for the Stirred Tank Reactor

In the chapter 5 we observed that the modified discrete projection method performs in a very advantageous manner delivering good multigrid rates and faster convergence to the steady state solution. Nevertheless, it is a known fact that very often the mesh complexity plays a crucial role in the numerical behaviour of the algorithm. Taking this fact into consideration one might ask a fair question. Namely, what if the proposed algorithm is applied to some complex real-life 3D model, will it produce the same numerical behaviour as we expect and which we observed for the unit cube geometry?

To give an answer on this question we decided to consider two configurations from a class of Stirred Tank Reactors (STR). This choice was due to the following reasons:

- *Geometrical*: STR possesses complex enough geometry.
- *Transforming*: STR makes it is possible to transform from the usual inertial frame of reference to the nonitertial one, and thus to substitute the system of the Navier-Stokes equations by a new one with rotational forces.

- *Scientific*: STR is widely used in industry and was intensively explored in the series of works: modeling and simulation of drop size distributions in stirred liquid-liquid systems [52, 91, 92] and flow control of turbulence models [41].

Drafts of the STR configurations 1 and 2 are presented in Figures. 8.1 and 8.1, respectively. Mesh characteristics are given in Table 8.1, where notation is similar to those used in Table 5.1.

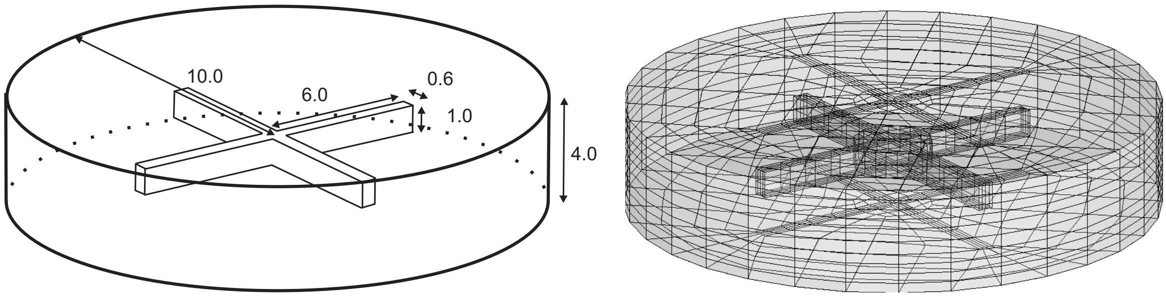

Fig. 8.1. STR configuration 1, (LEFT) Design draft, (RIGHT) Mesh on the 2d level.

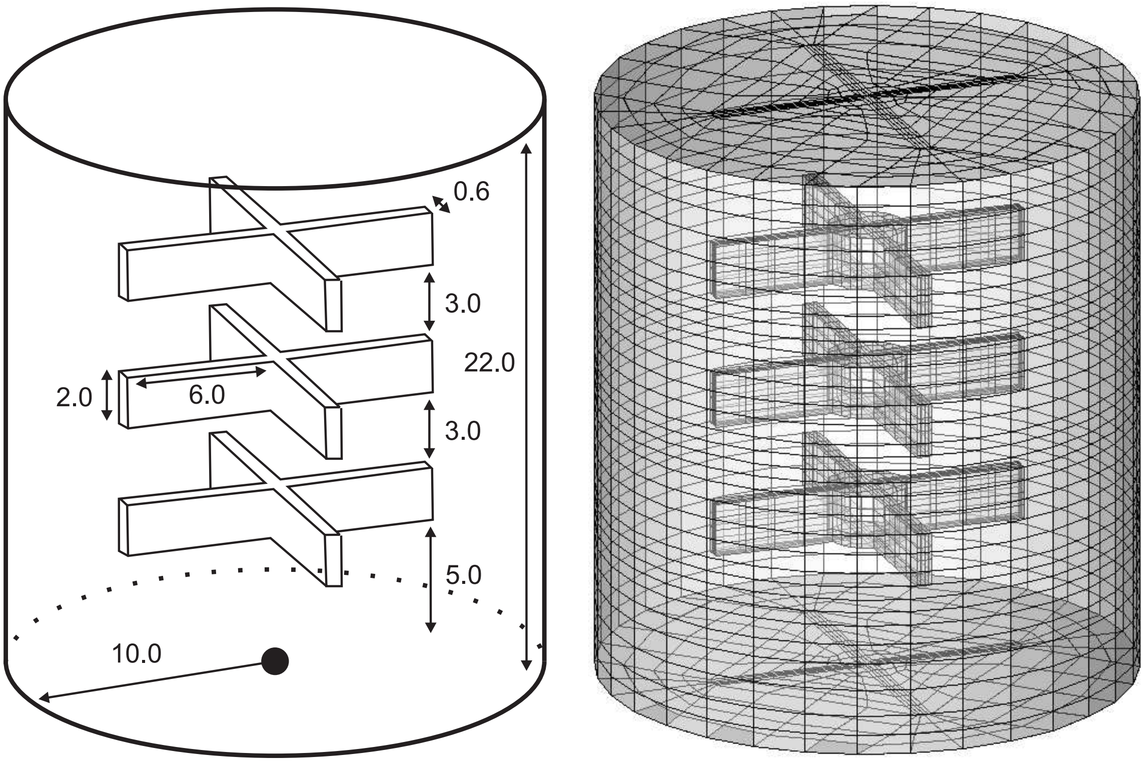

Fig. 8.2. STR configuration 2, (LEFT) Design draft, (RIGHT) Mesh on the 2d level.

Table 8.1. Characteristics of STR meshes.

level	NVT	NAT	NEL	NEQ
One-propeller STR configuration				
1	510	1,216	352	4,000
2	3,450	9,088	2,816	30,080
3	25,074	70,144	22,528	232,960
4	190,434	550,912	180,224	1,832,960
Three-propellers STR configuration				
1	1,406	3,528	1,048	11,632
2	9,864	26,688	8,384	88,448
3	73,100	207,360	67,072	689,152
4	560,916	1,634,304	536,576	5,439,488

8.1.1 Multigrid with smoother C for velocity problems

In the same way as we did in § 5.2.1 we test three preconditioning approaches for solving the velocity subproblem (2.25) for the chosen STR configurations. The first two schemes include standard pointwise SOR methods with the following preconditioners:

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

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

The third variant is the block-diagonal preconditioner $\boldsymbol{C}_{coriol}$ from (2.26). Both configurations gives almost the same data results (see Table 8.2), which are equivalent to those presented in Table 5.2.

Table 8.2. Number of multigrid iterations of the momentum equation.

Preconditioner	$\omega\Delta t$	Meshing level		
		2	3	4
$\boldsymbol{C}_{SOR}$	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	4	4
$\boldsymbol{C}_{coriol}$	60	2	2	2
$\boldsymbol{C}_{SOR}$	600	div	div	div
$\boldsymbol{C}_{SORcoriol}$	600	>100	>100	>100
$\boldsymbol{C}_{coriol}$	600	2	2	2

8.1.2 Multigrid solver for the modified pressure equation

As the next step, we check the convergence behaviour of the modified pressure Poisson equation and show that the observed behaviour of multigrid convergence rates has the same tendency as those shown in Tables 5.3–5.6. It can be also seen that the mesh complexity increases values of the average multigrid factor, which are presented in Tables 8.3 and 8.4.

Table 8.3. Multigrid convergence rates for different preconditioners $P = B^T \boldsymbol{M}_{(\cdot)}^{-1} B$ with 4 smoothing steps, resp., 2 smoothing steps for BiCGStab, 3d level.

Smoother	$2\omega\varDelta t$			
	0.05	0.5	5.0	50.0
$M_{\text{(mass)}}$				
SOR	0.50+00	0.50+00	0.50+00	0.50+00
ILU(1)	0.17-01	0.17-01	0.17-01	0.17-01
ILU(3)	0.75-03	0.75-03	0.75-03	0.75-03
BiCGStab(ILU(1))	0.19-02	0.19-02	0.19-02	0.19-02
BiCGStab(ILU(3))	0.47-03	0.47-03	0.47-03	0.47-03
$M_{\text{(mass+coriol)}}$				
SOR	0.50+00	0.51+00	0.81+00	div
ILU(1)	0.17-01	0.19-01	0.59-01	0.26-01
ILU(3)	0.75-03	0.75-03	0.48-02	0.28-02
BiCGStab(ILU(1))	0.18-02	0.18-02	0.61-02	0.30-02
BiCGStab(ILU(3))	0.47-03	0.36-03	0.21-02	0.18-02
$M_{\text{(diag)}}$				
SOR	0.46+00	0.31+00	0.41+00	0.49+00
ILU(1)	0.13-01	0.32-01	0.20+00	0.35+00
ILU(3)	0.23-02	0.76-02	0.81-01	0.19+00
BiCGStab(ILU(1))	0.31-02	0.83-02	0.45-01	0.88-01
BiCGStab(ILU(3))	0.96-03	0.18-02	0.20-02	0.43-02
$M_{\text{(diag+coriol)}}$				
SOR	0.46+00	0.34+00	0.56+00	0.68+00
ILU(1)	0.13-01	0.34-01	0.14+00	0.16+00
ILU(3)	0.23-02	0.79-02	0.38-01	0.40-01
BiCGStab(ILU(1))	0.31-02	0.85-02	0.23-01	0.28-01
BiCGStab(ILU(3))	0.96-03	0.17-02	0.13-02	0.19-02

8.1.3 Unsteady simulation for the STR configurations

Now we are ready to perform full unsteady numerical simulations for two configurations of the Stirred Tank Reactor. The main characteristics besides the presented in Fig. 8.1 (left) and Fig. 8.2 (left) are chosen to be as follows (all measures are given in non-dimensional form):

1. Configuration 1: Kinematic viscosity $\nu = 0.1$, velocity of inflow $\boldsymbol{u}_{inflow} = 40$, angular velocity $\omega = 2\pi$, Ekman number $Ek \approx 0.0004$, where $Ek := \dfrac{\nu}{\omega L_{blade}^2}$. In the case of mixers the Ekman number is such that $Ek = \dfrac{1}{Re}$, where Re is the Reynolds number.
2. Configuration 2: Kinematic viscosity $\nu = 0.1$, velocity of inflow $\boldsymbol{u}_{inflow} = 40$, angular velocity $\omega = 2\pi$, Ekman number $Ek \approx 0.0004$.

