The version of the following full text has not yet been defined or was untraceable and may differ from the publisher's version.

For additional information about this publication click this link. http://hdl.handle.net/2066/18748

Please be advised that this information was generated on 2020-06-20 and may be subject to change.
On a number of algorithms for solving non-stationary Stokes and Navier-Stokes equations

Sergey Finogenov

Report No. 9929 (June 1999)
On a number of algorithms for solving non-stationary Stokes and Navier-Stokes equations *

Sergey Finogenov †

Abstract

In the present work we consider a number of algorithms for solving non-stationary Stokes and Navier-Stokes equations based on sequence of locally refined triangular meshes. The spatial discretization of the problems is performed by the finite-element method while the discretization in time is based on the so-called splitting technique.

1 Introduction

This work is devoted to a number of algorithms for solving two-dimensional non-stationary Stokes and Navier-Stokes equations in domains of a rather complicated shape. When solving the above class of problems the generation of proper meshes is an important issue. Normally, solutions contain boundary layers and, thus, the mesh has to be refined in the vicinity of the boundary of the domain.

We consider two spatial discretization procedures: the discretization based on the conforming piecewise-linear finite elements for both the pressure and the velocity components with velocity components discretized on a once uniformly refined mesh [5], [7] as well as the discretization based on the piecewise-linear finite elements for the pressure and the Nedelec basis functions for the velocity [4]. We confine ourselves to the case of triangular meshes.

For the discretization in time we exploit a scheme [8] which requires during each of the time steps a) inversion of an operator spectrally equivalent to the Helmholtz operator \((-\Delta + \omega), \omega > 0\) with homogeneous Dirichlet boundary conditions to compute the velocity vector field and b) inversion of an operator spectrally equivalent to the Laplace operator with homogeneous Neumann boundary conditions to determine the pressure field. Since the value of \(\omega\) is normally large, the step a) can be easily performed by using (unpreconditioned) iterative solution method. The step b), however, requires the use of a preconditioner to be efficient. We consider a preconditioner based on the fictitious domain and multilevel ideas [1], [2], [3]. We assume that the

---

*This work was partly supported by the NWO Dutch-Russian collaborative research program 047.003.017
†Institute of Numerical Mathematics, 8 Gubkina Str., GSP-1, 117333 Moscow, Russia
discretization mesh is hierarchical to be able to construct a multilevel structure within
the algorithm.

2 Standard solution method

In this section we present the standard approach to solving the non-stationary Stokes
and Navier-Stokes equations defined on a hierarchical sequence of locally refined tri­
angular finite-element meshes. We describe both the algorithm for constructing the
sequence of meshes and the algorithm for solving the discrete problem. In the fol­
lowing we assume that the computational domain is polygonal and, thus, the coarse
mesh in the above hierarchy of meshes can be easily constructed.

2.1 An algorithm for generation a hierarchical sequence of lo­
cally refined triangular finite-element meshes

The following conditions are imposed on the mesh:

- the mesh is hierarchical,
- the refinement degree could be different in different subdomains as well as near
  the subdomain boundaries,
- the boundary of the computational domain is approximated with the second
degree w.r.t. $h$, where $h$ denotes a characteristic mesh step size.

To construct the sequence of meshes we have to define first the coarse mesh. Then
the coarse mesh is uniformly refined a number of times by dividing each triangle into
4 congruent ones. Next, the local refinement procedure is applied in the vicinity of
the domain boundary: the triangles to be refined are divided into 4 congruent parts.
After performing this step the mesh becomes non-congruent i.e. it starts containing
“hanging” nodes. To avoid this, the triangles which contain a single “hanging” node
on one of their edges are split into two sub-triangles by means of dissection while
the triangles which contain more than one “hanging” node are again divided into 4
congruent sub-triangles. The process of “removing” the “hanging” nodes from the
mesh might take a few iterations depending on the structure of the local refinement.
Finally, the mesh is locally adapted to the boundary by shifting the nodes belonging
to the edges which cross the boundary onto the boundary.

