Electronic Transactions on Numerical Analysis.
Volume 32, pp. 49-62, 2008.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

NUMERICAL STUDY OF A DISCRETE PROJECTION METHOD FOR ROTATING INCOMPRESSIBLE FLOWS*

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

Abstract. This paper presents a numerical analysis for complex 3D simulations of the Stirred Tank Reactor (STR) model by a modified discrete projection method (DPM) for rotating incompressible flow. For several prototypical configurations of the STR model, we examine the multigrid behavior for momentum and pressure Poisson subproblems for different values of the time step, the angular velocity, etc., and we give examples of the convergence behavior of the (outer) DPM scheme. For a prototypical application, we visualize the complex flow behavior by injecting sources of particle tracers into the obtained flow field to observe their mixing distribution.

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

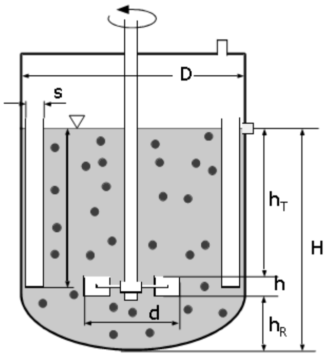
AMS subject classifications. 35Q30, 76D05, 76D07

1. Introduction.

The considered fluid motion is modelled by the nonstationary incompressible Navier-Stokes equations

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

for given force $\boldsymbol{f}$ and kinematic viscosity $\nu > 0$, with some prescribed boundary values and an initial condition.



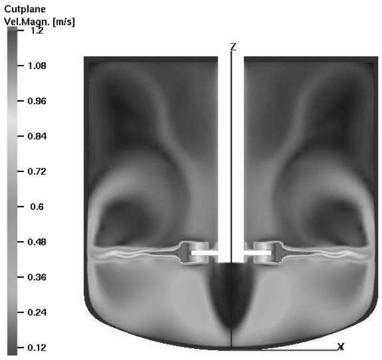


FIG. 1.1. *(left) STR geometry; (right) Numerical simulation (cutplane of velocity).*

If one wants to solve CFD problems of the numerical simulation for models with moving boundary parts, one has to think about the proper treatment of the moving object. As an example of such models, we take the Stirred Tank Reactor shown in Figure 1.1, which is a nice candidate to examine the efficiency of the proposed discrete projection method [10] for the simulation of the real-life 3D problems.

---

*Received November 28, 2007. Accepted for publication March 11, 2008. Published online on December 11, 2008. Recommended by G. Haase. This research was supported by the German Research Foundation and the Russian Foundation for Basic Research through the grant DFG-RFBR 08-01-91957 and TU 102/21-1.

†Institut für Angewandte Mathematik, TU Dortmund, 44227 Dortmund, Germany (asokolow@math.uni-dortmund.de, ture@featflow.de).

‡Department of Mechanics and Mathematics, Moscow State University, 119899 Moscow, Russia (maxim.olshanskii@mtu-net.ru).

ETNA
Kent State University
etna@mcs.kent.edu

All approaches proposed in the literature for treating moving boundary parts have some drawbacks, which one may wish to avoid during the numerical simulation. For example, the Arbitrary Lagrangian Eulerian method [3], while ensures mesh alignment along the boundaries may perform poorly when large deformations or substantial changes in body position in space are required. The Fictitious Boundary method [14] allows to simulate an arbitrarily moving object such that snapshots can demonstrate the realistic movement of the time-dependent tread patterns. Nevertheless, a large amount of CPU time is required to simulate even 2D benchmark models with acceptable accuracy due to the required large amount of grid points for high quality. Moreover, the handling of geometry and meshes of the method serves as a source of additional errors in velocity and pressure fields, since the Fictitious Boundary approach often uses a fixed mesh and therefore may capture boundaries of a moving object not sufficiently accurate unless the mesh is very fine; see Figure 1.2 (right). Luckily, there is a large class of "rotating" models, when the application of the above methods can be avoided by some modifications of the underlying PDEs and/or by special transformations of the model that allow considering a static computational domain. We follow this latter approach in this work for the simulation of the fluid flow in the Stirred Tank Reactor configurations.

