DELFT UNIVERSITY OF TECHNOLOGY

REPORT 13-07

On preconditioning incompressible non-Newtonian flow problems

Xin He, Maya Neytcheva and C. Vuik

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2013

Abstract

This paper deals with fast and reliable numerical solution methods for the incompressible non-Newtonian Navier-Stokes equations. To handle the nonlinearity of the governing equations, the Picard and Newton methods are used to linearize these coupled partial differential equations. For space discretization we use the finite element method and utilize the two-by-two block structure of the matrices in the arising algebraic systems of equations. The Krylov subspace iterative methods are chosen to solve the linearized discrete systems and the development of computationally and numerically efficient preconditioners for the two-by-two block matrices is the main concern in this paper. In non-Newtonian flows, the viscosity is not constant and its variation is an important factor that effects the performance of some already known preconditioning techniques. In this paper we examine the performance of several preconditioners for variable viscosity applications, and improve them further to be robust with respect to variations in viscosity.

Mathematics subject classification: 65F10 65F08 65N30.

Key words: non-Newtonian flows, Navier-Stokes equations, two-by-two block systems, Krylov subspace methods, preconditioners.

1 Introduction

In this paper, we assume that the velocity $\mathbf{u}$ and the pressure p satisfy the following generalized stationary incompressible Navier-Stokes equations:

$$\begin{aligned} -\nabla\cdot(2\nu(D_{\mathrm{II}}(\mathbf{u}),p)\mathbf{D}\mathbf{u}) + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f}, &\quad \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0, &\quad \text{in } \Omega \end{aligned} \tag{1}$$

with boundary conditions given by

$$\begin{aligned} \mathbf{u} = \mathbf{g}, &\quad \text{on } \partial\Omega_D \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p = 0. &\quad \text{on } \partial\Omega_N \end{aligned}$$

Here Ω is a bounded and connected domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$, and $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ is its boundary, where $\partial\Omega_D$ and $\partial\Omega_N$ denote the parts of the boundary where Dirichlet and Neumann boundary conditions for $\mathbf{u}$ are imposed, respectively. The terms $\mathbf{f} : \Omega \to \mathbb{R}^d$ and $\mathbf{g}$ are a given force field and Dirichlet boundary data for the velocity. The term $\mathbf{n}$ denotes the outward-pointing unit normal to the boundary.

The term $\mathbf{D}\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^T\mathbf{u})$ denotes the rate-of-deformation tensor and $\nu(\cdot)$ denotes the kinematic viscosity which depends on the second invariant of the rate-of-deformation tensor $D_{\mathrm{II}}(\mathbf{u}) = \frac{1}{2}\mathrm{tr}(\mathbf{D}^2\mathbf{u})$ and the pressure p. The following models are most often used in non-Newtonian fluids:

- [M1] $\nu(D_{\text{II}}(\mathbf{u}), p) = \nu_0 + \tau(D_{\text{II}}(\mathbf{u}))^{\frac{\alpha}{2}}$ describing "power law" non-Newtonian fluids (e.g., [10]);
- [M2] $\nu(D_{\text{II}}(\mathbf{u}), p) = \nu_\infty + (\nu_0 - \nu_\infty)(1 + \beta D_{\text{II}}(\mathbf{u}))^{\frac{\alpha}{2}}$ describing "Carreau law" non-Newtonian fluids;
- [M3] $\nu(D_{\text{II}}(\mathbf{u}), p) = \sqrt{2} \sin \phi p(D_{\text{II}}(\mathbf{u}))^{-\frac{1}{2}}$ describing "Schaeffer law" non-Newtonian fluids (e.g., [31]);
- [M4] non-Newtonian fluids with pressure and shear dependent viscosity (e.g., [19]),

with appropriate parameters ν_0, ν_∞, α, β, τ, ϕ.

