Numerical Analysis and Scientific Computing
Preprint Seria

ILU preconditioners for non-symmetric saddle point matrices with application to the incompressible Navier-Stokes equations

I.N. Konshin M.A. Olshanskii Yu.V. Vassilevski

Preprint #34

Department of Mathematics
University of Houston

March 2015

ILU PRECONDITIONERS FOR NON-SYMMETRIC SADDLE POINT MATRICES WITH APPLICATION TO THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS[*]

IGOR N. KONSHIN[†], MAXIM A. OLSHANSKII[‡], AND YURI V. VASSILEVSKI[§]

Abstract. Motivated by the numerical solution of the linearized incompressible Navier–Stokes equations, we study threshold incomplete LU factorizations for non-symmetric saddle point matrices. The resulting preconditioners are used to accelerate the convergence of a Krylov subspace method applied to finite element discretizations of fluid dynamics problems in three space dimensions. The paper presents and examines an extension for non-symmetric matrices of the Tismenetsky–Kaporin incomplete factorization. It is shown that in numerically challenging cases of higher Reynolds number flows one benefits from using this two-parameter modification of a standard threshold ILU preconditioner. The performance of the ILU preconditioners is studied numerically for a wide range of flow and discretization parameters, and the efficiency of the approach is shown if threshold parameters are chosen suitably. The practical utility of the method is further demonstrated for the haemodynamic problem of simulating a blood flow in a right coronary artery of a real patient.

Key words. iterative methods, preconditioning, threshold ILU factorization, Navier–Stokes equations, finite element method, haemodynamics

AMS subject classifications. 65F10, 65N22, 65F50.

1. Introduction.

This research is motivated by the numerical solution of the Navier–Stokes equations governing the flow of viscous incompressible Newtonian fluids. For a bounded domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ with boundary $\partial\Omega$, time interval $[0, T]$, and data $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{u}_0$, the goal is to find a velocity field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ and pressure field $p = p(\mathbf{x}, t)$ such that

$$\left\{\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f} &\quad \text{in } \Omega \times (0, T] \\ \operatorname{div}\, \mathbf{u} = 0 &\quad \text{in } \Omega \times [0, T] \\ \mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_0 \times [0, T], \qquad -\nu(\nabla\mathbf{u})\cdot\mathbf{n} + p\mathbf{n} = \mathbf{0} &\quad \text{on } \Gamma_{\rm N} \times [0, T] \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) &\quad \text{in } \Omega \end{aligned}\right. \tag{1.1}$$

where ν is the kinematic viscosity, Δ is the Laplacian, ∇ is the gradient and div is the divergence; $\partial\Omega = \overline{\Gamma}_0 \cup \overline{\Gamma}_{\rm N}$ and $\Gamma_0 \neq \varnothing$. Implicit time discretization and linearization of the Navier–Stokes system (1.1) by Picard fixed-point iteration result in a sequence of (generalized) Oseen problems of the form

$$\left\{\begin{aligned} \alpha\mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{w}\cdot\nabla)\mathbf{u} + \nabla p = \hat{\mathbf{f}} &\quad \text{in } \Omega \\ \operatorname{div}\, \mathbf{u} = \hat{g} &\quad \text{in } \Omega \\ \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_0, \qquad -\nu(\nabla\mathbf{u})\cdot\mathbf{n} + p\mathbf{n} = \mathbf{0} &\quad \text{on } \Gamma_{\rm N} \end{aligned}\right. \tag{1.2}$$

where $\mathbf{w}$ is a known velocity field from a previous iteration or time step and α is proportional to the reciprocal of the time step ($\alpha = 0$ for a steady problem), and the

[*] This work has been supported by Russian Science Foundation through the grant 14-31-00024.

[†] Institute of Numerical Mathematics, Institute of Nuclear Safety, Russian Academy of Sciences, Moscow; igor.konshin@gmail.com

[‡] Department of Mathematics, University of Houston; molshan@math.uh.edu

[§] Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow Institute of Physics and Technology, Moscow; yuri.vassilevski@gmail.com

right-hand side accounts for non-homogenous boundary conditions in the non-linear problem.

Finite element spatial discretization of (1.2) results in large, sparse systems of the form

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.3}$$

where u and p represent the discrete velocity and pressure, respectively, $A \in \mathbb{R}^{n\times n}$ is the discretization of the diffusion, convection, and time-dependent terms, $B^T \in \mathbb{R}^{n\times m}$ is the discrete gradient, B is the (negative) discrete divergence, $C \in \mathbb{R}^{m\times m}$ is a matrix resulting from possible pressure stabilization terms, and f and g contain forcing and boundary terms. If a discretization satisfies the LBB ('inf-sup') stability condition [15], no pressure stabilization is required and so $C = 0$ holds. If the LBB condition is not satisfied, the stabilization matrix $C \neq 0$ is symmetric and positive semidefinite and the actual choice of C depends on the particular finite element pair being used. For a symmetric positive definite A, solving (1.3) is equivalent to finding the saddle point of a Lagrangian, and so the system (1.3) is often referred to as *saddle point* system. In the literature, it is now common to refer to (1.3) as non-symmetric or generalized saddle point system if $A \neq A^T$.

The efficient solution of systems of the form (1.3) necessitates rapidly convergent iterative methods. Thus, in the last decade, considerable work has been done in developing efficient preconditioners for Krylov subspace methods applied to incompressible flow problems; see the comprehensive treatments in [3, 12, 29]. It is typical for the preconditioning to exploit explicitly the block structure of the system (1.3). A popular approach builds upon preconditioners to the sub-matrix A and pressure Schur complement matrix $S = BA^{-1}B^T + C$, see [13, 30, 41] for recent developments. Related to this class of methods are preconditioners based on the augmented Lagrangian reformulation of the saddle point problem [5]. Block preconditioners based on an additive splitting include the Hermitian and skew-Hermitian splitting approach [2] and a dimensional split approach [4]. Constraint block preconditioners for nonsymmetric saddle point matrices are treated in [7]. While the block preconditioners have proven to be effective in many cases, they are not yet completely robust with respect to variations of viscosity parameter, properties of advective velocity field $\mathbf{w}$, grid size and anisotropy ratio. The discussion of geometric and algebraic multigrid preconditioners for the Oseen problem can be found in [39, 42]. For the assessment of block preconditioners in the haemodynamics context we referee to the recent paper [10].

An interesting alternative to block preconditioners for the Oseen problem is the preconditioning based on *elementwise* incomplete factorizations of the 2×2 block matrix from (1.3). Relatively little research is found in the literature on ILU preconditioners for the discrete Oseen system and, more general, for saddle point linear algebraic systems. A review of incomplete Cholesky type preconditioners applicable for symmetric saddle-point systems can be found in the recent report [33] (symmetric system results from (1.2) if one sets $\mathbf{w} = 0$). For non-symmetric saddle-point systems that arise from the finite element discretization of incompressible Navier–Stokes equations the authors of [8, 40] developed ILU preconditioners, where the fill-in is allowed based on the connectivity of nodes rather than actual non-zeros in the matrix. The papers [34, 40] studied several reordering techniques for ILU factorization of (1.3) and found that some of the resulting preconditioners are competitive with the most advanced block preconditioners, while being more straightforward to implement in standard finite element codes.

The present paper focuses on incomplete LU factorizations with thresholds. As far as we are aware, threshold ILU factorizations for non-symmetric saddle point problems resulting from fluid dynamics applications have not been well studied in the literature. The present paper carries out a systematic study of ILU(τ)-type preconditioner performance depending on the threshold parameter τ, viscosity coefficient ν, as well as mesh discretization and time step parameters. The properties of advective velocity field $\mathbf{w}$ often also influence the performance of preconditioners, since the algebraic connectivity of nodes may be strongly influenced by local direction of flow. To assess the performance of ILU preconditioners, we experiment with unidirectional and complex 3D circulating flows including those arising in haemodynamics applications.

The paper also devises estimates for the LU factorization numerical stability for non-symmetric saddle-point matrices. We show that if the (1,1)-block A is a positive definite matrix, then the (exact) LU factorization of the (1.3) exists and its numerical stability is determined by the ellipticity constant of A and a quantity characterizing a ratio of symmetric and skew-symmetric parts of A. The analysis is applied to the discrete linearized Navier–Stokes equations and we discuss possible implications of this analysis for the stability of incomplete LU factorizations.

While in many situations ILU(τ) with optimized τ provides inexpensive (in terms of fill-in) and efficient (in terms of iteration counts) preconditioners for (1.3), for higher Reynolds number flows (small ν) further developments are required. In such cases, we show that a two-parameter variant of the threshold ILU factorization ILU(τ_1,τ_2) may lead to a significant improvement. For symmetric positive definite matrices, this factorization is also known in the literature as *the second order* or *limited-memory* or *Tismenetsky–Kaporin* IC factorization. For both ILU(τ) and ILU(τ_1,τ_2), the choice of optimal τ-s depends on problem parameters. Numerical experiments show that a choice of quasi-optimal parameters is feasible, leading to a preconditioner performance fairly insensitive to the variation of α, grid anisotropy, complexity of $\mathbf{w}$ and depending mildly on ν. Finally, we consider a test case of a flow in a digitally reconstructed right coronary artery of a real patient for a set of parameters describing a physiologically relevant blood flow scenario. The paper reports on the performance of ILU preconditioners for this practically interesting problem.

The remainder of the paper is organized as follows. In section 2 we give necessary details on the discretization method. Section 3 discusses LU factorizations for non-symmetric saddle point systems and its stability. Sufficient conditions on the existence of the LU factorization and an estimate on the entries of the LU factors are given here in terms of the properties of the (1,1)-block A. Further, this analysis is applied to the discretized system (1.2). Here sufficient conditions for positive definiteness of the A-block are derived. These conditions are sufficient for the existence of an LU factorization without pivoting. In section 4, we introduce a two-parameter Tismenetsky–Kaporin variant of the threshold ILU factorization for non-symmetric non-definite problems, which is used further for numerical experiments. In section 5 we consider two benchmark problems: a 3D flow in a cylindrical vessel and a 3D analog of the Beltrami flow proposed in [14]. For the discretization we use P2-P1 inf-sup stable finite elements. For each of the problems we run experiments for a variety of physical and discretization parameters and on a sequence of refined tetrahedral discretizations. Conclusions are made about the performance of preconditioners and the suitable range of threshold parameters. Further we present results for the test case of a flow in a right coronary artery. Section 6 collects conclusions and a few closing remarks.

2. Finite element method. In this paper, we consider an inf-sup stable conforming Finite Element (FE) method. To formulate it, we first need the weak formulation of the Oseen problem. Let $\mathbf{V} := \{\mathbf{v} \in H^1(\Omega)^3 \,:\, \mathbf{v}|_{\Gamma_0} = \mathbf{0}\}$. Given $\mathbf{f} \in \mathbf{V}'$, find $\mathbf{u} \in \mathbf{V}$ and $p \in L^2(\Omega)$ such that

$$\begin{aligned}\mathcal{L}(\mathbf{u}, p; \mathbf{v}, q) \; &= \; (\mathbf{f}, \mathbf{v})_* + (g, q) \qquad \forall\, \mathbf{v} \in \mathbf{V},\; q \in L^2(\Omega)\,,\\ \mathcal{L}(\mathbf{u}, p; \mathbf{v}, q) &:= \alpha(\mathbf{u}, \mathbf{v}) + \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + ((\mathbf{w}\cdot\nabla)\,\mathbf{u}, \mathbf{v}) - (p, \operatorname{div}\mathbf{v}) + (q, \operatorname{div}\mathbf{u})\,,\end{aligned}$$

where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product and $(\cdot,\cdot)_*$ is the duality paring for $\mathbf{V}'\times\mathbf{V}$.

We assume T_h to be a collection of tetrahedra which is a consistent tetrahedrization of Ω satisfying the regularity condition

$$\max_{\tau\in T_h} \operatorname{diam}(\tau)/\rho(\tau) \le C_T, \tag{2.1}$$

where $\rho(\tau)$ is the diameter of a subscribed ball in τ. A constant C_T measures the maximum anisotropy ratio for T_h. Further we denote $h_{\min} = \min_{\tau\in T_h} \operatorname{diam}(\tau)$. Given conforming FE spaces $\mathbb{V}_h \subset \mathbf{V}$ and $\mathbb{Q}_h \subset L^2(\Omega)$, the Galerkin FE discretization of (1.2) is based on the weak formulation: Find $\{\mathbf{u}_h, p_h\} \in \mathbb{V}_h \times \mathbb{Q}_h$ such that

$$\mathcal{L}(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h) = (\mathbf{f}, \mathbf{v}_h)_* + (g, q_h) \qquad \forall\, \mathbf{v}_h \in \mathbb{V}_h,\; q_h \in \mathbb{Q}_h\,. \tag{2.2}$$

In our experiments we shall use P2-P1 Taylor–Hood FE pair, which satisfies the LBB compatibility condition for $\mathbb{V}_h$ and $\mathbb{Q}_h$ [15] and hence ensures well-posedness and full approximation order for the FE linear problem. If one enumerates velocity unknowns first and further pressure unknowns, then the resulting discrete system has the 2×2-block form (1.3).