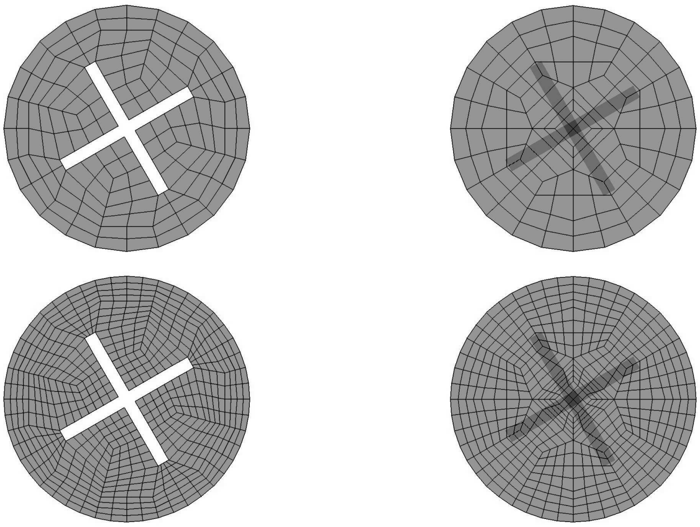


FIG. 1.2. *(top-left) body-fitted mesh on a coarse level; (top-right) mesh for the fictitious boundary approach on a coarse level; (bottom-left) body-fitted mesh on a finer level; (bottom-right) finer mesh for the fictitious boundary approach.*

Transformation of the system of coordinates from the inertial frame into the noninertial frame, rotating with the blades, leads to a new velocity $\boldsymbol{u} = \boldsymbol{v} + (\boldsymbol{\omega} \times \boldsymbol{r})$, where $\boldsymbol{\omega}$ is the angular velocity vector and $\boldsymbol{r}$ is the radius vector from the center of coordinates. The velocity $\boldsymbol{u}$ satisfies homogeneous Dirichlet boundary values on the blades of the propeller, while on the outside wall of the tank one obtains $\boldsymbol{u} = \boldsymbol{\omega} \times \boldsymbol{r}$. Thus, in the new reference frame the system (1.1) can be rewritten as

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

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.

ETNA
Kent State University
etna@mcs.kent.edu

For a more detailed derivation of (1.2); see, e.g., [2] or [11]. Using the equality

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

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

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

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

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

where $\boldsymbol{u} = (u_1, u_2, u_3)^T$ is the discrete velocity, p is the discrete pressure; B and B^T are discrete gradient and divergence operators and $\boldsymbol{S}$ is a block matrix, which stems from the discretized velocity operators in the momentum equation. The matrix $\boldsymbol{S}$ has the following block structure

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

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

The considered algorithm was implemented in the **Pp3d** module of the open-source CFD package **Featflow**[1].

2. Discrete projection method. The original discrete projection method for the time integration of (1.3) can be written as follows (see [10, 12, 13]): At every time step $t_n \to t_{n+1}$ do

1. For given $p^n = p(t_n)$ find an auxiliary velocity $\widetilde{\boldsymbol{u}}$ from

$$\boldsymbol{S}\widetilde{\boldsymbol{u}} = \boldsymbol{g} - \Delta t B p^n. \tag{2.1}$$

2. Solve the discrete pressure Poisson problem

$$P\, q := B^T M_L^{-1} B q = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}}, \tag{2.2}$$

where M_L is the mass matrix of the finite element velocity approximation.

3. Update the pressure and project the velocity via

$$p^{n+1} = p^n + q\,, \tag{2.3}$$

$$\boldsymbol{u}^{n+1} = \widetilde{\boldsymbol{u}} - \Delta t \boldsymbol{S}^{-1} B q\,. \tag{2.4}$$

The velocity matrix $\boldsymbol{S}$ is assumed to be obtained by linearization via a fixed point or Newton-like method and to be invertible.

We modify the above method so that it takes into account the possibly dominant convective and Coriolis force terms. Some theoretical aspects and the detailed derivation can be found in [10]. The final form of the modified DPM reads:

[1] See `www.featflow.de`.

ETNA
Kent State University
etna@mcs.kent.edu

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

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

with a special multigrid method. This multigrid method involves smoothing iterations with the special preconditioner $\boldsymbol{C}$:

$$\boldsymbol{C} := \boldsymbol{C}_{coriol} = \begin{bmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{bmatrix}, \tag{2.5}$$

where M_L is a lumped mass matrix. A similar multigrid method was suggested in [7], where convergence estimates were shown for the 2D model case under certain assumptions on the discretization. Taking into account the fact that all blocks of $\boldsymbol{C}_{coriol}$ are diagonal matrices, one can explicitly find the inverse $\boldsymbol{C}^{-1}_{coriol}$, which has again the same structure; see [9].