In this work we only consider the Bingham model, namely, $\nu(D_{\text{II}}(\mathbf{u})) = \nu_0 + \tau(D_{\text{II}}(\mathbf{u}))^{-\frac{1}{2}}$ (M1 with $\alpha = 1$), which is a special case of "power law" non-Newtonian fluids. Due to the possible singularity of $D_{\text{II}}(\mathbf{u})$, some regularization techniques are required. Here we utilize a widely used regularization method, namely, $\nu(D_{\text{II}}(\mathbf{u})) = \nu_0 + \tau(D_{\text{II}}(\mathbf{u}) + \varepsilon^2)^{-\frac{1}{2}}$ (c.f., [10, 14]). In practice, in order to characterize well the Bingham flow, one needs to choose ε as small as possible. On the other hand, in numerical experiments we see that reasonably small values of ε lead to higher computational complexities for the numerical solution methods.

Since the viscosity function $\nu(D_{\text{II}}(\mathbf{u}))$ also depends on the velocity $\mathbf{u}$, two terms in (1) exhibit a nonlinear behavior: $\nabla \cdot (2\nu(D_{\text{II}}(\mathbf{u}))\mathbf{D}\mathbf{u})$ and $(\mathbf{u} \cdot \nabla \mathbf{u})$. Thus, a linearization technique is needed. As it turns out, for various linearizations, the variable viscosity Oseen- or Stokes-type problem arises. In all cases, however, the finite element discretization of the linearized problems results in discrete linear systems of two-by-two block form. In this paper, Krylov subspace methods with appropriate preconditioners are chosen to solve the resulting linear systems. As far as the authors know, efficient preconditioners are only proposed for the variable viscosity Stokes-type problems before, e.g., [14, 28]. In this paper, fast and reliable preconditioning techniques are considered for both linearized problems, i.e., the Oseen- and Stokes-type problems with variable viscosity. In the past decades, the most often used preconditioners for the incompressible Navier-Stokes equations are originally proposed and analysed for the constant viscosity case, c.f., the surveys [5, 8] and the books [1, 12, 30]. Some preconditioners can be straightforwardly utilized for the variable viscosity applications due to their algebraic constructions. In this paper we choose the augmented Lagrangian preconditioner for the Oseen-type problem (Section 3) and the block lower-triangular and the SIMPLER preconditioners for the Stokes-type problem (Section 4). On the other hand, the variation of viscosity is an important factor effecting the efficiency of those preconditioners, and in this case their robustness with respect to a variable viscosity is a crucial objective for the fast and reliable preconditioners. In order to fully achieve this objective, we modify the above mentioned preconditioners and also propose some computational improvements. The comparison between the targeted preconditioners and the efficiency of the Oseen- and Stokes-type problems are illustrated in Section 5. Conclusions and future work are outlined in Section 6.

2 Problem formulation and linearization

For the weak formulation of the stationary Navier-Stokes equations (1), we define the approximate solution and test spaces for the velocity as

$$\begin{aligned}
\mathbf{H}^1_E &= \{\mathbf{u} \in \mathcal{H}^1(\Omega)^d | \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega_D\}, \\
\mathbf{H}^1_{E_0} &= \{\mathbf{v} \in \mathcal{H}^1(\Omega)^d | \mathbf{v} = \mathbf{0} \quad \text{on } \partial\Omega_D\}, \\
\mathcal{H}^1(\Omega)^d &= \{u_i : \Omega \to \mathbb{R}^d \mid u_i, \frac{\partial u_i}{\partial x_j} \in L_2(\Omega), i, j = 1, \cdots, d\},
\end{aligned}$$

and for the pressure as

$$L_2(\Omega) = \{p : \Omega \to \mathbb{R} \mid \int_\Omega p^2 < \infty\}.$$

Then the weak formulation reads as follows:
Find $\mathbf{u} \in \mathbf{H}^1_E$ and $\mathbf{p} \in L_2(\Omega)$ such that