3. LU factorization and properties of A and S. If the sub-matrices A and C of (1.3) are positive definite and positive semi-definite, respectively, the whole 2×2-block matrix is not sign definite. If $C = 0$, its diagonal has zero entries. In general, LU factorization of such matrices requires pivoting (rows and columns permutations) for stability reasons. However, exploiting the block structure and the properties of blocks A and C, one readily verifies that the LU factorization

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & -U_{22} \end{pmatrix} \tag{3.1}$$

with low (upper) triangle matrices L_{11}, L_{22} (U_{11}, U_{22}) exists without pivoting, once $\det(A) \ne 0$ and there exist LU factorizations for the (1,1)-block

$$A = L_{11}U_{11}$$

and the Schur complement matrix $S := BA^{-1}B^T + C$ is factorized as

$$S = L_{22}U_{22}.$$

To check (3.1), one lets $U_{12} = L_{11}^{-1}B^T$ and $L_{21} = BU_{11}^{-1}$.

An LU factorization of A exists if the matrix is positive definite, however its numerical stability (the relative size of entries in factors L_{11} and U_{11}) may depend on how large is the skew-symmetric part of A comparing to the symmetric part. Indeed, denote $A_{\mathrm{S}} = \frac{1}{2}(A + A^T)$, $A_{\mathrm{N}} = A - A_{\mathrm{S}}$ (we shall use similar notation for the symmetric and skew-symmetric parts of S). Denote by $\|\cdot\|_F$ the Frobenius matrix norm. Theorem 4.2.4 from [16] gives the bound on the size of elements of L and U:

$$\||L_{11}||U_{11}|\|_F \le n\left(\|A_\mathrm{S}\| + \|A_\mathrm{N} A_\mathrm{S}^{-1} A_\mathrm{N}\|\right),$$

where $|C| = \{|c_{ij}|\}$ for a matrix $C = \{c_{ij}\}$. Using $\|A_\mathrm{S}\| \le \|A\|$, the symmetry and negative definiteness of $A_\mathrm{N} A_\mathrm{S}^{-1} A_\mathrm{N}$, one can estimate

$$\begin{aligned}\|A_\mathrm{N} A_\mathrm{S}^{-1} A_\mathrm{N}\| &= -\sup_{x\in\mathbb{R}^n} \frac{\langle A_\mathrm{N} A_\mathrm{S}^{-1} A_\mathrm{N} x, x\rangle}{\|x\|^2} = \sup_{x\in\mathbb{R}^n} \frac{\|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12} x\|^2}{\|A_\mathrm{S}^{-\frac12} x\|^2}\\ &\le \sup_{x\in\mathbb{R}^n} \frac{\|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12} x\|^2 \|A_\mathrm{S}^{\frac12}\|^2}{\|x\|^2} = \|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|^2 \|A_\mathrm{S}^{\frac12}\|^2\\ &= \|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|^2 \|A_\mathrm{S}\| \le \|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|^2 \|A\|.\end{aligned}$$

Hence, we deduce the following stability bound for the LU-factorization of the positive definite matrix A:

$$\frac{\||L_{11}||U_{11}|\|_F}{\|A\|} \le n\left(1 + \|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|^2\right). \tag{3.2}$$

The positive definiteness of A implies that the Schur complement matrix is also positive definite, once B^T has full column rank and $C \ge 0$. This is easy to see from the identity

$$\langle Sq, q\rangle = \langle Bv, q\rangle + \langle Cq, q\rangle = \langle v, B^T q\rangle + \langle Cq, q\rangle = \langle Av, v\rangle + \langle Cq, q\rangle, \tag{3.3}$$

which is true for $q \in \mathbb{R}^m$ and $v := A^{-1}B^T q \in \mathbb{R}^n$. Therefore, if A is positive definite, then S is also positive definite and the factorization $S = L_{22}U_{22}$ enjoys the stability bound similar to (3.2):

$$\frac{\||L_{22}||U_{22}|\|_F}{\|S\|} \le m\left(1 + \|S_\mathrm{S}^{-\frac12} S_\mathrm{N} S_\mathrm{S}^{-\frac12}\|^2\right).$$

Thus, in the case of positive definite (1,1)-block, the quotients $\|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|$ and $\|S_\mathrm{S}^{-\frac12} S_N S_\mathrm{S}^{-\frac12}\|$ are largely responsible for the stability of the LU factorization for (1.3). The following lemma shows that it is sufficient to estimate $\|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\|$.

Lemma 3.1. *Let $A \in \mathbb{R}^{n\times n}$ be positive definite, then it holds*

$$\|S_\mathrm{S}^{-\frac12} S_N S_\mathrm{S}^{-\frac12}\| \le \|A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}\| =: C_A. \tag{3.4}$$

Proof. Let $\widetilde{A}_N = A_\mathrm{S}^{-\frac12} A_\mathrm{N} A_\mathrm{S}^{-\frac12}$. We need the following identities [11]:

$$\begin{aligned}\frac12\left(A^{-1} + A^{-T}\right) &= A_\mathrm{S}^{-\frac12}(I - \widetilde{A}_N^2)^{-1} A_\mathrm{S}^{-\frac12},\\ \frac12\left(A^{-1} - A^{-T}\right) &= A_\mathrm{S}^{-\frac12}(I + \widetilde{A}_N)^{-1}\widetilde{A}_N(I - \widetilde{A}_N)^{-1} A_\mathrm{S}^{-\frac12}.\end{aligned} \tag{3.5}$$

Note that due to the skew-symmetry of $S_\mathrm{S}^{-\frac12} S_N S_\mathrm{S}^{-\frac12}$ it holds $|\lambda| = |\mathrm{Im}(\lambda)|$ for $\lambda \in \mathrm{sp}(S_\mathrm{S}^{-\frac12} S_N S_\mathrm{S}^{-\frac12})$, where we use $\mathrm{sp}(\cdot)$ to denote the spectrum. We apply Bendixson's theorem [36] to estimate

$$\begin{aligned}\|S_{\rm S}^{-\frac{1}{2}}S_N\,S_{\rm S}^{-\frac{1}{2}}\| &= \max\{|\lambda|\,:\,\lambda\in \mathrm{sp}(S_{\rm S}^{-\frac{1}{2}}S_N S_{\rm S}^{-\frac{1}{2}})\}\\ &= \max\{|\mathrm{Im}(\lambda)|\,:\,\lambda\in \mathrm{sp}(S_{\rm S}^{-\frac{1}{2}}S_N S_{\rm S}^{-\frac{1}{2}})\}\\ &\le \sup_{q\in\mathbb{C}^m}\frac{|\langle S_N q,q\rangle|}{\langle S_{\rm S}q,q\rangle}.\end{aligned} \tag{3.6}$$

Employing identities from (3.5), we can write

$$\begin{aligned}S_{\rm S} &= BA_{\rm S}^{-\frac{1}{2}}(I-\widetilde{A}_N^*)^{-1}(I-\widetilde{A}_N)^{-1}A_{\rm S}^{-\frac{1}{2}}B^T+C,\\ S_N &= BA_{\rm S}^{-\frac{1}{2}}(I-\widetilde{A}_N^*)^{-1}\widetilde{A}_N(I-\widetilde{A}_N)^{-1}A_{\rm S}^{-\frac{1}{2}}B^T.\end{aligned}$$

With the help of the substitution $v_q=(I-\widetilde{A}_N)^{-1}A_{\rm S}^{-\frac{1}{2}}B^Tq$ in the right-hand side of (3.6) and recalling that C is non-negative definite, we obtain

$$\|S_{\rm S}^{-\frac{1}{2}}S_N\,S_{\rm S}^{-\frac{1}{2}}\|\le \sup_{q\in\mathbb{C}^m}\frac{\left|\langle \widetilde{A}_N v_q,v_q\rangle\right|}{\langle v_q,v_q\rangle+\langle Cq,q\rangle}\le \sup_{q\in\mathbb{C}^m}\frac{\left|\langle \widetilde{A}_N v_q,v_q\rangle\right|}{\|v_q\|^2}\le\|\widetilde{A}_N\|.$$

□

An estimate on the entries of U_{12} and L_{21} factors in (3.1) would form a complete picture of numerical stability of the factorization. The entries of these off-diagonal blocks can be estimated as follows. Using $\|AB\|_F\le\|A\|\|C\|_F$ we get

$$\|U_{12}\|_F=\|L_{11}^{-1}B^T\|_F\le\|L_{11}^{-1}\|\|B^T\|_F=\|U_{11}A^{-1}\|\|B^T\|_F\le\|U_{11}\|\|A^{-1}\|\|B^T\|_F.$$

With the help of (3.5) and noting $\|(I-\widetilde{A}_N)^{-1}\|\le 1$ for a skew-symmetric $\widetilde{A}_N$, one also estimates

$$\|A^{-1}\|\le\frac{1}{2}\left(\|A^{-1}+A^{-T}\|+\|A^{-1}-A^{-T}\|\right)\le\frac{1}{2}\|A_{\rm S}^{-\frac{1}{2}}\|^2(1+C_A)=\frac{1+C_A}{2c_A},$$

with the matrix A ellipticity constant $c_A=\lambda_{\min}(A_{\rm S})$. Repeating same arguments to estimate $\|L_{21}\|_F$, we arrive at the following bound

$$\frac{\|U_{12}\|_F+\|L_{21}\|_F}{(\|U_{11}\|+\|L_{11}\|)\|B\|_F}\le\frac{m(1+C_A)}{2c_A}.$$

The above analysis indicates that to judge about the stability of the LU factorization for (1.3) one should ensure the positive definiteness of the (1,1) block A and estimate the constant C_A which measures the ratio of skew-symmetry for A and the ellipticity constant c_A. In section 3.1 below, we estimate C_A and c_A for the discrete linearized Navier–Stokes system. In section 4, we argue why these analysis is still of interest if one focuses on incomplete factorization.

3.1. Properties of A and S. To study matrix properties, we invoke the FE formulation from section 2. Let $\{\varphi_i\}_{1\le i\le n}$ and $\{\psi_j\}_{1\le j\le m}$ be bases of $\mathbb{V}_h$ and $\mathbb{Q}_h$, respectively. For arbitrary $v\in\mathbb{R}^n$ and corresponding $\mathbf{v}_h=\sum_{i=1}^n v_i\varphi_i$, it holds:

$$\langle Av,v\rangle=\alpha\|\mathbf{v}_h\|^2+\nu\|\nabla\mathbf{v}_h\|^2+\frac{1}{2}\int_{\Gamma_{\rm N}}(\mathbf{w}\cdot\mathbf{n})|\mathbf{v}_h|^2\,ds+\frac{1}{2}((\mathrm{div}\,\mathbf{w})\mathbf{v}_h,\mathbf{v}_h), \tag{3.7}$$

where $\mathbf{n}$ is an outward normal on $\Gamma_{\rm N}$. We shall also need the velocity mass and stiffness matrices M, K: $M_{ij} = (\varphi_i, \varphi_j)$, $K_{ij} = (\nabla\varphi_i, \nabla\varphi_j)$ and the pressure mass matrix M_p: $(M_p)_{ij} = (\psi_i, \psi_j)$.

While the first two terms on the right-hand side of (3.7) are positive, handling the rest terms requires some care. If $\Gamma_{\rm N}$ is an outflow part of the boundary, i.e. $(\mathbf{w}\cdot\mathbf{n}) > 0$, then the boundary integral is non-negative. However, in practical fluid dynamics simulations, it is not uncommon when $(\mathbf{w}\cdot\mathbf{n}) < 0$ on a *part* of $\Gamma_{\rm N}$, and one likely can find such $\mathbf{v}_h$ that the boundary integral in (3.7) is negative. Hence, we shall estimate this term using a FE trace inequality. We remark that modifications of boundary conditions from (1.1) on $\Gamma_{\rm N}$ are known, which insure the resulting boundary integral to be always non-negative, see, e.g., [6]. Other artificial outflow boundary conditions, which lead to Dirichlet conditions to be prescribed in (1.2) on the entire boundary are also common in fluid dynamics, see, e.g., [28,32], in this case $\Gamma_{\rm N} = \varnothing$.

