INFORMATICA, 2024, Vol. 35, No. 2, 341–361


DOI: https://doi.org/10.15388/24-INFOR543

# MINI Element for the Navier–Stokes System in 3D: Vectorized Codes and Superconvergence

Radek KUČERA[1,*], Vladimír ARZT[2], Jonas KOKO[3]

[1] *VSB – Technical University of Ostrava, Department of Mathematics and Descriptive Geometry, FME, 17. listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic*
[2] *VSB – Technical University of Ostrava, Department of Applied Mathematics, FEECS, 17. listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic*
[3] *Université Clermont Auvergne, Clermont Auvergne INP, CNRS, LIMOS, 63000 Clermont–Ferrand, France*
*e-mail: radek.kucera@vsb.cz, vladimir.arzt@vsb.cz, jonas.koko@uca.fr*

Received: May 2023; accepted: February 2024

**Abstract.** A fast vectorized codes for assembly mixed finite element matrices for the generalized Navier–Stokes system in three space dimensions in the MATLAB language are proposed by the MINI element. Vectorization means that the loop over tetrahedra is avoided. Numerical experiments illustrate computational efficiency of the codes. An experimental superconvergence rate for the pressure component is established.

**Key words:** Navier–Stokes system, mixed finite element method, MINI element, convergence rate, MATLAB.

## 1. Introduction

Numerical solution algorithms for the Navier Stokes equations is the rapidly developing field, in which various cost-effective methods are being proposed. This can be domain decomposition methods (Rønquist, 1996; Girault and Wheeler, 2008), methods based on splitting (Henriksen and Holmen, 2002; Viguerie and Veneziani, 2018), asymptotic expansions (Panasenko, 1998; Hoanga and Martinez, 2018), multigrid methods (Griebel *et al.*, 1998; Pernice and Tocci, 2001) etc. In this paper we use a simple Schur complement algorithm. A survey of the Schur complement methods can be found in Loghin and Wathen (2002).

The paper focuses on the generalized Navier–Stokes system in three space dimensions (3D) approximated by the mixed finite element method using the MINI element called also the P1-bubble/P1 pair (Arnold *et al.*, 1984). The main contribution of the paper consists in the development of the vectorized codes for fast assembly finite element matrices in the MATLAB language so that the loop over tetrahedra is avoided. These codes

*Corresponding author.

are very fast and enable to perform experiments with relatively large scale problems. Vectorized codes were proposed for different differential operators, e.g. see Rahman and Valdman (2015) and references therein. Our research is inspired by Koko (2019), where the vectorized codes for the generalized Stokes problem are proposed. However, an extension to the Navier–Stokes problem is not trivial. Formally, it consists of adding the nonlinear convective term in the momentum equation. This nonlinearity is typically treated iteratively using the Oseen or the Newton type linearization (Elman *et al.*, 2014). In the first case, the Oseen convection matrix depends on the nodal velocity field from the previous iteration that is presented in computations. The situation for the Newton convective matrix is more involved, since it depends, in addition, on the partial derivatives of the velocity field that are not immediately presented. We approximate them from appropriate directional derivatives so that computed approximations are invariant with respect to local renumbering of nodes and can by easily vectorized. Another ingredient in the assembling process is the bubble component elimination that is performed by the Schur complement reduction on the element level. This elimination requires inverting blocks of the local matrices that is done by the vectorized Cramer rule. The elimination itself uses a vectorized variant of linear combinations of vectors and of sums of vector outer products. Note that the vectorized codes for the Navier–Stokes problem play an important role in the whole solution process, since the finite element matrices for the linearized subproblems are repeatedly assembled in each iterative step that have to be fast.

Numerical experiments with our codes show superconvergence rate of the finite element approximation of the pressure component that is close to $\mathcal{O}(h^{3/2})$ or higher. Similar results were observed for the pure Stokes problem by Cioncolini and Boffi (2022).

The rest of the paper is organized as follows. Section 2 presents the classical and weak formulation of the problem and introduces the basic iterative schemes. In Section 3, we describe mixed finite element approximation based on the MINI element, for which we derive local linear systems that are split on the bubble and non-bubble components. We present also an idea how to approximate partial derivatives from the discrete vector field of the previous iteration. In Section 4, we discuss element matrices in more details so that their final forms may be coded by vectorized operations. Section 5 introduces ideas of the vectorization and refers to our free available codes. In Section 6, we describe the dual implementation of the iterative scheme that we use in computations. Section 7 is devoted to numerical experiments. First, we demonstrate low time requirements of the vectorized codes and than we compute the convercence rates. Finally, we conclude with some remarks and comments in Section 8.

To better understand our presentation we use different font styles. The mathematics bold symbols are used for the vector functions or their vector arguments, e.g. $\boldsymbol{u}$, $\boldsymbol{p}$. The "text up" symbols are used for matrices, vectors and scalars on the level of finite elements, e.g. $\mathrm{A}_{kk}$, $\mathrm{B}_l$, p. The "bold up" symbols are used for global matrices and vectors, e.g. $\mathbf{A}$, $\mathbf{B}_U$, $\mathbf{p}$. Finally, the typewriter style is used for codes, e.g. `for p = 1:4`.

## 2. Formulation

Let $\Omega \subset \mathbb{R}^3$ be a bounded domain with a sufficiently smooth boundary $\Gamma = \partial\Omega$. We consider the steady Navier–Stokes system with the homogeneous Dirichlet boundary condition:

$$\left.\begin{aligned} -\nu\Delta \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \alpha\boldsymbol{u} + \nabla p &= \boldsymbol{f} && \text{in } \Omega,\\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega,\\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{on } \Gamma, \end{aligned}\right\} \tag{1}$$

where $\nu > 0$ is the kinematic viscosity, $\alpha > 0$ is a constant from a time discretization of the unsteady problem, and $\boldsymbol{f} : \Omega \to \mathbb{R}^3$ represents the external forces. We are searching for the vector velocity field $\boldsymbol{u} : \Omega \to \mathbb{R}^3$ and the scalar pressure field $p : \Omega \to \mathbb{R}$. The existence and uniqueness of a weak solution to (1) are discussed in Girault and Raviart (1986).

We will consider two iterative methods for solving (1) with different linearizations of the convection term $\boldsymbol{u}\cdot\nabla\boldsymbol{u}$. Let $\boldsymbol{w}$ be an approximation of $\boldsymbol{u}$ and $\delta\boldsymbol{w}$ be an increment such that $\boldsymbol{u} = \boldsymbol{w} + \delta\boldsymbol{w}$. The Oseen type linearization is based on the omission of the linear term with $\delta\boldsymbol{w}$:

$$\boldsymbol{u}\cdot\nabla\boldsymbol{u} = (\boldsymbol{w}+\delta\boldsymbol{w})\cdot\nabla\boldsymbol{u} = \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \delta\boldsymbol{w}\cdot\nabla\boldsymbol{u} \approx \boldsymbol{w}\cdot\nabla\boldsymbol{u}.$$

In the Newton type linearization we omit the quadratic term with $\delta\boldsymbol{w}$:

$$\begin{aligned} \boldsymbol{u}\cdot\nabla\boldsymbol{u} &= \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \delta\boldsymbol{w}\cdot\nabla(\boldsymbol{w}+\delta\boldsymbol{w}) = \boldsymbol{w}\cdot\nabla\boldsymbol{u} + (\boldsymbol{u}-\boldsymbol{w})\cdot\nabla\boldsymbol{w} + \delta\boldsymbol{w}\cdot\nabla\delta\boldsymbol{w}\\ &\approx \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{w} - \boldsymbol{w}\cdot\nabla\boldsymbol{w}. \end{aligned}$$

We arrive at the following linearizations of (1):

$$\left.\begin{aligned} -\nu\Delta\boldsymbol{u} + \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \rho\boldsymbol{u}\cdot\nabla\boldsymbol{w} + \alpha\boldsymbol{u} + \nabla p &= \boldsymbol{f} + \rho\boldsymbol{w}\cdot\nabla\boldsymbol{w} && \text{in } \Omega,\\ \nabla\cdot\boldsymbol{u} &= 0 && \text{in } \Omega,\\ \boldsymbol{u} &= \boldsymbol{0} && \text{on } \Gamma, \end{aligned}\right\} \tag{2}$$

where $\boldsymbol{w} : \Omega \to \mathbb{R}^3$ is given. For $\rho = 0/1$, we get the Oseen/Newton linearization of (1).

The weak formulation of (2) requires the following spaces:

$$\boldsymbol{V} = \left(H_0^1(\Omega)\right)^3, \qquad Q = \left\{q \in L^2(\Omega) : \int_\Omega q\,\mathrm{d}\boldsymbol{x} = 0\right\}$$

and the forms:

$$a(\boldsymbol{u},\boldsymbol{v}) = \nu \sum_{i=1}^{3} \int_\Omega \nabla u_i \cdot \nabla v_i \,\mathrm{d}\boldsymbol{x}, \quad c(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) = \int_\Omega (\boldsymbol{w}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x},$$
$$m(\boldsymbol{u},\boldsymbol{v}) = \alpha \int_\Omega \boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x}, \quad b(\boldsymbol{v},q) = -\int_\Omega q(\nabla\cdot\boldsymbol{v})\,\mathrm{d}\boldsymbol{x},$$
$$a_\rho(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) = a(\boldsymbol{u},\boldsymbol{v}) + c(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) + \rho c(\boldsymbol{u};\boldsymbol{w},\boldsymbol{v}) + m(\boldsymbol{u},\boldsymbol{v}),$$
$$l_\rho(\boldsymbol{w};\boldsymbol{v}) = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \rho c(\boldsymbol{w};\boldsymbol{w},\boldsymbol{v}),$$

where $\boldsymbol{u},\boldsymbol{v},\boldsymbol{w} \in (H^1(\Omega))^3$, $q \in L^2(\Omega)$, and $\boldsymbol{u} = (u_1,u_2,u_3)$, $\boldsymbol{v} = (v_1,v_2,v_3)$.