In some cases an additional so-called “balancing” of the mesh nodes is useful. The
“balancing” procedure is defined as follows. Consider a cluster of all mesh cells which
have an internal mesh node in common (i.e. a node which does not belong to neither
$\partial \Omega$ nor $\partial \Pi$). The above cluster of triangles forms a polygon with the selected node
in the interior. To “balance” the mesh, this node is moved into the gravity center of
the polygon. The above procedure can be performed a number of times for each of
the mesh nodes.

The above steps of generating the sequence of meshes are illustrated in Figure 1.
The sub-figure a) presents a non-conforming mesh refined in the vicinity of $\partial \Omega$; the
Figure 1: Steps of the mesh generation procedure
sub-figure b) illustrates the algorithm for “removing” the “hanging” nodes; the sub-figure c) shows how the mesh can be locally adapted to the boundary and, finally, the sub-figure d) illustrates the “balancing” procedure.

2.2 Model problem. Standard formulation.

Consider a domain $\Omega \in \mathbb{R}^2$. The Navier-Stokes equations describing the flow of the incompressible viscous fluid can be written in the following form:

$$\frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla)u + \nabla p = 0 \quad \text{in} \quad \Omega,$$

$$\nabla \cdot u = 0 \quad \text{in} \quad \Omega,$$

$$u(x,0) = u_0(x), \quad x \in \Omega, \quad (\nabla \cdot u_0 = 0),$$

$$u = g_0 \quad \text{on} \quad \partial \Omega,$$

where $u$ and $p$ denote the velocity and the pressure field, respectively, $\nu > 0$ is the viscosity coefficient, $u_0$ is the initial velocity distribution at $t = 0$ and $g_0$ is a trace of $u_0$ onto the boundary $\partial \Omega$. We assume that the overall flux of $u$ through the boundary is zero, i.e.

$$\int_{\partial \Omega} g_0 \cdot n \, ds = 0,$$

where $n$ denotes an outward normal vector to $\partial \Omega$.

2.3 Variational formulation

Introduce the following functional spaces.

$$V_{g_0} = \{v | v \in (H^1(\Omega))^2, \quad v_0 = g_0 \quad \text{on} \quad \partial \Omega\},$$

$$V_0 = (H_0^1(\Omega))^2,$$

$$L^2_0(\Omega) = \{q | q \in L^2(\Omega), \quad \int_\Omega q \, dx = 0\}$$

The variational counterpart of problem (1)-(4) is:

For all $t \geq 0$ find a pair $\{u(t), p(t)\} \in V_{g_0}(\Omega) \times L^2_0(\Omega)$ such that

$$\int_{\Omega} \frac{\partial u}{\partial t} \cdot v \, dx + \nu \int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} (u \cdot \nabla)u \cdot v \, dx - \int_{\Omega} p \nabla \cdot v \, dx = 0,$$

$$\int_{\Omega} q \nabla \cdot u \, dx = 0,$$

$$u(0) = u_0,$$

$$u = g_0 \quad \text{on} \quad \partial \Omega,$$
2.4 Finite-element discretization

Let the domain $\Omega$ be contained in a "simple" domain $\Pi$, i.e. $\Omega \subset \Pi$. Let us assume that in the domain $\Pi$ the mesh refinement procedure has been performed (including the local refinement near the boundary) thus leading to a mesh $\tilde{\tau}_h$. Let the mesh $\tilde{\tau}_h$ be non-conforming. Then, by "removing" the "hanging" nodes and performing the local adaptation procedure we arrive at the mesh $\tau_h$. Next, we construct a conforming mesh $\tau_h$ by removing the triangles from $\Pi \setminus \Omega$. The meshes $\tilde{\tau}_h$, $\tau_h$ and $\tau_h$ have the following properties:

- the meshes $\tilde{\tau}_h$ and $\tau_h$ have the same number of nodes
- $\tau_h \subset \tilde{\tau}_h$
- the mesh $\tilde{\tau}_h$ is hierarchical
- the boundary of $\Omega$ is approximated by the edges of $\tau_h$ with the second degree of accuracy w.r.t. $h$.
- the mesh $\tilde{\tau}_h$ may contain a number of levels of the local refinement in $\Omega$.