Table 8.4. Multigrid convergence rates for the preconditioner $P = B^T \boldsymbol{M}^{-1}_{(diag+coriol)} B$ for different levels with 4 smoothing steps, resp., 2 smoothing steps for BiCGStab.

level	$2\omega \Delta t$			
	0.05	0.5	5.0	50.0
SOR				
2	0.35+00	0.35+00	0.57+00	0.65+00
3	0.46+00	0.34+00	0.56+00	0.68+00
4	0.40+00	0.40+00	0.60+00	0.65+00
BiCGStab(ILU(1))				
2	0.85-03	0.91-03	0.45-02	0.76-02
3	0.31-02	0.85-02	0.23-01	0.28-01
4	0.53-02	0.98-02	0.23-01	0.38-01

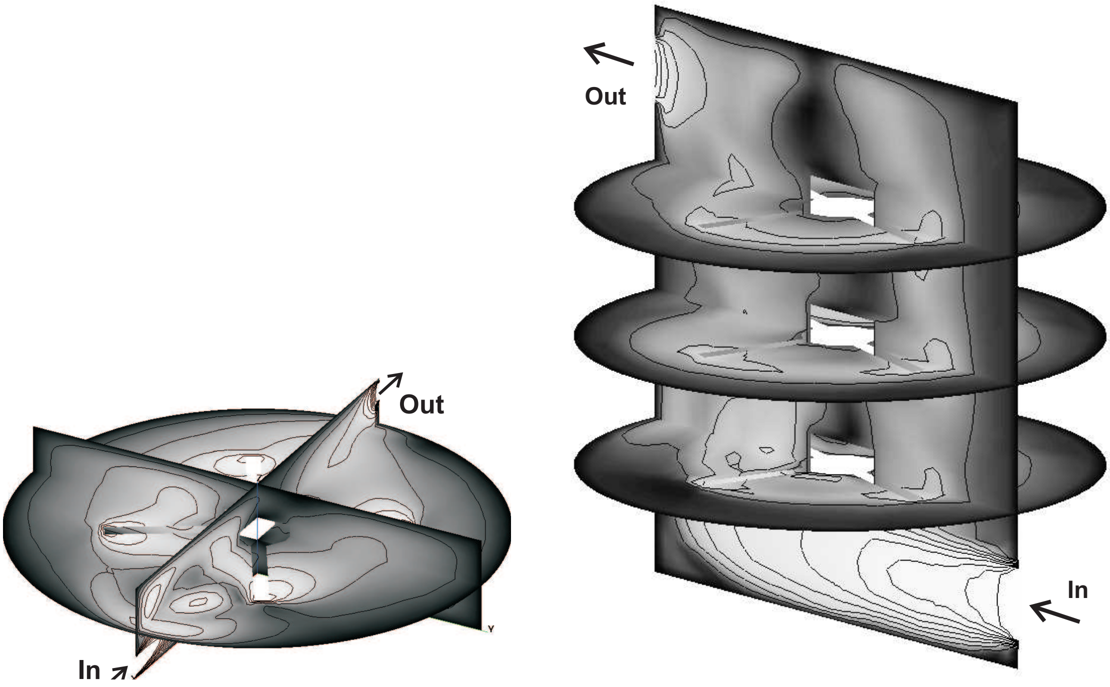

Fig. 8.3. Numerical simulation, (LEFT) For the configuration 1, (RIGHT) For the configuration 2.

In every case fluid enters the tank through an inlet near the bottom, then it is 'mixed' by the rotating propeller and leaves the stirred tank through an outlet located on the top, see Fig. 8.3. Inlet and outlet are prescribed on the outside wall as inhomogeneous Dirichlet and homogeneous Neumann boundary conditions, resp. The inlet is prescribed as a triangular region:

- *Configuration 1*: width $\phi \in [-\frac{\pi}{32}, \frac{\pi}{32}]$, height $Z \in [0.2, 1.2]$.
- *Configuration 2*: width $\phi \in [-\frac{\pi}{32}, \frac{\pi}{32}]$, height $Z \in [1.0, 4.0]$.

Outlet is located on the opposite side of the tank and defined as follows:

- *Configuration 1*: width $\phi \in [\pi - \frac{3\pi}{64}, \pi + \frac{3\pi}{64}]$, height $Z \in [2.8, 3.8]$.
- *Configuration 2*: width $\phi \in [\pi - \frac{3\pi}{64}, \pi + \frac{3\pi}{64}]$, height $Z \in [18.0, 21.0]$.

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, see Fig. 8.4. Movies of the velocity field can be found at www.mathematik.uni-dortmund.de/lsiii/download/sokolow.

Particle tracing

Usually, it might be a difficult task to make concluding remarks about the flow field in the 3D geometry. Moreover, very often the main interest of the simulation is not the flow field itself, but a mixture of some sources/species inside of the reactor. Injection of the particle tracers into the geometry of the STR helps to evaluate both the propagation of the velocity field and the mixture of the particles.

For these reasons we took the explicit time-stepping particle tracing tool `GMVPT` [1], which was developed by Jens Acker at TU Dortmund. `GMVPT` matches with `Featflow` output data in the `GMV` format and uses these data to calculate propagation of particles in the obtained flow field. For the best visualization one can choose number of sources, their colours, location of source injectors and other helpful features.

Here we demonstrate a pair of postprocessing simulations done by a particle tracing tool `GMVPT`. In the first case prescribed near the inlet three sources of particles, which can be distinguished by its colour: green, yellow and red, respectively (see Fig. 8.5). Six snapshots at the succeeding time steps give a realistic understanding of the flow motion and show a good mixing process of particles. In the second case, the domain of prescription we leave the same, but now we colour particles with respect to their duration of stay in the tank: starting from blue as the youngest particle and following the RGB colour scale up to red as the oldest particle. Six snapshots at the succeeding time steps are shown in Fig. 8.6. Movies of particle distributions can be found at www.mathematik.uni-dortmund.de/lsiii/download/sokolow.

As a remark, we would like to point out that the simulated STR configurations can be significantly more complex (curvature and number of blades, shape of the tank, etc.) without any loss from the side of the numerical behaviour of the proposed DPM.

The considered discrete projection method shows a robust and accurate behaviour for such complex unsteady problems as a stirred tank reactor. The developed code also exploits advanced CFD techniques such as stable non-conforming finite elements [88], robust high-resolution stabilization of the convective term [56], multigrid solvers [105], etc. Furthermore, the approach can be extended to population balance models or turbulent flows ($k - \varepsilon$ turbulence model), which will be discussed shortly in sections § 8.2 and § 8.3.

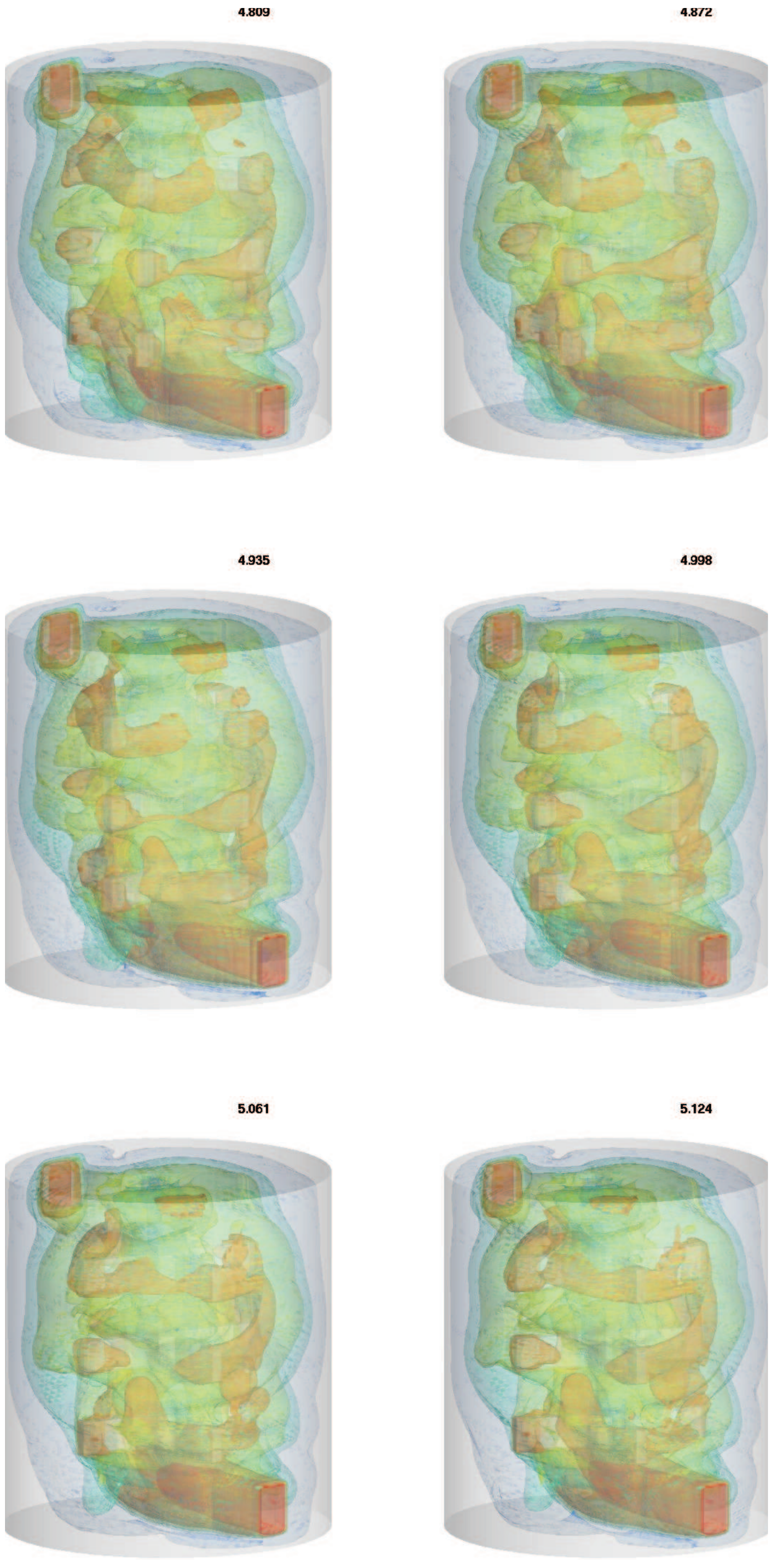

Fig. 8.4. Distribution of the velocity field in the STR.