Next, if one assumes the incompressibility condition (second equation in (1.1)) to hold true for the advection velocity field $\mathbf{w}$, then the fourth term on the right-hand side vanishes. In practice, however, $\mathbf{w}$ is typically a *finite element* velocity field, i.e., $\mathbf{w} \in \mathbb{V}_h$, which satisfies only weak divergence free constraint: $(\operatorname{div}\mathbf{w}, q_h) = 0 \quad \forall\, q \in \mathbb{Q}_h$. For most of stable FE for fluids and, in particular, for P2-P1 elements this weak divergence free equation does *not* imply $\operatorname{div}\mathbf{w} = 0$ pointwise (see, however, [18,27] and references therein for recent attempts to deal with this problem). Another possible way of getting rid of the $(\operatorname{div}\mathbf{w})$-dependent term in (3.7) is to 'skew-symmetrize' the bilinear form by adding the consistent term $\frac{1}{2}((\operatorname{div}\mathbf{w})\mathbf{u}_h, \mathbf{v}_h)$ to the FE formulation [37]. Otherwise the last term on the right-hand side of (3.7) should be controlled. We make the above conclusions more precise in Theorem 3.2 below. The theorem gives estimates on the ellipticity constant c_A and the stability constant C_A from (3.4).

To avoid the repeated use of generic but unspecified constants, in the remainder of the paper the binary relation $x \lesssim y$ means that there is a constant c such that $x \le c\,y$, and c does not depend on the parameters which x and y may depend on, e.g., ν, α, mesh size, and properties of $\mathbf{w}$. Obviously, $x \gtrsim y$ is defined as $y \lesssim x$.

Theorem 3.2. *Assume that $\mathbf{w} \in L^\infty(\Omega)$, problem and discretization parameters satisfy* (3.13). *Then the matrix A is positive definite and it holds*

$$\langle Av, v\rangle \ge \frac{1}{4}\langle(\alpha M + \nu K)v, v\rangle \quad \forall\, v \in \mathbb{R}^n \quad \textit{and} \quad C_A \lesssim 1 + \frac{\|\mathbf{w}\|_{L^\infty(\Omega)}}{\sqrt{\nu\alpha} + \nu + h_{\min}\alpha}, \tag{3.8}$$

where C_A is the constant defined in (3.4), *and hence $c_A \ge \frac{1}{4}\lambda_{\min}(\alpha M + \nu K)$. Furthermore, matrix $S := BA^{-1}B^T + C$ is also positive definite and it holds*

$$\langle Sq, q\rangle \gtrsim \frac{\langle M_p q, q\rangle}{(\nu + \alpha + \|\mathbf{w}\|_{L^\infty(\Gamma_{\rm N})} + \|\operatorname{div}\mathbf{w}\|_{L^\infty(\Omega)})(1 + C_A^2)} \quad \forall\, q \in \mathbb{R}^m.$$

Proof. First, recall the trace inequality

$$\int_{\Gamma_{\rm N}} |\mathbf{v}_h|^2\, ds \le C_0\|\nabla\mathbf{v}_h\|^2 \quad \forall\, \mathbf{v}_h \in \mathbb{V}_h, \tag{3.9}$$

which allows the control of the boundary term in (3.7) by the diffusion term, if ν is sufficiently large. To exploit the zero order term in (3.7) , consider the FE trace and inverse inequalities

$$\int_{\partial\tau} \mathbf{v}_h^2\, ds \le C_{\rm tr} h_\tau^{-1}\|\mathbf{v}_h\|_\tau^2, \quad \|\nabla\mathbf{v}_h\|_\tau \le C_{\rm in} h_\tau^{-1}\|\mathbf{v}_h\|_\tau \quad \forall\, \tau \in T_h,\ \mathbf{v}_h \in \mathbb{V}_h, \tag{3.10}$$

where the constants $C_{\rm tr}$, $C_{\rm in}$ depend only on the polynomial degree k and the shape regularity constant C_T from (2.1). In addition, denote by $C_{\rm f}$ the constant from the Friedrichs inequality:

$$\|\mathbf{v}_h\| \le C_{\rm f}\|\nabla\mathbf{v}_h\| \quad \forall\ \mathbf{v}_h \in \mathbb{V}_h. \tag{3.11}$$

Let $C_{\mathbf{w}} := \|(\mathbf{w}\cdot\mathbf{n})_-\|_{L^\infty(\Gamma_{\rm N})}$. Applying (3.9) and (**??**) in (3.7), we deduce

$$\begin{aligned}\langle Av, v\rangle \ge\ & \alpha\|\mathbf{v}_h\|^2 + \nu\|\nabla\mathbf{v}_h\|^2 - \frac{C_{\mathbf{w}}}{2}\int_{\Gamma_{\rm N}}|\mathbf{v}_h|^2\,ds - \frac{1}{2}\|\operatorname{div}\mathbf{w}\|_{L^\infty(\Omega)}\|\mathbf{v}_h\|^2\\ \ge\ & \alpha\|\mathbf{v}_h\|^2 + \nu\|\nabla\mathbf{v}_h\|^2 - \frac{C_{\mathbf{w}}}{2}\min\{C_0\|\nabla\mathbf{v}_h\|^2, C_{\rm tr}h_{\min}^{-1}\|\mathbf{v}_h\|^2\}\\ & - \frac{1}{2}\|\operatorname{div}\mathbf{w}\|_{L^\infty(\Omega)}\|\mathbf{v}_h\|^2.\end{aligned} \tag{3.12}$$

To ensure the right-hand side is positive, we assume the following conditions on problem parameters and coefficients:

$$\begin{cases} C_{\mathbf{w}}C_{\rm tr}h_{\min}^{-1} \le \dfrac{\alpha}{4} \ \text{ or } \ C_{\mathbf{w}}C_0 \le \dfrac{\nu}{4},\\ \|\operatorname{div}\mathbf{w}\|_{L^\infty(\Omega)} \le \dfrac{1}{4}\max\{\alpha, \nu C_f^{-1}\},\end{cases} \tag{3.13}$$

with constants defined in (3.9) and (3.11). Employing conditions (3.13) in (3.12), we deduce

$$\langle Av, v\rangle \ge \frac{1}{4}\left(\alpha\|\mathbf{v}_h\|^2 + \nu\|\nabla\mathbf{v}_h\|^2\right) = \frac{1}{4}\left(\alpha\langle Mv, v\rangle + \nu\langle Kv, v\rangle\right) \quad \forall\ v \in \mathbb{R}^n. \tag{3.14}$$

Further, we estimate

$$\begin{aligned} C_A := \|A_{\rm S}^{-\frac12}A_{\rm N}\,A_{\rm S}^{-\frac12}\| &= \max\{|\lambda|\ :\ \lambda \in {\rm sp}(A_{\rm S}^{-\frac12}A_{\rm N}A_{\rm S}^{-\frac12})\}\\ &= \max\{|\lambda|\ :\ \lambda \in {\rm sp}(A_{\rm S}^{-1}A_{\rm N})\}\\ &\le \|A_{\rm S}^{-1}A_{\rm N}\|_*,\end{aligned} \tag{3.15}$$

and for $\|\cdot\|_*$ we choose a matrix norm induced by the vector norm $\langle(\alpha M + \nu K)\cdot,\cdot\rangle^{\frac12}$. For a given $v \in \mathbb{R}^n$ and $u = A_{\rm S}^{-1}A_{\rm N}\,v$ consider their finite element counterparts $\mathbf{v}_h, \mathbf{u}_h \in \mathbb{V}_h$. Then $A_{\rm S}u = A_{\rm N}\,v$ can be written in a finite element form as

$$\begin{aligned}\nu(\nabla\mathbf{u}_h, \nabla\boldsymbol{\varphi}_h) + \alpha(\mathbf{u}_h, \boldsymbol{\varphi}_h) + \frac{1}{2}\int_{\Gamma_{\rm N}}(\mathbf{w}\cdot\mathbf{n})\mathbf{u}_h\cdot\boldsymbol{\varphi}_h\,ds + \frac{1}{2}((\operatorname{div}\mathbf{w})\mathbf{u}_h, \boldsymbol{\varphi}_h)\\ = \frac{1}{2}[(\mathbf{w}\cdot\nabla\mathbf{v}_h, \boldsymbol{\varphi}_h) - (\mathbf{w}\cdot\nabla\boldsymbol{\varphi}_h, \mathbf{v}_h)] \quad \forall\boldsymbol{\varphi}_h \in \mathbb{V}_h.\end{aligned} \tag{3.16}$$

We set $\boldsymbol{\varphi}_h = \mathbf{u}_h$. For the left-hand side of (3.16) the lower bound (3.14) holds. To estimate the right-hand side, we apply the Cauchy–Schwarz inequality:

$$[(\mathbf{w}\cdot\nabla\mathbf{v}_h, \boldsymbol{\varphi}_h) - (\mathbf{w}\cdot\nabla\boldsymbol{\varphi}_h, \mathbf{v}_h)] \le \|\mathbf{w}\|_{L^\infty(\Omega)}(\|\nabla\mathbf{v}_h\|\|\mathbf{u}_h\| + \|\nabla\mathbf{u}_h\|\|\mathbf{v}_h\|) \tag{3.17}$$

and estimate terms on the right-hand side by employing Young's, Friedrichs, and finite element inverse inequalities:

$$
\begin{aligned}
\|\mathbf{w}\|_{L^\infty(\Omega)}\|\nabla\mathbf{v}_h\|\|\mathbf{u}_h\| &\le \frac{1}{16}(\nu\|\nabla\mathbf{u}_h\|^2+\alpha\|\mathbf{u}_h\|^2)\\
&\qquad + 4\|\mathbf{w}\|^2_{L^\infty(\Omega)}\min\left\{\frac{1}{\alpha\nu},\frac{C_{\mathrm{f}}^2}{\nu^2},\frac{C_{\mathrm{in}}^2}{\alpha^2 h_{\min}^2}\right\}(\nu\|\nabla\mathbf{v}_h\|^2+\alpha\|\mathbf{v}_h\|^2),\\
\|\mathbf{w}\|_{L^\infty(\Omega)}\|\nabla\mathbf{u}_h\|\|\mathbf{v}_h\| &\le \frac{1}{16}(\nu\|\nabla\mathbf{u}_h\|^2+\alpha\|\mathbf{u}_h\|^2)\\
&\qquad + 4\|\mathbf{w}\|^2_{L^\infty(\Omega)}\min\left\{\frac{1}{\alpha\nu},\frac{C_{\mathrm{f}}^2}{\alpha^2},\frac{C_{\mathrm{f}}^2}{\nu^2}\right\}(\nu\|\nabla\mathbf{v}_h\|^2+\alpha\|\mathbf{v}_h\|^2).
\end{aligned}
\tag{3.18}
$$

From (3.14)–(3.18) we derive using $\min\{a_1,a_2,a_3\}\le 3(a_1^{-1}+a_2^{-1}+a_3^{-1})^{-1}$, the estimate

$$
\nu\|\nabla\mathbf{u}_h\|^2+\alpha\|\mathbf{u}_h\|^2 \lesssim \left(1+\frac{\|\mathbf{w}\|^2_{L^\infty(\Omega)}}{\nu\alpha+\nu^2+h_{\min}^2\alpha^2}\right)(\nu\|\nabla\mathbf{v}_h\|^2+\alpha\|\mathbf{v}_h\|^2).
$$

Therefore, we proved

$$
C_A := \|A_{\mathrm{S}}^{-\frac12}A_{\mathrm{N}}A_{\mathrm{S}}^{-\frac12}\| \le \|A_{\mathrm{S}}^{-1}A_{\mathrm{N}}\|_* \lesssim \left(1+\frac{\|\mathbf{w}\|_{L^\infty(\Omega)}}{\sqrt{\nu\alpha}+\nu+h_{\min}\alpha}\right). \tag{3.19}
$$

Denote $\tilde{c}_{\mathbf{w}} := \|\mathbf{w}\|_{L^\infty(\Gamma_{\mathrm{N}})}$, $\hat{c}_{\mathbf{w}} = \|\operatorname{div}\mathbf{w}\|_{L^\infty(\Omega)}$ To show the ellipticity estimate for Schur complement matrix, we note that (3.7), (3.9), (3.11) and the LBB stability of the finite element spaces yield the following relations,

$$
\begin{aligned}
\langle BA_{\mathrm{S}}^{-1}B^Tq,q\rangle &= \sup_{v\in\mathbb{R}^n}\frac{\langle Bv,q\rangle^2}{\langle A_{\mathrm{S}}v,v\rangle}\\
&\ge \sup_{\mathbf{v}_h\in\mathbb{V}_h}\frac{(\operatorname{div}\mathbf{v}_h,q_h)^2}{\nu\|\nabla\mathbf{v}_h\|^2+\alpha\|\mathbf{v}_h\|^2+C_0\tilde{c}_{\mathbf{w}}\|\nabla\mathbf{v}_h\|^2+\hat{c}_{\mathbf{w}}\|\mathbf{v}_h\|^2}\\
&\gtrsim \sup_{\mathbf{v}_h\in\mathbb{V}_h}\frac{(\operatorname{div}\mathbf{v}_h,q_h)^2}{(\nu+\alpha+\tilde{c}_{\mathbf{w}}+\hat{c}_{\mathbf{w}})\|\nabla\mathbf{v}_h\|^2} \gtrsim \frac{\|q_h\|^2}{\nu+\alpha+\tilde{c}_{\mathbf{w}}+\hat{c}_{\mathbf{w}}}\\
&= \frac{\langle M_pq,q\rangle}{\nu+\alpha+\tilde{c}_{\mathbf{w}}+\hat{c}_{\mathbf{w}}}.
\end{aligned}
\tag{3.20}
$$