Consider also a mesh $\tau_{h/2}$ which is obtained from the mesh $\tau_h$ by performing a single step of a uniform refinement procedure (i.e. each triangle of the mesh $\tau_h$ is assumed to be divided into 4 smaller congruent sub-triangles). Clearly, the mesh $\tau_{h/2}$ is conforming. Define the following finite-element spaces which approximate the spaces $V_{g_0}$, $V_0$, $L^2(\Omega)$ and $L^2_0(\Omega)$:

\[
V_{g_0h} = \{ v_h | v_h \in C_0^0(\Omega), \quad v_h|_T \in P_1 \times P_1, \quad \forall T \in \tau_{h/2}, \quad v_h|_{\Gamma} = g_{0h} \},
\]
(12)

\[
V_{0h} = \{ v_h | v_h \in C_0^0(\Omega), \quad v_h|_T \in P_1 \times P_1, \quad \forall T \in \tau_{h/2}, \quad v_h|_{\Gamma} = 0 \},
\]
(13)

\[
L^2_h = \{ q_h | q_h \in C_0^0(\Omega), \quad q_h|_T \in P_1, \quad \forall T \in \tau_h \}
\]
(14)

\[
L^2_{0h} = \{ q_h | q_h \in L^2_h, \quad \int_{\Omega} q_h dx = 0 \}.
\]
(15)

On the basis of the finite element spaces we define the finite element counterparts of the Laplace operator, the divergence operator and the identity operator:

\[
(Lu_h, v_h) \approx \int_{\Omega} \nabla u_h \cdot \nabla v_h d\Omega,
\]
(16)

\[
(Cu_h, q_h) \approx \int_{\Omega} q_h \nabla \cdot u_h d\Omega,
\]
(17)
We also define the diagonal operator $D$ as a lumped operator $\tilde{D}$.

Let us note that the operator $L$ is applied to functions $u_h$ and $v_h$ from the spaces $V_{goh}$ and $V_{oh}$ which satisfy the Dirichlet boundary conditions (homogeneous b.c. for $v_h$). For the operator $C$ defined by (17), the function $q_h \in L^2_{oh}$ satisfies the natural b.c. Thus, the finite element operator $L$ is a mapping from $V_{goh}$ (or $V_{oh}$) into $V_{oh}$, the operator $C$ acts from $V_{goh}$ (or $V_{oh}$) to $L^2_{oh}$, while the operator $C^T$ acts from $L^2_{oh}$ to $V_{oh}$.

2.5 Time discretization of the problem.

For approximate solution of system (1)-(4) we apply the following time discretization procedure:

$$\frac{\tilde{u}^{k+1} - \tilde{u}^k(x(-))}{\tau} - \nu \Delta \tilde{u}^{k+1} + \nabla(2p^k - p^{k-1}) = 0, \quad (19)$$

$$\Delta(p^{k+1} - p^k) = \frac{1}{\tau} \nabla \tilde{u}^{k+1}, \quad (20)$$

$$u^{k+1} = \tilde{u}^{k+1} - \tau \nabla(p^{k+1} - p^k) \quad (21)$$

Here $\tau$ is a time step, $u^k, p^k$ are the velocity and the pressure fields at $t = k\tau$ and $\tilde{u}^k$ is an auxiliary vector.

The notation $\tilde{u}^k(x(\cdot))$ implies the reconstruction by the method of characteristics.

The scheme is stable if $\tau = O(h)$ and its order of approximation accuracy is $O(\tau + h^2)$.