The weak formulations of (2) read as follows:

$$\left.\begin{aligned} &\text{Find } (\boldsymbol{u},p) \in \boldsymbol{V}\times Q \text{ such that for all } (\boldsymbol{v},q)\in \boldsymbol{V}\times Q,\\ &a_\rho(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = l_\rho(\boldsymbol{w};\boldsymbol{v}),\\ &b(\boldsymbol{u},q) = 0. \end{aligned}\right\} \tag{3}$$

The problem (1) can be solved by the following iterative scheme:

$$\left.\begin{aligned} &\text{Given } \left(\boldsymbol{u}^{(0)},p^{(0)}\right) \in \boldsymbol{V}\times Q.\\ &\text{For } \kappa \geqslant 1, \text{ find } (\boldsymbol{u}^{(\kappa)},p^{(\kappa)}) \in \boldsymbol{V}\times Q \text{ solving (3) with } \boldsymbol{w} = \boldsymbol{u}^{(\kappa-1)}. \end{aligned}\right\} \tag{4}$$

This scheme will be called the Oseen/Newton iteration for $\rho = 0/1$, respectively. A comprehensive discussion of convergence results can be found in Elman *et al.* (2014), pp. 344–346. The Oseen iteration are also caled the Picard iteration.

## 3. Mixed Finite Element Approximation with the MINI Element

We approximate (3) by the mixed finite element method. It requires to choose a finite element pair satisfying the *inf-sup* condition (Brezzi and Fortin, 1991). Here, we use the P1-bubble/P1 pair called also the MINI element proposed by Arnold *et al.* (1984).

We suppose that $\Omega$ is a polyhedral domain. Let $\mathcal{T}_h$ be a regular partition of $\overline{\Omega}$ given by tetrahedra $T_j \in \mathcal{T}_h$, $1 \leqslant j \leqslant n_t$, Let $\boldsymbol{x}_i \in \overline{\Omega}$, $1 \leqslant i \leqslant n_p$, be finite element nodes. Each tetrahedron $T = T_j$ has 4 vertices $\boldsymbol{x}_{i_1},\boldsymbol{x}_{i_2},\boldsymbol{x}_{i_3},\boldsymbol{x}_{i_4} \in T$. Each vertex of $T$ is associated with the local linear basis function $\phi_j = \phi_j(\boldsymbol{x})$ such that $\phi_j(\boldsymbol{x}_{i_k}) = \delta_{jk}$, $1 \leqslant j,k \leqslant 4$. The bubble function is defined on $T$ by the product: $\phi_b = \phi_b(\boldsymbol{x}) = 4^4\phi_1(\boldsymbol{x})\phi_2(\boldsymbol{x})\phi_3(\boldsymbol{x})\phi_4(\boldsymbol{x})$, $\boldsymbol{x}\in T$. We denote also $\phi_5 = \phi_b$. Integrals over $T$ can be evaluated by the formula:

$$\int_T \phi_1^{\alpha_1}(\boldsymbol{x})\phi_2^{\alpha_2}(\boldsymbol{x})\phi_3^{\alpha_3}(\boldsymbol{x})\phi_4^{\alpha_4}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 6|T|\frac{\alpha_1!\alpha_2!\alpha_3!\alpha_4!}{(\alpha_1+\alpha_2+\alpha_3+\alpha_4)!},$$

where $|T|$ is the tetrahedron volume.

On $\mathcal{T}_h$ we introduce the space of the bubble functions and the piecewise linear functions:

$$B_h = \{v_h \in C(\bar{\Omega}) : v_{h|T} = c_T \phi_b,\ c_T \in \mathbb{R} \quad \forall T \in \mathcal{T}_h\},$$
$$W_h = \{v_h \in C(\bar{\Omega}) : v_{h|T} \in P^1(T) \quad \forall T \in \mathcal{T}_h\},$$

respectively, where $P^1(T)$ is the space of all polynomials on $T$ of the degree at least one. We approximate $\boldsymbol{V}$ and $Q$ as follows:

$$\boldsymbol{V}_h = \{\boldsymbol{v}_h \in (W_h \oplus B_h)^3 : \boldsymbol{v}_h(\boldsymbol{x}_i) = \boldsymbol{0} \quad \forall \boldsymbol{x}_i \in \Gamma\},$$
$$Q_h = \left\{q_h \in W_h : \int_\Omega q_h \,\mathrm{d}\boldsymbol{x} = 0\right\},$$

respectively. The finite element approximation of (3) reads as follows:

$$\left.\begin{aligned} &\text{Find } (\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h \text{ such that for all } (\boldsymbol{v}_h, p_h) \in \boldsymbol{V}_{0h} \times Q_h, \\ &a_\rho(\boldsymbol{w}_h; \boldsymbol{u}_h, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h) = l_\rho(\boldsymbol{w}_h; \boldsymbol{v}_h), \\ &b(\boldsymbol{u}_h, q_h) = 0, \end{aligned}\right\} \tag{5}$$

where $\boldsymbol{w}_h \in \boldsymbol{V}_h$ is an approximation of $\boldsymbol{w}$ from (3).

On $T \in \mathcal{T}_h$ the components of $\boldsymbol{u}_h = (u_{h1}, u_{h2}, u_{h3})$ are the linear combinations of the basis functions:

$$u_{hk}(\boldsymbol{x}) = \sum_{j=1}^{4} \mathrm{u}_{kj}\phi_j(\boldsymbol{x}) + \mathrm{u}_{kb}\phi_b(\boldsymbol{x}), \quad 1 \leqslant k \leqslant 3, \quad \text{and} \quad p_h(\boldsymbol{x}) = \sum_{j=1}^{4} \mathrm{p}_j \phi_j(\boldsymbol{x}).$$

The linear systems arising from (5) over one element $T$ read as:

$$\begin{pmatrix} \bar{\mathrm{A}}_{11} & \bar{\mathrm{A}}_{12} & \bar{\mathrm{A}}_{13} & \bar{\mathrm{B}}_1^\top \\ \bar{\mathrm{A}}_{21} & \bar{\mathrm{A}}_{22} & \bar{\mathrm{A}}_{23} & \bar{\mathrm{B}}_2^\top \\ \bar{\mathrm{A}}_{31} & \bar{\mathrm{A}}_{32} & \bar{\mathrm{A}}_{33} & \bar{\mathrm{B}}_3^\top \\ \bar{\mathrm{B}}_1 & \bar{\mathrm{B}}_2 & \bar{\mathrm{B}}_3 & 0 \end{pmatrix} \begin{pmatrix} \bar{\mathrm{u}}_1 \\ \bar{\mathrm{u}}_2 \\ \bar{\mathrm{u}}_3 \\ \mathrm{p} \end{pmatrix} = \begin{pmatrix} \bar{\mathrm{f}}_1 \\ \bar{\mathrm{f}}_2 \\ \bar{\mathrm{f}}_3 \\ 0 \end{pmatrix}, \tag{6}$$

where $\bar{\mathrm{u}}_k = (\mathrm{u}_{k1}, \mathrm{u}_{k2}, \mathrm{u}_{k3}, \mathrm{u}_{k4}, \mathrm{u}_{kb})^\top$, $1 \leqslant k \leqslant 3$, and $\mathrm{p} = (\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \mathrm{p}_4)^\top$. The matrices $\bar{\mathrm{A}}_{kl} \in \mathbb{R}^{5\times 5}$ are algebraic counterparts of the form $a_\rho$:

$$\bar{\mathrm{A}}_{kk} = \bar{\mathrm{R}} + \bar{\mathrm{C}} + \rho\bar{\mathrm{N}}_{kk} + \bar{\mathrm{M}}, \qquad \bar{\mathrm{A}}_{kl} = \rho\bar{\mathrm{N}}_{kl} \quad \text{for } k \neq l,$$

where $\bar{\mathrm{R}}$ is the diffusion matrix, $\bar{\mathrm{C}}$ is the Oseen convection matrix, $\bar{\mathrm{N}}_{kl}$ are the Newton convection matrices, $1 \leqslant k, l \leqslant 3$, and, $\bar{\mathrm{M}}$ is the mass matrix. Further, $\bar{\mathrm{B}}_k \in \mathbb{R}^{4\times 5}$ are the components of the divergence matrix representing the form $b$ and the right-hand sides

$\bar{\mathrm{f}}_k \in \mathbb{R}^5$, $1 \leqslant k \leqslant 3$, correspond to the form $l_\rho$. The entries are given as follows:

$$
\begin{aligned}
(\bar{\mathrm{R}})_{ij} &= \nu \int_T \nabla\phi_j(\boldsymbol{x}) \cdot \nabla\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad (\bar{\mathrm{C}})_{ij} = \sum_{k=1}^{3} \int_T w_{hk}(\boldsymbol{x})\partial_k\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \\
(\bar{\mathrm{N}}_{kl})_{ij} &= \int_T \partial_l w_{hk}(\boldsymbol{x})\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad (\bar{\mathrm{M}})_{ij} = \alpha \int_T \phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \\
(\bar{\mathrm{f}}_k)_i &= \int_T f_k(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \rho \sum_{l=1}^{3} \int_T w_{hl}(\boldsymbol{x})\partial_l w_{hk}(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}
\end{aligned}
\tag{7}
$$

for $1 \leqslant i, j \leqslant 5$, where $w_{hk}$ are the components of $\boldsymbol{w}_h$, and

$$
(\bar{\mathrm{B}}_k)_{ij} = -\int_T \partial_k\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{8}
$$

for $1 \leqslant i \leqslant 4$ and $1 \leqslant j \leqslant 5$.