With the help of the first identity from (3.5) and (3.20) we obtain

$$
\begin{aligned}
\langle Sq,q\rangle = \langle A^{-1}B^Tq,B^Tq\rangle &= \langle (I-(A_{\mathrm{S}}^{-\frac12}A_{\mathrm{N}}\,A_{\mathrm{S}}^{-\frac12})^2)^{-1}A_{\mathrm{S}}^{-\frac12}B^Tq,A_{\mathrm{S}}^{-\frac12}B^Tq\rangle\\
&\ge \frac{\langle A_{\mathrm{S}}^{-\frac12}B^Tq,A_{\mathrm{S}}^{-\frac12}B^Tq\rangle}{1+\|(A_{\mathrm{S}}^{-\frac12}A_{\mathrm{N}}\,A_{\mathrm{S}}^{-\frac12})^2\|} = \frac{\langle BA_{\mathrm{S}}^{-1}B^Tq,q\rangle}{1+\|(A_{\mathrm{S}}^{-\frac12}A_{\mathrm{N}}\,A_{\mathrm{S}}^{-\frac12})^2\|}\\
&\gtrsim \frac{1}{(\nu+\alpha+\tilde{c}_{\mathbf{w}}+\hat{c}_{\mathbf{w}})(1+\|(A_{\mathrm{S}}^{-\frac12}A_{\mathrm{N}}\,A_{\mathrm{S}}^{-\frac12})\|^2)}\langle M_pq,q\rangle.
\end{aligned}
\tag{3.21}
$$

Now we combine (3.21) and (3.19) to show the desired ellipticity estimate for S. □

We are in position to discuss conditions (3.13), which guarantee the matrices A and S to be positive definite and so the saddle-point matrix admits LU factorization without pivoting. The *first condition* in (3.13) is effective only if $\Gamma_{\mathrm{N}}\neq\varnothing$. Also if the entire Γ_{N} is outflow boundary then $C_{\mathbf{w}}=0$ and the condition is trivially satisfied.

Otherwise, either the Reynolds number should be sufficiently small (creeping flows) or a Courant type condition $(\Delta t) \le c\,h_{\min}$ should hold with a problem dependent constant c (we recall that $\alpha \approx (\Delta t)^{-1}$). From the first look, the *second condition* in (3.13) is not restrictive. For example, for P2-P1 Taylor–Hood elements and a second order time discretization, the FE velocity gradient converges quadratically to the one of true solution, and hence one may expect that $\|\mathrm{div}\,\mathbf{w}\|_{L^\infty(\Omega)} \le \widetilde{C}(h^2 + (\Delta t)^2)$. This would make the left-hand side of the second condition small. On the other hand, the constant $\widetilde{C}$ is data dependent, and for ν small enough the constant can be large. Fortunately, for any fixed unsteady problem one can choose such small Δt that the second condition holds due to $\alpha \sim (\Delta t)^{-1}$.

4. A two-parameter threshold ILU factorization. In this section we proceed with incomplete LU factorizations of (1.3). Few remarks are in order.

Any threshold incomplete factorization can be written in the form $A = LU - E$, with an error matrix E. How small is the matrix E is ruled by a threshold parameter $\tau > 0$. The error matrix E largely defines the quality of preconditioning, see, for example, [21] for estimates on GMRES method convergence written in terms of $\|E\|$ and subject to a proper pre-scaling of A and the diagonalizability assumption. Furthermore, if A is positive definite, then there exists such a small τ that LU is also positive definite and so estimates from [16] can be applied to assess the numerical stability of the incomplete factorization. For $c_A = \lambda_{\min}(A_{\mathrm{S}})$, a sufficient condition is $\tau < c_A n^{-1}$. Although in practice this estimate is often too pessimistic, for realistic τ and non-symmetric matrices, non-positive or close to zero pivots may encounter, and breakdown of an algorithm may happen. A number of remedies have been proposed in the literature to deal with the problem of breakdown. A concise review of these techniques and further references can be found in [1]. Although most of the techniques were developed for the SPD case, some of them can be applied to non-symmetric matrices. These are pivot modification, diagonal shifting, matrix scaling, unknowns reordering, the Ajiz–Jennings modification. Among those, we found the matrix two-side scaling to be the most important in our applications. We shall review this technique later in this section. Now let us look at the situation with ILU factorization for saddle point matrices with positive definite (1,1)-block.

It was observed in [33, 43] for symmetric saddle-point systems that the block factorization as in (3.1) can be used to construct an incomplete factorization. One way to do this is first to compute an ILU factorization for the (1,1)-block, $A \approx \widetilde{L}_{11}\widetilde{U}_{11}$, set $\widetilde{U}_{12} = \widetilde{L}_{11}^{-1}B^T$ and $\widetilde{L}_{21} = B\widetilde{U}_{11}^{-1}$, and define $\widetilde{L}_{22}$ and $\widetilde{U}_{22}$ as incomplete factors for the inexact Schur complement:

$$B(\widetilde{L}_{11}\widetilde{U}_{11})^{-1}B^T + C \approx \widetilde{L}_{22}\widetilde{U}_{22}.$$

As we noted before, $A > 0$ implies $\widetilde{L}_{11}\widetilde{U}_{11} > 0$, at least for sufficiently small τ, and so inexact Schur complement is also positive definite. In the present paper, we apply a global incomplete factorization of the matrix instead of the above block-wise factorization. We also avoid pivoting, i.e. we preserve the ordering when velocity unknowns are numbered before pressure unknowns, and we still observe stable factorizations.

Theorem 3.2 shows that for certain flow regimes the stability constant C_A from (3.8) may become large and the ellipticity constant c_A approaches zero, which means the non-symmetric part of the matrix dominates over the symmetric one. Even for advanced threshold ILU factorizations this drives the threshold parameter τ to be smaller and hence increases the fill-in. Results of the next section demonstrate that exactly this behaviour of the algorithm is observed in numerical experiments. To ameliorate the performance of the preconditioning in such extreme situations, we consider the two-parameter Tismenetsky–Kaporin variant of the threshold ILU factorization. The factorization was introduced and first studied in [20,38] for symmetric positive definite case. Below we consider an extension of the Tismenetsky–Kaporin factorization for the case of non-symmetric and saddle-point matrices and give further motivation for it.

Given a matrix $A \in \mathbb{R}^{n\times n}$, consider the factorization of the form

$$A = LU + LR_u + R_\ell U - E, \tag{4.1}$$

where R_u and R_ℓ are strictly upper and lower triangular matrices, while U and L are upper and lower triangular matrices, respectively. Given two small parameters τ_1 and τ_2, we shall assume that the entries absolute values of R_ℓ and R_u do not exceed τ_1, and E is an error matrix with entries of order $O(\tau_2)$. We shall call (4.1) the ILU(τ_1, τ_2) factorization of A. Of course, a generic ILU(τ) can be viewed as (4.1) with $R_u = R_\ell = 0$ and $\tau_2 = \tau$. The important improvement the two-parameter ILU factorization gives over a generic ILU(τ) is that the fill-in of L and U is ruled by the first threshold parameter τ_1, while the quality of the resulting preconditioner is mainly defined by τ_2, once $\tau_1^2 \lesssim \tau_2$ holds. Roughly speaking, taking $\tau_2 = \tau_1^2 := \tau^2$ one expects the fill-in of ILU(τ_1, τ_2) to be similar to that of ILU(τ), while the convergence of preconditioned Krylov subspace method is better and asymptotically (for $\tau \to 0$) can be comparable to the one with ILU(τ^2) preconditioner. This statement is made more precise in [20] for symmetric positive definite matrices, where estimates on the eigenvalues and K-condition number of $L^{-1}AU^{-1}$ were derived with $L^T = U$ and $R_\ell = R_u^T$. However, not much analysis of the decomposition (4.1) is known for a general non-symmetric case. We note that the estimate (2.5) from [17] applied to the matrix $(L + R_\ell)(U + R_u) = A + R_\ell R_u + E$ yields the low bound for the pivots of the (4.1) factorization

$$|L_{ii}U_{ii}| \geq \min_{v\in\mathbb{R}^n} \frac{\langle (A + R_\ell R_u + E)v, v\rangle}{\|v\|^2} \geq c_A - \|R_\ell R_u\| - \|E\|,$$

with the ellipticity constant c_A and the norms $\|R_\ell R_u\|$, $\|E\|$ proportional to τ_1^2 and τ_2, respectively. Thus the stability of system solution with matrices L and U is ruled by the values the second parameter and the *square* of the first parameter, while the fill-in is defined by τ_1 rather than τ_1^2. Using ILU(τ_1, τ_2) becomes important for the efficiency of the ILU preconditioning, when the problem setup is such that the estimates from Theorem 3.2 predicts that the stability constant C_A is large and c_A is small.

Similar to the situation with ILU(τ) factorization, an ILU(τ_1, τ_2) factorization for the saddle-point matrix $\mathcal{A}$ can be built based on two-parameter ILU factorizations (without pivoting) for the (1,1) block

$$A = L_1U_1 + L_1R_{u1} + R_{\ell 1}U_1 - E_1 \tag{4.2}$$

and the inexact Schur complement matrix $\widetilde{S} = C + B[(L_1 + R_{l1})(U_1 + R_{u1})]^{-1}B^T$

$$\widetilde{S} = L_2U_2 + L_2R_{u2} + R_{\ell 2}U_2 - E_2. \tag{4.3}$$

For a matrix $C \in \mathbb{R}^{n\times m}$ and real $\tau \geq 0$ denote $\{C\}^\tau \in \mathbb{R}^{n\times m}$ with entries $\{C\}^\tau_{ij} = C_{ij}$, if $|C_{ij}| \geq \tau$, and $\{C\}^\tau_{ij} = 0$ otherwise; let $[C]^\tau = C - \{C\}^\tau$. Given (4.2) and (4.3) one may check the following factorization for the saddle-point matrix $\mathcal{A}$:

$$\mathcal{A} = \mathcal{L}\mathcal{U} + \mathcal{L}\mathcal{R}_u + \mathcal{R}_\ell\mathcal{U} - \mathcal{E} \tag{4.4}$$

with sparse block factors $\mathcal{L}$ and $\mathcal{U}$:

$$\mathcal{L} = \begin{pmatrix} L_1 & 0 \\ \{B(U_1 + R_{u1})^{-1}\}^{\tau_1} & L_2 \end{pmatrix}, \qquad \mathcal{U} = \begin{pmatrix} U_1 & \{(L_1 + R_{l1})^{-1}B^T\}^{\tau_1} \\ 0 & -U_2 \end{pmatrix},$$

strictly upper triangle matrices $\mathcal{R}_\ell^T$ and $\mathcal{R}_u$:

$$\mathcal{R}_\ell = \begin{pmatrix} R_{\ell 1} & 0 \\ \left[B(U_1 + R_{u1})^{-1}\right]^{\tau_1} & R_{\ell 2} \end{pmatrix}, \quad \mathcal{R}_u = \begin{pmatrix} R_{u1} & [(L_1 + R_{l1})^{-1}B^T]^{\tau_1} \\ 0 & -R_{u2} \end{pmatrix},$$

and the error matrix

$$\mathcal{E} = \begin{pmatrix} E_1 & -R_{\ell 1}[(L_1 + R_{l1})^{-1}B^T]^{\tau_1} \\ -[B(U_1 + R_{u1})^{-1}]^{\tau_1} R_{u1} & -E_2 - [B(U_1 + R_{u1})^{-1}]^{\tau_1}[(L_1 + R_{l1})^{-1}B^T]^{\tau_1} \end{pmatrix}.$$

If (4.2) and (4.3) are ILU(τ_1, τ_2) factorizations, then the formulae above show the existence of a Tismenetsky–Kaporin type incomplete factorization of (1.3), with the error matrix having elements of order $O(\tau_2 + \tau_1^2)$. In practice, we do not exploit the block form and neither matrix $\widetilde{S}$, nor factorization (4.3) are generated explicitly.