2.6 Matrix formulation of the discrete problem.

The time discretization scheme (19)-(21) can be written in the following matrix form:

$$D\tilde{u}^{k+1} - D\tilde{u}^k(x(\cdot)) + \nu L\tilde{u}^{k+1} - \tau C^T(2p^k - p^{k-1}) = 0, \quad (22)$$

$$CD^{-1}C^T(p^{k+1} - p^k) = -\tau^{-1}C\tilde{u}^{k+1}, \quad (23)$$

$$u^{k+1} = \tilde{u}^{k+1} + \tau D^{-1}C^T(p^{k+1} - p^k), \quad (24)$$

where $L, C, D$ are the matrices corresponding to the Laplace operator, divergence and the identity operators, respectively.
Note, that $D\tilde{u}^{k+1} - D\tilde{u}^{k}(x(\cdot)) \in V_{\Omega}$ and, thus, the equation (22) is correctly posed. Clearly, $CD^{-1}CT$ is symmetric and positive semidefinite. Since $C\tilde{u}^{k+1} \in L^2_{\Omega h}$, the equation (23) is also well defined. Therefore, the problem (22)-(24) is well posed if the boundary conditions are satisfied.

Let us rewrite the equation (22):

$$(D + \tau \nu L)\tilde{u}^{k+1} = D\tilde{u}^{k}(x(\cdot)) + \tau C^{T}(2p^{k} - p^{k-1}).$$  \hspace{1cm} (25)$$

Since the condition number of $D + \tau \nu L$ satisfies

$$\text{cond}(D + \tau \nu L) = O(\tau \nu/h^2),$$

we can apply the conjugate gradient method without preconditioning for solving (25).

Matrix $CD^{-1}CT$ is spectrally equivalent to the discrete counterpart of the Laplace operator with Neumann boundary conditions on $\partial\Omega$. Thus we can apply the preconditioned conjugate gradient method for solving (23). The preconditioner we use in the present work is based on the multigrid method coupled with the fictitious domain technique.

### 2.7 Multigrid/fictitious domain preconditioner.

We construct the preconditioner on the grid $\tilde{\tau}_{h}$ defined on $\Pi$ as shown in Figure 1a). We note that the grid $\tilde{\tau}_{h}$ is not conforming. The conformal triangulation $\tau_{h}$ of $\Omega$ is obtained from $\tilde{\tau}_{h}$ by shifting several nodes on $\partial\Omega$, splitting certain cells to have a conformal mesh, and removing the triangles outside $\Omega$.

In every cell of $\tilde{\tau}_{h}$ we define the local stiffness matrix

$$\tilde{A} = \begin{pmatrix}
1 & -0.5 & -0.5 \\
-0.5 & 1 & -0.5 \\
-0.5 & -0.5 & 1
\end{pmatrix}$$  \hspace{1cm} (26)$$

and the local mass matrix

$$\tilde{M} = \begin{pmatrix}
4/3 & 0 & 0 \\
0 & 4/3 & 0 \\
0 & 0 & 4/3
\end{pmatrix}$$  \hspace{1cm} (27)$$

The global matrices $A$ and $M$ are defined by assembling over the cells the respective local matrices. The system of algebraic equations

$$Ax = f$$  \hspace{1cm} (28)$$

is then a finite difference counterpart of the Poisson equation

$$-\Delta x = \tilde{f}$$  \hspace{1cm} (29)$$

with Neumann boundary conditions on $\partial\Pi$.  

7
The preconditioner $H$ is then defined as an application of the V-cycle multigrid method (Fedorenko-Bakhvalov) to equation (28). A single Jacobi iteration with matrix $M$ is used as a smoother.

It can be proved that the matrix $H$ defined above is symmetric and positive definite.

Numerical experiments show that $\text{cond}(HA) < 12$ in the space orthogonal to the kernel of $A$.