Recall that $\boldsymbol{w}_h$ represents the velocity field from the previous iteration of discretized analogy of (4). In the discrete case we know only nodal values of $\boldsymbol{w}_h$ at the vertices $\boldsymbol{x}_i$ of $T$, $1 \leqslant i \leqslant 4$. In $(\bar{\mathrm{C}})_{ij}$, $(\bar{\mathrm{N}}_{kl})_{ij}$, and $(\bar{\mathrm{f}}_k)_i$ we will use constant approximations of $w_{hk}$ and $\partial_l w_{hk}$ on $T$. The approximation of $w_{hk}$ is defined by the arithmetic mean:

$$
\mathrm{w}_k = \frac{1}{4}(\mathrm{w}_{k1} + \mathrm{w}_{k2} + \mathrm{w}_{k3} + \mathrm{w}_{k4}), \quad 1 \leqslant k \leqslant 3,
$$

where $\mathrm{w}_{ki} = w_{hk}(\boldsymbol{x}_i)$, $1 \leqslant i \leqslant 4$. The approximation of $\partial_l w_{hk}$ denoted by $\delta_l \mathrm{w}_k$ will be taken by their value at $\boldsymbol{x}_1$: $\delta_l \mathrm{w}_k = \partial_l w_{hk}(\boldsymbol{x}_1)$. The relations between the gradient $\nabla w_{hk}(\boldsymbol{x}_1) = (\delta_1 \mathrm{w}_k, \delta_2 \mathrm{w}_k, \delta_3 \mathrm{w}_k)$ and the derivatives in the directions $\boldsymbol{x}_{i+1} - \boldsymbol{x}_1$ lead to three linear systems of the form:

$$
\nabla w_{hk}(\boldsymbol{x}_1) \cdot (\boldsymbol{x}_{i+1} - \boldsymbol{x}_1) = w_{hk}(\boldsymbol{x}_{i+1}) - w_{hk}(\boldsymbol{x}_1), \quad 1 \leqslant i \leqslant 3, \tag{9}
$$

from which $\delta_l \mathrm{w}_k$, $1 \leqslant k, l \leqslant 3$, can be computed. Note that the solutions are invariant with respect to (local) renumbering of the vertices $\boldsymbol{x}_i$ of $T$. Using the same symbols for the respective approximations of $(\bar{\mathrm{C}})_{ij}$, $(\bar{\mathrm{N}}_{kl})_{ij}$, and $(\bar{\mathrm{f}}_k)_i$, we get:

$$
(\bar{\mathrm{C}})_{ij} = \sum_{k=1}^{3} \mathrm{w}_k \int_T \partial_k\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad (\bar{\mathrm{N}}_{kl})_{ij} = \delta_l \mathrm{w}_k \int_T \phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{10}
$$

$$
(\bar{\mathrm{f}}_k)_i = \int_T f_k(\boldsymbol{x})\phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \rho \sum_{l=1}^{3} \mathrm{w}_l \delta_l \mathrm{w}_k \int_T \phi_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{11}
$$

Hence, $\bar{\mathrm{C}}$ is the linear combination of $\bar{\mathrm{B}}_k$ and $\bar{\mathrm{N}}_{kl}$ are multiples of $\bar{\mathrm{M}}$ so that the Ossen and Newton matrices may be assembled from the matrices of the Stokes problem.

The element matrices and vectors will be divided on the non-bubble and bubble components:

$$\bar{\mathrm{R}} = \begin{pmatrix} \mathrm{R} & \mathrm{r} \\ \mathrm{r}^\top & \omega_R \end{pmatrix}, \quad \bar{\mathrm{C}} = \begin{pmatrix} \mathrm{C} & \mathrm{c}_U \\ \mathrm{c}_L^\top & \omega_C \end{pmatrix}, \quad \bar{\mathrm{N}}_{kl} = \begin{pmatrix} \mathrm{N}_{kl} & \mathrm{n}_{kl} \\ \mathrm{n}_{kl}^\top & \omega_{N_{kl}} \end{pmatrix},$$

$$\bar{\mathrm{M}} = \begin{pmatrix} \mathrm{M}(\alpha) & \mathrm{m}(\alpha) \\ \mathrm{m}(\alpha)^\top & \omega_M(\alpha) \end{pmatrix}, \quad \bar{\mathrm{B}}_k = (\mathrm{B}_k, \mathrm{B}_{kb}), \quad \bar{\mathrm{f}}_k = \begin{pmatrix} \mathrm{f}_k \\ \mathrm{f}_{kb} \end{pmatrix}, \quad \bar{\mathrm{u}}_k = \begin{pmatrix} \mathrm{u}_k \\ \mathrm{u}_{kb} \end{pmatrix},$$

where $\mathrm{R}, \mathrm{M}(\alpha), \mathrm{B}_k, \mathrm{C}, \mathrm{N}_{kl} \in \mathbb{R}^{4\times 4}$, $\mathrm{r}, \mathrm{m}(\alpha), \mathrm{B}_{kb}, \mathrm{c}_U, \mathrm{c}_L, \mathrm{n}_{kl}, \mathrm{f}_k, \mathrm{u}_k \in \mathbb{R}^4$, and $\omega_R$, $\omega_M(\alpha), \omega_C, \omega_{N_{kl}}, \mathrm{f}_{kb}, \mathrm{u}_{kb} \in \mathbb{R}$ for $1 \leqslant k, l \leqslant 3$.

## 4. Element Matrices

The tetrahedron $T \in \mathcal{T}_h$ is represented by four vertices $\boldsymbol{x}_i = (x_i, y_i, z_i)$, $1 \leqslant i \leqslant 4$. We denote the entries of $\boldsymbol{x}_i - \boldsymbol{x}_j$ by $x_{[ij]} = x_i - x_j$, $y_{[ij]} = y_i - y_j$, $z_{[ij]} = z_i - z_j$ and $\mathrm{w}_{k[ij]} = \mathrm{w}_{ki} - \mathrm{w}_{kj}$, $i \neq j$.

### 4.1. *Derivatives of the Basis Functions*

The constant values of the basis functions derivatives on $T$ are given by the following formulas (see Arzt, 2019):

$$\begin{pmatrix} \partial_1\phi_1 \\ \partial_1\phi_2 \\ \partial_1\phi_3 \\ \partial_1\phi_4 \end{pmatrix} = \frac{1}{\det(\mathrm{X})}\mathrm{x}, \quad \begin{pmatrix} \partial_2\phi_1 \\ \partial_2\phi_2 \\ \partial_2\phi_3 \\ \partial_2\phi_4 \end{pmatrix} = \frac{1}{\det(\mathrm{X})}\mathrm{y}, \quad \begin{pmatrix} \partial_3\phi_1 \\ \partial_3\phi_2 \\ \partial_3\phi_3 \\ \partial_3\phi_4 \end{pmatrix} = \frac{1}{\det(\mathrm{X})}\mathrm{z},$$

where

$$\mathrm{x} = \begin{pmatrix} y_{[42]}z_{[32]} - y_{[32]}z_{[42]} \\ y_{[31]}z_{[41]} - y_{[41]}z_{[31]} \\ y_{[41]}z_{[21]} - y_{[21]}z_{[41]} \\ y_{[21]}z_{[31]} - y_{[31]}z_{[21]} \end{pmatrix}, \quad \mathrm{y} = \begin{pmatrix} x_{[32]}z_{[42]} - x_{[42]}z_{[32]} \\ x_{[41]}z_{[31]} - x_{[31]}z_{[41]} \\ x_{[21]}z_{[41]} - x_{[41]}z_{[21]} \\ x_{[31]}z_{[21]} - x_{[21]}z_{[31]} \end{pmatrix},$$

$$\mathrm{z} = \begin{pmatrix} x_{[42]}y_{[32]} - x_{[32]}y_{[42]} \\ x_{[31]}y_{[41]} - x_{[41]}y_{[31]} \\ x_{[41]}y_{[21]} - x_{[21]}y_{[41]} \\ x_{[21]}y_{[31]} - x_{[31]}y_{[21]} \end{pmatrix}, \quad \mathrm{X} = \begin{pmatrix} x_{[21]} & x_{[31]} & x_{[41]} \\ y_{[21]} & y_{[31]} & y_{[41]} \\ z_{[21]} & z_{[31]} & z_{[41]} \end{pmatrix}.$$

The volume $|T|$ of the tetrahedron $T$ can be computed by:

$$|T| = \frac{1}{6}\big|\det(\mathrm{X})\big| = \frac{1}{6}|x_{[21]}\mathrm{x}_2 + x_{[31]}\mathrm{x}_3 + x_{[41]}\mathrm{x}_4|.$$

4.2. *Approximation of the Velocity Derivatives*

The linear systems (9) read as follows:

$$\begin{pmatrix} x_{[21]} & y_{[21]} & z_{[21]} \\ x_{[31]} & y_{[31]} & z_{[31]} \\ x_{[41]} & y_{[41]} & z_{[41]} \end{pmatrix} \begin{pmatrix} \delta_1 \mathrm{w}_k \\ \delta_2 \mathrm{w}_k \\ \delta_3 \mathrm{w}_k \end{pmatrix} = \begin{pmatrix} \mathrm{w}_{k[21]} \\ \mathrm{w}_{k[31]} \\ \mathrm{w}_{k[41]} \end{pmatrix} \quad \text{for } 1 \leqslant k \leqslant 3.$$

Using Cramer's rule, we get:

$$\begin{aligned} \delta_1 \mathrm{w}_k &= (\mathrm{w}_{k[21]}\mathrm{x}_2 + \mathrm{w}_{k[31]}\mathrm{x}_3 + \mathrm{w}_{k[41]}\mathrm{x}_4)/\det(\mathrm{X}), \\ \delta_2 \mathrm{w}_k &= (\mathrm{w}_{k[21]}\mathrm{y}_2 + \mathrm{w}_{k[31]}\mathrm{y}_3 + \mathrm{w}_{k[41]}\mathrm{y}_4)/\det(\mathrm{X}), \\ \delta_3 \mathrm{w}_k &= (\mathrm{w}_{k[21]}\mathrm{z}_2 + \mathrm{w}_{k[31]}\mathrm{z}_3 + \mathrm{w}_{k[41]}\mathrm{z}_4)/\det(\mathrm{X}) \end{aligned}$$

for $1 \leqslant k \leqslant 3$.