4.1. The algorithm. In what follows, the algorithm makes no specific use of the block structure of the matrix $\mathcal{A}$, but can be formally applied to a generic non-symmetric $A \in \mathbb{R}^{n \times n}$ (for a general matrix it can fail, of course). Thus, for the sake of notation, we denote by A below some given non-symmetric square matrix, rather than the (1,1)-block of $\mathcal{A}$; and $n = \dim(A)$.

4.1.1. Two-side scaling of A. The derivation of the ILU(τ_1,τ_2) preconditioner in SPD case assumes such a scaling of the matrix and unknowns that all diagonal elements are equal to 1, see [20]. Clearly, in a non-symmetric case such scaling is not always possible. However, *for the performance of the method, we found it very important to re-scale a given matrix.* Thus, we look for a scaling vectors $\ell, r \in \mathbb{R}^n$ such that the matrix $A' = \mathrm{diag}(\ell) A \, \mathrm{diag}(r)$ has nearly balanced Euclidean norms of rows and columns. To accomplish this task, we apply the Sinkhorn algorithm [35] to the nonnegative matrix $F = [a_{kj}^2]_{kj=1}^n$. The Sinkhorn method is an iterative algorithm recalled below. One iteration of the algorithm reads:

$$\begin{aligned} \mathrm{diag}(r^{(k+1)}) &= \mathrm{diag}(F^T \ell^{(k)})^{-1}, \\ \mathrm{diag}(\ell^{(k+1)}) &= \mathrm{diag}(F r^{(k+1)})^{-1}. \end{aligned}$$

We use the starting vector $\ell^{(0)}$ of all ones. All experiments in the next sections perform 5 iterations to find the scaling vectors, before any incomplete factorization was computed. The importance of a proper two-side scaling for a quality of ILU factorizations for non-symmetric matrices is discussed in [21], see also [9, 22, 25, 26].

If an incomplete factorization is computed for the scaled matrix A' so that $L'U' \approx A'$, the triangular factors for the original matrix have to be re-scaled:

$$LU \approx A, \quad L = (\mathrm{diag}(\ell))^{-1} L', \quad U = U' (\mathrm{diag}(r))^{-1}.$$

In what follows, we will refer to matrices $\mathrm{diag}(\ell)$ and $\mathrm{diag}(r)$ as the left and right scaling matrices D_L and D_R, respectively.

4.1.2. Row-wise ILU(τ_1,τ_2) factorization. In general, a two-parameter threshold ILU factorization algorithm we are using is similar to that of RIC2 from [20]. It was suggested and implemented by Sergei Goreinov in the open source software [23, 24]. The main differences with RIC2 are the *row oriented* data storage of involved matrices and extention of the method to *non-symmetric* matrices.

Assume that the input matrix $A \in \mathbb{R}^{n\times n}$ to be factorized is given in the compressed sparse row (CSR) format. Dropping for a moment the error matrix, consider the $(i+1)$-th step of the row-wise ILU(τ_1,τ_2) algorithm in the block-matrix form:

$$\begin{aligned}\begin{bmatrix} A^i & a^i & \widetilde{A}^i \\ \widehat{a}^i & \alpha^i & \widetilde{a}^i \\ * & * & * \end{bmatrix} &= \begin{bmatrix} L^i & & \\ l^i & \lambda^i & \\ * & * & * \end{bmatrix} \begin{bmatrix} U^i & u^i & \widetilde{U}^i \\ & \mu^i & \widetilde{u}^i \\ & & * \end{bmatrix} \\ &+ \begin{bmatrix} L^i & & \\ l^i & \lambda^i & \\ * & * & * \end{bmatrix} \begin{bmatrix} R^i_u & r^i & \widetilde{R}^i \\ & 0 & \widetilde{r}^i \\ & & * \end{bmatrix} + \begin{bmatrix} R^i_\ell & & \\ \widehat{r}^i & 0 & \\ * & * & * \end{bmatrix} \begin{bmatrix} U^i & u^i & \widetilde{U}^i \\ & \mu^i & \widetilde{u}^i \\ & & * \end{bmatrix}.\end{aligned}$$

Here we use the convention to denote matrices and vectors (row or column) by Latin uppercase (capitals) and lowercase letters, respectively, and scalars by Greek symbols. All objects in the first row are known from the previous step, while $l^i, \lambda^i, \mu^i, \widetilde{u}^i$ have to be computed. The second row gives the set of equations:

$$\widehat{a}^i = (l^i + \widehat{r}^i)U^i + l^i R^i_u, \tag{4.5}$$

$$\alpha^i = (l^i + \widehat{r}^i)u^i + l^i r^i + \lambda^i \mu^i, \tag{4.6}$$

$$\widetilde{a}^i = (l^i + \widehat{r}^i)\widetilde{U}^i + l^i \widetilde{R}^i_u + \lambda^i(\widetilde{u}^i + \widetilde{r}^i). \tag{4.7}$$

Once one defines a rule for splitting a row vector $z = l^i + \widehat{r}^i \in \mathbb{R}^i$ into two structurally orthogonal parts l^i and $\widehat{r}^i$ (i.e. $l^i_k \widehat{r}^i_k = 0$ for $k = 1, \ldots, i$), the equation (4.6) is uniquely solvable for l^i and $\widehat{r}^i$. The ILU(τ_1,0) method imposes the splitting: $l^i_k = z_k$ if $|z_k| > \tau_1$, and $l^i_k = 0$, otherwise. Recalling that R^i_u is strictly upper triangle, the vectors l^i and $\widehat{r}^i$ can be computed as is shown in steps (3)-(4) of the ILU(τ_1,τ_2) algorithm below, where vector z is a part of a full size accumulator vector $v \in \mathbb{R}^n$.

After the vectors l^i and $\widehat{r}^i$ are found, $\mu^i, \widetilde{u}^i$ can be computed from (4.6), (4.7) up to the scaling of λ^i ($\widetilde{u}^i$ is determined from the vector $\widetilde{z} = \widetilde{u}^i + \widetilde{r}^i$ using the same splitting rule). In our implementation, we set $\lambda^i = \|\widetilde{u}^i\|_{\ell^\infty}$. Finally, the entries of the factors not exceeding τ_2 are dropped out and ignored in computations as in the standard threshold ILU strategy [31]. Pivots with absolute values smaller than τ_2 are modified. The pseudo-code of the resulting method is given below.

4.1.3. ILU(τ_1, τ_2) algorithm pseudo-code. Input: a sparse non symmetric matrix A, left and right scaling diagonal matrices D_L and D_R (see section 4.1.1), threshold parameters $0 < \tau_2 \le \tau_1 < 1$. For a matrix C, $P(C)$ denotes the subset of indexes (i, j) such that $C_{ij} = 0$. Since R_ℓ is not computed in the course of the factorization, we use below the notation R for the upper triangular error factor R_u; $v \in \mathbb{R}^n$ is an auxiliary vector initially set equal to 0.

```
(1) Main loop by rows of $A$ to compute the rows of $L$ and $U$:
    for $i = 1, \ldots, n$:
(2) Initialize the row accumulator vector $v$ by the $i$th row of the balanced matrix $A$:
        for $j = 1, \ldots, n$ and if $(i, j) \notin P(A)$:
            $v_j := (D_L)_{ii} a_{ij} (D_R)_{jj}$
        end for
```

```
(3) Loop over all already computed rows of $U$:
        **for** $k = 1, \dots, i-1$ and if $v_k \neq 0$:
(4) Update the accumulator vector:
            $v_k := v_k / U_{kk}$
            **if** $|v_k| > \tau_2$ **then**
                **for** $j = k+1, \dots, n$ and if $(k, j) \notin P(U)$:
                    $v_j := v_j - v_k U_{kj}$
                **end for**
            **end if**
            **if** $|v_k| > \tau_1$ **then**
                **for** $j = k+1, \dots, n$ and if $(k, j) \notin P(R)$:
                    $v_j := v_j - v_k R_{kj}$
                **end for**
            **end if**
        **end for**
(5) Rescale the $i$th row of $U$:
        $\lambda_i := \max_{k=i,\dots,n} |v_k|$
        **if** $\lambda_i < \tau_2$ **then**
            $\lambda_i := \tau_2$
        **end if**
        **for** $j = i, \dots, n$ and if $v_j \neq 0$:
            $v_j := v_j / \lambda_i$
        **end for**
(6) Compute the $i$th row of $L$:
        $L_{ii} = \lambda_i$
        **for** $j = 1, \dots, i-1$ and if $|v_j| > \tau_1$:
            $L_{ij} := v_j$
        **end for**
(7) Compute the $i$th row of $U$ and $R$:
        **if** $|v_i| < \tau_2$ **then**
            $v_i := \tau_2 \cdot \mathrm{sign}(v_i)$
        **end if**
        $U_{ii} = v_i$
        **for** $j = i+1, \dots, n$ and if $v_j \neq 0$:
            **if** $|v_j| > \tau_1$ **then**
                $U_{ij} := v_j$
            **else if** $|v_j| > \tau_2$ **then**
                $R_{ij} := v_j$
            **end if**
        **end for**
(8) Clear nonzero elements of the row accumulator $v$:
        **for** $j = 1, \dots, n$ and if $v_j \neq 0$:
            $v_j := 0$
        **end for**
    **end for**
(9) Perform the final re-scaling of the incomplete factors $L$ and $U$:
    **for** $i = 1, \dots, n$:
        **for** $j = 1, \dots, i$ and if $(i, j) \notin P(L)$:
            $L_{ij} := L_{ij} / (D_L)_{ii}$
        **end for**
        **for** $j = i, \dots, n$ and if $(i, j) \notin P(U)$:
            $U_{ij} := U_{ij} / (D_R)_{jj}$
        **end for**
```

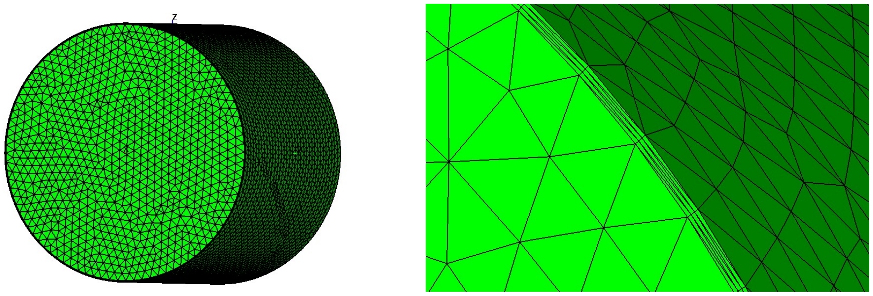

FIG. 5.1. *Cylindrical domain with mesh 5 is shown on the left. The right picture zooms the mesh near the lateral boundary to show anisotropic elemets, otherwise not seen on the left picture.*

```
        end for
```

Note that the row-wise variant of the two-parameter ILU factorization drops off elements of matrix R_ℓ after processing ith row of A. This essentially saves the required working memory.

If an available working memory limit is exhausted in the course of computations, one can discard those entries of R_u which are never used later in computations. Therefore, the factorization can be continued with (partially) compressed factors. In the present implementation of ILU(τ_1,τ_2), the sparsity of matrices is exploited as follows: L and U are stored in the CSR format using separate integer pointers. All the inner loops are made along the sparsity structure indices. Other loops over row accumulator vector v are based on linked-list data structure.

We remark that ILU(τ_1,τ_2) with $\tau_1 = \tau_2$ is similar to the ILUT(p,τ) dual parameter incomplete factorization [31] with $p = n$ (all elements passing the threshold criterion are kept in the factors). If no small pivots modification is done, the only differences between the algorithms, are the scaling of pivots, and a row dependent scaling of threshold values used in ILUT. Recall that we also preprocess the matrix by a proper two-side scaling.

5. Numerical results.

In this section, we show results of several numerical experiments with different values of fluid, discretization and threshold parameters. We look for optimal values of ILU thresholds and how is sensitive the preconditioner performance to deviations of τ-s from this optimal values. The stopping criterion in all experiments is the decrease of the residual by 10 orders of magnitude. Three flow problems of increasing computational complexity are considered in this section.

5.1. Pipe flow.

First, we consider a flow in a cylinder of circular cross-section. The length of the cylinder is 2, the diameter is 1, $\mathbf{w}$ is the Poiseuille flow with $\max_{\Gamma_0} |\mathbf{w}| = 1$. We prescribe zero no-slip conditions on the lateral boundary of the cylinder. The parabolic inflow profile is prescribed on the inlet of the cylinder and $-\nu(\nabla \mathbf{u}) \cdot \mathbf{n} + p\mathbf{n} = \mathbf{0}$ on the outlet.