For solving (23) we apply the fictitious domain method with zero extension and preconditioner $H$. In this case the preconditioned conjugate gradient method is proven to be convergent and the rate of convergence is bounded by a constant independent of the mesh step size.

3 On a solution method based on the Nedelee discretization procedure

In this chapter we describe an algorithm for solving the Stokes and the Navier-Stokes equations discretized by means of the Nedelee vector functions. We demonstrate that this type of discretization leads to more efficient solution method than in the case when the standard discretization procedure is exploited. In particular, better efficiency of the method is achieved due to smaller memory requirements of the algorithm.

3.1 Formulation of the model problem

Consider the model problem defined in subsection 2.2. Let us recall the following well known equalities:

\[ \nabla \times \nabla \times \mathbf{u} = -\Delta \mathbf{u} + \nabla (\nabla \cdot \mathbf{u}); \]

\[ (\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \times \mathbf{u} \times \mathbf{u} + \nabla (\mathbf{u}^2/2). \]

Using the above equations the problem (1)-(4) can be reformulated as follows:

\[ \frac{\partial \mathbf{u}}{\partial t} + \nu \nabla \times \nabla \times \mathbf{u} - \mathbf{u} \times \nabla \times \mathbf{u} + \nabla \bar{p} = 0 \quad \text{in} \quad \Omega, \quad (30) \]

\[ \nabla \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega, \quad (31) \]

\[ \mathbf{u}(x,0) = \mathbf{u}_0(x), \quad x \in \Omega, \quad (\nabla \cdot \mathbf{u}_0 = 0), \quad (32) \]

\[ \mathbf{u} = \mathbf{g}_0 \quad \text{on} \quad \partial \Omega, \quad (33) \]

where $\mathbf{u}$ and $\bar{p}$ denote the velocity and the dynamic pressure, $\bar{p} = p + \mathbf{u}^2/2$ and $\nu > 0$ is a viscosity coefficient.

We assume that

\[ \int_{\partial \Omega} \mathbf{g}_0 \cdot \mathbf{n} ds = 0. \quad (34) \]
3.2 Variational formulation of the problem

Let us introduce the following functional spaces.

\[ V_{g_0} = \{ v | v \in H^{rot}(\Omega), \ v = g_0 \text{ on } \partial\Omega \}, \]  
\[ V_0 = H_0^{rot}(\Omega), \]  
\[ L_0^2(\Omega) = \{ q | q \in L^2(\Omega), \ \int_{\Omega} q dx = 0 \} \]

The variational counterpart of problem (30)-(33) is the following:

For all \( t \geq 0 \) find a pair \( \{ u(t), \rho(t) \} \in V_{g_0}(\Omega) \times L_0^2(\Omega) \) such that

\[
\int_\Omega \frac{\partial u}{\partial t} \cdot v d\Omega + \nu \int_\Omega \nabla \times u \cdot \nabla \times v d\Omega - \int_\Omega u \times \nabla \times u \cdot v d\Omega + \int_\Omega \nabla \rho \cdot v d\Omega = 0, \tag{38}
\]

\[
\int_\Omega \nabla q \cdot u d\Omega = \int_{\partial\Omega} g_0 q ds, \tag{39}
\]

\[
u(0) = u_0, \tag{40}
\]

\[ u = g_0 \text{ on } \partial\Omega, \tag{41} \]

Note, that the functions from \( H^{rot} \) and \( H_0^{rot} \) are assumed to have only those derivatives which are present in the rotor operator.