2. Solve the discrete pressure problem

$$Pq = \frac{1}{\Delta t} B^T \widetilde{\boldsymbol{u}}\ , \tag{2.6}$$

where P can be interpreted as a special preconditioner to the pressure Schur complement operator $B^T\boldsymbol{S}^{-1}B$. We assume that the matrix P takes the form $B^T\boldsymbol{M}^{-1}_{(\cdot)}B$, with $\boldsymbol{M}_{(\cdot)}$ as an approximation of the velocity matrix $\boldsymbol{S}$. This particular form of P ensures that the update of the velocity in (2.11) is actually a projection into the subspace of discretely divergence-free functions. To take into account convection and the Coriolis force term we suggest to use:

$$\boldsymbol{M}_{(\cdot)} := \boldsymbol{M}_{(\text{diag+coriol})} = \begin{bmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & diag(A) \end{bmatrix}. \tag{2.7}$$

In our numerical experiments we also test other choices for $\boldsymbol{M}^{-1}_{(\cdot)}$:

$$\boldsymbol{M}_{(\text{mass+coriol})} = \begin{bmatrix} M_L & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & M_L & 0 \\ 0 & 0 & M_L \end{bmatrix}, \tag{2.8}$$

$$\boldsymbol{M}_{(\text{diag})} = \begin{bmatrix} diag(A) & 0 & 0 \\ 0 & diag(A) & 0 \\ 0 & 0 & diag(A) \end{bmatrix}, \tag{2.9}$$

$$\boldsymbol{M}_{(\text{diagXY+coriol})} = \begin{bmatrix} diag(A) & -2\omega\Delta t M_L & 0 \\ 2\omega\Delta t M_L & diag(A) & 0 \\ 0 & 0 & M_L \end{bmatrix}. \tag{2.10}$$

3. Calculate the pressure and the velocity approximations as (M_p is the pressure mass matrix)

$$\begin{aligned} p &= p^n + q + \alpha M_p^{-1} B^T \widetilde{\boldsymbol{u}}, \\ \boldsymbol{u} &= \widetilde{\boldsymbol{u}} - \Delta t \boldsymbol{M}^{-1}_{(\cdot)} Bq, \end{aligned} \tag{2.11}$$

with $\alpha = 0$ or $\alpha = \nu$. We set $p^{n+1} = p$, $\boldsymbol{u}^{n+1} = \boldsymbol{u}$. One can also perform several loops of steps 1.–3. to recover the fully coupled solution at time t_{n+1}.

ETNA
Kent State University
etna@mcs.kent.edu

3. Numerical experiments. In this section we analyze the numerical properties of the modified DPM for the system of the Stokes and Navier-Stokes equations with the Coriolis force term. We constructed two configurations of the STR model (see Table 3.1 and Figure 3.1) to test the multigrid behavior for the momentum and Pressure Poisson equations and to examine the overall convergence behavior of this discrete projection method. In Table 3.1 we adopt the following notation: NEL is the number of elements, NAT is the number of faces, NVT and NEQ are the number of vertices and the total number of unknowns (degrees of freedoms) on different grid levels.

TABLE 3.1
Characteristics of STR meshes.

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

3.1. Multigrid with smoother C for velocity problems. Discretizing the system of Navier-Stokes equations with the Coriolis force term (1.3) in time and space, we obtain the linearized momentum equation of the following form with similar structure as in (1.5)

$$\boldsymbol{S}\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{u}^n, p^n, \boldsymbol{f}^{n+1}, \boldsymbol{f}^n). \tag{3.1}$$

We test three preconditioning approaches for solving (3.1). The first two include standard pointwise SOR methods with the following preconditioners:

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

and

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

The third variant is block-diagonal preconditioner $\boldsymbol{C}_{coriol}$ from (2.5). Both $\boldsymbol{C}_{SORcoriol}$ and $\boldsymbol{C}_{coriol}$ matrices take into account convective and Coriolis force terms. However, only $\boldsymbol{C}_{coriol}$ uses the full Coriolis force terms and, at the same time, we can explicitly construct its inverse. Table 3.2 shows the typical number of multigrid iterations for the momentum equation for several problem parameters for the above preconditioners. The STR configuration 1 was used.