To discretize the problem, we build several tetrahedra subdivisions of Ω (the lateral boundary is approximated by a triangulated surface). First, three increasingly fine meshes with regular tetrahadra elements are constructed. The corresponding number of degrees of freedom and average number of non-zero entries per row in the saddle-point matrix from (1.3) are the following: d.o.f. = 7330, nz$(A)/n = 19.5$ (Mesh 1), d.o.f. = 42066, nz$(A)/n = 27.3$ (Mesh 2), d.o.f. = 296715, nz$(A)/n = 34.1$ (Mesh 3). Further, two more meshes are build, each of these two contains 3 layers of

Table 5.1

The dependence of ILU(τ) performance on the choice of threshold parameter for the pipe flow; results are shown for $\nu = 0.001$, $\alpha = 1$, Meshes 3 and 5.

τ	fill_{LU}	#it	T_{build}	T_{it}	T_{CPU}	fill_{LU}	#it	T_{build}	T_{it}	T_{CPU}
			Mesh 3					Mesh 5		
0.100	0.387	135	1.2	13.5	14.7					
0.080	0.497	94	1.5	10.2	11.7	0.385	129	2.3	23.3	25.6
0.070	0.571	76	1.7	8.7	10.3	0.434	115	2.5	21.6	24.1
0.060	0.667	60	1.9	7.3	9.2	0.519	69	2.9	13.7	16.6
0.050	0.793	52	2.3	6.8	**9.1**	0.640	62	3.4	13.2	16.6
0.040	0.969	49	2.9	7.0	9.9	0.798	52	4.2	12.1	**16.4**
0.030	1.239	44	3.9	7.2	11.1	1.003	43	5.4	11.2	16.6
0.020	1.722	30	5.9	5.9	11.8	1.360	31	7.7	9.5	17.3
0.010	2.917	22	12.3	6.1	18.4	2.209	24	15.0	10.0	25.0
0.007	3.754	18	17.8	5.9	23.8	2.766	18	21.0	8.7	29.7
0.005	4.700	16	25.1	6.2	31.3	3.384	16	28.8	8.9	37.7
0.003	6.472	13	41.6	6.3	47.9	4.520	12	46.5	8.2	54.7
0.002	8.207	11	61.3	6.4	67.7	5.612	12	67.4	9.6	77.0
0.001	11.954	9	115.5	7.0	122.5	8.007	10	125.4	10.6	135.9

anisotropic elements aligned along the lateral boundary. These two meshes mimic the situation when one has to adapt a mesh to a boundary layer. The data for these two meshes are given by d.o.f. = 501639, $\text{nz}(A)/n = 37.0$, anisotropy ratio is equal to 5 (Mesh 4), d.o.f. = 528598, $\text{nz}(A)/n = 37.1$, anisotropy ratio is equal to 10 (Mesh 5). The later mesh is illustrated in Figure 5.1.

In all experiments in this section, the resulting linear algebraic systems are solved by the preconditioned BiCGstab method with either ILU(τ) or ILU(τ_1, τ_2) preconditioners, with zero initial guess. The ILU(τ_1,τ_2) preconditioner is computed by the algorithm from section 4.1.3, and ILU(τ):=ILU(τ,τ). All presented results are computed with 5 iterations to balance the matrix, as described in section 4.1.1. Using only 1 iteration we experienced slightly worse performance of preconditioners. However, without the pre-processing both ILU(τ) and ILU(τ_1, τ_2) fail for most of the examples treated in the numerical section.

In our first series of experiments, we study the τ-dependence of the ILU(τ) preconditioner performance. The computations were run on the finest mesh 3 for $\nu = 0.001$ and $\alpha = 1$. The results are presented in Table 5.1. T_{build} and T_{it} show CPU time spent for building ILU factorization (including the two-side scaling) and iterations, respectively; $T_{\text{CPU}} = T_{\text{build}} + T_{\text{it}}$, and #it is the number of BiCGstab iterations needed to satisfy the stoping criterion. The ratio of fill-in is computed from

$$\text{fill}_{LU} = (\text{nz}(L) + \text{nz}(U)) / \text{nz}(A), \qquad \text{nz}(A) = \sum_{ij} \text{sign}|A_{ij}|.$$

Note that $\text{fill}_{LU} \leq 1$ means that the number of non-zero elements in factors is less then in ILU(0), the commonly used ILU factorization by position. For smaller values of τ we observe the increase of fill-in and T_{build}, but the decrease of iteration numbers and T_{it}; both facts are expected. The optimal τ is found to be 0.05 and its variation ($\tau \in [0.03, 0.08]$) gives a minor increase of total computation time. We repeated the same experiments with Mesh 5, which contains anisotropic elements. The results are shown in Table 5.1. We observe that the performance of the preconditioner does not change significantly, the optimal value of τ was found to be about the same. We run

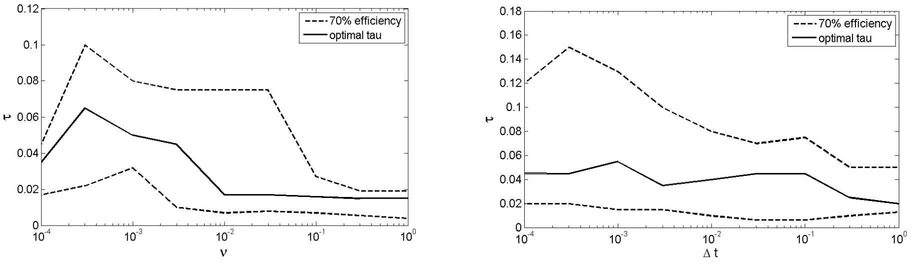

FIG. 5.2. *The dependence of the optimal values of the threshold parameter τ in ILU(τ) on the viscosity ν (left) and $\alpha = \frac{1}{\Delta t}$ (right). Both plots also show the bounds on τ where the efficiency of the preconditioner is at least 70% of the optimal case.*

the same set of experiments with meshes 1, 2, and 4, and observed that the optimal values of all τ-s are almost grid independent.

Further, we study the dependence of optimal threshold parameters with respect to the variation of ν and α. The results are presented in Figure 5.2: on the left plot we vary ν for fixed $\alpha = 10$ and given mesh 3, while on the right plot we vary α for fixed $\nu = 0.01$ and the same mesh. Optimal τ-s were found with respect to total computational time, i.e. $T_{\text{CPU}} = T_{\text{build}} + T_{\text{it}}$. We also compute a range of "quasi-optimal" τ-s, which is defined as the set of all parameters τ such that the efficiency of ILU(τ) decreases at most by 30% compared to the case of the optimal value. From the plots we see that the optimal threshold values do depend on ν and α. However, the range of acceptable values is rather wide, though it decreases for the diffusion dominated case. For further experiments, we choose a quasi-optimal value $\tau = 0.03$, independent of parameters. Table 5.2 collects the results of experiments with this quasi-optimal threshold value, showing the rate of fill-in, the number of iterations and T_{CPU} for all five meshes, different ν-s, and α-s. One observes convergent iterations for all meshes and parameters, with certain loss of performance for the strongly convection dominated Oseen problem discretized on strongly anisotropic mesh. It is interesting that a moderately convection dominated problem, i.e. $\nu \in \{0.1; 0.01; 0.001\}$ appear to be more amenable for efficient ILU preconditioning than diffusion dominated case.

We repeat the same set of experiments, but now with the two-parameter ILU preconditioner. We set $\tau_1 = 0.03$ (equal to the quasi-optimal value in ILU(τ) preconditioner) and $\tau_2 = c_0\tau_1^2$, with $c_0 = 7$. We note that in the symmetric positive definite case, the author of [20] recommends an *ad hoc* choice of $c_0 = 10$, while we found some decreasing of c_0 beneficial for the ILU(τ_1,τ_2) performance. The results are reported in Table 5.3 and they appear to be largely comparable to those obtained with ILU(τ).

5.2. The Ethier–Steinman problem. Next we consider the well known Ethier-Steinman solution for the Navier-Stokes equations from [14]. For chosen parameters a, d and viscosity ν, the exact solution is given in $[-1, 1]^3$ by

$$u_1 = -a\left(e^{ax}\sin(ay+dz) + e^{az}\cos(ax+dy)\right)e^{-\nu d^2 t}$$
$$u_2 = -a\left(e^{ay}\sin(az+dx) + e^{ax}\cos(ay+dz)\right)e^{-\nu d^2 t}$$
$$u_3 = -a\left(e^{az}\sin(ax+dy) + e^{ay}\cos(az+dx)\right)e^{-\nu d^2 t}$$

Table 5.2

The performance of the one-parameter ILU($\tau = 0.03$) preconditioner for the pipe flow test case. The results are shown for various values of viscosity ν, α, and different meshes.

	ν:	1		10^{-1}		10^{-2}		10^{-3}		10^{-4}	
Mesh	α:	10	100	10	100	10	100	10	100	10	100
						fill_{LU}					
1		0.88	0.73	0.74	0.80	0.80	1.06	1.06	1.18	1.17	1.20
2		0.89	0.78	0.78	0.62	0.72	0.94	1.26	1.19	1.71	1.24
3		0.89	0.85	0.85	0.66	0.72	0.72	1.24	1.08	2.86	1.25
4		0.89	0.86	0.86	0.74	0.77	0.71	1.00	0.92	1.83	1.14
5		0.83	0.81	0.80	0.70	0.73	0.73	1.00	0.99	1.91	1.02
						#it					
1		12	10	9	12	11	14	13	15	15	15
2		48	21	19	19	19	25	23	26	27	26
3		170	61	59	34	32	38	44	42	79	52
4		169	62	56	34	31	43	41	67	87	73
5		177	67	59	36	32	50	43	59	99	136
						T_{CPU}					
1		0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06
2		0.82	0.44	0.42	0.38	0.42	0.58	0.71	0.68	1.00	0.70
3		25.1	10.2	10.0	5.86	5.91	7.15	11.0	9.33	33.2	11.8
4		43.8	18.2	16.8	10.7	10.3	13.0	15.6	21.0	44.2	25.1
5		51.4	19.6	17.7	11.4	10.7	15.3	16.9	20.5	51.8	41.9

and

$$\begin{aligned}p = -\frac{a^2}{2}(&e^{2ax} + e^{2ay} + e^{2az} + 2\sin(ax+dy)\cos(az+dx)e^{a(y+z)}\\ &+2\sin(ay+dz)\cos(ax+dy)e^{a(z+x)}\\ &+2\sin(az+dx)\cos(ay+dz)e^{a(x+y)})e^{-2\nu d^2 t}.\end{aligned}$$

In our experiments we set $a = \pi/4$, $d = \pi/2$ and vary ν. This problem was developed as a 3D analogue to the Taylor vortex problem, for the purpose of benchmarking. Although unlikely to be physically realized, it is a good test problem because it has analitically known solution, the flow has no principle direction, but has a non-trivial vortical structure.

For the purpose of testing the algebraic solver, we do not perform time-stepping, but linearize the Navier–Stokes equation over the analytical solution at $t = 0.1$. For the discretization, a regular tetrahedrization of the cube $[-1,1]^3$ is build. The coarsest mesh is uniformly refined three times. This results in four gradually refined meshes. The corresponding number of degrees of freedom and average number of non-zero entries per row in the saddle-point matrix from (1.3) were the following: d.o.f. = 2251, $\text{nz}(A)/n = 17.3$ (Mesh 1), d.o.f. = 12420, $\text{nz}(A)/n = 25.8$ (Mesh 2), d.o.f. = 75660, $\text{nz}(A)/n = 32.5$ (Mesh 3), d.o.f. = 522220, $\text{nz}(A)/n = 37.5$ (Mesh 4). Similar to the previous test, the resulting linear algebraic system was solved by BiCGstab method with either ILU(τ_1) or ILU(τ_1, τ_2) preconditioners and zero initial guess.

Figure 5.3 demonstrates the dependence of ILU(τ) performance with respect to the choice of the threshold parameter τ. The experiments were run with $\nu = 0.01$, $\alpha = 10$, and for all four meshes. The vertical axis shows the total CPU time per degree of freedom. We observe certain dependence of optimal τ on the mesh size,

TABLE 5.3
The performance of the two-parameter ILU($\tau_1 = 0.03, \tau_2 = 7\tau_1^2$) preconditioner for the pipe flow test case. The results are shown for various values of viscosity ν, α and different meshes.