$$\begin{aligned}
\int_\Omega 2\nu(D_{\rm II}(\mathbf{u}))\mathbf{D}\mathbf{u} : \mathbf{D}\mathbf{v} d\Omega + \int_\Omega (\mathbf{u} \cdot \nabla\mathbf{u})\mathbf{v} d\Omega - \int_\Omega p\nabla \cdot \mathbf{v} d\Omega &= \int_\Omega \mathbf{f}\mathbf{v} d\Omega, \\
\int_\Omega q\nabla \cdot \mathbf{u} d\Omega &= 0,
\end{aligned} \tag{2}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and all $q \in L_2(\Omega)$. The pressure is uniquely defined only up to a constant term. To make it unique, one usually imposes the additional constraint $\int_\Omega p \; d\Omega = 0$. We also assume that the discretization is done using a stable pair of FEM spaces, satisfying the LBB condition [12].

As mentioned, the nonlinearity of the considered problem is handled by some linearization methods. The two well-known and most often used methods are the Newton and Picard methods [12], briefly introduced below.

Let $(\mathbf{u}_0, p_0)$ be an initial guess and let $(\mathbf{u}_k, p_k)$ be the approximate solution at the kth nonlinear step. Then we update the velocity and the pressure on the $(k+1)$ step as $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}_k$, $p_{k+1} = p_k + \delta p_k$ for $k = 0, 1, \cdots$ until convergence, where $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ (provided $\mathbf{u}_k \in \mathbf{H}^1_E$ and $p_k \in L_2(\Omega)$). Substituting $\mathbf{u}_{k+1}$ and p_{k+1} into the weak formulation (2), the correction $(\delta\mathbf{u}_k, \delta p_k)$ should satisfy the following problem:
Find $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\begin{aligned}
\int_\Omega 2\nu(D_{\rm II}(\mathbf{u}_k))\mathbf{D}\delta\mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega &+ \int_\Omega 2\nu'(D_{\rm II}(\mathbf{u}_k))[\mathbf{D}\mathbf{u}_k : \mathbf{D}\delta\mathbf{u}_k][\mathbf{D}\mathbf{u}_k : \mathbf{D}\mathbf{v}] d\Omega \\
+ \int_\Omega (\mathbf{u}_k \cdot \nabla\delta\mathbf{u}_k) \cdot \mathbf{v} d\Omega + \int_\Omega (\delta\mathbf{u}_k \cdot \nabla\mathbf{u}_k) \cdot \mathbf{v} d\Omega &- \int_\Omega \delta p_k \; (\nabla \cdot \mathbf{v}) d\Omega = R_k \\
\int_\Omega q \; (\nabla \cdot \delta\mathbf{u}_k) d\Omega &= P_k,
\end{aligned} \tag{3}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and $q \in L_2(\Omega)$. The residual terms are obtained as

$$\begin{aligned} R_k = \int_\Omega \mathbf{f}\cdot\mathbf{v}d\Omega - \int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_k))\mathbf{D}\mathbf{u}_k : \mathbf{D}\mathbf{v}d\Omega - \int_\Omega (\mathbf{u}_k\cdot\nabla\mathbf{u}_k)\cdot\mathbf{v}d\Omega + \int_\Omega p_k\nabla\cdot\mathbf{v}d\Omega \\ P_k = -\int_\Omega q\ (\nabla\cdot\mathbf{u}_k)d\Omega. \end{aligned} \tag{4}$$

This procedure is refereed to as the Newton linearization method. In the regularized Bingham model, i.e., $\nu(D_{\mathrm{II}}(\mathbf{u})) = \nu_0 + \tau(D_{\mathrm{II}}(\mathbf{u})+\varepsilon^2)^{-\frac{1}{2}}$, the derivative $\nu'(D_{\mathrm{II}}(\mathbf{u}))$ in terms of $D_{\mathrm{II}}(\mathbf{u})$ is $\nu'(D_{\mathrm{II}}(\mathbf{u})) = -\frac{1}{2}\tau(D_{\mathrm{II}}(\mathbf{u})+\varepsilon^2)^{-\frac{3}{2}}$. More details on the Newton method can be found, for example, in [12, 17].