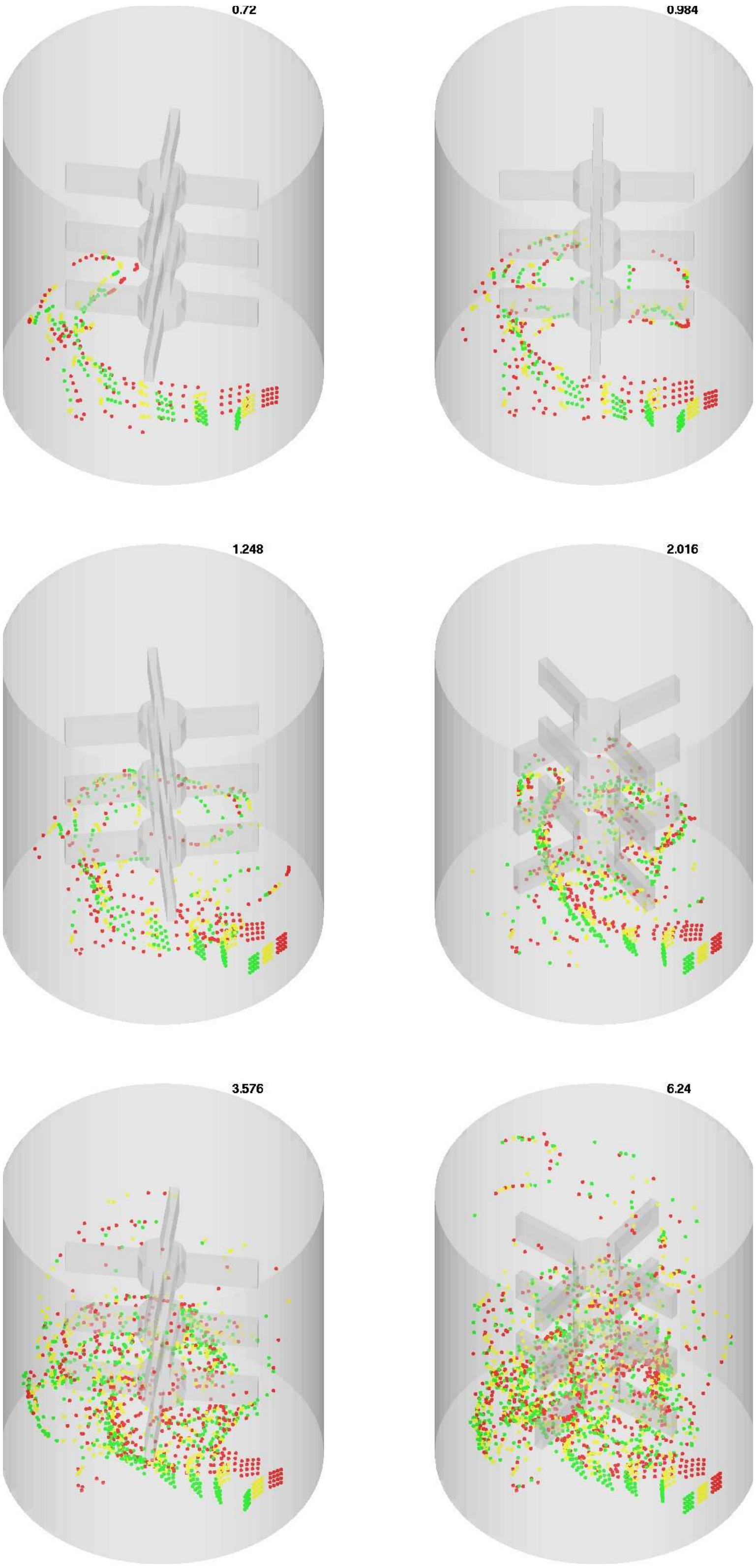

Fig. 8.5. Distribution of particles at consecutive time steps in the STR. Particles are colored by sources.

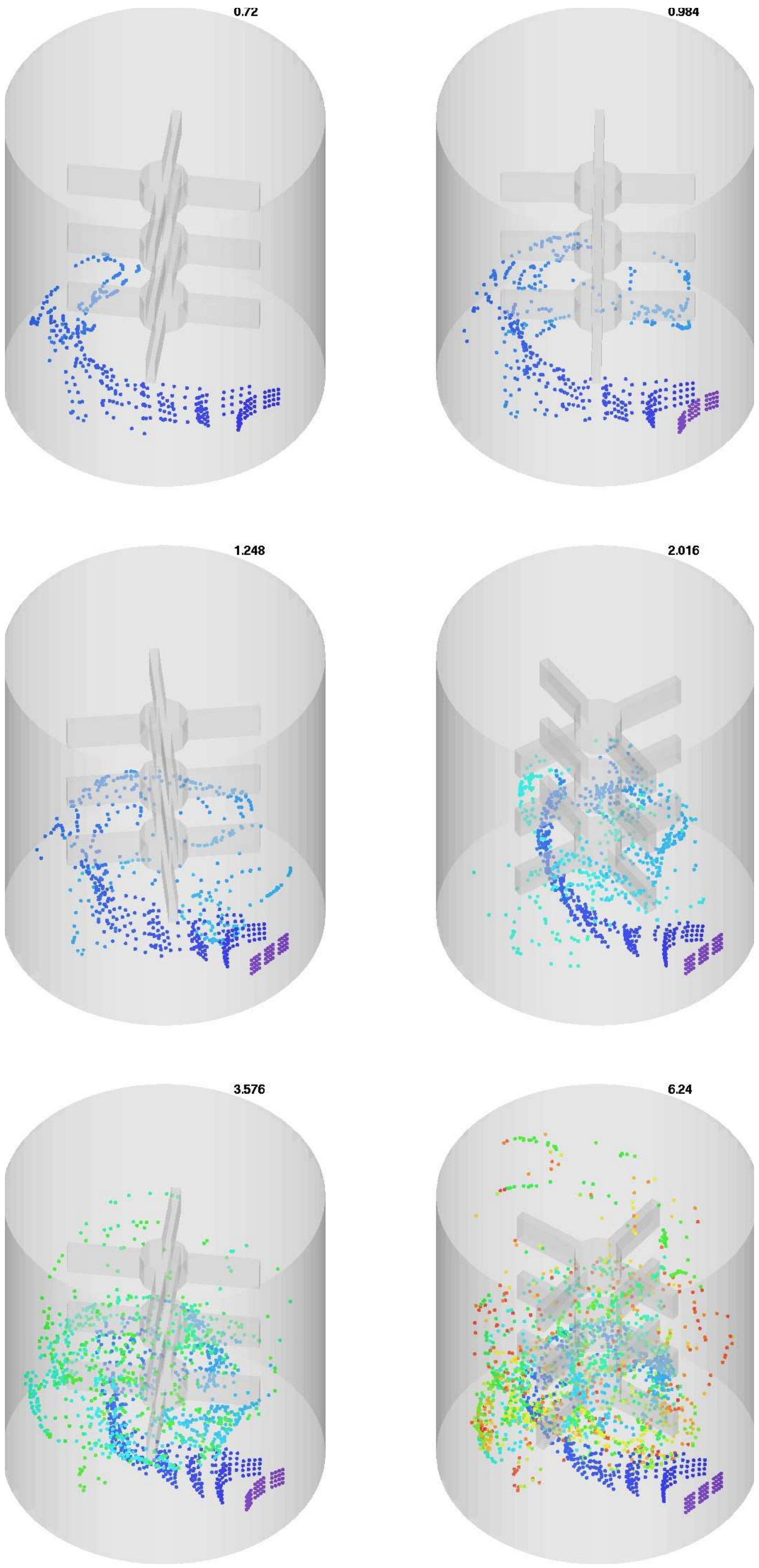

Fig. 8.6. Distribution of particles at consecutive time steps in the STR. Particles are colored by the duration of their stay in the tank.

8.2 Turbulence model

To model the relevant processes appearing in such a system as a stirred tank reactor, one has to account for the turbulent flow. Mathematically the turbulent flow is defined by a system of Navier-Stokes equations with a high Reynolds number ($Re = UL/\nu$, where U is a characteristic velocity, L is a characteristic length and ν is a kinematic viscosity). While dealing with three-dimensional flows at high Reynolds numbers, the numerical costs of DNS (Direct Numerical Simulation) are extremely high. For example, for a 3D simulation of the stirred tank reactor with Reynolds number $Re = 30\,000$ this would require about $Re^{\frac{9}{4}} \geq 10\,000\,000\,000$ nodes (see e. g. [36], [92]), which is not feasible with today's computing power. Therefore, some kind of turbulence modelling has to be used.

A k-ε turbulence model was added to FEATFLOW, in order to make the calculation of such flows on meshes of moderate size possible. The corresponding CFD code `pp3d-ke` was developed by Kuzmin building on the laminar `Featflow` version [57]. The mathematical foundations of the program can be described as follows.