	ν:	1		10^{-1}		10^{-2}		10^{-3}		10^{-4}	
Mesh	α:	10	100	10	100	10	100	10	100	10	100
						fill_{LU}					
1		0.91	0.73	0.74	0.76	0.77	0.96	1.01	1.07	1.11	1.09
2		0.93	0.79	0.80	0.62	0.72	0.84	1.21	1.03	1.65	1.09
3		0.93	0.88	0.88	0.67	0.73	0.70	1.20	0.95	2.59	1.10
4		0.91	0.88	0.87	0.74	0.77	0.69	0.97	0.84	1.69	1.01
5		0.86	0.83	0.83	0.71	0.74	0.71	0.97	0.88	1.74	0.94
						#it					
1		10	9	7	11	9	12	11	15	12	13
2		36	19	16	15	14	20	20	20	22	25
3		157	50	42	30	24	35	31	36	47	39
4		171	50	44	32	24	35	30	51	54	63
5		127	55	42	29	22	36	32	53	45	83
						T_{CPU}					
1		0.06	0.05	0.05	0.06	0.06	0.08	0.10	0.10	0.11	0.09
2		0.95	0.62	0.58	0.49	0.59	0.81	1.39	0.99	2.11	1.08
3		26.9	11.7	10.7	7.21	7.27	8.75	17.9	12.7	61.1	15.0
4		49.5	20.0	18.6	13.5	12.5	14.4	23.0	22.7	68.6	29.5
5		39.7	21.5	18.3	13.1	12.2	15.5	24.2	24.3	68.3	34.3

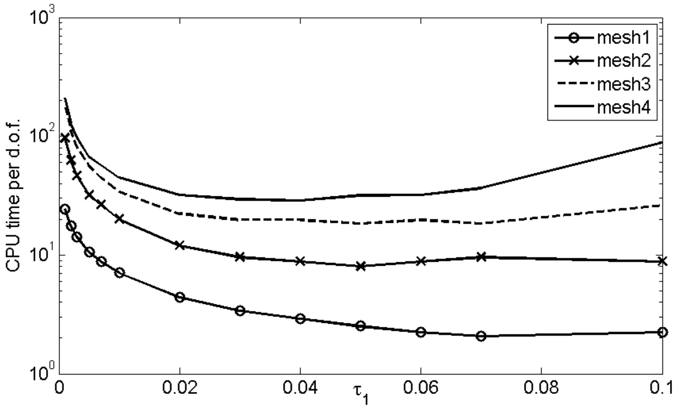

FIG. 5.3. *Dependence of ILU(τ) on the threshold parameter τ for the Ethier–Steinman test case; $\nu = 0.01$, $\alpha = 10$.*

but the range of quasi-optimal parameters is wide and $\tau \in [0.02, 0.08]$ would be a reasonable choice for all meshes. We set $\tau = 0.02$ and run computation with ILU(τ) and ILU($\tau, 7\tau^2$) for different values of viscosity coefficient $\nu \in \{1, 0.1, 0.01, 0.001\}$ and parameter $\alpha \in \{1, 0.1, 0.01\}$. The results for two fine meshes are collected in Table 5.4. From the results in this table, we see that in the range of moderate viscosity values, both preconditions demonstrate very similar behaviour with ILU(τ) being somewhat cheaper during the setup phase. For the diffusion dominated case ($\nu = 1$, $\alpha = 1$), when the matrix becomes more symmetric, the two-parameter preconditioning wins in terms of iteration number and total CPU time. The convection dominated case

Table 5.4

The performance of the ILU($\tau = 0.02$) and ILU($\tau_1 = 0.02, \tau_2 = 7\tau_1^2$) preconditioners for the Ethier–Steinman flow. The results are shown for various values of ν, α and two different meshes.

	ν: 1			10^{-1}			10^{-2}			10^{-3}			
Mesh	α: 1	10	100	1	10	100	1	10	100	1	10	100	ILU
						fill$_{LU}$							
3	1.22	1.20	1.07	1.19	1.08	0.81	1.97	1.47	1.21	79.96	5.83	1.78	ILU1
3	1.21	1.19	1.07	1.20	1.08	0.81	1.98	1.46	1.12	20.62	4.93	1.64	ILU2
4	1.22	1.22	1.17	1.21	1.17	0.93	1.48	1.27	0.97	n/c	6.53	1.89	ILU1
4	1.20	1.20	1.16	1.20	1.16	0.93	1.53	1.30	0.96	9.28	5.33	1.80	ILU2
						#it							
3	72	46	18	29	14	17	12	15	24	n/c	38	28	ILU1
3	58	37	16	24	14	13	12	14	15	58	19	17	ILU2
4	337	296	50	119	38	26	27	22	31	n/c	83	44	ILU1
4	201	158	36	95	31	26	22	24	26	47	38	31	ILU2
						T_{CPU}							
3	4.4	3.4	1.5	2.2	1.4	1.3	2.1	1.7	1.8	n/c	13.6	2.9	ILU1
3	4.2	3.5	2.8	3.3	2.7	2.0	7.6	5.2	3.4	492	46.1	6.2	ILU2
4	170.1	149.7	31.0	58.5	25.7	17.5	20.8	16.7	16.8	n/c	234.1	38.4	ILU1
4	93.8	96.9	35.7	69.9	35.1	26.8	51.2	42.8	35.1	2174	735	89.5	ILU2

appears to be the hardest. Here ILU(τ) fails for $\alpha \in \{1, 10\}$, while the usage of the two-parameter preconditioner leads to a convergent method.

Finally we have a closer look at the most hard case from Table 5.4, i.e. $\nu = 0.001$ and $\alpha = 1$, and experiment with different values of the threshold parameters. Table 5.5 shows the result of this experiments for $\nu = 0.001$ and $\alpha = 1$ on a fixed given Mesh 3. We see that similar to the pipe flow case, optimal parameter for ILU(τ) decreases. Interesting enough, that the decrease of ν and α by 10 times resulted in the 10 times decrease of τ_{opt}, which is consistent with the ellipticity bound on matrix A in Theorem 3.2. Also a 'comfortable' zone around τ_{opt} shrinks making an overshoot in choosing quasi-optimal τ easily possible. For this convection dominated problem, one clearly benefits from using the two-parameter ILU preconditioner. For two-parameter ILU, we fixed $\tau_1 = 0.02$ and vary the scaling factor c_0 in $\tau_2 = c_0\tau_1^2$. The optimal $c_0 = 8$ is close to $c_0 = 7$ we found suitable in the case of pipe flow. Overall, *the two-parameter ILU factorization leads to more efficient preconditioner in terms of memory usage (fill-in) and iteration counts, but with more expensive set-up stage, compared to the standard ILU(τ).*

5.3. Flow in a right coronary artery. Finally, we study the performance of the ILU preconditioner for a model hemodynamic problem of a blood flow in a right coronary artery. The geometry of the flow domain was recovered from a real patient coronary CT angiography, the ANI3D package [24] was used to generate the tetrahedral mesh (see Figure 5.5) and to build the finite element systems (1.3). The diameter of the inlet cross-section is about 0.27 cm and the whole domain can be embedded in a parallelogram with sides 6.5 cm × 6.8 cm × 5 cm. The mesh consists of 120 191 tetrahedra leading to the discrete Navier–Stokes system with 623 883 of unknowns. Other model parameters are $\nu = 0.04\,\mathrm{cm}^2/\mathrm{s}$, $\rho = 1\,\mathrm{g/cm}$, one cardiac cycle period was 0.735 s. The inlet velocity waveform is shown in Figure 5.4 (top-left); it was suggested in [19] on the basis of clinical measurements. This waveform

TABLE 5.5

The performance of ILU(τ) and ILU(τ_1, τ_2) depending on the choice of threshold parameters for the Ethier–Steinman flow; results are shown for $\nu = 0.001$, $\alpha = 1$, Mesh 3.

τ	fill$_{LU}$	#it	T_{CPU}	τ	c_0	fill$_{LU}$	#it	T_{CPU}
	ILU(τ)					ILU($\tau, c_0\tau^2$)		
0.0065	76.041	n/c						
0.0060	76.816	107	656.8	0.02	12.5	27.239	n/c	
0.0055	78.068	55	632.5	0.02	10	23.764	553	570.7
0.0050	79.769	34	655.2	0.02	9	22.609	145	450.8
0.0045	82.126	26	676.0	0.02	**8**	21.537	86	438.2
0.0040	85.046	18	724.1	0.02	7.5	21.084	73	439.7
0.0030	93.718	12	868.7	0.02	7	20.616	58	440.1
0.0020	108.269	8	1135.7	0.02	6	19.963	50	448.8
0.0015	119.858	7	1383.4	0.02	5	18.967	45	470.1
0.0010	137.594	5	1781.1	0.02	4	18.108	39	508.4

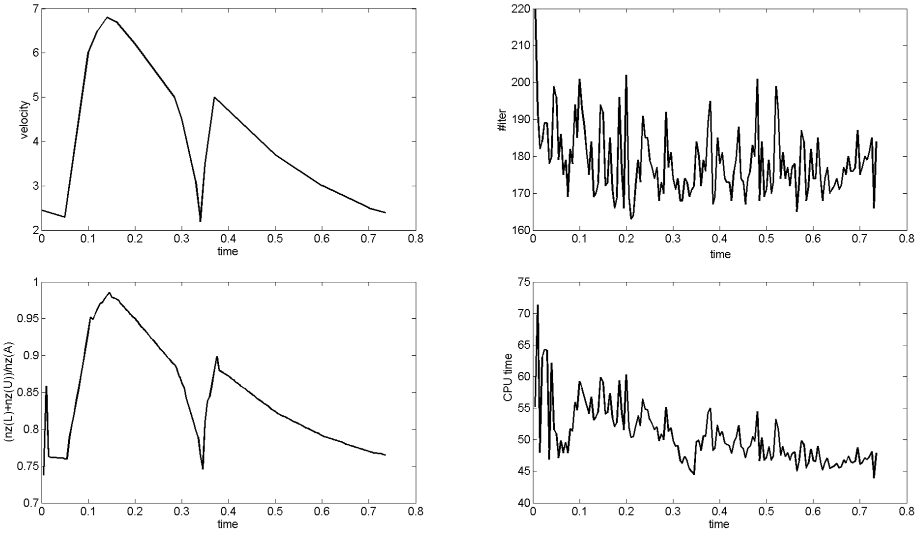

FIG. 5.4. *Right coronary artery test case: The top-left plot shows the velocity waveform on the inflow, the top-right plot shows the number of BiCGStab iterations, the bottom-left plot shows the fill-in ratio, and the bottom-right plot shows linear system solution CPU times. All shown data are functions of time.*

was used to define the flow rate through the inflow cross-section, while for the inflow velocity profile we prescribed the Poiseuille flow. The Neumann boundary condition $-\nu(\nabla \mathbf{u}) \cdot \mathbf{n} + p\mathbf{n} = \mathbf{0}$ was imposed on all outflow boundaries. No elasticity model was used for the vessel walls, i.e., the walls were treated as rigid.

The Navier–Stokes system (1.1) was integrated in time using a semi-implicit second order method with $\Delta t = 0.005$. The Oseen problem (1.2) was solved on every time step with the preconditioned BiCGstab method. The solution from the previous time step was used as the initial guess. For the preconditioner we used the two-parameter ILU factorization with the choice of parameters $\tau_1 = 0.03$, $\tau_2 = 7\tau_1^2$. Recall that these are quasi-optimal parameters for pipe flows from section 5.1. This choice of the preconditioner and parameters results in stable computations over the whole cardiac cycle. The preconditioner performance data is shown in Figure 5.4. It is interesting to note that the graph of the fill-in rate for the LU-factors repeats remarkably well the waveform of the inflow velocity. Thanks to this adaptive feature of the threshold factorization, the variations of the iteration numbers and computational times per linear solve are rather modest, see the right plots in Figure 5.4. The computed solution was physiologically relevant; it is illustrated in Figure 5.5.

6. Closing remarks and conclusions. In this paper, we studied threshold ILU preconditioners for the discrete linearized Navier-Stokes system. Incomplete elementwise factorization preconditioners have a clear advantage of being rather insensitive to several factors, such as a choice of discretization, boundary conditions for governing PDEs, domain geometry, and flow directions, which otherwise influence the performance of many other algebraic solvers for the fluid dynamics problem. Furthermore, the presented method does not need a choice of subsolvers or inner iterations in contrast to many block preconditioners. It is well-known that for discrete elliptic problems, ILU preconditioners do not scale optimally with respect to the number of unknowns. We observed this non-optimality in our numerical experiments as well. However, in numerical experiments this mesh dependence was more pronounced for diffusion dominated flows and less evident when convection plays an important role. For 3D problems, when the number of grid refinement levels is not large, such dependence can be an acceptable price for other robustness properties of the preconditioner.

Small values of viscosity parameters cause problems for most, if not all, known preconditioners for (1.3). Our results show that the threshold ILU is not an exception. At the same time, we found that the performance range with respect to ν of ILU(τ) and, especially, of ILU(τ_1, τ_2) is rather impressive, and likely covers most of laminar flows. Introducing subgrid models for higher Re numbers (e.g., turbulent) flows changes the discrete system, and since such models are commonly dissipative, this improves algebraic properties of discrete system and should make the presented preconditioning also feasible. We observed such an improvement if SUPG stabilization added to the finite element formulation of the Ethier–Steinman problem for $\nu = 10^{-3}$, but do not include these extra results in the report.