4.3. *Stokes Matrices*

The following formulas are adopted from Koko (2019). For the diffusion matrix we get:

$$\begin{aligned} \mathrm{R} &= \frac{\nu}{36|T|}\left(\mathrm{x}\mathrm{x}^\top + \mathrm{y}\mathrm{y}^\top + \mathrm{z}\mathrm{z}^\top\right), \\ \omega_R &= \frac{2048\nu}{8505|T|}\big(\mathrm{x}_1^2 + \mathrm{y}_1^2 + \mathrm{z}_1^2 - \mathrm{x}_2(\mathrm{x}_3 + \mathrm{x}_4) - \mathrm{x}_3\mathrm{x}_4 - \mathrm{y}_2(\mathrm{y}_3 + \mathrm{y}_4) - \mathrm{y}_3\mathrm{y}_4 - \\ &\quad - \mathrm{z}_2(\mathrm{z}_3 + \mathrm{z}_4) - \mathrm{z}_3\mathrm{z}_4\big), \end{aligned}$$

and $\mathrm{r} = 0$. For the mass matrix we get:

$$\begin{aligned} \mathrm{M}(\alpha) &= \frac{\alpha|T|}{20}\begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}, \quad \mathrm{m}(\alpha) = \frac{8\alpha|T|}{105}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \\ \omega_M(\alpha) &= \frac{8192\alpha|T|}{51975}. \end{aligned}$$

For the divergence matrix we get:

$$\begin{aligned} \mathrm{B}_1 &= -\frac{S}{24}\begin{pmatrix} \mathrm{x}^\top \\ \mathrm{x}^\top \\ \mathrm{x}^\top \\ \mathrm{x}^\top \end{pmatrix}, \quad \mathrm{B}_2 = -\frac{S}{24}\begin{pmatrix} \mathrm{y}^\top \\ \mathrm{y}^\top \\ \mathrm{y}^\top \\ \mathrm{y}^\top \end{pmatrix}, \quad \mathrm{B}_3 = -\frac{S}{24}\begin{pmatrix} \mathrm{z}^\top \\ \mathrm{z}^\top \\ \mathrm{z}^\top \\ \mathrm{z}^\top \end{pmatrix}, \\ \mathrm{B}_{1b} &= \frac{16S}{315}\mathrm{x}, \quad \mathrm{B}_{2b} = \frac{16S}{315}\mathrm{y}, \quad \mathrm{B}_{3b} = \frac{16S}{315}\mathrm{z}, \end{aligned}$$

where $S = \mathrm{sgn}(\det(\mathrm{X}))$. The signum-function is due to $|\det(\mathrm{X})|/\det(\mathrm{X})$.

4.4. *Oseen and Newton Convection Matrices*

Using Green's formula and $\phi_b = \phi_5 = 0$ on $\partial T$, we obtain:

$$(\bar{\mathrm{C}})_{i,5} = \sum_{k=1}^{3} \mathrm{w}_k \int_T \partial_k \phi_b(\boldsymbol{x}) \phi_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = -\sum_{k=1}^{3} \mathrm{w}_k \int_T \partial_k \phi_i(\boldsymbol{x}) \phi_b(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = -(\bar{\mathrm{C}})_{5,i}$$

for $1 \leqslant i \leqslant 4$. It implies $\mathrm{c}_L = -\mathrm{c}_U$. Further:

$$\omega_C = (\bar{\mathrm{C}})_{5,5} = \sum_{k=1}^{3} \mathrm{w}_k \int_T \partial_k \phi_b(\boldsymbol{x}) \phi_b(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \sum_{k=1}^{3} \mathrm{w}_k \sum_{i=1}^{4} \partial_k \phi_i(\boldsymbol{x}) I_i = 0,$$

since $\sum_{i=0}^{4} \partial_k \phi_i(\boldsymbol{x}) = 0$ on $T$ and $I_i = 4^4 \int_T \prod_{j \neq i} \phi_j(\boldsymbol{x}) \phi_b(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$ does not depend on $i$. Comparing the first equality in (10) with (8), we can write $\mathrm{C} = -\sum_{k=1}^{3} \mathrm{w}_k \mathrm{B}_k$ and $\mathrm{c}_U = -\sum_{k=1}^{3} \mathrm{w}_k \mathrm{B}_{kb}$. For the Oseen convection matrix we get:

$$\mathrm{C} = \frac{S}{24} \left( \mathrm{w}_1 \begin{pmatrix} \mathrm{x}^\top \\ \mathrm{x}^\top \\ \mathrm{x}^\top \\ \mathrm{x}^\top \end{pmatrix} + \mathrm{w}_2 \begin{pmatrix} \mathrm{y}^\top \\ \mathrm{y}^\top \\ \mathrm{y}^\top \\ \mathrm{y}^\top \end{pmatrix} + \mathrm{w}_3 \begin{pmatrix} \mathrm{z}^\top \\ \mathrm{z}^\top \\ \mathrm{z}^\top \\ \mathrm{z}^\top \end{pmatrix} \right),$$

$$\mathrm{c}_U = -\frac{16S}{315} (\mathrm{w}_1 \mathrm{x} + \mathrm{w}_2 \mathrm{y} + \mathrm{w}_2 \mathrm{z}),$$

$\mathrm{c}_L = -\mathrm{c}_U$, and $\omega_C = 0$.

Comparing $\tilde{\mathrm{N}}$ in (10) with $\bar{\mathrm{M}}$ in (7), we get for the Newton convection matrices:

$$\mathrm{N}_{kl} = \mathrm{M}(\delta_l \mathrm{w}_k), \quad \mathrm{n}_{kl} = \mathrm{m}(\delta_l \mathrm{w}_k), \quad \omega_{N_{kl}} = \omega_M(\delta_l \mathrm{w}_k) \quad \text{for } 1 \leqslant k, l \leqslant 3.$$

4.5. *Right-Hand Side Vectors*

The functions $f_k$ are approximated on $T$ by the mean values. Denoting:

$$\tilde{\mathrm{f}}_k = \frac{1}{4} \sum_{i=1}^{4} f_k(\boldsymbol{x}_i) + \rho \sum_{l=1}^{3} \mathrm{w}_l \delta_l \mathrm{w}_k,$$

we get from (11):

$$\mathrm{f}_k = \frac{|T|}{4} \tilde{\mathrm{f}}_k \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathrm{f}_{kb} = \frac{32\,|T|}{105} \tilde{\mathrm{f}}_k \quad \text{for } 1 \leqslant k \leqslant 3.$$

4.6. *Bubble Components Elimination*

Let us permute the system (6) as follows:

$$\left(\begin{array}{ccc|ccc|c} \mathrm{A}_{11} & \mathrm{A}_{12} & \mathrm{A}_{13} & \mathrm{Z}_{U,11} & \mathrm{Z}_{U,12} & \mathrm{Z}_{U,13} & \mathrm{B}_1^\top \\ \mathrm{A}_{21} & \mathrm{A}_{22} & \mathrm{A}_{23} & \mathrm{Z}_{U,21} & \mathrm{Z}_{U,22} & \mathrm{Z}_{U,23} & \mathrm{B}_2^\top \\ \mathrm{A}_{31} & \mathrm{A}_{32} & \mathrm{A}_{33} & \mathrm{Z}_{U,31} & \mathrm{Z}_{U,32} & \mathrm{Z}_{U,33} & \mathrm{B}_3^\top \\ \hline \mathrm{Z}_{L,11}^\top & \mathrm{Z}_{L,21}^\top & \mathrm{Z}_{L,31}^\top & \omega_{11} & \omega_{12} & \omega_{13} & \mathrm{B}_{1b}^\top \\ \mathrm{Z}_{L,12}^\top & \mathrm{Z}_{L,22}^\top & \mathrm{Z}_{L,32}^\top & \omega_{21} & \omega_{22} & \omega_{23} & \mathrm{B}_{2b}^\top \\ \mathrm{Z}_{L,13}^\top & \mathrm{Z}_{L,23}^\top & \mathrm{Z}_{L,33}^\top & \omega_{31} & \omega_{32} & \omega_{33} & \mathrm{B}_{3b}^\top \\ \hline \mathrm{B}_1 & \mathrm{B}_2 & \mathrm{B}_3 & \mathrm{B}_{1b} & \mathrm{B}_{2b} & \mathrm{B}_{3b} & 0 \end{array}\right) \left(\begin{array}{c} \mathrm{u}_1 \\ \mathrm{u}_2 \\ \mathrm{u}_3 \\ \mathrm{u}_{1b} \\ \mathrm{u}_{2b} \\ \mathrm{u}_{3b} \\ \mathrm{p} \end{array}\right) = \left(\begin{array}{c} \mathrm{f}_1 \\ \mathrm{f}_2 \\ \mathrm{f}_3 \\ \mathrm{f}_{1b} \\ \mathrm{f}_{2b} \\ \mathrm{f}_{3b} \\ 0 \end{array}\right), \tag{12}$$

where $\mathrm{A}_{kk} = \mathrm{R} + \mathrm{C} + \rho\mathrm{N}_{kk} + \mathrm{M}(\alpha)$, $\mathrm{Z}_{U,kk} = \mathrm{c}_U + \rho\mathrm{n}_{kk} + \mathrm{m}(\alpha)$, $\mathrm{Z}_{L,kk} = -\mathrm{c}_U + \rho\mathrm{n}_{kk} + \mathrm{m}(\alpha)$, $\omega_{kk} = \omega_R + \rho\omega_{N_{kk}} + \omega_M(\alpha)$, and $\mathrm{A}_{kl} = \rho\mathrm{N}_{kl}$, $\mathrm{Z}_{U,kl} = \mathrm{Z}_{L,kl} = \rho\mathrm{n}_{kl}$, $\omega_{kl} = \rho\omega_{N_{kl}}$, $k \neq l$, for $1 \leqslant k, l \leqslant 3$. For the Oseen linearization ($\rho = 0$), the blocks (1, 1), (1, 2), (2, 1), (2, 2) (denoted by lines) of the matrix in (12) are block diagonal. For this simpler case the bubble component elimination is described in Koko (2019). Let us consider the Newton linearization ($\rho \neq 0$). The middle block equations in (12) gives:

$$\left(\begin{array}{c} \mathrm{u}_{1b} \\ \mathrm{u}_{2b} \\ \mathrm{u}_{3b} \end{array}\right) = W^{-1} \left( \left(\begin{array}{c} \mathrm{f}_{1b} \\ \mathrm{f}_{2b} \\ \mathrm{f}_{3b} \end{array}\right) - \left(\begin{array}{ccc} \mathrm{Z}_{L,11}^\top & \mathrm{Z}_{L,21}^\top & \mathrm{Z}_{L,31}^\top \\ \mathrm{Z}_{L,12}^\top & \mathrm{Z}_{L,22}^\top & \mathrm{Z}_{L,32}^\top \\ \mathrm{Z}_{L,13}^\top & \mathrm{Z}_{L,23}^\top & \mathrm{Z}_{L,33}^\top \end{array}\right) \left(\begin{array}{c} \mathrm{u}_1 \\ \mathrm{u}_2 \\ \mathrm{u}_3 \end{array}\right) - \left(\begin{array}{c} \mathrm{B}_{1b}^\top \\ \mathrm{B}_{2b}^\top \\ \mathrm{B}_{3b}^\top \end{array}\right) \mathrm{p} \right), \tag{13}$$

where $W = (\omega_{ij}) \in \mathbb{R}^{3\times 3}$. The inverse $W^{-1} = (\hat{\omega}_{ij})$ is given by Cramer's rule:

$$\begin{aligned} \det(W) &= \omega_{11}\omega_{22}\omega_{33} + \omega_{12}\omega_{23}\omega_{31} + \omega_{13}\omega_{21}\omega_{32} - \omega_{13}\omega_{22}\omega_{31} - \\ &\quad - \omega_{12}\omega_{21}\omega_{33} - \omega_{11}\omega_{23}\omega_{32}, \\ \hat{\omega}_{11} &= (\omega_{22}\omega_{33} - \omega_{23}\omega_{32})/\det(W), \quad \hat{\omega}_{12} = (\omega_{13}\omega_{32} - \omega_{12}\omega_{33})/\det(W), \\ \hat{\omega}_{13} &= (\omega_{12}\omega_{23} - \omega_{13}\omega_{22})/\det(W), \\ \hat{\omega}_{21} &= (\omega_{23}\omega_{31} - \omega_{21}\omega_{33})/\det(W), \quad \hat{\omega}_{22} = (\omega_{11}\omega_{33} - \omega_{13}\omega_{31})/\det(W), \\ \hat{\omega}_{23} &= (\omega_{13}\omega_{21} - \omega_{11}\omega_{23})/\det(W), \\ \hat{\omega}_{31} &= (\omega_{21}\omega_{32} - \omega_{22}\omega_{31})/\det(W), \quad \hat{\omega}_{32} = (\omega_{12}\omega_{31} - \omega_{11}\omega_{32})/\det(W), \\ \hat{\omega}_{33} &= (\omega_{11}\omega_{22} - \omega_{12}\omega_{21})/\det(W). \end{aligned}$$

Applying (13) in the first and the last block equations of (12) we arrive at:

$$\left(\begin{array}{ccc|c} \hat{\mathrm{A}}_{11} & \hat{\mathrm{A}}_{12} & \hat{\mathrm{A}}_{13} & \hat{\mathrm{B}}_{U,1}^\top \\ \hat{\mathrm{A}}_{21} & \hat{\mathrm{A}}_{22} & \hat{\mathrm{A}}_{23} & \hat{\mathrm{B}}_{U,2}^\top \\ \hat{\mathrm{A}}_{31} & \hat{\mathrm{A}}_{32} & \hat{\mathrm{A}}_{33} & \hat{\mathrm{B}}_{U,3}^\top \\ \hline \hat{\mathrm{B}}_{L,1} & \hat{\mathrm{B}}_{L,2} & \hat{\mathrm{B}}_{L,3} & -\hat{\mathrm{E}} \end{array}\right) \left(\begin{array}{c} \mathrm{u}_1 \\ \mathrm{u}_2 \\ \mathrm{u}_3 \\ \mathrm{p} \end{array}\right) = \left(\begin{array}{c} \hat{\mathrm{f}}_1 \\ \hat{\mathrm{f}}_2 \\ \hat{\mathrm{f}}_3 \\ \hat{\mathrm{g}} \end{array}\right), \tag{14}$$

where

$$\hat{\mathrm{A}}_{kl} = \mathrm{A}_{kl} - \sum_{j=1}^{3} \hat{\mathrm{Z}}_{U,kj} \mathrm{Z}_{L,lj}^{\top}, \quad \hat{\mathrm{f}}_k = \mathrm{f}_k - \sum_{j=1}^{3} \hat{\mathrm{Z}}_{U,k1} \mathrm{f}_{kb},$$

$$\hat{\mathrm{B}}_{U,k} = \mathrm{B}_k - \sum_{j=1}^{3} \mathrm{B}_{jb} \hat{\mathrm{Z}}_{U,kj}^{\top}, \quad \hat{\mathrm{B}}_{L,k} = \mathrm{B}_k - \sum_{j=1}^{3} \hat{\mathrm{B}}_{jb} \mathrm{Z}_{L,kj}^{\top},$$

$$\hat{\mathrm{E}} = \sum_{j=1}^{3} \hat{\mathrm{B}}_{bj} \mathrm{B}_{bj}^{\top}, \quad \hat{\mathrm{g}} = -\sum_{j=1}^{3} \hat{\mathrm{B}}_{bj} \mathrm{f}_{bj},$$

$$\hat{\mathrm{Z}}_{U,kl} = \sum_{j=1}^{3} \mathrm{Z}_{U,kj} \hat{\omega}_{jl}, \quad \hat{\mathrm{B}}_{kb} = \sum_{j=1}^{3} \mathrm{B}_{jb} \hat{\omega}_{jk}$$

for $1 \leqslant k, l \leqslant 3$. These formulas can be easily vectorized, since they are given by linear combinations of vectors or by outer products of vectors.

## 5. Vectorized Coding

The local matrices and vectors $\hat{\mathrm{A}}_{kl}$, $\hat{\mathrm{B}}_{U,k}$, $\hat{\mathrm{B}}_{L,k}$, $\hat{\mathrm{E}}$, $\hat{\mathrm{f}}_k$, $\hat{\mathrm{g}}$ on $T_j \in \mathcal{T}_h$, $1 \leqslant j \leqslant n_t$, derived in (14), are associated with the global ones $\mathbf{A}_{kl}$, $\mathbf{B}_{U,k}$, $\mathbf{B}_{L,k}$, $\mathbf{E}$, $\mathbf{f}_k$, $\mathbf{g}$, $1 \leqslant k, l \leqslant 3$, respectively, through the ordered index-set $\mathcal{J} = [j_1, j_2, j_3, j_4]$ determined by the vertices $\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \boldsymbol{x}_{j_3}, \boldsymbol{x}_{j_4}$ of $T_j$. For instance:

$$(\mathbf{A}_{kl})_{\mathcal{J}\mathcal{J}} := (\mathbf{A}_{kl})_{\mathcal{J}\mathcal{J}} + \hat{\mathrm{A}}_{kl}, \quad (\mathbf{f}_k)_{\mathcal{J}} := (\mathbf{f}_k)_{\mathcal{J}} + \hat{\mathrm{f}}_k,$$

and analogously for other matrices and vectors. An usual assembly procedure uses three loops: the outer loop over all $T_j$ and two (one in case of vectors) inner loops over indices of $\mathcal{J}$. In vectorized coding we interchange positions of the loops so that the short loops over $\mathcal{J}$ are the outer ones while the inner loop over all $T_j$ is replaced by appropriate vectorized operations. To demonstrate this process, we show in an abstract setting how to vectorize outer products of vectors appearing in $\hat{\mathrm{A}}_{kl}$, $\hat{\mathrm{B}}_{U,k}$, $\hat{\mathrm{B}}_{L,k}$, and $\hat{\mathrm{E}}$.

We assume that for $T_j \in \mathcal{T}_h$, $1 \leqslant j \leqslant n_t$, the indices of $\mathcal{J}$ are stored in the $j$-th row of the array `t` of the size $n_t \times 4$ so that:

$$\mathcal{J} = \mathtt{t(j, 1:4)}.$$

Let $\mathrm{v}_j, \mathrm{w}_j \in \mathbb{R}^4$ be column vectors that define the local contribution to the (abstract) global matrix $\mathbf{A}$ corresponding to $T_j$ so that:

$$\mathbf{A}_{\mathcal{J}\mathcal{J}} := \mathbf{A}_{\mathcal{J}\mathcal{J}} + \mathrm{v}_j \mathrm{w}_j^{\top}.$$

We assume also that $\mathrm{v}_j$, $\mathrm{w}_j$ are stored in the $j$-th rows of the arrays v, w of the size $n_t \times 4$ so that:

$$\mathrm{v}_j^\top = \mathtt{v(j,1:4)}, \quad \mathrm{w}_j^\top = \mathtt{w(j,1:4)},$$

respectively. Using three loops we update A (the MATLAB representation of $\mathbf{A}$) as follows:

```
for i = j:nt
   for p = 1:4
      for q = 1:4
         A(t(j,p),t(j,q)) = A(t(j,p),t(j,q))+v(j,p)*w(j,q).
```

The same effect can be achieved by the following vectorized code:

```
for p = 1:4
   for q = 1:4
      A(t(:,p),t(:,q)) = A(t(:,p),t(:,q))+v(:,p).*w(:,q).
```

Here, the dot-product "`.*`" and the addition "+" are the vectorized MATLAB operations that are performed on the low level and, therefore, are fast. For more details, we refer to Koko (2019) and to our free available codes (Kučera *et al.*, 2023).