3.3 Finite element discretization of the problem

We define the following finite dimensional spaces which approximate \( V_{g_0}, V_0, L^2(\Omega), L_0^2(\Omega) \):

\[ V_{g_0 h} = \{ v_h | v_h \in H_h^{rot}(\Omega), \ v_h|_\Gamma = g_{0h} \}, \]  
\[ V_{0h} = \{ v_h | v_h \in H_h^{rot}(\Omega), \ v_h|_\Gamma = 0 \}, \]  
\[ L^2_h = \{ q_h | q_h \in C^0(\Omega), \ q_h|_T \in P_1, \ \forall T \in \tau_h \} \]

\[ L_{0h}^2 = \{ q_h | q_h \in L_{0h}^2, \ \int_{\Omega} q_h dx = 0 \}. \]
The space $H^{\text{rot}}(\Omega)$ will be approximated by a finite-element subspace spanned by the Nedelec vector basis functions [4]. The Nedelec basis functions are associated with the mesh edges rather than with the mesh nodes as it was in the case of the standard finite element discretization described in the previous section. Note, that the dimension of the space $H^{\text{rot}}_h$ equals $N_e$ as compared to the dimension of the space (5) which equals $2N_e + 2N_v$, where $N_e$ denotes the number of edges in the mesh and $N_v$ is a number of mesh nodes.

Let us define the Nedelec vector basis function $\varphi_i \in H^{\text{rot}}_h$ associated with a mesh edge $e_i$ as follows:

- $\varphi_i$ is a linear functions on the triangles adjacent to the edge $e_i$;
- the tangential component $\varphi_i^\| \parallel T$ of $\varphi_i$ equals one at the associated edge $e_i$, $\varphi_i^\| = 1$;
- $\varphi_i = 0$ at the rest of the mesh edges;
- $\varphi_i$ is non-zero only at the mesh nodes associated with the edge $e_i$ and is equal to zero at the rest of the mesh nodes.

We note that the above functions are associated with the edges rather than the nodes of the mesh.

Introduce the finite-element counterparts $R$, $C^T$ and $\tilde{D}$ of the operators $\nabla \times \nabla$, $\nabla$ and the identity operator as well as the operators for the convective term:

\[(R u_h, v_h) \approx \int_{\Omega} \nabla \times u_h \cdot \nabla \times v_h \, d\Omega, \quad (46)\]

\[(C^T \tilde{p}_h, v_h) \approx \int_{\Omega} \nabla \tilde{p}_h \cdot v_h \, d\Omega, \quad (47)\]

\[(\tilde{D} u_h, v_h) \approx \int_{\Omega} u_h \cdot v_h \, d\Omega, \quad (48)\]

\[(f_p, q_h) \approx \int_{\partial \Omega} g_0 q_h \, ds, \quad (49)\]

\[(B(b) u_h, v_h) \approx \int_{\Omega} \nabla \times u_h \cdot b \times v_h \, d\Omega, \quad (50)\]

\[(\tilde{B}(b) u_h, v_h) \approx \int_{\Omega} \nabla \times b \cdot u_h \times v_h \, d\Omega, \quad (51)\]

We also define the operator $D$ as a lumped operator $\tilde{D}$.  

3.4 Time discretization of the Navier-Stokes equations

Consider first the time discretization procedure of the Stokes equations. Namely, let us consider the scheme used in the previous section:

\[
\frac{\tilde{u}^{k+1} - \tilde{u}^k}{\tau} + \nu \nabla \times \nabla \times \tilde{u}^{k+1} + \nabla (2p^{k} - p^{k-1}) = 0, \tag{52}
\]

\[
\Delta (p^{k+1} - p^k) = \frac{1}{\tau} \nabla \tilde{u}^{k+1}, \tag{53}
\]

\[
\tilde{u}^{k+1} = \tilde{u}^{k+1} - \tau \nabla (p^{k+1} - p^k) \tag{54}
\]

As one can easily see, the above scheme is fully implicit. Indeed, by substituting \(\tilde{u}^{k+1}\) from (54) into (52) and using the obvious formulae

\[
\nabla \times \nabla \varphi = 0,
\]

which holds for every scalar function \(\varphi\), we obtain:

\[
\frac{u^{k+1} - u^k}{\tau} + \nu \nabla \times \nabla \times u^{k+1} + \nabla p^{k+1} = 0, \tag{55}
\]

The equation (53) can be obtained by applying the divergence operator to the equation (54). Thus, the equation (53) is equivalent to

\[
\nabla \cdot u^{k+1} = 0.
\]

Therefore, the scheme is fully implicit.