ETNA
Kent State University
etna@mcs.kent.edu

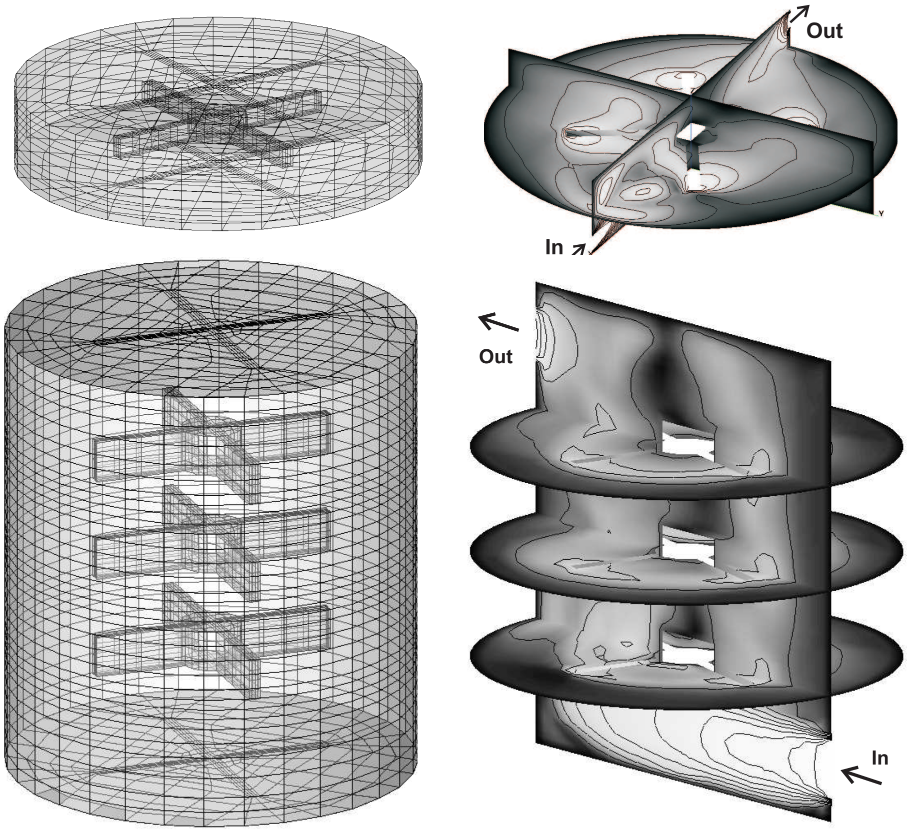


FIG. 3.1. *(top-left) Configuration 1 of the STR, 2d level; (top-right) velocity for configuration 1; (bottom-left) Configuration 2 of the STR, 2d level; (bottom-right) velocity for configuration 2.*

TABLE 3.2
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

For small values of $\omega\Delta t$, the explicit construction of $\boldsymbol{C}_{coriol}$ might not be advantageous. However, for larger values of $\omega\Delta t$, the approximation of the velocity matrix by the preconditioner $\boldsymbol{C}_{coriol}$, with its upper and lower parts of the Coriolis force term on the off-diagonal matrices, shows more advantages. Moreover, the diagonal nature of $\boldsymbol{C}_{coriol}$ makes it possible to find its inverse explicitly and thus, avoids further loss of efficiency.

3.2. Multigrid solver for the modified pressure equation (2.6). 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 Section 3.1 show that the geometric multigrid method with special smoothing 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 (2.6). In the paper [10], we show that the matrix $P = B^T \boldsymbol{M}_{(\cdot)}^{-1} B$ with $\boldsymbol{M}_{(\cdot)}^{-1}$ from (2.8)–(2.10) 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)).

TABLE 3.3
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	$2\omega\Delta t$			
	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

The procedure to measure the multigrid convergence rates was chosen as follows: for

ETNA
Kent State University
etna@mcs.kent.edu

given ω, we calculate until some prescribed stopping criteria are satisfied. Then, the obtained steady state solution $(\widetilde{\boldsymbol{u}}, \widetilde{p})$ is used as an initial solution so that $diag(A) = diag(A(\widetilde{\boldsymbol{u}}))$, and we solve only the Pressure Poisson equation for various preconditioners and the values of $\omega\Delta t$; see Table 3.3. Again, we used the STR configuration 1 to calculate values presented in the table. The multigrid convergence for the STR configuration 2 shows similar behavior.

TABLE 3.4
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				
level 2	0.35+00	0.35+00	0.57+00	0.65+00
level 3	0.46+00	0.34+00	0.56+00	0.68+00
level 4	0.40+00	0.40+00	0.60+00	0.65+00
BiCGStab(ILU(1))				
level 2	0.85-03	0.91-03	0.45-02	0.76-02
level 3	0.31-02	0.85-02	0.23-01	0.28-01
level 4	0.53-02	0.98-01	0.23-01	0.38-01

3.3. Numerical analysis of the new DPM. For the numerical analysis of the computational performance of the new DPM we consider two different cases. First, 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. Then, in the next section, the DPM is used to compute the fully unsteady case for the STR problem. In order 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 expect the convergence of the solution to the steady case to 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 of the form $P = B^T \boldsymbol{M}^{-1}_{(\cdot)} B$ with $M^{-1}_{(\cdot)}$ equals $\boldsymbol{M}_{(\text{mass+coriolis})}$, $\boldsymbol{M}_{(\text{diag})}$ and $\boldsymbol{M}_{(\text{diag+coriol})}$ might give slower convergence, compared with the exact one, but should deliver faster convergence compared with the original unmodified preconditioner $\boldsymbol{M}_{(\text{mass})}$. Moreover, convergence is faster, when larger values $\omega\Delta t$ are used. In the following, we perform tests for each of the discussed choices. The following graphs are for the unit cube geometry; see [10] for details. For the STR configurations the convergence of the DPM has similar behavior as for the unit cube geometry, though due to the higher mesh complexity of the STR the relevant upper bound of the $\omega\Delta t$ value is smaller than in the case of the unit cube.