We consider the following system of *Reynolds-averaged Navier-Stokes* (RANS) equations:

$$\begin{array}{c} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nabla \cdot ((\nu_0 + \nu_{\mathrm{T}}) \mathcal{D}(\mathbf{u})) \\ \nabla \cdot \mathbf{u} = 0 \end{array} \tag{8.1}$$

where $\mathbf{u} = (u_1, u_2, u_3)^T$ is a time-averaged velocity and $p = p(t, \boldsymbol{x})$ is a time-averaged pressure, both defined on a time-space domain $(0, T) \times \Omega$ with $T > 0$ and $\Omega \subset \mathbb{R}^3$. $\mathcal{D}(\mathbf{u}) = \nabla \mathbf{u} + (\nabla \mathbf{u})^T$ is a strain tensor and $\nu_{\mathrm{T}} = C_\mu \frac{k^2}{\varepsilon}$ is a turbulent eddy viscosity. The turbulent kinetic energy k and its dissipation rate ε are modelled by the following scalar transport equations:

$$\begin{aligned} \frac{\partial k}{\partial t} + \nabla \cdot \left(k\mathbf{u} - \frac{\nu_T}{\sigma_k} \nabla k \right) &= P_k - \varepsilon, \\ \frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left(\varepsilon\mathbf{u} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) &= \frac{\varepsilon}{k}(C_1 P_k - C_2 \varepsilon), \end{aligned} \tag{8.2}$$

where $P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2$. The default values of the involved empirical constants are: $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$.

Additionally, appropriate boundary conditions for $\mathbf{u}$, k and ε have to be prescribed $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{wall}} \cup \Gamma_{sym}$.

As usual, Dirichlet boundary conditions for $\mathbf{u}$, k and ε are prescribed on the inflow boundary Γ_{in}:

$$\mathbf{u} = \mathbf{g}, \quad k = c_{ab} |\mathbf{u}|^2, \quad \varepsilon = C_\mu \frac{k^{3/2}}{l_0}, \tag{8.3}$$

where c_{ab} is an empirical constant [58] and l_0 is a mixing length.

Let us denote $\mathbf{n}$ to be the unit outward normal to the boundary. At the outlet Γ_{out} the following 'do-nothing' boundary conditions are prescribed:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0, \quad \frac{\partial k}{\partial \mathbf{n}} = 0 \quad \frac{\partial \varepsilon}{\partial \mathbf{n}} = 0, \quad \mathbf{n} \cdot [pI - \nu_T \mathcal{D}(\mathbf{u})] = 0. \tag{8.4}$$

In the $k - \varepsilon$ model the behaviour of a fluid near an impervious solid wall is modelled by wall functions. The computational wall boundary Γ_{wall} is located at a distance δ from the real geometrical wall boundary. In our case we assume that the computational domain is already reduced by a layer of width δ, which is a user-defined parameter.

We set the following boundary conditions on Γ_{wall}:

$$\mathbf{n}\cdot\mathbf{u}=0,\quad \mathbf{n}\cdot\mathcal{D}(\mathbf{u})=-\frac{u_\tau^2}{\nu_T}\frac{\mathbf{u}}{|\mathbf{u}|},\quad k=\frac{u_\tau^3}{\sqrt{C_\mu}}\quad,\quad \varepsilon=\frac{u_\tau}{\kappa\delta},\tag{8.5}$$

where $\kappa = 0.41$ is the von Kármán constant and u_τ is the friction velocity. Here u_τ is the solution of the logarithmic wall law equation:

$$|\mathbf{u}| = u_\tau(\frac{1}{\kappa}\log y^+ + 5.5),\tag{8.6}$$

where $y^+ = u_\tau\delta/\nu$ is the local Reynolds number.

A detailed mathematical derivation of the $k-\varepsilon$ turbulence model can be found, e.g. in [61, 69]. The realization of the numerical algorithm is described in [58].

We would like to note that the wall boundary conditions (8.5) and (8.6) in the $k-\varepsilon$ turbulent model are based on the experimentally measured behaviour of flow in a channel. Since no data about wall boundaries in the stirred tank is available, we have to assume that they do not differ substantially from those in the channel flow. The coordinate transformation led not to the rotation of the blades of the propeller, but to the outside wall, which shape is more similar to the plain deck, especially if the curvature of the wall of the tank is small. Moreover, since the outside wall of the tank rotates (again, if to consider from the noninertial frame of reference), the main role plays not normal, but the tangential component of velocity $\boldsymbol{u}$. Thus, the boundary conditions (8.5) and (8.6) are modified in the following way:

$$\mathbf{n}\cdot\mathbf{u}=\mathbf{n}\cdot\dot{\boldsymbol{x}},\quad \left|\mathbf{n}^\perp\cdot(\boldsymbol{u}-\dot{\boldsymbol{x}})\right| = u_\tau(\frac{1}{\kappa}\log y^+ + 5.5),\tag{8.7}$$

where $\dot{\boldsymbol{x}}$ is velocity of the wall and $\boldsymbol{u}\cdot\mathbf{n}^\perp = \boldsymbol{u}-(\mathbf{n}\cdot\boldsymbol{u})\mathbf{n}$.
Basing on the Reynolds-averaged Navier-Stokes equations (8.1) and the generalized boundary conditions (8.7) we can propose the coordinate transformation concept for the better treatment of a rotating propeller boundaries. This can lead to the application of the proposed modified discrete projection method. Careful numerical analysis of the modified DPM in the direction of the Reynolds-averaged turbulence modeling we leave as a topic for future.

8.3 Population balance

Population balance is a well-established method used to analyse the size of the dispersed phase during our modelling of the phenomena that occur in the stirred tank. We have to describe the behaviour of this dispersed phase and as a result to account for the population dynamical processes of the drops: to describe how the number of drops of a certain size changes with time. On the one hand, when two drops collide, they may form a larger drop. That means a new larger drop is generated, whereas two smaller drops disappear. On the other hand, a drop may also collide with an eddy such that it breaks up into some smaller drops. In this case, some smaller drops are formed and the larger, the so-called mother drop, is destroyed. The confluence of drops is called coalescence, the breakup is called dispersion.

In simulation the following assumptions can be made:

- Only binary coalescence is considered (since the probability that three or more drops collide in a time interval (t,$t+dt$) at a certain point is negligibly small compared to the probability that two drops meet).
- Source term in the population balance equation are assumed to be only due to break-up and coalescence (expansion, growth and other processes are ignored).

The continuity equation for the dispersed phase, accounting for the changes in the particle size, is of the form [86], [97]:

$$\frac{\partial}{\partial t}\rho\alpha + \nabla\cdot\rho\alpha\boldsymbol{u} + \frac{\partial}{\partial r}\rho\alpha\dot{r} = 0, \tag{8.8}$$

where t is time, ρ is density, α is volume fraction, $\boldsymbol{u}$ is velocity, r is a particle radius and $\dot{r}$ is the rate of change in particle radius. The first term of this equation represents changes of particle concentration in time, the second term the convection in physical space and the third term the changes in size.

Let us denote

$$S = -\frac{\partial}{\partial r}\rho\alpha\dot{r}$$

to be the source/sink term $S = S(V, \boldsymbol{x}, t)$ and

$$f = -\rho\alpha$$

to be the average number density function $f = f(V, \boldsymbol{x}, t)$, i.e. f describes how many drops of volume V are at a certain point $\boldsymbol{x}$ at time t.

Applying Reynolds averaging to the population balance equation (8.8), we obtain:

$$\frac{\partial f}{\partial t} + \nabla\cdot(\mathbf{u}f) - \nabla\cdot(c_t\nabla f) = S, \tag{8.9}$$

where the diffusion term $-\nabla\cdot(c_t\nabla f)$ is due to the modelling $\nabla\cdot(\boldsymbol{u}'f') = -\nabla\cdot(c_t\nabla f)$. Here, $\boldsymbol{u}'$ is velocity of small eddies, f' is an average number density function of the small eddies, $c_t = \nu_T/(\rho Sc_t)$, where ν_T is a turbulent eddy viscosity and Sc_t is the turbulent Schmidt number. Further information can be found in the book by D. Ramkrishna [86]

The coalescence and breakup phenomena are taken into account by means of the terms in the right-hand side of this integro-differential equation. There are different approaches can be found in literature on how to model source/sink term S. In our case we will assume that sink is only due to the sink of coalescence s^-_{coal} and breakup s^-_{break}, and, in the same way, source is only due to the source of coalescence s^+_{coal} and breakup s^+_{break}. Therefore $S(V, \boldsymbol{x}, t) = s^+_{\text{coal}} + s^-_{\text{coal}} + s^+_{\text{break}} + s^-_{\text{break}}$.

The terms due to coalescence are modeled by:

$$\begin{aligned} s^-_{\text{coal}}(\mathbf{x}, V, t) &= -f(\mathbf{x}, V, t)\int_0^{V_{\max}-V} R_{\text{coal}}(V, V', \mathbf{y}(\mathbf{x}, t))f(\mathbf{x}, V', t)dV', \\ s^+_{\text{coal}}(\mathbf{x}, V, t) &= \int_0^{V} R_{\text{coal}}(V', V'', \mathbf{y}(\mathbf{x}, t))f(\mathbf{x}, V', t)f(\mathbf{x}, V'', t)dV'. \end{aligned}$$

Here, $R_{\text{coal}}(V', V'', \mathbf{y}(\mathbf{x}, t))$ denotes the coalescence rate, which describes the probability that two drops with volumes V' and V'' coalesce. Vector $\mathbf{y}$ is the so-called continuous phase vector, which is dependent on the properties of the continuous phase that influence the coalescence and breakage processes. Thus, the vector $\mathbf{y}$ may consist of pressure, temperature, or other values that we get from the calculation of the flow field.