Let us now consider a number of different ways of presenting the convective term \(u \times \nabla \times u\) in the Navier-Stokes equations.

(i) First, consider the case when this term is given in the explicit form:

\[
\frac{\tilde{u}^{k+1} - \tilde{u}^k}{\tau} + \nu \nabla \times \nabla \times u^{k+1} - u^k \times \nabla \times u^k + \nabla (2p^{k} - p^{k-1}) = 0, \tag{56}
\]

\[
\Delta (\tilde{p}^{k+1} - \tilde{p}^k) = \frac{1}{\tau} \nabla \tilde{u}^{k+1}, \tag{57}
\]

\[
\tilde{u}^{k+1} = \tilde{u}^{k+1} - \tau \nabla (\tilde{p}^{k+1} - \tilde{p}^k) \tag{58}
\]

Clearly, the above scheme coincides with the following semi-explicit scheme:

\[
\frac{u^{k+1} - u^k}{\tau} + \nu \nabla \times \nabla \times u^{k+1} - u^k \times \nabla \times u^k + \nabla \tilde{p}^{k+1} = 0, \tag{59}
\]
\[ \nabla \cdot \mathbf{u}^{k+1} = 0. \]

(ii) Next, consider the case when the convection term is represented as \( \mathbf{u}^k \times \nabla \times \mathbf{u}^{k+1} \), i.e. the case when it enters the equations (59) as

\[ \mathbf{u}^k \times \nabla \times \mathbf{u}^{k+1}. \]

Note, that in this case we also obtain a well known scheme.

(iii) Consider the case when the convective term is represented as \( \mathbf{u}^{k+1} \times \nabla \times \mathbf{u}^k \). In this case it enters the equations (59) as

\[ \mathbf{u}^{k+1} \times \nabla \times \mathbf{u}^k + \tau \nabla (\mathbf{p}^{k+1} - \mathbf{p}^k) \times \nabla \times \mathbf{u}^k. \]

There are no estimates (at least up the author’s knowledge) for approximation and stability properties of the above scheme. These estimates will not be addressed in the present work as well. The present work will not address also a fully implicit scheme for Navier-Stokes equations.

3.5 Matrix formulation of the discrete problem

Consider the arising algebraic system for different approaches for approximating the convective term. First, let us consider the matrix formulation of the Stokes equation:

\[
D\mathbf{u}^{k+1} - D\mathbf{u}^k + \nu R\mathbf{u}^{k+1} + \tau CT(2\mathbf{p}^k - \mathbf{p}^{k-1}) = 0, \tag{60}
\]

\[
CD^{-1}CT(\mathbf{p}^{k+1} - \mathbf{p}^k) = -\tau^{-1}C\mathbf{u}^{k+1} + f_p, \tag{61}
\]

\[
\mathbf{u}^{k+1} = \mathbf{u}^{k+1} - \tau D^{-1}CT(\mathbf{p}^{k+1} - \mathbf{p}^k), \tag{62}
\]

where \( R, CT, D \) are the matrices corresponding to \( \nabla \times \nabla, \nabla \) and the identity operators, respectively.

Since the condition number of \( D + \tau \nu R \) satisfies

\[ \text{cond}(D + \tau \nu R) = O(\tau \nu / h^2). \]

We apply the conjugate gradient method without preconditioning for solving (60).

For solving (61) we apply the preconditioned conjugate gradient method. As in the previous section the preconditioner is based on the multigrid method coupled with the fictitious domain technique.