3.3.1. The Stokes case with Coriolis force. Let us consider the Stokes system of equations:

$$\begin{array}{rcl} \boldsymbol{u}_t - \nu\Delta\boldsymbol{u} + 2\boldsymbol{\omega}\times\boldsymbol{u} + \nabla P & = & \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} & = & 0, \end{array} \quad \text{in } \Omega\times(0,T]. \tag{3.2}$$

Pressure operators of the form $P = B^T M^{-1}_{(\cdot)} B$ are examined with the following choices of $M_{(\cdot)}$:

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

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. Conversely, if $\omega\Delta t$ decreases, the iterative behavior of the DPM as a solver with the matrix $P = B^T\boldsymbol{M}^{-1}_{(\text{mass+coriol})}B$ in step 2 is similar to the performance of the standard Chorin-like scheme with the matrix $P = B^T\boldsymbol{M}^{-1}_{(\text{mass})}B$. We illustrate this in Figure 3.2.

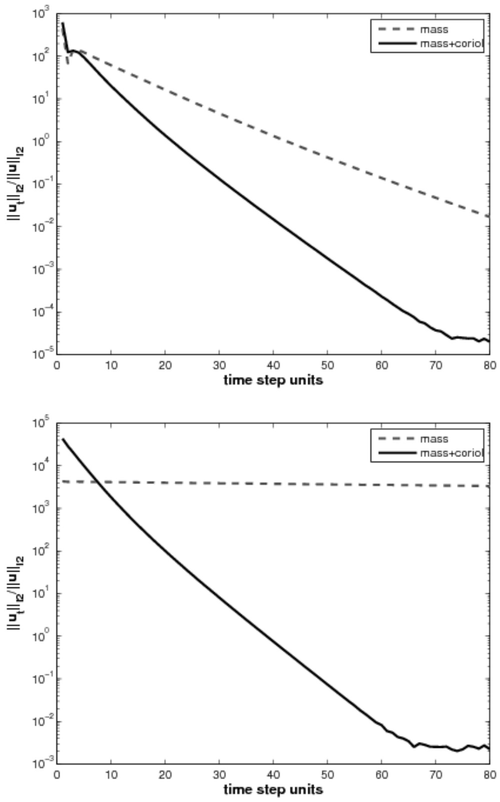


FIG. 3.2. *Stokes equations (top)* $2\omega\Delta t = 0.5$; *(bottom)* $2\omega\Delta t = 10.0$.

3.3.2. Schur complement preconditioners for the Navier-Stokes case. When considering the system of the Navier-Stokes equations (1.3), we can expect to gain a substantial improvement in the convergence rates by applying the pressure operator P with the matrix $\boldsymbol{M}_{(\text{mass+coriol})}$. However, in this case we also have to care about the effect of convective terms on the choice of P. 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 $M_{(\cdot)}$ as follows:

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

$$\text{or} \quad \boldsymbol{M}_{(\text{diagXY})} = \begin{bmatrix} diag(A) & 0 & 0 \\ 0 & diag(A) & 0 \\ 0 & 0 & M_L \end{bmatrix}.$$

ETNA
Kent State University
etna@mcs.kent.edu

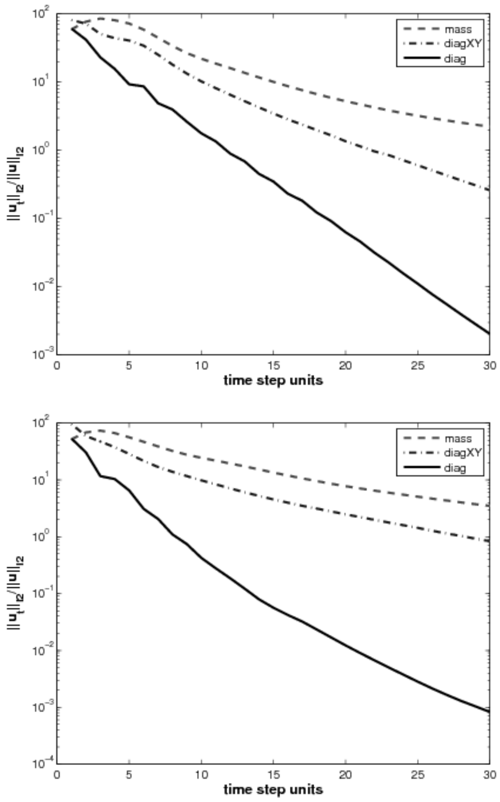


FIG. 3.3. *Navier-Stokes equations (top)* $2\omega\Delta t = 1.5$*; (bottom)* $2\omega\Delta t = 2.5$.