## 6. Algebraic Iterative Scheme

The algebraic version of the iterative scheme (4) reads as follows:

$$\left.\begin{array}{l} \text{Given } (\mathbf{u}^{(0)}, \mathbf{p}^{(0)}) \in \mathbb{R}^{3n_p} \times \mathbb{R}^{n_p}. \\ \text{For } \kappa \geqslant 1, \text{ solve} \\ \begin{pmatrix} \mathbf{A}(\mathbf{u}^{(\kappa-1)}) & \mathbf{B}_U^\top(\mathbf{u}^{(\kappa-1)}) \\ \mathbf{B}_L(\mathbf{u}^{(\kappa-1)}) & -\mathbf{E}(\mathbf{u}^{(\kappa-1)}) \end{pmatrix} \begin{pmatrix} \mathbf{u}^{(\kappa)} \\ \mathbf{p}^{(\kappa)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{u}^{(\kappa-1)}) \\ \mathbf{g}(\mathbf{u}^{(\kappa-1)}) \end{pmatrix}, \end{array}\right\} \tag{15}$$

where $\mathbf{A} = \mathbf{A}(\mathbf{u}^{(\kappa-1)})$, $\mathbf{B}_U = \mathbf{B}_U(\mathbf{u}^{(\kappa-1)})$, $\mathbf{B}_L = \mathbf{B}_L(\mathbf{u}^{(\kappa-1)})$, and $\mathbf{f} = \mathbf{f}(\mathbf{u}^{(\kappa-1)})$ are composed from the blocks $\mathbf{A}_{kl}, \mathbf{B}_{U,k}, \mathbf{B}_{L,k}$, and $\mathbf{f}_k$, $1 \leqslant k, l \leqslant d$, respectively. Dependency of all blocks on the velocity field from the previous iteration $\mathbf{u}^{(\kappa-1)}$ is due to the bubble component elimination (on the element level). Other artefacts of this elimination are the presence of the blocks $\mathbf{E} = \mathbf{E}(\mathbf{u}^{(\kappa-1)})$, $\mathbf{g} = \mathbf{g}(\mathbf{u}^{(\kappa-1)})$ and the fact that $\mathbf{B}_U \neq \mathbf{B}_L$. The linear system in (15) is naturally adapted by the homogeneous Dirichlet boundary data of $(1)_3$. We initialize (15) by $\mathbf{u}^{(0)} = \mathbf{0}$, $\mathbf{p}^{(0)} = \mathbf{0}$.

Our implementation of (15) is based on an inexact dual strategy. In each step of (15) we solve iteratively the Schur complement linear systems:

$$\mathbf{S}\mathbf{p}^{(\kappa)} = \mathbf{d}, \tag{16}$$

where $\mathbf{S} = \mathbf{B}_L\mathbf{A}^{-1}\mathbf{B}_U^\top + \mathbf{E}$, $\mathbf{d} = \mathbf{B}_L\mathbf{A}^{-1}\mathbf{f} - \mathbf{g}$. The precision of $\mathbf{p}^{(\kappa)}$ computed from (16) is driven adaptively with respect to the precision achieved in the outer iterations of (15). One iteration in (15) requires the following points:

- Assembling $\mathbf{A}$, $\mathbf{B}_U$, $\mathbf{B}_L$, $\mathbf{E}$, $\mathbf{f}$, and $\mathbf{g}$ by a vectorized code.
- Computing LU-factorization of $\mathbf{A}$ with the complete pivoting that results in the lower, upper triangular matrices $\mathbf{L}$, $\mathbf{U}$, respectively, and in two permutation matrices $\mathbf{P}$, $\mathbf{Q}$ such that $\mathbf{PAQ} = \mathbf{LU}$.
- Assembling $\mathbf{d} = \mathbf{B}_L(\mathbf{Q}(\mathbf{U}^{-1}(\mathbf{L}^{-1}(\mathbf{Pf})))) - \mathbf{g}$ and activating the procedure for matrix-vector products $\mathbf{Sq}$ based on $\mathbf{Sq} = \mathbf{B}_L(\mathbf{Q}(\mathbf{U}^{-1}(\mathbf{L}^{-1}(\mathbf{P}(\mathbf{B}_U^{\top}\mathbf{q}))))) + \mathbf{Eq}$.
- Solving (16) using the *BiCGSTAB* method (Elman *et al.*, 2014) which applies the matrix-vector procedure from the previous point. The *BiCGSTAB* iterations start from the initial approximation $\mathbf{p}^{(\kappa-1)}$ and end if the adaptive inner terminating tolerance is achieved:

$$tol_{BiCGSTAB}^{(\kappa)} = \left\{ r_{tol} \times err^{\,(\kappa-1)}, c_{fact} \times tol_{BiCGSTAB}^{(\kappa-1)} \right\},$$

  where $0 < r_{tol} < 1$, $0 < c_{fact} < 1$, and $err^{\,(\kappa-1)}$ is the outer terminating criterion ($err^{\,(0)} = 1$ and $tol_{BiCGSTAB}^{(0)} = r_{tol}/c_{fact}$). The mass matrix is the preconditioner in the *BiCGSTAB* method (Elman *et al.*, 2014).
- Computing $\mathbf{u}^{(\kappa)} = \mathbf{Q}(\mathbf{U}^{-1}(\mathbf{L}^{-1}(\mathbf{P}(\mathbf{f} - \mathbf{B}_U^{\top}\mathbf{p}^{(\kappa)}))))$.
- Stop if the outer terminating criterion is sufficiently small:

$$err^{\,(\kappa)} := \frac{\|(\mathbf{u}^{(\kappa)}, \mathbf{p}^{(\kappa)}) - (\mathbf{u}^{(\kappa-1)}, \mathbf{p}^{(\kappa-1)})\|}{\|(\mathbf{u}^{(\kappa)}, \mathbf{p}^{(\kappa)})\| + 1} \leqslant \varepsilon,$$

  otherwise perform the next outer iteration with $\kappa := \kappa + 1$.

## 7. Numerical Experiments

We consider three test problems defined on the unit cube with known analytic solutions. They are 3D extensions of the well-known test problems of computational fluid dynamics in 2D. First, we examine time requirements of the assembly operations for different discretizations. Then we investigate convergence properties of the finite element approximation. All computations are done in *MATLAB R2021a* on supercomputer *Karolina* (IT4Innovations, 2023). Meshes are generated by free available *iso2mesh* generator (Fang, 2018). The structured partition $\mathcal{T}_h$ of $\Omega = [0, 1]^3$ is defined so that $\Omega$ is first divided onto $n^3$ cubes of the same size and, then, each of them is divided onto five tetrahedra. In this case we have the mesh norm $h = \sqrt{3}/n$. The mesh norms for unstructured partitions are computed as maxima over all tetrahedra. See Fig. 1 for examples of structured and unstructured meshes. Recall that $n_p$ is the number of the finite element nodes and $n_t$ is the number of the tetrahedra. The iterative scheme (15) uses the following parameters: $\varepsilon = 10^{-5}$ and $r_{tol} = c_{fact} = 0.9$.

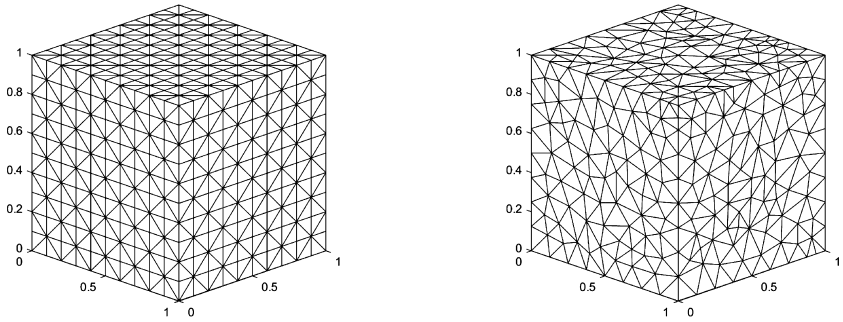


Fig. 1. Structured (left) and unstructured (right) mesh for test problems.

### 7.1. *Test Problem #1*

We consider the functions $\boldsymbol{u} = (u_1, u_2, u_3)$ and $p$ in $\Omega = [0, 1]^3$ as follows:

$$
\begin{aligned}
u_1(x, y, z) &= 4\big(1 - \cos(2\pi x)\big)\sin(2\pi y)z(1 - z),\\
u_2(x, y, z) &= 4\sin(2\pi x)\big(\cos(2\pi y) - 1\big)z(1 - z),\\
u_3(x, y, z) &= 0,\\
p(x, y, z) &= 2\pi\big(-\cos(2\pi x) + 2\cos(2\pi y) - \cos(2\pi z)\big).
\end{aligned}
$$

These functions solve the problem (1) with $\boldsymbol{f} := -\nu\Delta\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \alpha\boldsymbol{u} + \nabla p$. The problem corresponds to a flow formed by a vortex rotating around the central axis of the cube so that the velocity field is parallel to the $xy$ plane with the maximal magnitude in the central plan $z = 0.5$.

### 7.2. *Test Problem #2*

We consider the functions $\boldsymbol{u} = (u_1, u_2, u_3)$ and $p$ in $\Omega = [0, 1]^3$ as follows:

$$
\begin{aligned}
u_1(x, y, z) &= \big(1 - \cos(2\pi x)\big)\sin(2\pi y)\sin(2\pi z),\\
u_2(x, y, z) &= 2\sin(2\pi x)\big(\cos(2\pi y) - 1\big)\sin(2\pi z),\\
u_3(x, y, z) &= \sin(2\pi x)\sin(2\pi y)\big(1 - \cos(2\pi z)\big),\\
p(x, y, z) &= 2\pi\big(-\cos(2\pi x) + 2\cos(2\pi y) - \cos(2\pi z)\big).
\end{aligned}
$$

These functions solve the problem (1) with $\boldsymbol{f}$ defined as in the problem #1. The flow is a vortex of a fusiform character around the central axis of the cube, i.e. rising and falling spirals above the $xy$ plane.

### 7.3. *Test Problem #3*

We consider the functions $\boldsymbol{u} = (u_1, u_2, u_3)$ and $p$ in $\Omega = [0, 1]^3$ as follows:

$$
\begin{aligned}
u_1(x, y, z) &= x^2(1-x)^2 2y(1-y)(1-2y)2z(1-z)(1-2z),\\
u_2(x, y, z) &= 2y^2(1-y)^2 2x(1-x)(2x-1)2z(1-z)(1-2z),\\
u_3(x, y, z) &= z^2(1-z)^2 2y(1-y)(1-2y)2x(1-x)(1-2x),\\
p(x, y, z) &= x(1-x)y(1-y)(1-z).
\end{aligned}
$$

These functions solve the problem (1) with $\boldsymbol{f}$ defined as in the problem #1. The flow character is a vortex of the form as in the problem #2 but now described by the polynomial functions instead of the trigonometric ones.