On the other hand, the source and sink terms due to breakage $s^{\pm}_{\text{break}}(\mathbf{x}, V, t)$ are modelled by:

$$\begin{aligned} s^-_{\text{break}}(\mathbf{x}, V, t) &= -R_{\text{break}}(V, \mathbf{y}(\mathbf{x}, t))\ f(\mathbf{x}, V, t), \\ s^+_{\text{break}}(\mathbf{x}, V, t) &= \int_V^{V_{\max}} n(V', \mathbf{y}(\mathbf{x}, t))\gamma(V, V', \mathbf{y}(\mathbf{x}, t))R_{\text{break}}(V', \mathbf{y}(\mathbf{x}, t)) \\ &\qquad f(\mathbf{x}, V', t)dV'. \end{aligned}$$

Here, $R_{\text{break}}(V', \mathbf{y}(\mathbf{x}, t))$ denotes the breakage rate, which accounts for the probability that a drop with volume V' breaks up. Furthermore, $\gamma(V, V', \mathbf{y}(\mathbf{x}, t))$, the so-called distribution of daughter drops, describes the probability that the breakage of a drop with volume V' leads to at least one daughter drop with volume V. The quantity

$n(V', \mathbf{y}(\mathbf{x}, t))$ gives the number of daughter drops that are formed by the breakage of a drop with volume V'.

Physically, the following boundary conditions should be prescribed on $\varGamma_{\text{wall}} = \varGamma_{\text{stirrer}} \cup \varGamma_{\text{tank}}$:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_{\text{stirrer}} \ \text{ on } \varGamma_{\text{stirrer}} \\ \mathbf{u} &= 0 \ \text{ on } \varGamma_{\text{tank}} \\ f(\mathbf{x}, V, t) &= 0 \ \text{ on } \ \varGamma_{\text{wall}} \end{aligned}$$

where $\varGamma_{\text{stirrer}}$ describes the stirrer, and $\varGamma_{\text{tank}}$ the boundary of the tank. The initial conditions are chosen such that they are consistent with the boundary conditions. Further details about the modelling of stirred liquid-liquid systems can be found in [52, 91, 92].

In practice it is often assumed that drops belong to some fixed size groups, the number of which is finite. Hence, discretizing (8.8) in respect to size, we obtain a continuity equation for the size group-i:

$$\frac{\partial}{\partial t}\rho\alpha_i + \nabla \cdot \rho\alpha_i \boldsymbol{u} = S_i, \tag{8.10}$$

where S_i is the rate of mass transfer into or out of the size group due to break-up and coalescence. It is obvious, that the sum of all particle volume fractions equals the volume fraction of the dispersed phase:

$$\sum \alpha_i = \alpha.$$

As before, we can rewrite the individual size-group volume fraction in terms of the total as:

$$f_i\alpha = \alpha_i$$

and rewrite (8.10):

$$\frac{\partial}{\partial t}\rho\alpha f_i + \nabla \cdot \rho\alpha\boldsymbol{u} f_i = S_i, \tag{8.11}$$

where f_i is the fraction of the dispersed phase volume fraction in group-i. This equation has the form of the transport equation of a scalar variable f_i in the dispersed phase.

Due to one way coupling we can apply the coordinate transformation technique to calculate the velocity field $\boldsymbol{u}$, which will be used later in the population balance equation (8.9). Numerical experiments in this direction we will also leave as a topic for future.

8.4 Conclusions

From the previous two sections we observe that the coordinate transformation concept and therefore the modified discrete projection framework are applicable for the numerical simulation of population balance and 'Reynolds-averaged' turbulence models, where precise calculation of a flow field along moving boundaries is important. We showed that the modified discrete projection method is suitable for simulation of complex industrial models. We also noticed that for complex 3D simulations presented in this chapter the iterative solver demonstrates the same properties already observed for test models of the chapter 5. The particle tracing tool gives a realistic behaviour of the flow field in the stirred tank reactor. At the moment, the `Featflow` group at TU Dortmund continues intensive research in the directions of turbulent flows [53] and populations balance modeling [4]. As future extensions one can choose simulation of such complex models by a general-purpose scheme (4.8)–(4.10), where convection and the Coriolis force term are written in a form of the cross-product operator $\boldsymbol{w}(\boldsymbol{u}, \boldsymbol{\omega}, \cdot) \times \boldsymbol{u}$.

9

Conclusion and outlook

In this thesis we proposed a new discrete projection method for the incompressible Stokes and 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 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-like operator in a projection step was deduced using pressure Schur complement preconditioning technique. It appears to be much more efficient both in accuracy in time and in convergence to the steady state solution than the standard one since convective as well as rotational parts were taken into account. The numerical results showed that the modified DPM is more efficient and robust with respect to the variation in problem parameters than the standard projection scheme.

Furthermore we analysed the accuracy of the modified projection scheme. It was proven that the proposed DPM for the Navier-Stokes equations with the Coriolis force (1.3) has the same order of accuracy as the classical projection scheme for the Navier-Stokes equations (1.1). Namely, the velocity is weakly first-order approximation and the pressure is weakly order $\frac{1}{2}$ approximation.

As the next step we introduced the rotational form of convection. By doing so, we extended the framework of the modified scheme to the general case, which made it possible to treat any terms written in a form of the cross-product operator $\boldsymbol{w}(\boldsymbol{u}, \cdot) \times \boldsymbol{u}$. Though we did not gain advantageous numerical behaviour of the rotation form of convection with respect to those of the standard form, we showed that with the help of edge-oriented and $\nabla\ div$-stabilization techniques one can obtain sufficiently accurate results up to medium Reynolds numbers. As test models we took the lid-driven cavity and the flow around cylinder benchmark problems.

Finally, with the code for the modified discrete projection scheme we performed nonsteady simulations for two configurations of stirred tank reactor models. In the obtained flow field we injected virtual particles and observed their distribution and mixture. These tests showed that the proposed DPM can be successfully used for real-life models. We also showed possible applications of our DPM for turbulent flows in the stirred tank reactor, where prescription of boundary layers is of primary importance. Numerical analysis of this model by a general scheme with coriolis-convection operator $\boldsymbol{w}(\boldsymbol{u}, \boldsymbol{\omega}, \cdot) \times \boldsymbol{u}$ we leave as a topic for the future.

10

Appendix A

10.1 Derivation of the modified system of the Navier-Stokes equations

As we have already seen, for the case of the stirred tank reactor it is reasonable to perform a coordinate transformation in such a way that blades of the propeller remain stationary, but the outside wall of the stirred tank reactor rotates in a direction opposite to those of the propeller before coordinate transformation, see Fig. 10.1. In other words, we assume that the observer is placed into the middle of the propeller. By doing so, we have to turn to consider the system of the Navier-Stokes equations with the so-called rotational forces – forces, which arise in the system due to this coordinate transformation. In our derivations we make the following assumptions:

1. The propeller in the stirred tank rotates around the Z-axis. The general case can be easily obtained by accumulating the same idea, presented below.
2. Angular velocity is constant, i.e. $\boldsymbol{\omega} = (0, 0, \omega)^T$, where $\omega = const$.
3. The tank is filled with homogeneous liquid.

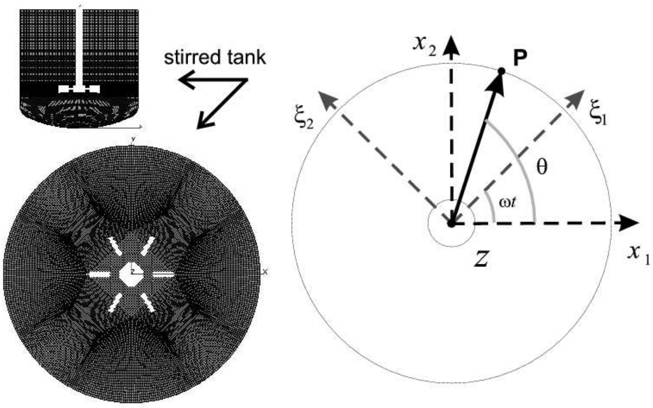

Fig. 10.1. Systems of coordinates

We say that an inertial frame $\boldsymbol{K}$ is a preferable frame of reference and anything causing a particle to be accelerated relative to it is regarded as a true force. We denote a basis in $\boldsymbol{K}$ by $\{x_1, x_2, z\}$.

At the same time, the non-inertial reference frame **S** (in which body violates Newton's Laws of Motion) has a common OZ-axis and rotates in an anticlockwise direction with some 'speed' ω relative to $\boldsymbol{K}$. Let us choose an arbitrary point $\mathcal{P} = \mathcal{P}(\boldsymbol{x}) = \mathcal{P}(\boldsymbol{\xi})$, where $\boldsymbol{x} = (x_1, x_2, z)^T$ are its coordinates in $\boldsymbol{K}$, and $\boldsymbol{\xi} = (\xi_1, \xi_2, z)^T$ are coordinates in **S**. There can be established the following relation between $\boldsymbol{x}$ and $\boldsymbol{\xi}$:

$$\begin{cases} x_1 = r\cos\theta \\ x_2 = r\sin\theta \\ \xi_1 = r\cos(\theta - \omega t) = r\cos\theta\cos\omega t + r\sin\theta\sin\omega t \\ \xi_2 = r\sin(\theta - \omega t) = r\sin\theta\cos\omega t - r\cos\theta\sin\omega t \\ z = z \end{cases} \tag{10.1}$$

where t is a time variable. Rewriting (10.1) in a matrix form, we get

$$\boldsymbol{\xi} = \boldsymbol{U}\boldsymbol{x}, \tag{10.2}$$

where $\boldsymbol{U} = \begin{pmatrix} \cos\omega t & sin\omega t & 0 \\ -\sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is a matrix of the coordinate transformation $\boldsymbol{x} \rightarrow \boldsymbol{\xi}$. One can easily check that $\boldsymbol{U}$ is orthogonal, i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = I$. Hence,

$$\boldsymbol{x} = \boldsymbol{U}^T\boldsymbol{\xi}. \tag{10.3}$$

Differentiating (10.2) with respect to t, one obtains

$$\begin{cases} \dot{\xi}_1 = \dot{x}_1\cos\omega t + \dot{x}_2\sin\omega t - x_1\omega\sin\omega t + x_2\omega\cos\omega t \\ \dot{\xi}_2 = -\dot{x}_1\sin\omega t + \dot{x}_2\cos\omega t - x_1\omega\cos\omega t - x_2\omega\sin\omega t \\ \dot{z} = \dot{z} \end{cases} \tag{10.4}$$