From the results in Figure 3.3, we can conclude that, on the one hand, using the matrix $M_{(\text{diagXY})}$ improves the convergence rates compared with those for $M_{(\text{mass})}$ and, on the other hand, $M_{(\text{diagXY})}$ makes the iterative process more robust compared to $M_{(\text{diag})}$. Our numerical tests show that the pressure Schur complement preconditioner $B^T M^{-1}_{(\text{diagXY})} B$ can be successfully used for the flow simulations with small velocity values in the Z-direction. In this case the convergence history of $||\boldsymbol{u}_t||_{l_2}/||\boldsymbol{u}||_{l_2}$ for simulations using $\boldsymbol{M}_{(\text{diag})}$ and $\boldsymbol{M}_{(\text{diagXY})}$ are quite close to each other. For the flow simulations with significant velocity values in the Z-direction, the case $M_{(\text{diagXY})}$ is not applicable.

Finally, in Figure 3.4, we perform the corresponding tests for the Navier-Stokes equation with the pressure operator P inside the DPM, where convection and Coriolis force terms are included:

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

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

For the last test case, we perform computations with the convective term being of the

ETNA
Kent State University
etna@mcs.kent.edu

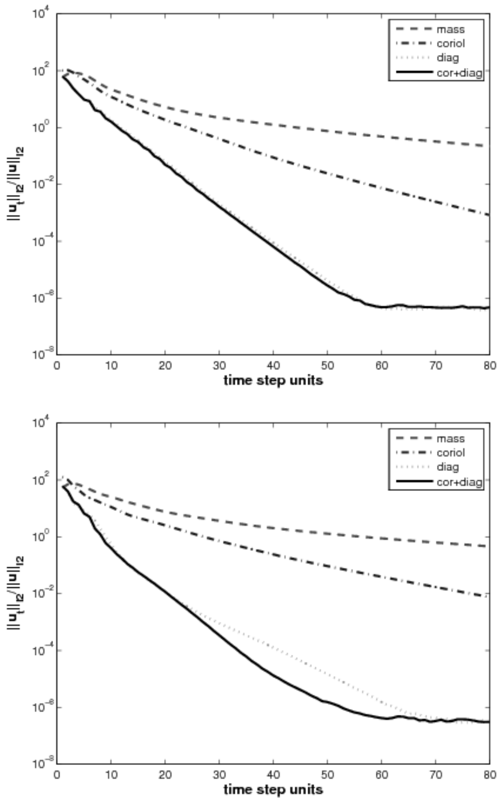


FIG. 3.4. *Navier-Stokes equations (top)* $2\omega\Delta t = 1.5$; *(bottom)* $2\omega\Delta t = 2.5$.

modified 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 now allows much higher values of $\omega\Delta t$, since the convection part is linear. The purpose is to demonstrate that in this case the matrix $\boldsymbol{M}_{(\text{diag+coriol})}$ in the operator P delivers significantly better convergence rates than $\boldsymbol{M}_{(\text{diag})}$. As expected, the matrix $\boldsymbol{M}_{(\text{diag+coriol})}$ gives better convergence rates than any other choice. Results are shown in Figure 3.5.

3.4. Simulation for the full STR configurations. We now present more realistic examples for unsteady numerical simulations for two configurations of the Stirred Tank Reactor; see Figure 3.1 (left). The main characteristics are as follows (all measures are given in non-dimensional form):

1. Configuration 1: Number of propellers = 1, height of the tank $H_{tank} = 4$, radius of the tank $R_{tank} = 10$, length of each blade $L_{blade} = 6$, width of each blade $W_{blade} = 1$, $\nu = 0.1$, $\boldsymbol{u}_{inflow} = 45$, $\omega = 2\pi$, $Ek \approx 0.0004$, where $Ek := \nu/(\omega L_{blade}^2)$ is the Ekman number. In the case of mixers, the Ekman number is such that $Ek = 1/Re$, where Re is the Reynolds number.
2. Configuration 2: Number of propellers = 3, $H_{tank} = 22$, $R_{tank} = 10$, $L_{blade} = 6$, $W_{blade} = 2$, $\nu = 0.1$, $\boldsymbol{u}_{inflow} = 45$, $\omega = 2\pi$, $Ek \approx 0.0004$.