### 7.4. *Example 1: Time Demands*

In Tables 1–2 we denote by *A_time(V)*, *A_time(L)* the CPU time for assembly operations when the vectorized code or the loop over tetrahedra is used, respectively. *S_time* is the CPU time for solving the respective linear system.

In Table 1 we report assembly time of the vectorized operations and the loop over tetrahedra on meshes of the cube $\Omega = [0, 1]^3$ computed by:

$$\textit{ratio\_1} = \textit{A\_time(L)/A\_time(V)}.$$

One can see that the loop over tetrahedra is extremely unefficient for large scale problems. These tests are computed for the Oseen linearization with $\nu = 0.5$ and $\alpha = 1$.

Table 1
Loop over tetrahedra versus vectorized code.

| $n_p$ | 729 | 2197 | 4913 | 9261 | 15625 | 29791 | 50653 | 91125 |
|---|---|---|---|---|---|---|---|---|
| $n_t$ | 2560 | 8640 | 20480 | 40000 | 69120 | 135000 | 233280 | 425920 |
| *A_time(V)* | 8.1e−03 | 2.8e−02 | 5.9e−02 | 1.3e−01 | 2.4e−01 | 6.4e−01 | 1.2e+00 | 2.1e+00 |
| *A_time(L)* | 2.4e−01 | 1.9e+00 | 2.0e+01 | 1.4e+02 | 4.7e+02 | 1.9e+03 | 6.0e+03 | 2.0e+04 |
| *ratio_1* | 29.6 | 66.8 | 333.6 | 1079.6 | 1952.3 | 3020.9 | 5079.3 | 9299.0 |

Table 2
Loop over tetrahedra: $\nu = 0.5$.

| | Oseen linearization | | Newton linearization | |
|---|---|---|---|---|
| $n_p$ | 4913 | 15625 | 4913 | 15625 |
| $n_t$ | 20480 | 69120 | 20480 | 69120 |
| *A_time(L)* | 2.17e+01 | 3.22e+02 | 2.56e+01 | 5.05e+02 |
| *S_time* | 1.57e−01 | 1.26e+00 | 4.28e−01 | 3.41e+00 |
| *ratio_2(L)* | 0.993 | 0.996 | 0.984 | 0.993 |

Table 3
Vectorized code: $\nu = 0.5; 0.05$.

| | Oseen linearization | | | Newton linearization | | |
|---|---|---|---|---|---|---|
| $n_p$ | 103823 | 166375 | 250047 | 103823 | 166375 | 250047 |
| $n_t$ | 486680 | 787320 | 1191640 | 486680 | 787320 | 1191640 |
| *A_time(V)* | 7.60e+00 | 1.52e+01 | 2.50e+01 | 6.77e+00 | 1.46e+01 | 2.18e+01 |
| *S_time* | 3.45e+01 | 7.53e+01 | 1.58e+02 | 9.21e+01 | 2.30e+02 | 5.13e+02 |
| *ratio_2(V)* | 0.180 | 0.168 | 0.137 | 0.069 | 0.060 | 0.041 |
| *A_time(V)* | 7.66e+00 | 1.50e+01 | 2.19e+01 | 7.19e+00 | 1.52e+01 | 2.28e+01 |
| *S_time* | 5.47e+01 | 1.12e+02 | 2.04e+02 | 1.21e+02 | 2.74e+02 | 6.78e+02 |
| *ratio_2(V)* | 0.123 | 0.118 | 0.097 | 0.056 | 0.052 | 0.032 |

Tables 2–3 are obtained by solving the problem #1. We report the ratio of the assembly operations per iteration of the scheme (15) computed by:

$$\mathit{ratio\_2(X)} = \mathit{A\_time(X)} \big/ \big(\mathit{A\_time}(X) + \mathit{S\_time}\big), \quad X = L, V.$$

Table 2 is computed by the loop over tetrahedra. It shows that about 99% computational time per iteration takes assembling of matrices. Similar behaviour was observed by Koko (2016) for linear elasticity problems.

Table 3 is computed by the vectorized codes. It is seen that the relative efficiency of the vectorized assembly operations is higher for large scale problems. Comparing *S_time* we see that the first order Oseen linearization is faster than the second order Newton linearization. A heuristic explanation of this fact consists in more complicated structure of the Newton matrices.

### 7.5. *Example 2: Convergence Rate on Structured Meshes*

In this example, we investigate experimentally convergence rates of finite element approximations computed by our vectorized codes on structured meshes. The following optimal convergence result is proved in Boffi *et al.* (2013) for the Stokes problem and the MINI element:

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1} + \|p - p_h\|_{L^2} \leqslant Ch\big(\|\boldsymbol{u}\|_{H^2} + \|p\|_{H^1}\big), \tag{17}$$

where $C > 0$ does not dependent on $h$. Cioncolini and Boffi (2019) and again Cioncolini and Boffi (2022) studied experimentally convergence rates of the Stokes problem in 2D and 3D, respectively. Note that the bound (17) is valid also for the Navier–Stokes problem as it follows from Girault and Raviart (1986) but under more complicated assumptions (pp. 101, Theorem 4.1). The formula (17) indicates that $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ as well as $\|p - p_h\|_{L^2}$ converge linearly. In Tables 4–9, we compute convergence rates for the test problems #1–#3 with $\nu = 0.5, 0.05$ and $\alpha = 0$. From the obtained results we can conclude that the experimental convergence rates are close to the following ones: $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2} = \mathcal{O}(h^2)$, $\|p - p_h\|_{L^2} = \mathcal{O}(h^{3/2})$, and $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1} = \mathcal{O}(h)$ and, in some cases, they are higher.

Table 4
Convergence rates for the test problem #1 with $\nu = 0.5$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 1.749e−2 | | 2.846e−1 | | 3.507e−1 | |
| 5.774e−2‖30 | 9.410e−3 | 2.00 | 1.785e−1 | 1.50 | 2.469e−1 | 1.13 |
| 4.558e−2‖38 | 5.856e−3 | 2.01 | 1.254e−1 | 1.50 | 1.909e−1 | 1.09 |
| 3.608e−2‖46 | 3.990e−3 | 2.01 | 9.422e−2 | 1.44 | 1.559e−1 | 1.06 |
| 3.093e−2‖54 | 2.891e−3 | 2.01 | 7.414e−2 | 1.50 | 1.318e−1 | 1.05 |
| 2.794e−2‖62 | 2.190e−3 | 2.01 | 6.030e−2 | 1.50 | 1.142e−1 | 1.04 |

Table 5
Convergence rates for the test problem #1 with $\nu = 0.05$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 1.583e−2 | | 6.059e−2 | | 3.515e−1 | |
| 5.774e−2‖30 | 8.452e−3 | 2.02 | 3.379e−2 | 1.88 | 2.467e−1 | 1.14 |
| 4.558e−2‖38 | 5.239e−3 | 2.02 | 2.180e−2 | 1.85 | 1.907e−1 | 1.09 |
| 3.608e−2‖46 | 3.560e−3 | 2.02 | 1.538e−2 | 1.83 | 1.557e−1 | 1.06 |
| 3.093e−2‖54 | 2.575e−3 | 2.02 | 1.151e−2 | 1.81 | 1.316e−1 | 1.05 |
| 2.794e−2‖62 | 1.949e−3 | 2.02 | 8.996e−3 | 1.79 | 1.141e−1 | 1.04 |

Table 6
Convergence rates for the test problem #2 with $\nu = 0.5$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 3.998e−2 | | 7.449e−1 | | 8.546e−1 | |
| 5.774e−2‖30 | 2.184e−2 | 1.95 | 4.726e−1 | 1.47 | 5.974e−1 | 1.15 |
| 4.558e−2‖38 | 1.370e−2 | 1.97 | 3.336e−1 | 1.47 | 4.604e−1 | 1.10 |
| 3.608e−2‖46 | 9.383e−3 | 1.98 | 2.514e−1 | 1.48 | 3.751e−1 | 1.07 |
| 3.093e−2‖54 | 6.822e−3 | 1.99 | 1.981e−1 | 1.49 | 3.168e−1 | 1.05 |
| 2.794e−2‖62 | 5.182e−3 | 1.99 | 1.613e−1 | 1.49 | 2.743e−1 | 1.04 |

Table 7
Convergence rates for the test problem #2 with $\nu = 0.05$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 3.505e−2 | | 1.826e−1 | | 8.442e−1 | |
| 5.774e−2‖30 | 1.898e−2 | 1.98 | 1.024e−1 | 1.87 | 5.927e−1 | 1.14 |
| 4.558e−2‖38 | 1.187e−2 | 1.99 | 6.576e−2 | 1.87 | 4.580e−1 | 1.09 |
| 3.608e−2‖46 | 8.117e−3 | 1.99 | 4.556e−2 | 1.92 | 3.738e−1 | 1.06 |
| 3.093e−2‖54 | 5.898e−3 | 1.99 | 3.435e−2 | 1.76 | 3.160e−1 | 1.05 |
| 2.794e−2‖62 | 4.481e−3 | 1.99 | 2.668e−2 | 1.83 | 2.738e−1 | 1.04 |

Especially, the convergence rate of the pressure component shows the superconvergence property. This result was theoretically proved for the pure Stokes problem in 2D by Eichel *et al.* (2011). It was experimentally confirmed in above mentioned papers of Cioncolini and Boffi but, again, for the Stokes problem. Our observation for the Navier–Stokes problem is new.