The matrix form of (10.4) is

$$\dot{\boldsymbol{\xi}} = U\dot{\boldsymbol{x}} - \boldsymbol{\omega} \times \boldsymbol{r}(\boldsymbol{\xi}), \tag{10.5}$$

where $\boldsymbol{r}(\boldsymbol{\xi}) = (\xi_1, \xi_2, z)^T$ is a radius vector. Again, differentiating (10.3) with respect to t, one gets

$$\dot{\boldsymbol{x}} = \boldsymbol{U}^T\dot{\boldsymbol{\xi}} + \boldsymbol{\omega} \times \boldsymbol{r}(\boldsymbol{x}). \tag{10.6}$$

Let us turn our attention to the relation between the rate of change of an arbitrary vector $\overline{O\mathcal{P}} = \boldsymbol{p}$ in **K** and its rate of change in **S**. If we fix $\boldsymbol{p}$, which can be in general a function of time, then its rate of change will appear different in both systems. If the tip of the vector is at rest in **S**, then, as seen from **K**, a perpendicular component $\boldsymbol{\omega} \times \boldsymbol{p}$ adds up per unit of time. In general, when $\boldsymbol{p}$ is not fixed in **S**, either in direction or in magnitude or both, we shall have:

$$\left(\frac{d\boldsymbol{p}}{dt}\right)_K = \left(\frac{d\boldsymbol{p}}{dt}\right)_S + \boldsymbol{\omega} \times \boldsymbol{p}. \tag{10.7}$$

Next, let us denote by

$$\boldsymbol{u} = \dot{\boldsymbol{x}} = \left(\frac{d\boldsymbol{r}}{dt}\right)_K \quad \text{and} \quad \boldsymbol{v} = \left(\frac{d\boldsymbol{r}}{dt}\right)_S = \begin{pmatrix} x_{1_{\xi_1}}\dot{\xi}_1 + x_{1_{\xi_2}}\dot{\xi}_2 \\ x_{2_{\xi_1}}\dot{\xi}_1 + x_{2_{\xi_2}}\dot{\xi}_2 \\ \dot{z} \end{pmatrix} = U^T\dot{\xi} \tag{10.8}$$

velocities in **K** and **S** frames, respectively. Then, from (10.6) we can derive

$$\left(\frac{d\boldsymbol{u}}{dt}\right)_K = \left(\frac{d\boldsymbol{v}}{dt}\right)_K + \boldsymbol{\omega} \times \boldsymbol{u}. \tag{10.9}$$

Substituting $\boldsymbol{u}$ into the last term of (10.9) we obtain

$$\left(\frac{d\boldsymbol{u}}{dt}\right)_K = \left(\frac{d\boldsymbol{v}}{dt}\right)_K + \boldsymbol{\omega} \times \boldsymbol{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}). \tag{10.10}$$

Next, applying (10.7) to $\boldsymbol{v}$ one gets

$$\left(\frac{d\boldsymbol{v}}{dt}\right)_K = \left(\frac{d\boldsymbol{v}}{dt}\right)_S + \boldsymbol{\omega} \times \boldsymbol{v}. \tag{10.11}$$

Substituting (10.11) into (10.10) we come to the final equality

$$\left(\frac{d\boldsymbol{u}}{dt}\right)_K = \left(\frac{d\boldsymbol{v}}{dt}\right)_S + 2(\boldsymbol{\omega} \times \boldsymbol{v}) + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}). \tag{10.12}$$

As we know, in the inertial frame of reference $\boldsymbol{K}$ the system of Navier-Stokes equations reads:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \nu \Delta \boldsymbol{u} \tag{10.13}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{10.14}$$

We perform a term-by-term modification to get its view in the noninertial frame **S**.

First, taking into account the incompressible property (10.14) we obtain

$$\Delta \boldsymbol{u} = \nabla \cdot \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) = 2\nabla \cdot \mathcal{D}(\boldsymbol{u}).$$

The tensor components of $D(\boldsymbol{u})$ are ‘transformed’ as follows:

$$u_{ik} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right) = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i}\right) +$$
$$+\frac{1}{2}\left[\frac{\partial}{\partial x_k}(\boldsymbol{\omega} \times \boldsymbol{r})_i + \frac{\partial}{\partial x_i}(\boldsymbol{\omega} \times \boldsymbol{r})_k\right] = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i}\right) = v_{ik},$$

since

$$\frac{\partial}{\partial x_k}(\boldsymbol{\omega} \times \boldsymbol{r})_i + \frac{\partial}{\partial x_i}(\boldsymbol{\omega} \times \boldsymbol{r})_k = \frac{\partial}{\partial x_k}(\varepsilon_{ilm}\omega_l x_m) +$$
$$+\frac{\partial}{\partial x_i}(\varepsilon_{klm}\omega_l x_m) = \omega_l\left(\varepsilon_{ilm}\delta_{mk} + \varepsilon_{klm}\delta_{mi}\right) = \omega_i(\varepsilon_{ilk} + \varepsilon_{kli}) = 0,$$

where $\varepsilon_{ilm} = 1$, if (i, l, m) is an even permutation, and $\varepsilon_{ilm} = -1$, if (i, l, m) is an odd permutation.

We see that the components v_{ik} are built from the vector field $\boldsymbol{v}$ in the same way as the components u_{ik} are built from the vector field $\boldsymbol{u}$. Because the pressure term is trivial to evaluate, the second and the last step to do is to ‘transform’ the equation (10.14) for the incompressibility as follows:

$$\nabla \cdot \boldsymbol{u} = \nabla \cdot (\boldsymbol{v} + (\boldsymbol{\omega} \times \boldsymbol{r})) = \nabla \cdot \boldsymbol{v} + \nabla \cdot (\boldsymbol{\omega} \times \boldsymbol{r}) =$$
$$= \nabla \cdot \boldsymbol{v} + \frac{\partial}{\partial x_k}(\boldsymbol{\omega} \times \boldsymbol{r})_k = \nabla \cdot \boldsymbol{v} + \frac{\partial}{\partial x_k}\varepsilon_{klm}\omega_l x_m = \nabla \cdot \boldsymbol{v}.$$

Thus, in the noninertial frame of reference **S** the system of Navier-Stokes equations becomes of the form:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} + 2\boldsymbol{\omega} \times \boldsymbol{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) = -\nabla p + \nu \Delta \boldsymbol{u} \tag{10.15}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{10.16}$$

where $2\boldsymbol{\omega} \times \boldsymbol{v}$ is the *Coriolis force* and $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})$ is the *Centrifugal force*.

At the end we would like to prove two auxiliary propositions, which were used in § 1 and throughout the thesis.

Proposition 1. *Let* $\boldsymbol{\omega} \perp \boldsymbol{r}$. *Then* $\boldsymbol{\omega} \times \ (\boldsymbol{\omega} \times \boldsymbol{r}) = -\omega^2 \boldsymbol{r}$

Proof. By direct calculation. □

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

Proof.

$$(\boldsymbol{\omega} \times \boldsymbol{r})^2 = (\boldsymbol{\omega} \times \boldsymbol{r}) \cdot (\boldsymbol{\omega} \times \boldsymbol{r}) = (\boldsymbol{r} \times (\boldsymbol{\omega} \times \boldsymbol{r})) \cdot \boldsymbol{\omega} = \omega^2 \boldsymbol{r}^2 - (\boldsymbol{r} \cdot \boldsymbol{\omega})^2$$

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

Proposition is proved. □

Nomenclature

A	velocity stiffness matrix, page 4
$\boldsymbol{A}$	Stokes operator, page 16
B	discrete gradient operator, page 4
B^T	discrete divergence operator, page 4
$\boldsymbol{C}_{(\cdot)}$	diagonal preconditioner for the velocity matrix in the velocity equation, page 8
CFD	computational fluid dynamics, page IX
DPM	discrete projection method, page 6
e	error, page 18
Ek	Ekman number $Ek = \dfrac{\nu}{\omega L^2}$, where ν is the kinematic viscosity, ω is the angular velocity and L is the characteristic length, page 76
$\boldsymbol{f}$	force, page X
$\boldsymbol{F}$	velocity matrix, page 4
h	mesh size, page 30
I_u, I_p	identity matrices on discrete velocity and pressure spaces, respectively, page 6
J	jump operator, page 40
k or $\varDelta t$	time step, page 15
L	discrete Laplacian operator, page 4
LED	local extremum diminishing, page 35
M or M_C	consistent mass matrix, page 10
$\boldsymbol{M}_{(\cdot)}$	approximation to the velocity matrix, page 10
M_L	lumped mass matrix, page 10
M_p	mass matrix for the pressure, page 10
$N(\boldsymbol{u})$	discrete convective operator, page 4
$\boldsymbol{n} = (n^1, n^2, n^3)^T$	outward unit normal to the boundary, page 12