Picard linearization is obtained in a similar way as Newton method, except that the terms, i.e., $\int_\Omega(\delta\mathbf{u}_k\cdot\nabla\mathbf{u}_k)\cdot\mathbf{v}d\Omega$ and $\int_\Omega 2\nu'(D_{\mathrm{II}}(\mathbf{u}_k))[\mathbf{D}\mathbf{u}_k : \mathbf{D}\delta\mathbf{u}_k][\mathbf{D}\mathbf{u}_k : \mathbf{D}\mathbf{v}]d\Omega$ are dropped. Thus, the linear problem in Picard method reads as follows:
Find $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\begin{aligned} \int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_k))\mathbf{D}\delta\mathbf{u}_k : \mathbf{D}\mathbf{v}d\Omega + \int_\Omega(\mathbf{u}_k\cdot\nabla\delta\mathbf{u}_k)\cdot\mathbf{v}d\Omega - \int_\Omega \delta p_k\ (\nabla\cdot\mathbf{v})d\Omega = R_k \\ \int_\Omega q\ (\nabla\cdot\delta\mathbf{u}_k)d\Omega = P_k, \end{aligned} \tag{5}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and $q \in L_2(\Omega)$. Similarly, we update the approximations as $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}_k$ and $p_{k+1} = p_k + \delta p_k$ for $k = 0, 1, \cdots$ until convergence.

3 The variable viscosity Oseen-type problem

Let $\mathbf{X}^h_{E_0}$ and P^h be finite dimensional subspaces of $\mathbf{H}^1_{E_0}$ and $L_2(\Omega)$, and let $\{\vec{\varphi}_i\}_{1\le i\le n_u}$ be the nodal basis of $\mathbf{X}^h_{E_0}$ and $\{\phi_i\}_{1\le i\le n_p}$ be the nodal basis of P^h. According to the Galerkin framework, the discrete corrections of the velocity and the pressure are represented as

$$\delta\mathbf{u}_h = \sum_{i=1}^{n_u}\delta\mathbf{u}_i\vec{\varphi}_i, \quad \delta\mathbf{p}_h = \sum_{i=1}^{n_p}\delta p_i\phi_i,$$

where n_u and n_p are the total number of degrees of freedom for the velocity and the pressure. The linear systems arising in Newton and Picard linearizations are of the form

$$\begin{bmatrix} F & B^T \\ B & O \end{bmatrix}\begin{bmatrix} \delta\mathbf{u}_h \\ \delta\mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{F}\mathbf{x} = \mathbf{b}, \tag{6}$$

where the system matrix $\mathcal{F} = \begin{bmatrix} F & B^T \\ B & O \end{bmatrix}$ is nonsymmetric and of a two-by-two block form. The matrix $B \in \mathbb{R}^{n_p\times n_u}$ corresponds to the (negative) divergence operator and B^T corresponds to the gradient operator (e.g., [12]). Here we assume either that the discrete LBB condition is satisfied, otherwise some stabilization is applied, resulting in a nonzero $(2, 2)$ block. When comparing Newton and Picard linearization methods, the difference appears in the pivot block $F \in \mathbb{R}^{n_u \times n_u}$, which is of the form $F = A_\nu + \delta_1 \widehat{A}_\nu + N + \delta_2 \widehat{N}$. Newton method corresponds to $\delta_1 = \delta_2 = 1$, while Picard method corresponds to $\delta_1 = \delta_2 = 0$. Given the approximation $\mathbf{u}_h$, the entries of A_ν, $\widehat{A}_\nu$, N and $\widehat{N}$ are