Incomplete threshold factorization is not a black-box method. A user should make at least a choice of threshold parameter(s), and many techniques have been suggested in the literature to improve the performance of ILU preconditioners. For fluid flows treated in this paper, we found that natural $\mathbf{u}$-p ordering of unknowns and matrix two-side scaling is sufficient for numerically stable factorizations. Further performance improvements by using, for example, matrix-band diminishing re-ordering of velocity unknowns, could be possible. Although optimal threshold parameters appear to be flow-dependent, quasi-optimal τ-s can be chosen and successfully used for a wide range of flow and discretization parameters.

We considered a Tismenetsky–Kaporin type incomplete two-parameter factorization for non-symmetric matrices and tested it for matrices arising in computational fluid dynamics. While for modest values of ν (leading to a parity between convection and diffusion terms) the performance of ILU(τ_1,τ_2) was similar to that of ILU(τ), for larger and smaller ν-s ILU(τ_1,τ_2) was found to provide a more efficient preconditioner. It was observed that ILU(τ_1, τ_2) preconditioner has a low fill-in and leads to faster convergent iterations for the expense of more time consuming set-up phase. This properties may make it an ideal choice for time-dependent computations, when one can re-use a preconditioner over several time steps.

A numerical analysis of incomplete factorizations for non-symmetric matrices is

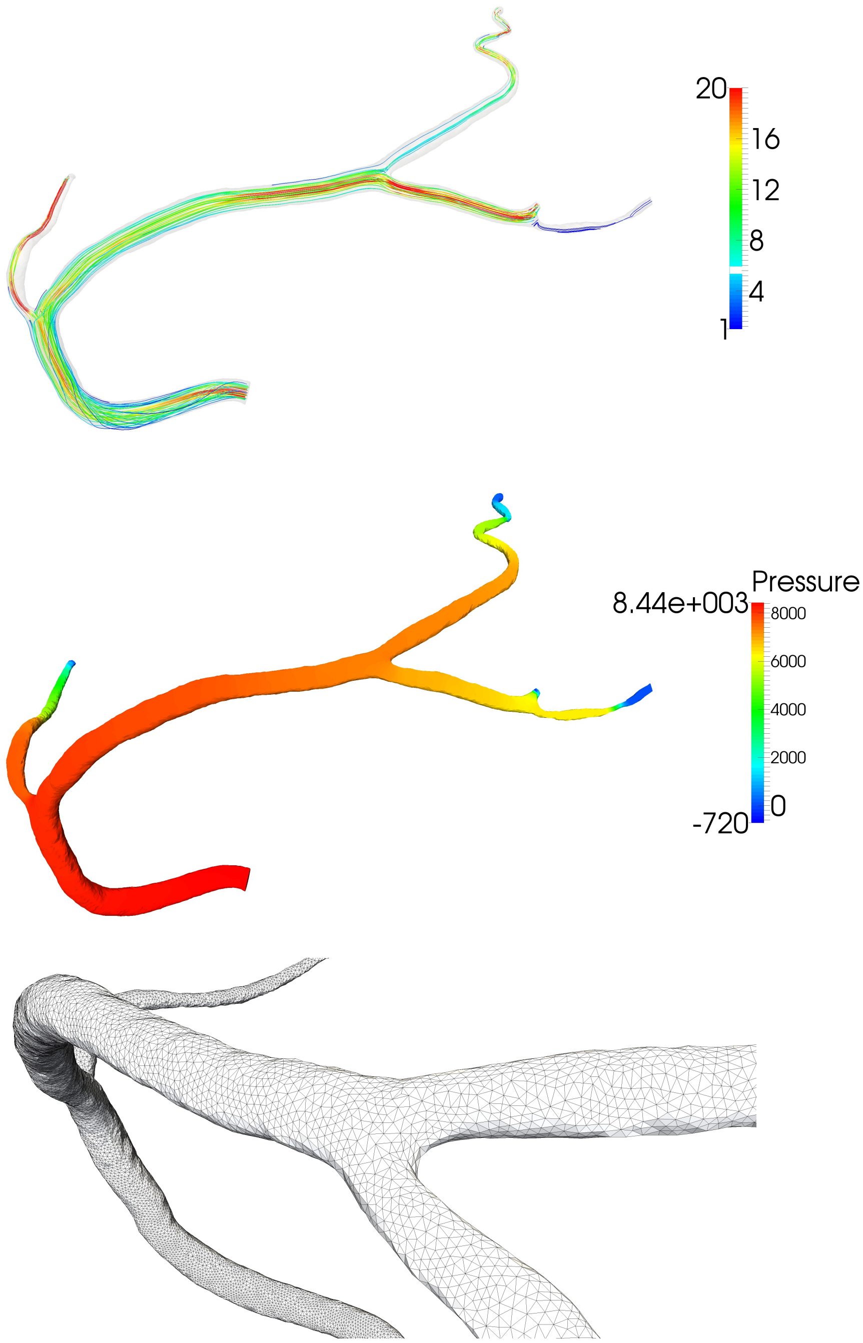

FIG. 5.5. *Right coronary artery: Top picture shows selected steamlines colored by the velocity absolute value at time 0.4s; Middle picture shows the pressure distribution at time 0.4s; Bottom picture illustrates the grid for this test case.*

still limited. This paper proves numerical stability bounds for the exact LU factorization of non-symmetric saddle-point matrices. We estimated the dependence of the constants in these bounds on the flow problem parameters. This might give some insight into the performance of incomplete factorizations applied to flow problems.

The two-parameter ILU preconditioner was applied to simulate a blood flow in a right coronary artery reconstructed from a real patient coronary CT angiography. We found the performance of the preconditioner satisfactory.

Acknowledgements. The authors thank Tatiana Dobroserdova and Alexander Danilov for the assistance in building tetrahedral meshes and finite element systems, and Sergei Goreinov for sharing his implementation of the row-wise variant of the ILU(τ_1,τ_2) factorization. Numerical implementation and experiments in sections 5.1, 5.2 have been supported by RFBR grant 14-01-00830. We are grateful to Igor Kaporin for his insightful comments on a draft of this report.

REFERENCES

[1] M. Benzi, *Preconditioning techniques for large linear systems: a survey*, Journal of Computational Physics, 182 (2002), pp. 418–477.

[2] ———, *A generalization of the hermitian and skew-hermitian splitting iteration*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 360–374.

[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[4] M. Benzi, M. Ng, Q. Niu, and Z. Wang, *A relaxed dimensional factorization preconditioner for the incompressible Navier–Stokes equations*, Journal of Computational Physics, 230 (2011), pp. 6185–6202.

[5] M. Benzi, M. A. Olshanskii, and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, International Journal for Numerical Methods in Fluids, 66 (2011), pp. 486–508.

[6] M. Braack, P. B. Mucha, and W. M. Zajaczkowski, *Directional do-nothing condition for the Navier–Stokes equations*, Journal of Computational Mathematics, 32 (2014), pp. 507–521.

[7] Z.-H. Cao, *A class of constraint preconditioners for nonsymmetric saddle point matrices*, Numerische Mathematik, 103 (2006), pp. 47–61.

[8] O. Dahl and S. Ø. Wille, *An ILU preconditioner with coupled node fill-in for iterative solution of the mixed finite element formulation of the 2D and 3D Navier-Stokes equations*, International Journal for Numerical Methods in Fluids, 15 (1992), pp. 525–544.

[9] V. F. de Almeida, A. M. Chapman, and J. J. Derby, *On equilibration and sparse factorization of matrices arising in finite element solutions of partial differential equations*, Numerical Methods for Partial Differential Equations, 16 (2000), pp. 11–29.

[10] S. Deparis, G. Grandperrin, and A. Quarteroni, *Parallel preconditioners for the unsteady Navier–Stokes equations and applications to hemodynamics simulations*, Computers & Fluids, 92 (2014), pp. 253–273.

[11] H. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations*, SIAM Journal on Scientific Computing, 17 (1996), pp. 33–46.

[12] H. C. Elman, D. Silvester, and A. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Oxford University Press, 2014.

[13] H. C. Elman and R. S. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier–Stokes equations*, Electronic Transactions on Numerical Analysis, 35 (2009), pp. 257–280.

[14] C. R. Ethier and D. Steinman, *Exact fully 3D Navier–Stokes solutions for benchmarking*, International Journal for Numerical Methods in Fluids, 19 (1994), pp. 369–375.

[15] V. Girault and P.-A. Raviart, *Finite element approximation of the navier-stokes equations*, Lecture Notes in Mathematics, Berlin Springer Verlag, 749 (1979).

[16] G. H. Golub and C. v. Loan, *Matrix computations*, Baltimore, MD: Johns Hopkins University Press, 1996.

[17] G. H. Golub and C. Van Loan, *Unsymmetric positive definite linear systems*, Linear Algebra and its Applications, 28 (1979), pp. 85–97.

[18] J. Guzmán and M. Neilan, *Conforming and divergence-free Stokes elements in three dimensions*, IMA Journal of Numerical Analysis, 34 (2014), pp. 1489–1508.

[19] J. Jung, A. Hassanein, and R. W. Lyczkowski, *Hemodynamic computation using multiphase flow dynamics in a right coronary artery*, Annals of biomedical engineering, 34 (2006), pp. 393–407.

[20] I. E. Kaporin, *High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition*, Numerical Linear Algebra with Applications, 5 (1998), pp. 483–509.

[21] ———, *Scaling, reordering, and diagonal pivoting in ilu preconditionings*, Russian Journal of Numerical Analysis and Mathematical Modelling rnam, 22 (2007), pp. 341–375.

[22] ———, *Scaling, preconditioning, and superlinear convergence in gmres-type iterations*, Matrix Methods: Theory, Algorithms, Applications (V. Olshevsky, E. Tyrtyshnikov, eds.), World Scientific Publ, (2010), pp. 273–295.

[23] K. Lipnikov, Y. Vassilevski, A. Danilov, et al., *Advanced Numerical Instruments 2D*, http://sourceforge.net/projects/ani2d.

[24] ———, *Advanced Numerical Instruments 3D*, http://sourceforge.net/projects/ani3d.

[25] O. E. Livne and G. H. Golub, *Scaling by binormalization*, Numerical Algorithms, 35 (2004), pp. 97–120.

[26] J. Mayer, *Symmetric permutations for i-matrices to delay and avoid small pivots during factorization*, SIAM Journal on Scientific Computing, 30 (2008), pp. 982–996.

[27] M. A. Olshanskii and A. Reusken, *Grad-div stablilization for Stokes equations*, Mathematics of Computation, 73 (2004), pp. 1699–1718.

[28] M. A. Ol'shanskii and V. M. Staroverov, *On simulation of outflow boundary conditions in finite difference calculations for incompressible fluid*, International Journal for Numerical Methods in Fluids, 33 (2000), pp. 499–534.

[29] M. A. Olshanskii and E. E. Tyrtyshnikov, *Iterative methods for linear systems: theory and applications*, SIAM, 2014.

[30] M. A. Olshanskii and Y. V. Vassilevski, *Pressure Schur complement preconditioners for the discrete Oseen problem*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2686–2704.

[31] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.

[32] R. L. Sani and P. M. Gresho, *Résumé and remarks on the open boundary condition minisymposium*, International Journal for Numerical Methods in Fluids, 18 (1994), pp. 983–1008.

[33] J. Scott and M. Tuma, *On signed incomplete Cholesky factorization preconditioners for saddle-point systems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2984–A3010.

[34] A. Segal, M. ur Rehman, and C. Vuik, *Preconditioners for incompressible Navier–Stokes solvers*, Numerical Mathematics: Theory, Methods and Applications, 3 (2010), pp. 245–275.

[35] G. W. Soules, *The rate of convergence of Sinkhorn balancing*, Linear Algebra and its Applications, 150 (1991), pp. 3–40.

[36] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, Springer, New York, 1993.

[37] R. Temam, *Navier–Stokes equations: theory and numerical analysis*, vol. 343, American Mathematical Soc., 2001.

[38] M. Tismenetsky, *A new preconditioning technique for solving large sparse linear systems*, Linear Algebra and its Applications, 154 (1991), pp. 331–353.

[39] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approache*, vol. 6, Springer Science & Business Media, 1999.

[40] C. Vuik, G. Segal, et al., *A comparison of preconditioners for incompressible Navier–Stokes solvers*, International Journal for Numerical Methods in Fluids, 57 (2008), pp. 1731–1751.

[41] ———, *Simple-type preconditioners for the Oseen problem*, International Journal for Numerical Methods in Fluids, 61 (2009), pp. 432–452.

[42] M. Wabro, *AMGe—coarsening strategies and application to the Oseen equations*, SIAM Journal on Scientific Computing, 27 (2006), pp. 2077–2097.

[43] J. Zhao, *The generalized Cholesky factorization method for saddle point problems*, Applied Mathematics and Computation, 92 (1998), pp. 49–58.