NAT	number of faces, page 45
NDF	number of degrees of freedom, page 42
NEL	number of elements, page 45
NEQ	number of unknowns, page 45
NVT	number of vertices, page 45
p	pressure, page X
P	new or Bernoulli pressure, page X
$P_{(\cdot)} = B^T \boldsymbol{M}_{(\cdot)}^{-1} B$	modified Schur Complement operator, page 10
q	intermediate pressure, page 13
$\boldsymbol{r} = (x_1, x_2, x_3)^T$	radius vector, page X
Re	Reynolds number $Re = \dfrac{U\,L}{\nu}$, where U and L are, respectively, characteristic velocity and length scales, and ν is the kinematic viscosity, page 59
R^n	integral residual of the Taylor series, page 18
Ro	Rossby number $Ro = \dfrac{U}{L\,f}$, where U and L are, respectively, characteristic velocity and length scales, and $f = 2\,\omega\,\sin\varphi$ is the Coriolis frequency, where ω is the angular velocity and φ is the latitude, page 8
$\mathcal{S} = B^T \boldsymbol{F}^{-1} B$	Schur Complement operator, page 4
STR	stirred tank reactor, page 74
t	time, page 4
TVD	total variation diminishing, page 27
$\boldsymbol{u} = (u_1, u_2, u_3)^T$	velocity, page X
$\widetilde{\boldsymbol{u}}$	intermediate velocity, page 13
$\boldsymbol{w} = (w_1, w_2, w_3)^T$	rotational operator, page 28
ν	kinematic viscosity, page IX
α	parameter, page 55
$\mathit{\Gamma}$ or $\partial\mathit{\Omega}$	boundary of the domain, page 15
γ_{div}	∇div stabilization parameter, page 60
γ_{edge}	edge-oriented stabilization parameter, page 59
$\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^T$	angular velocity, page X
$\mathit{\Omega}$	domain, page IX
$\boldsymbol{\phi}$	test function for the velocity space, page 12
$\mathit{\Phi}$	correction factor, page 32
$\boldsymbol{\psi}$	test function for the pressure space, page 12

References

1. J. F. Acker and S. Turek. Postprocessing of FEATFLOW Data with the Particle Tracing Tool GMVPT (Version 1.2). Ergebnisberichte des Instituts für Angewandte Mathematik, Nr. 193, FB Mathematik, Universität Dortmund, 2000.
2. J. D. Anderson. *Computational fluid dynamics: the basics with applications*. McGraw Hill, 6th edition, 1995. ISBN 0070016852.
3. B. Y. Ballal and Rivlin R. S. Flow of a Newtonian fluid between eccentric rotating cylinders: inertial effects. *Archive for Rational Mechanics and Analysis*, 62(3):237–294, 1976.
4. E. Bayraktar, O. Mierka, F. Platte, D. Kuzmin, and S. Turek. Modeling and numerical aspects of population balance equations in gas/liquid-liquid two phase flows. *submitted to Applied Mathematical Modelling journal*, 2008.
5. Ch. Becker and S. Turek. Featflow finite element software for the incompressible Navier-Stokes equations - user manual release 1.1. Technical report, Universität Heidelberg, 1998.
6. Ch. Becker and S. Turek. Featflow - finite element software for the incompressible Navier-Stokes equations - user manual release 1.2. User manual, Universität Dortmund, 1999.
7. Murali Beddhu, Lafayette K. Taylor, and David L. Whitfield. Strong conservative form of the incompressible Navier-Stokes equations in a rotating frame with a solution procedure. *Journal of Computational Physics*, 128:427–437, 1996.
8. M. Benzi, G. H. Golub, and J. Liessen. Numerical solution of saddle point problems. *Acta Numerica*, pages 1–137, Cambridge University Press, 2005. Available at `http://www.math.tu-berlin.de/~liesen//Publicat/BenGolLie05.pdf`.
9. M. Benzi and J. Liu. An efficient solver for the incompressible Navier-Stokes equations in rotation form. *SIAM Journal on Scientific Computing*, 29(5):1959–1981, 2007.
10. H. Blum, J. Harig, S. Müller, and S. Turek. FEAT2D finite element analysis tools. User Manual. Release 1.3. Technical report, Universität Heidelberg, 1992.
11. J. P. Boris and D. L. Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1971.
12. O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers and Fluids*, 27(4):421–433, 1998.
13. E. Burman, M. A. Fernandez, and P. Hansbo. Edge stabilization for the incompressible Navier-Stokes equations: a continuous interior penalty finite element method. Technique report, Institut National de Recherche en Informatique et en Automatique, Institut d'Analyse et Calcul Scientifique (CMCS/IACS), Ecole Polytechnique Federale de Lausanne, Switzerland, 2004.
14. E. Burman and P. Hansbo. Stabilized Crouzeix-Raviart element for the Darcy-Stokes problem. Technical report, Chalmers Finite Element Center, 2003. Preprint 2003-15.
15. E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1437–1453, 2004.
16. E. Burman and P. Hansbo. A stabilized non-conforming finite element method for incompressible flow. *Computer Methods in Applied Mechanics and Engineering*, 195(23–24):2881–99, 2006.
17. W. Chang, F. Giraldo, and J. B. Perot. Analysis of an exact fractional step method. *Journal of Computational Physics*, 180:183–199, 2002.

18. A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 23:745–762, 1969.
19. A. J. Chorin and J. E. Marsden. *A mathematical introduction to fluid mechanics, third edition*. Springer, 1992.
20. T. J. Chung. *Computational fluid dynamics*. Cambridge University Press, 2001.
21. R. Codina. Finite element solution of the stokes problem with dominating coriolis force. *Computer Methods in Applied Mechanics and Engineering*, 142:215–234, 1997.
22. R. Codina. Numerical solution of the incompressible Navier-Stokes equations with coriolis forced based on the discretization of the total time derivative. *Journal of Computational Physics*, 148:567–496, 1999.
23. R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190:2681–2706, 2001.
24. M. Crouzeix and P. A. Raviart. Conforming and non-conforming finite element methods for solving the stationary Stokes equations. *RAIRO*, pages 77–104, 1973.
25. Che Dahong. *Numerische simulation von strömungsvorgängen mit der "Arbitrary Lagrangian-Eulerian method" (ALE-Methode)*. Aachen: Mainz, 1997.
26. deal.II. *See* `http://www.dealii.org/`, as to 01.08.2008.
27. J. Douglas and Jr. T. Dupont. *Interior penalty procedures for elliptic and parabolic Galerkin methods, in: Computing methods in applied sciences (Second Internat. Sympo., Versailles, 1975)*, volume 58. Springer, Berlin, 1976. Lecture Note in Phys.
28. H. C. Elman. Preconditioning for the stead-state Navier-Stokes equations with low viscosity. *SIAM Journal on Scientific Computing*, 20:1299–1316, 1999.
29. H. C. Elman. Preconditioning strategies for models of incompressible flow. *SIAM Journal on Scientific Computing*, 25:347–366, 2005.
30. H.C. Elman, D. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, Oxford, UK, 2005.
31. J. H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer, 1996.
32. Fluent Inc. FLUENT Flow Modeling Software. *See* `http://www.fluent.com/software/fluent/`, as to 01.08.2008.
33. V. Girault and P. A. Raviart. *Finite element methods for Navier–Stokes equations*. Springer, Berlin, 1986. Berlin-Heidelberg.
34. R. Glowinski, T. W. Pan, T. I. Helsa, D. D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169:363–426, 2001.
35. S. K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik*, 47:271–306, 1959.
36. M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*. SIAM Monographs on Mathematical Modeling and Computation. SIAM, Philadelphia, 1998.
37. J. Guermond and J. Shen. On the error estimates for the rotational pressure-correction projection methods. *Mathematics of Computation*, 73(248).
38. J. L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195:6011–6045, 2006.
39. P. Harig, P. Schreiber, and S. Turek. FEAT3D - Finite Element Analysis Tools in 3 Dimensions User Manual. Release 1.2. Preprints SFB 359, Nr. 94-19, Universität Heidelberg, March 1994.
40. A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.
41. L. Henning, D. Kuzmin, V. Mehrmann, M. Schmidt, A. Sokolov, and S. Turek. Flow control on the basis of a Featflow-Matlab coupling. In 1st Active Flow Control Conference (AFC 2006), editor, *R. King*. Springer, Berlin, 2006.
42. J. G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem. I. Regularity of solutions and second-order error estimates for spatial discretization. *SIAM Journal on Numerical Analysis*, 19:275–311, 1982.
43. D. Hysom and A. Pothen. Level-based Incomplete LU Factorization: Graph Model and Algorithms (pdf file). Preprint UCRL-JC-150789, U.S. Department of Energy, November 2002. Available at `http://www.cs.odu.edu/~pothen/papers.html`.