$$\begin{aligned}
A_\nu \in \mathbb{R}^{n_u \times n_u}, \quad & [A_\nu]_{i,j} = \int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_h))\mathbf{D}\vec{\varphi}_i : \mathbf{D}\vec{\varphi}_j, \\
\widehat{A}_\nu \in \mathbb{R}^{n_u \times n_u}, \quad & [\widehat{A}_\nu]_{i,j} = \int_\Omega 2\nu'(D_{\mathrm{II}}(\mathbf{u}_h))[\mathbf{D}\mathbf{u}_h : \mathbf{D}\vec{\varphi}_j][\mathbf{D}\mathbf{u}_h : \mathbf{D}\vec{\varphi}_i], \\
N \in \mathbb{R}^{n_u \times n_u}, \quad & [N]_{i,j} = \int_\Omega (\mathbf{u}_h \cdot \nabla\vec{\varphi}_j)\vec{\varphi}_i, \\
\widehat{N} \in \mathbb{R}^{n_u \times n_u}, \quad & [\widehat{N}]_{i,j} = \int_\Omega (\vec{\varphi}_j \cdot \nabla\mathbf{u}_h)\vec{\varphi}_i.
\end{aligned} \tag{7}$$

In this paper the linear system (6) arising in Newton (3) or Picard method (5) is referred to as the Oseen-type problem with variable viscosity.

Computing the solutions of the linear systems in (6) is the kernel and most time-consuming part in the numerical simulations. Therefore, fast and reliable solution techniques are critical. As is well known, direct solution methods are highly robust with respect to both problem and discretization parameters, and are, therefore, a preferred choice in the numerical simulations performed by engineers and applied scientists. The limiting factors for the sparse direct solvers are most often the computer memory demands and the need to repeatedly factorize matrices, which are recomputed during the simulation process, as for instance, the Jacobians in nonlinear problems. For real industrial applications where the models are mostly in three space dimensions and result in very large scale linear systems of the type (6), rapidly convergent iterative methods, accelerated by a proper preconditioner become the methods of choice. In this work, we consider preconditioned Krylov subspace methods, see the books [1, 12, 30].

3.1 Preconditioning the variable viscosity Oseen-type problem

As already mentioned, the linear systems in (6) are of two-by-two block form, and how to precondition such systems have been intensively studied in the past decades. In this work we limit ourselves to preconditioners, based on approximate block factorizations of the coefficient matrix. The literature on this class of preconditioners is huge. We refer for more details to the articles [2, 3, 4, 20, 25], the surveys [5, 8, 9, 32] and the books [1, 12, 30], with numerous references therein. In general, the exact factorization of a two-by-two block matrix reads

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & O \\ A_{21} & S \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1}A_{12} \\ O & I_2 \end{bmatrix} = \begin{bmatrix} I_1 & O \\ A_{21}A_{11}^{-1} & I_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ O & S \end{bmatrix}, \tag{8}$$

where I_1 and I_2 are identity matrices of proper dimensions. The pivot block A_{11} is assumed to be nonsingular and $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ is the exact Schur complement matrix. In our case, $A_{11} = F$, $A_{12} = B^T$, $A_{21} = B$ and $A_{22} = O$. So, $S = -BF^{-1}B^T$.

As preconditioners for such matrices of two-by-two block form, block lower- or upper-triangular approximate factors are often used

$$\begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{S} \end{bmatrix}, \qquad \begin{bmatrix} \widetilde{A}_{11} & A_{12} \\ O & \widetilde{S} \end{bmatrix}. \tag{9}$$

Here the matrix $\widetilde{A}_{11}$ denotes some approximation of A_{11}, given either in an explicit form or implicitly defined via an inner iterative solution method with a proper stopping tolerance. The matrix $\widetilde{S}$ is some approximation of the exact Schur complement S.

The results in [3] show that the quality of the preconditioners in (9) can be improved by making a sufficient number of inner iterations when implicitly approximating A_{11} and by choosing a sufficiently accurate approximation $\widetilde{S}$. The most challenging task turns out to be the construction of numerically and computationally efficient approximations of the Schur complement, which is in general dense and it is not practical to form it explicitly.

For the two-by-two block system arising in the incompressible Navier-Stokes equations with constant viscosity, several state-of-art approximations of the Schur complement are proposed and analysed, c.f., [6, 7, 11, 13, 15, 18, 22, 26, 28, 29, 33]. Due to their algebraic construction, some of them can be straightforwardly used for the variable viscosity case. However, the variation of viscosity is an important factor and effects the efficiency of the available preconditioners. We choose the augmented Lagrangian (AL) method (see e.g., [2, 6, 7]) to check the impact of variations in viscosity, and to even improve further its performance.