If the first approach is used for approximating the convective term then the following scheme is obtained:

\[
D\mathbf{u}^{k+1} - D\mathbf{u}^k + \nu R\mathbf{u}^{k+1} + B(\mathbf{u}^k)\mathbf{u}^k + \tau CT(2\mathbf{p}^k - \mathbf{p}^{k-1}) = 0, \tag{63}
\]
The algorithm for solving the above system coincides with the algorithm for solving the system (60)-(62) as they differ only in the right hand side. If the second approach is used then we have

\[ D\tilde{u}^{k+1} - D\tilde{u}^k + \tau \nu R\tilde{u}^{k+1} + B(u^k)u^{k+1} + \tau C^T(2p^k - p_0) = 0, \quad (66) \]

\[ CD^{-1}C^T(\tilde{p}^{k+1} - \tilde{p}^k) = -\tau^{-1}C\tilde{u}^{k+1} + f_p, \quad (67) \]

\[ u^{k+1} = u^{k+1} - \tau D^{-1}C^T(\tilde{p}^{k+1} - \tilde{p}^k), \quad (68) \]

In this case the equations (61) and (67) coincide while the operator of the equation (66) has the form

\[ S = D + \tau \nu R + \tau B(u^k). \]

Since \( B(u^k) \) is non-symmetric, the conjugate gradient method can not be applied for inverting it. The problems with the operator \( S \) will be solved by means of the generalized minimal residual (GMRES) method. Since the properties of \( S \) depend on the solution \( u^k \), we can say nothing about the convergence rate of the method.

If the third approach is used for approximating the convective term then the operator \( S \) takes the following form:

\[ S = D + \tau \nu R + \tau \bar{B}(u^k). \]

In this case the operator \( S \) is also non-symmetric, but \( \bar{B}(u^k) \) is skew-symmetric. Thus, the spectrum of \( S \) is located in the right semi-plane and \( Re(\lambda) > \lambda_{\min} \), where \( \lambda \) is an eigenvalue of \( S \) and \( \lambda_{\min} \) is a minimal eigenvalue of the s.p.d. operator \( D + \tau \mu R \). This guarantees the convergence of the GMRES method.

Therefore, if we use the first approach for approximating the convective term then we have certain restrictions on the time step (due to the stability reasons) but on the each time step we have to solve the problems with well-conditioned s.p.d. operators and, thus, efficient iterative methods can be used. If we use the second, semi-explicit, approach then the properties of the operator \( S \) are unknown, and the available algorithms for solving the problems on each of the time steps are less efficient. In the third case the convergence of the iterative scheme is guaranteed, but the finite-difference scheme we come up with is perturbed.

4 Conclusions

In this section we compare the above solution strategies. Since the dimension of \((H^1_h(\Omega))^2\) is twice as large as the dimension of \(H^{rot}(\Omega)\), the dimension of the problem
(25) is twice as large as the dimension of (60), and the solution times are also expected to be related similarly. For the problems (23) and (61) for the pressure the dimensions of the problems coincide and, thus, the computing times are also equal. This is confirmed by a number of numerical tests.

The Stokes equations were solved using the two approaches on the identical meshes. The difference between the solutions was in the order of $1 - 2\%$ for the mesh with about 5000 nodes (the pressure fields were used for the comparison).

When solving the Navier-Stokes equations the described above algorithms for approximating the convective term have produced very close (w.r.t. to the modeling accuracy) results. Arithmetic cost was higher for the cases 2 and 3 as compared to the case 1. Stability of the computations using the developed schemes was higher than for the standard one. Some instabilities were observed when using both approaches when the parameter $\nu$ was very close to 0. This can be explained either by the imperfectness of the time discretization scheme or by the insufficient quality of the discretization meshes, which were insufficiently refined near the boundary.

Acknowledgments

The author is grateful to prof. O. Axelsson and dr. Yu. Vassilevsky for the time they invested in a number of fruitful discussions about the method. The author would like also to thank dr. A. Padiy for the assistance in preparing the manuscript.

References