Table 8
Convergence rates for the test problem #3 with $\nu = 0.5$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 3.648e−5 | | 1.386e−3 | | 1.114e−3 | |
| 5.774e−2‖30 | 1.990e−5 | 1.95 | 8.827e−4 | 1.45 | 7.985e−4 | 1.07 |
| 4.558e−2‖38 | 1.248e−5 | 1.98 | 6.257e−4 | 1.46 | 6.233e−4 | 1.15 |
| 3.608e−2‖46 | 8.539e−6 | 1.99 | 4.733e−4 | 1.46 | 5.115e−4 | 1.03 |
| 3.093e−2‖54 | 6.205e−6 | 1.99 | 3.740e−4 | 1.47 | 4.339e−4 | 1.03 |
| 2.794e−2‖62 | 4.711e−6 | 1.99 | 3.052e−4 | 1.47 | 3.769e−4 | 1.02 |

Table 9
Convergence rates for the test problem #3 with $\nu = 0.05$, structured mesh.

| $h\|n$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 7.873e−2‖22 | 3.649e−5 | | 1.409e−4 | | 1.114e−3 | |
| 5.774e−2‖30 | 1.990e−5 | 1.95 | 8.934e−5 | 1.47 | 7.986e−4 | 1.07 |
| 4.558e−2‖38 | 1.248e−5 | 1.98 | 6.309e−5 | 1.47 | 6.233e−4 | 1.05 |
| 3.608e−2‖46 | 8.539e−6 | 1.99 | 4.773e−5 | 1.46 | 5.115e−4 | 1.03 |
| 3.093e−2‖54 | 6.205e−6 | 1.99 | 3.769e−5 | 1.47 | 4.339e−4 | 1.03 |
| 2.794e−2‖62 | 4.711e−6 | 1.99 | 3.073e−5 | 1.48 | 3.769e−4 | 1.02 |

Table 10
Convergence rates for the test problem #2 with $\nu = 0.5$, unstructured mesh.

| $h$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 9.4515e−2 | 2.275e−2 | | 4.502e−1 | | 3.572e−1 | |
| 7.6129e−2 | 1.435e−2 | 2.13 | 3.044e−1 | 1.81 | 2.706e−1 | 1.28 |
| 6.2010e−2 | 8.915e−3 | 2.32 | 2.297e−1 | 1.37 | 2.060e−1 | 1.33 |
| 4.8458e−2 | 5.542e−3 | 1.93 | 1.677e−1 | 1.28 | 1.595e−1 | 1.04 |
| 3.8869e−2 | 3.481e−3 | 2.11 | 1.210e−1 | 1.48 | 1.242e−1 | 1.16 |
| 3.1102e−2 | 2.177e−3 | 2.11 | 9.098e−2 | 1.28 | 9.756e−2 | 1.08 |
| *G_mean* | – | 2.12 | – | 1.43 | – | 1.17 |

### 7.6. *Example 3: Convergence Rate on Unstructured Meshes*

In Tables 10–11, we present results analogous to Tables 6–7 computed for the test problem #2 but now on unstructured meshes. Since the convergence rates are scattered, we characterize them by the geometrical mean in the rows labelled *G_mean*. Surprisingly, the conference rates are in many cases better than on the structured meshes.

## 8. Conlusions and Comments

In this paper, we present main ideas for vectorized coding of matrices and vectors describing mixed finite element approximation based on the MINI element for the Navier–Stokes system in 3D. It is shown that the vectorized operations are considerably faster than the loop over tetrahedra. This allows to experiment with this problem in the user-friendly Mat-

Table 11
Convergence rates for the test problem #2 with $\nu = 0.05$, unstructured mesh.

| $h$ | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}$ | Rate | $\|p - p_h\|_{L^2}$ | Rate | $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1}$ | Rate |
|---|---|---|---|---|---|---|
| 9.4515e−2 | 2.275e−2 | | 4.502e−1 | | 3.572e−1 | |
| 7.6129e−2 | 1.435e−2 | 2.13 | 3.044e−1 | 1.81 | 2.706e−1 | 1.28 |
| 6.2010e−2 | 8.915e−3 | 2.32 | 2.297e−1 | 1.37 | 2.060e−1 | 1.33 |
| 4.8458e−2 | 5.542e−3 | 1.93 | 1.677e−1 | 1.28 | 1.595e−1 | 1.04 |
| 3.8869e−2 | 3.481e−3 | 2.11 | 1.210e−1 | 1.48 | 1.242e−1 | 1.14 |
| 3.1102e−2 | 2.177e−3 | 2.11 | 9.098e−2 | 1.28 | 9.756e−2 | 1.08 |
| *G_mean* | – | 2.11 | – | 1.43 | – | 1.17 |

lab environment. Note that our codes are freely available (Kučera *et al.*, 2023) and include also 2D case.

The dual implementation of the basic iterative schemes in Section 6 is a starting point for more sophisticated problem with the stick-slip boundary condition, describing hydrophobia effect, e.g. in which the dual formulation is a natural tool (see Haslinger *et al.*, 2021). This scheme works well for small Reynold's numbers.

Finally, we should point out that the results of our experiments are in agreement with the theoretical convergence rates of the finite element approximation. Moreover, it extends observations of other authors on a superconvergence rate of the pressure component. It confirms, among others, correctness of our codes.

## Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140) and the internal VSB-TUO SGS-project.

## References

Arnold, D.N., Brezzi, F., Fortin, M. (1984). A stable finite element for the Stokes equations. *Calcolo*, 21, 337–344.

Arzt, V. (2019). *Finite Element Meshes and Assembling of Stiffness Matrices*. Master's thesis, VŠB-TU Ostrava, Czech Republic (in Czech). https://dspace.vsb.cz/bitstream/handle/10084/137486/ARZ0009_USP_B3968_3901R076_2019.pdf?sequence=1&isAllowed=y [online].

Boffi, D., Brezzi, F., Fortin, M. (2013). *Mixed Finite Element Methods and Applications, Springer Series in Computational Mathematics*. Springer Verlag, Heidelberg, New York, Dordrecht, London.

Brezzi, F., Fortin, M. (1991). *Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics*. Springer Verlag, Berlin.

Cioncolini, A., Boffi, D. (2019). The MINI mixed finite element for the Stokes problem: an experimental investigation. *Computers and Mathematics with Applications*, 77(9), 2432–2446.

Cioncolini, A., Boffi, D. (2022). Superconvergence of the MINI mixed finite element discretization of the Stokes problem: an experimental study in 3D. *Finite Elements in Analysis and Design*, 201, 103706.

Eichel, H., Tobiska, L., Xie, H. (2011). Supercloseness and superconvergence of stabilized low-order finite element discretizations of the Stokes problem. *Mathematics of Computations*, 80, 697–722.

Elman, H.C., Silvester, D.J., Wathen, A.J. (2014). *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, Numerical Mathematics and Scientific Computation*. Oxford University Press, Oxford.

Fang, Q. (2018). Iso2mesh: a 3D surface and volumetric mesh generator for MATLAB/Octave. http://iso2mesh.sourceforge.net [online].

Girault, V., Raviart, P.A. (1986). *Finite Element Methods for Navier-Stokes Equations, Springer Series in Computational Mathematics*. Springer Verlag, Berlin.

Girault, V., Wheeler, M.F. (2008). Discontinuous Galerkin methods. In: Glowinski, R., Neittaanmäki, P. (Eds.), *Partial Differential Equations. Computational Methods in Applied Sciences*, Vol. 16. Springer, Dordrecht, pp. 2–26.

Griebel, M., Neunhoeffer, T., Regler, H. (1998). Algebraic multigrid methods for the solution of the Navier–Stokes equations in complicated geometries. *International Journal for Numerical Methods in Fluids*, 26(3), 281–301.

Haslinger, J., Kučera, R., Sassi, T., Šátek, V. (2021). Dual strategies for solving the Stokes problem with stick-slip boundary conditions in 3D. *Mathematics and Computers in Simulation*, 189, 191–206.

Henriksen, M.O., Holmen, J. (2002). Algebraic splitting for incompressible Navier–Stokes equations. *Journal of Computational Physics*, 175(2), 438–453.

Hoanga, L.T., Martinez, V.R. (2018). Asymptotic expansion for solutions of the Navier–Stokes equations with non-potential body forces. *Journal of Mathematical Analysis and Applications*, 462, 84–113.

IT4Innovations (2023). https://www.it4i.cz/en/infrastructure/karolina [online].

Koko, J. (2016). Fast MATLAB assembly of FEM matrices in 2D and 3D using cell array approach. *International Journal of Modeling, Simulation, and Scientific Computing*, 7(2), 1650010.

Koko, J. (2019). Efficient MATLAB codes for the 2D/3D Stokes equation with the mini-element. *Informatica*, 30(2), 243–268.

Kučera, R., Arzt, V., Koko, J. (2023). Free available vectorized codes. https://homel.vsb.cz/~kuc14/programs/ReferenceAssembling.zip [online].

Loghin, D., Wathen, A.J. (2002). Schur complement preconditioners for the Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 40(3–4), 403–412.

Panasenko, G.P. (1998). Asymptotic expansion of the solution of Navier-Stokes equation in a tube structure. *Comptes Rendus De L Academie Des Sciences Serie Ii Fascicule B-mecanique Physique Astronomie*, 326(12), 867–872.

Pernice, M., Tocci, M.D. (2001). A multigrid-preconditioned Newton–Krylov method for the incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 23, 398–418.

Rahman, T., Valdman, J. (2015). Fast MATLAB assembly of FEM matrices in 2D and 3D: edge elements. *Applied Mathematics and Computation*, 267, 252–263.

Rønquist, E.M. (1996). A domain decomposition solver for the steady Navier-Stokes equations. In: Ilin, A., Scott, R. (Eds.), *Proceedings of the 3rd International Conference on Spectral and High-Order Methods*. HJM, Houston, pp. 469–485.

Viguerie, A., Veneziani, A. (2018). Algebraic splitting methods for the steady incompressible Navier–Stokes equations at moderate Reynolds numbers. *Computer Methods in Applied Mechanics and Engineering*, 330, 271–291.

**R. Kučera** is a full professor at Department of Mathematics and Descriptive Geometry at the VSB–Technical University of Ostrava, Czech Republic. His research interests include numerical linear algebra and numerical optimization with applications to contact problems for linear elastic bodies and flow problems with stick-slip boundary conditions.

**V. Arzt** is a PhD student at Department of Applied Mathematics at the VSB–Technical University of Ostrava, Czech Republic. His research interests include numerical optimization with applications to solving PDEs with the stick-slip boundary conditions and shape optimization problems.

**J. Koko** is an associate professor of applied mathematics in School of Computer Science at Université Clermont Auvergne, France. His research interests include scientific/parallel computing, numerical optimization and applications to nonlinear mechanics.