Following the AL framework, we first algebraically transform the system (6) into an equivalent one as follows

$$\begin{bmatrix} F + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_h \\ \delta \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{f}} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{F}_\gamma \mathbf{x} = \widehat{\mathbf{b}}, \tag{10}$$

where $\widehat{\mathbf{f}} = \mathbf{f} + \gamma B^T W^{-1}\ \mathbf{g}$, and $\gamma > 0$ and W are suitable scalar and matrix parameters. Clearly, the transformed system (10) has the same solution as (6) for any value of γ and any nonsingular matrix W.

The equivalent system (10) is what we intend to solve and the AL-type preconditioner proposed for $\mathcal{F}_\gamma$ in (10) is of a block lower-triangular form

$$\mathcal{P}_\gamma = \begin{bmatrix} F + \gamma B^T W^{-1} B & 0 \\ B & -\frac{1}{\gamma} W \end{bmatrix}. \tag{11}$$

To distinguish from the modified AL preconditioner introduced later, the preconditioner $\mathcal{P}_\gamma$ is referred to as the ideal AL preconditioner. It can be seen that the exact Schur complement $S_{\mathcal{F}_\gamma} = -B(F + \gamma B^T W^{-1} B)^{-1} B^T$ of the transformed matrix $\mathcal{F}_\gamma$ is approximated by $-\frac{1}{\gamma} W$. We analyse the ideal AL preconditioner using the technique in [7, 15], for instance. Consider the following generalized eigenvalue problem

$$\mathcal{F}_\gamma \mathbf{v} = \lambda \mathcal{P}_\gamma \mathbf{v}. \tag{12}$$

We see that

$$\mathcal{P}_\gamma^{-1}\mathcal{F}_\gamma = \begin{bmatrix} I & (F+\gamma B^TW^{-1}B)^{-1}B^T \\ 0 & \gamma W^{-1}B(F+\gamma B^TW^{-1}B)^{-1}B^T \end{bmatrix}.$$

Thus, the eigenvalues λ in (12) are either equal to 1 (with multiplicity equal to the dimension of $F \in \mathbb{R}^{n_u\times n_u}$) or coincide with those of the matrix $\gamma W^{-1}B(F+\gamma B^TW^{-1}B)^{-1}B^T$. Applying Sherman-Morrison-Woodbury's formula to $(F+\gamma B^TW^{-1}B)^{-1}$, we have

$$\widetilde{Q} \equiv \gamma W^{-1}B(F+\gamma B^TW^{-1}B)^{-1}B^T = \gamma Q - \gamma Q(I+\gamma Q)^{-1}\gamma Q,$$

where $Q = W^{-1}BF^{-1}B^T$. The matrix $BF^{-1}B^T$ is the negative Schur complement of the original system matrix $\mathcal{F}$ in (6). We state the following theorem, which has been shown in [16] and is included here only for completeness.

Theorem 3.1 *Let $\mu = a + i\,b$ be an eigenvalue of $Q = W^{-1}BF^{-1}B^T$, λ be an eigenvalue of the eigenproblem (12) and δ be an eigenvalue of the matrix $\widetilde{Q} = \gamma W^{-1}B(F+\gamma B^TW^{-1}B)^{-1}B^T$. Then the following holds:*

(1) The matrices Q and $\widetilde{Q}$ have the same eigenvectors and the eigenvalues of $\widetilde{Q}$ are equal to

$$\delta = \frac{\gamma\mu}{1+\gamma\mu} = \frac{1}{1+\frac{1}{\gamma\mu}}. \tag{13}$$

(2) The eigenvalues λ equal

$$\lambda = \begin{cases} 1, & \text{with multiplicity } n_u \\ \delta. & \end{cases}$$

When $\gamma \to \infty$ all nonzero eigenvalues λ converge to 1.