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 Figure 3.1

ETNA
Kent State University
etna@mcs.kent.edu

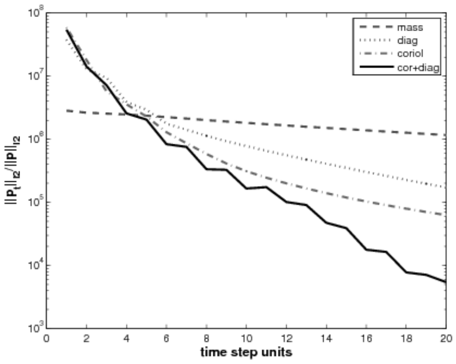


FIG. 3.5. *Navier-Stokes equations with* $\boldsymbol{U}\cdot\nabla\boldsymbol{u}$, $2\omega\Delta t = 10.0$.

(right). 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 correspondingly to provide the user with the 'standard' motion of the propeller in the stirred tank reactor. For the methods discussed here, the movies can be found in the supplement of this paper,
http://etna.math.kent.edu/vol.32.2008/pp49-62.dir/particle.html

It is usually 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 tracer into the geometry of the STR helps to evaluate both the propagation of the velocity field and the mixture of the particles. We used an explicit time-stepping particle tracing tool **GMVPT** [1]. Near the inlet we prescribed three sources of particles, which can be distinguished by their colors: green, yellow and red, respectively; see Figure 4.1. Snapshots at six consecutive time steps give a realistic understanding of the flow motion and the good mixture of the particles. 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 degradation of the numerical behavior of the proposed DPM.

The discrete projection method, considered in this article, shows a very robust and accurate behavior for such complex unsteady problems. The developed code also exploits such advanced CFD techniques as stable non-conforming finite elements [8], robust high-resolution stabilization of the convective term [5], multigrid solvers [13], etc. Furthermore, the approach can be extended to population balance models or turbulent flows (k-ε turbulence model) which is our current research; see [4, 6].

4. Conclusions. In this article we tested the numerical efficiency of a new Discrete Projection Method for the incompressible Navier-Stokes equations with Coriolis force due to a rotating system. As a test model we took a complex 3D geometry of the Stirred Tank Reactor. We examined the multigrid behavior for the momentum and pressure Poisson equations. We showed that the speedup in the convergence to the steady state solution for time-independent problems depends on the choice of the matrix P in the 'pressure Poisson' step of the method. This matrix should account for convection and Coriolis force terms. Finally, we performed nonsteady simulations for the two configurations of the STR model. In the obtained flow field, we injected virtual particle sources and observed their distribution and mixture. All