44. ANSYS Inc. ANSYS CFX. *See* `http://www.ansys.com/products/cfx.asp`, as to 01.08.2008.
45. A. Jameson. Analisys and design of numerical schemes for gas dynamics. 1. Artificial diffusion, upwinding biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics*, 4:171–218, 1995.
46. V. John, J. M. Maubach, and L. Tobiska. Nonconforming streamline-diffusion-finite-element-method for convection-diffusion problems. *Numerische Mathematik*, 78:165–188, 1997.
47. W. P. Jones and B. E. Launder. The prediction of laminarisation with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 15:301–314, 1972.
48. G. M. Kobelkov and M. A. Olshanskii. Effective preconditioning of Uzawa type schemes for a generalized Stokes problem. *Numerische Mathematik*, 86:443–470, May 2000.
49. D. Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. *Journal of Computational Physics*, 219:513–531, 2006.
50. D. Kuzmin. Algebraic Flux Correction for finite element discretizations of coupled systems. In M. Papadrakakis, E. Onate, and B. Schrefler, editors, *Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, 2007.
51. D. Kuzmin, R. Löhner, and S. Turek. *Flux-Corrected Transport*. Scientific Computation. Springer, Berlin, 2005. Subtitle: Principles, Algorithms and Applications, ISBN 3-540-23730-5.
52. D. Kuzmin, V. Mehrmann, S. Schlauch, A. Sokolov, and S. Turek. Population balances coupled with the CFD-code Featflow. Ergebnisberichte des Instituts für Angewandte Mathematik, Nr. 324, FB Mathematik, Universität Dortmund, June 2006.
53. D. Kuzmin, O. Mierka, and Turek S. On the implementation of the k-epsilon turbulence model in incompressible flow solvers based on a finite element discretization. *International Journal of Computing Science and Mathematics*, 1:193–206, 2007.
54. D. Kuzmin and M. Möller. Algebraic flux correction i. Scalar conservation laws. Ergebnisberichte des Instituts für Angewandte Mathematik, Nr. 249, FB Mathematik, Universität Dortmund, March 2004.
55. D. Kuzmin and S. Turek. Finite element discretization and iterative solution techniques for multiphase flows in gas-liquid reactors. In *Conjugate Gradient Algorithms and Finite Element Methods*, pages 297–324. Springer, Berlin, 2004. Jyväskylä, Finland, June 11-12,2002; (ISBN-Nr. 3-540-21319-8).
56. D. Kuzmin and S. Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics*, 198:131–158, 2004.
57. D. Kuzmin and S. Turek. Numerical simulation of turbulent bubbly flows. Ergebnisberichte des Instituts für Angewandte Mathematik, Nr. 254, FB Mathematik, Universität Dortmund, April 2004.
58. D. Kuzmin and S. Turek. Numerical simulation of turbulent bubbly flows. In *submitted to 3rd International Symposium on Two-Phase Flow Modelling and Experimentation*, 2004. Pisa, 22. - 24. September 2004.
59. O. A. Ladyzhenskaya. *The boundary value problems of mathematical physics*. Applied Mathematical Sciences, New York: Springer, 1985.
60. B. E. Launder and D. B. Spalding. *Mathematical models of turbulence*. Academic Press, London, 1972.
61. R. E. Launder and D. R. Spalding. The numerical simulation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3:269–289, 1974.
62. W. Layton, C. C. Manica, Neda M., M. A. Olshanskii, and L. G. Rebholz. A note on the accuracy of the rotation skew symmetric form in simulations of the NSE. *Availabe online at* `http://www.mathcs.emory.edu/~molshan/OLSH/publ.html`, 2008.
63. C. Li and C. Vuik. Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow. *Numerical Linear Algebra with Applications*, 11:511–523, 2004.
64. J. Liu. *Preconditioned Krylov Subspace Methods for Incompressible Flow Problems*. Phd thesis, Department of Mathematics and Computer Science, Emory University, Atlanta, 2006. Available at `http://http://www.uwf.edu/jliu/research.htm`.
65. G. Lube and M. A. Olshanskii. Stable finite element calculation of incompressible flows using the rotation form of convection. *IMA Journal of Numerical Analysis*, 22:437–461, 2002.
66. P. R. M. Lyra. *Unstructured grid adaptive algorithms for fluid dynamics and heat conduction*. PhD thesis, University of Wales, 1994.
67. P. R. M. Lyra, K. Morgan, J. Peraire, and J. Peiro. TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *International Journal for Numerical Methods in Fluids*, 19:827–847, 1994.

68. Grid3D 3.0.21 Manual. *See* `http://www.feast.uni-dortmund.de/downloads.html`, as to 01.08.2008.
69. B. Mohammadi and O. Pironneau. Analysis of the k-epsilon turbulence model. *Recherches en Mathematiques Appliquees*, 1994.
70. MooNMD-Mathematics and object-oriented Numerics in Magdeburg. *See* `http://www-ian.math.uni-magdeburg.de/home/mitkova/MooNMD_html/`, as to 01.08.2008.
71. A.C. de Niet and F. W. Wubs. Two preconditioners for saddle point problems in fluid flows. *International Journal for Numerical Methods in Fluids*, 54:355–377, 2007.
72. M. A. Olshanskii. An iterative solver for the Oseen problem and numerical solution of incompressible Navier-Stokes equations. *Numerical Linear Algebra and Applications*, 6:353–378, 1999.
73. 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. *Computer Methods in Applied Mechanics and Engineering*, 191:5515–5536, 2002.
74. 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*, 23(5):1683–1706, 2002.
75. M. A. Olshanskii, A. Sokolov, and S. Turek. Error analysis of a projection method for the Navier-Stokes equations with Coriolis force. *to appear in Journal of Mathematical Fluid Mechanics*, 2008.
76. M. A. Olshanskii and Yu. V. Vassilevski. Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM Journal on Scientific Computing*, 29:2686–2704, 2008. 1064-8275.
77. OpenFOAM: The Open Source CFD Toolbox. *See* `http://www.opencfd.co.uk/openfoam/`, as to 01.08.2008.
78. F. A. Ortega. General mesh viewer. 2003.
79. A. Ouazzi. *Finite element simulation of nonlinear fluids with application to granular material and powder*. Phd thesis, Universität Dortmund, December 2005.
80. A. Ouazzi and S. Turek. Multigrid methods for stabilized nonconforming finite element for incompressible flow, October 2001. Vortrag, AMFLOW 2001, Workshop on Adaptive Methods for Flow Computation.
81. A. Ouazzi and S. Turek. Efficient multigrid and data structures for edge-oriented FEM stabilization. In *Numerical Mathematics and Advanced Applications Enumath 2005*, pages 520–527. Springer, 2006.
82. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108:51–58, 1993.
83. O. Pironneau. *Finite element methods for fluids*. Wiley, Masson, Paris, 1989.
84. A. Prohl. *Projection and quasi-compressibility methods for solving the incompressible Navier-Stokes equations*. BG Teubner, 1997.
85. A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 188:505–526, 2000.
86. D. Ramkrishna. *Population balances. Theory and applications to particulate systems in engineering*. Academic Press, San Diego, 2000.
87. R. Rannacher. *On chorin's projection method for the incompressible Navier-Stokes equations*. Springer Berlin / Heidelberg, 2006.
88. R. Rannacher and S. Turek. A simple nonconforming quadrilateral Stokes element. *Numerical Methods for Partial Differential Equations*, 8:97–111, 1992.
89. L. Rebholz. An energy and helicity conserving finite element scheme for the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 45(4):1622–1638, 2007.
90. F. Saleri and A. Veneziani. Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 43:174–194, 2005.
91. S. Schlauch. Modeling of stirred liquid-liquid dispersions. Preprint 41-2004, Institut für Mathematik, Technische Universität Berlin, 2004.
92. S. Schlauch. *Modeling and simulation of drop size distributions in stirred liquid-liquid Systems*. PhD thesis, TU Berlin, 2007. Available at `http://opus.kobv.de/tuberlin/volltexte/2007/1518/`.
93. F. Schunxin, L. Qibing, and F. Song. On the orbital motion of a rotating inner cylinder in annular flow. *International Journal for Numerical Methods in Fluids*, 54:155–173, 2007.
94. J. Shen. On error estimates of projection methods for Navier-Stokes equations: first-order schemes. *SIAM Journal on Numerical Analysis*, 29:57–77, 1992.
95. J. Shen. Remarks on the pressure error estimate for the projection methods. *Numerische Mathematik*, 67:513–520, 1994.

96. J. Shen. On error estimates of the projection methods for Navier-Stokes equations: second-order schemes. *Mathematics of Computation*, 65(215):1039–1065, 1996.
97. L. Simon. Submitted for the course: application of population balance to CFD modelling of gas-liquid reactors. *Trends in Numerical and Physical Modelling for Industrial Multiphase Flows*, September 27-29, 2000.
98. A. Sokolov, M. A. Olshanskii, and S. Turek. A discrete projection method for incompressible viscous flow with Coriolis force. *Computer Methods in Applied Mechanics and Engineering*, 197(51–52):4512–4520, 2008.
99. A. Sokolov, S. Turek, and M. A. Olshanskii. Numerical study of a new discrete projection method for rotating incompressible flows. *to appear in Electronic Transactions on Numerical Analysis*, 2008.
100. CD-adapco. STAR-CD. *See* `http://www.cd-adapco.com/products/STAR-CD/`, as to 01.08.2008.
101. R. Temam. Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des pas fractionnaires II. *Archive for Rational Mechanics and Analysis*, 33:377–385, 1969.
102. R. Temam. Navier-Stokes equations and nonlinear functional analysis. *CBMS-NSF Regional Conference Ser., Society for Industrial and Applied Mathematics, Philadelphia, PA*, 1983.
103. S. Turek. A comparative study of time-stepping techniques for the incompressible Navier-Stoke equations: From fully implicit non-linear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids*, 22:987–1011, 1996.
104. S. Turek. On discrete projection methods for the incompressible Navier-Stokes equations: an algorithmical approach. *Computer Methods in Applied Mechanics and Engineering*, 143:271–288, 1997.
105. S. Turek. *Efficient solvers for incompressible flow problems: an algorithmic and computational approach*. Springer, Berlin, 1999.
106. 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 E. Bänsch, editor, *Challenges in Scientific Computing CISC 2002*, LNCSE, pages 37–68, Berlin, 2002. Springer, Berlin.
107. S. Turek and A. Ouazzi. Unified edge-oriented stabilization of nonconforming FEM for incompressible flow problems: numerical investigations. *Journal of Numerical Mathematics*, 15:299–322, 2007.
108. S. Turek, A. Ouazzi, and J. Hron. On pressure separation algorithms (PSepA) for improving the accuracy of incompressible flow simulations. *Accepted to International Journal for Numerical Methods in Fluids*, 2008.
109. S. Turek, A. Ouazzi, and R. Schmachtel. Multigrid methods for stabilized nonconforming finite elements for incompressible flow involving the deformation tensor formulation. *Journal of Numerical Mathematics*, 10:235–248, 2002.
110. S. Turek and M. Schäfer. Benchmark computations of laminar flow around cylinder. In E.H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics*, pages 547–566. Vieweg, 1996. co. F. Durst, E. Krause, R. Rannacher.
111. S. P. Vanka. Implicit multigrid solutions of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65:138–158, 1985.
112. A. Veneziani. Block factorized preconditioners for high-order accurate in time approximation to the Navier-Stokes equations. *Numer. Methods Partial Differential Equations*, 19:487–510, 2003.
113. P. Wesseling. *Principles of computational fluid dynamics*. Springer, Berlin, 2001. ISBN 3-540-67853-0.
114. D. C. Wilcox. Multiscale model for turbulent flows. *AIAA Journal*, 26, 1988.
115. S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluid. *Journal of Computational Physics*, 31:335–362, 1979.