(3) Assume that μ is bounded in a rectangular box, i.e., there exist constants $\{a_{\min}, a_{\max}, b_{\max}\}$, independent of the mesh size parameter h, such that

$$\left\{ \begin{array}{l} a_{\min} \le a \le a_{\max}, \\ \mid b \mid \le b_{\max}. \end{array} \right\}$$

Then λ is also bounded in a rectangular box with sizes, independent of h. Furthermore, there holds

$$\delta = 1 - \frac{1+\gamma\,a}{(1+\gamma\,a)^2+\gamma^2b^2} + i\frac{\gamma\,b}{(1+\gamma\,a)^2+\gamma^2b^2}. \tag{14}$$

For any $\gamma \ge 1$, and any value of a and b, we have

$$1 - \frac{1+\gamma|a|}{(1+\gamma\,a)^2+\gamma^2b^2} < \mathcal{R}(\delta) < 1 \text{ and } |\mathcal{I}(\delta)| = \frac{\gamma|b|}{(1+\gamma\,a)^2+\gamma^2b^2} < 1, \tag{15}$$

where $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ are the real and the imaginary part of a complex number.

As mentioned, the transformation (10) holds true for any nonsingular matrix W. In practice W is often chosen to be the pressure mass matrix M as in [6], or to be the identity matrix as in [2, 9]. For the Oseen-type problem with variable viscosity, it has been proved in [16] that with $W = M$ or even $\mathrm{diag}(M)$ (the diagonal of M), the eigenvalues μ of $Q = W^{-1}BF^{-1}B^T$ are bounded in a rectangular box with bounds independent of the mesh size. In [16] the viscosity is considered to be a function of space and time, and this type of variation of viscosity arises in multiphase flow problems. Although the viscosity is dependent of different parameters, Theorem 3.1 always holds true. Then, based on (3) in Theorem 3.1 the spectrum of the eigenvalue problem (12) is also bounded in a rectangular box and the bounds are independent of the mesh size. Thus, the ideal AL preconditioner with $W = M$ or $\mathrm{diag}(M)$ is fully robust with respect to the space discretization parameter.

For the variable viscosity Oseen-type problems arising in multiphase flow, numerical results in [16] show that the AL preconditioner with $W = \mathrm{diag}(M)$ is indeed independent of the mesh size as expected. However, robustness with respect to the variation of viscosity is not guaranteed. In order to also achieve this objective, it is natural to let W incorporate the "information" of the variable viscosity. Therefore, here we choose W as the scaled pressure mass matrix, i.e., $M_\nu = \{(M_\nu)_{i,j}\} \in \mathbb{R}^{n_p \times n_p}$ with $(M_\nu)_{i,j} = (\nu^{-1}\phi_i, \phi_j)$. This is one of the main contributions of this paper, related to the AL preconditioner utilised in the Bingham flow. Numerical experiments in Section 5 show that the AL preconditioner with $W = \mathrm{diag}(M_\nu)$ is rather robust with respect to the variation of viscosity and results in a much faster convergence than $W = \mathrm{diag}(M)$. Following the proof of Theorem 3.1 in [16], it is straightforward to conclude that the AL preconditioner with $W = M_\nu$ or $W = \mathrm{diag}(M_\nu)$ is also independent of the mesh refinement.

Indeed, the convergence rate is effected by the ratio of the extremal values of the viscosity,

$$\nu_{min} = \inf_{\Omega} \nu(\cdot), \quad \nu_{max} = \sup_{\Omega} \nu(\cdot).$$

Especially, in the regularized Bingham model, we have $\nu_{\min} = O(1)$ and $\nu_{\max} = O(\varepsilon^{-1})$ [14]. The robustness of the AL preconditioner with respect to the parameter ε is not theoretically explored in this paper, and is considered as a direction for future research. Instead, in Section 5 we provide thorough numerical experiments and illustrate that the AL preconditioner with $W = \mathrm{diag}(M_\nu)$ is robust with respect to the parameters ε, τ in the Bingham model and, as predicted, is fully independent of the mesh size.