ETNA
Kent State University
etna@mcs.kent.edu

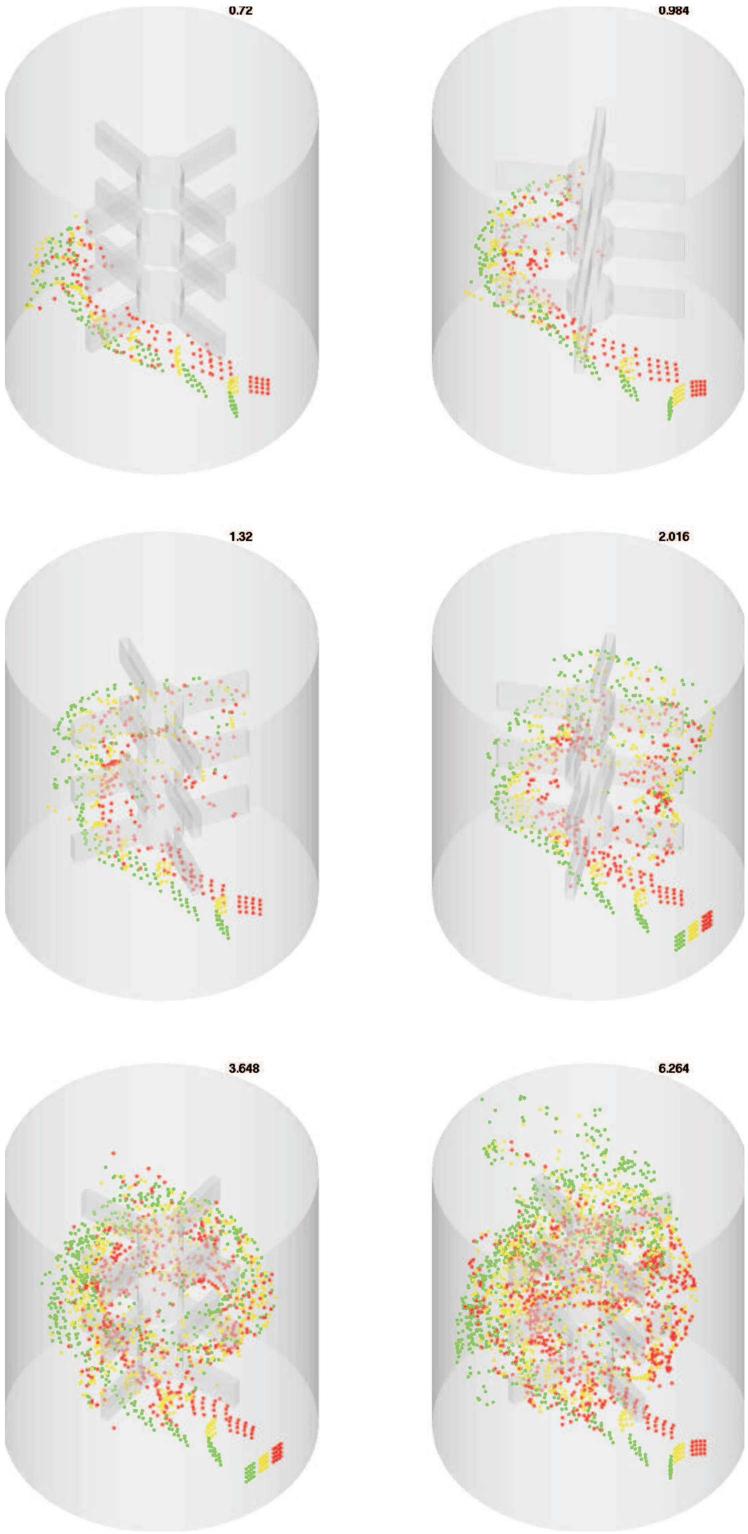


FIG. 4.1. *Distribution of particles at consecutive time steps.*

numerical results show that the modified DPM is more efficient and robust with respect to the variation in problem parameters than standard projection schemes.

REFERENCES

[1] J. F. ACKER AND S. TUREK, *Postprocessing of FEATFLOW data with particle tracing tool GMVPT (Version 1.2)*, Nr. 193, FB Mathematik, TU Dortmund, December 2000. http://www.mathematik.uni-dortmund.de/lsiii

[2] J. AHARONI, *Lectures on Mechanics*, Oxford University Press, Oxford, 1972.

[3] J. DONEA, *Arbitrary Lagrangian-Eulerian finite element methods*, in Computational Methods for Transient Analysis, T. Belytschko, T. Hughes, and K.-J. Bathe, eds., North-Holland, Amsterdam, 1983, pp. 473–516.

[4] 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, TU Dortmund, Germany, June 2006. http://www.mathematik.uni-dortmund.de/lsiii

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

[6] D. KUZMIN, O. MIERKA, AND S. TUREK, *On the implementation of the k-ε turbulence model in incompressible flow solvers based on a finite element discretization*, Internat. J. Comput. Sci. Math., 1 (2007), pp. 193–206.

[7] M. A. OLSHANSKII AND A. REUSKEN, *Navier-Stokes equations in rotation form: A robust multigrid solver for the velocity problem*, SIAM J. Sci. Comput., 23 (2002), pp. 1683–1706.

[8] R. RANNACHER AND S. TUREK, *A simple nonconforming quadrilateral Stokes element*, Numer. Methods Partial Differential Equations, 8 (1992), pp. 97–111.

[9] A. SOKOLOV, *Analysis and Numerical Implementation of Discrete Projection Methods for Rotating Incompressible Flows*, PhD thesis, TU Dortmund, Germany, 2008.

[10] A. SOKOLOV, M. A. OLSHANSKII, S. TUREK, *A new discrete projection method for incompressible viscous flow with Coriolis force*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 4512–4520.

[11] I. E. TARAPOV, *Mehanika Sploshnoy Sredi*, Kharkov National University Press, Kharkiv, Ukraine, 2002.

[12] S. TUREK, *On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach*, Comput. Methods Appl. Mech. Engrg., 143 (1997), pp. 271–288.

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

[14] S. TUREK, W. DECHENG AND L. RIVKIND, *The fictitious boundary method for the implicit treatment of Dirichlet boundary conditions with applications to incompressible flow simulation*, in Challenges in Scientific Computing CISC 2002, E. Bänsch, ed., Lecture Notes in Computational Science and Engineering, Vol. 35, Springer, Berlin, 2002, pp. 37–68,