The second parameter in the AL scheme is the scalar γ. As pointed out in Theorem 3.1, with $\gamma \to \infty$ and for any nonsingular matrix W, all the eigenvalues of the preconditioned matrix $\mathcal{P}_\gamma^{-1}\mathcal{F}_\gamma$ cluster at one. This result means that for large values of γ and provided that we solve the sub-systems with the modified pivot block $F_\gamma = F + \gamma B^T W^{-1} B$ accurately enough, the AL preconditioner ensures a very fast convergence, within a few iterations. However, with increasing γ the modified pivot block F_γ becomes increasingly ill-conditioned and computing solutions of systems with F_γ becomes more and more difficult. In earlier related publications (c.f., [7]), relatively small values of γ have been used since in that case one can use known methods that are efficient when solving systems with F. The choice of γ is discussed in more details at the end of this section.

Although the matrices F and B are sparse, the modified pivot block F_γ is in general much denser. Furthermore, F_γ contains discretizations of mixed derivatives, and F_γ is not block-diagonal. Besides, the mixed derivatives bring additional difficulties for the numerical solution methods. How to efficiently solve systems with F_γ in the AL framework is in general still an open question and more research efforts need to be invested here. In this work, we utilize the approach proposed in [7] and illustrate it for a problem in two space dimensions. In 2D, F is of the form $F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$, where each block is square and of order $n_u/2$. Denoting $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$, we have

$$\begin{aligned} F_\gamma &= F + \gamma B^T W^{-1} B \\ &= \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} + \gamma \begin{bmatrix} B_1^T \\ B_2^T \end{bmatrix} W^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} \\ &= \begin{bmatrix} F_{11} + \gamma B_1^T W^{-1} B_1 & F_{12} + \gamma B_1^T W^{-1} B_2 \\ F_{21} + \gamma B_2^T W^{-1} B_1 & F_{22} + \gamma B_2^T W^{-1} B_2 \end{bmatrix} \\ &:= \begin{bmatrix} F_{\gamma,11} & F_{\gamma,12} \\ F_{\gamma,21} & F_{\gamma,22} \end{bmatrix}. \end{aligned}$$

A possible approach is to approximate F_γ by a block lower-triangular matrix

$$\widetilde{F}_\gamma = \begin{bmatrix} \widetilde{F}_{\gamma,11} & O \\ F_{\gamma,21} & \widetilde{F}_{\gamma,22} \end{bmatrix},$$

and replacing F_γ by $\widetilde{F}_\gamma$ in the ideal AL preconditioner (11) we obtain the modified AL preconditioner as follows

$$\widetilde{\mathcal{P}}_\gamma = \begin{bmatrix} \widetilde{F}_\gamma & O \\ B & -\frac{1}{\gamma} W \end{bmatrix} = \begin{bmatrix} \widetilde{F}_{\gamma,11} & O & O \\ F_{\gamma,21} & \widetilde{F}_{\gamma,22} & O \\ B_1 & B_2 & -\frac{1}{\gamma} W \end{bmatrix}, \tag{16}$$

where the terms $\widetilde{F}_{\gamma,11}$ and $\widetilde{F}_{\gamma,22}$ denote approximations of $F_{\gamma,11}$ and $F_{\gamma,22}$, for instance, obtained via an inner iterative solution method with a proper stopping tolerance.

The modified AL preconditioner offers two main advantages compared to the ideal one. When solving systems with $\widetilde{F}_\gamma$ one needs to solve two sub-systems with $F_{\gamma,11}$ and $F_{\gamma,22}$. In this way, the size of the linear system to be solved is reduced. Besides, as already mentioned, there are approximations of mixed derivatives in F_γ, i.e., $F_{\gamma,21}$ and $F_{\gamma,12}$. This can be an obstacle when applying known solution techniques, such as algebraic multigrid (AMG) methods. Here we use AMG as a block solver and the details are presented in Section 5. A comparison between applying the multigrid solver for the whole block F_γ and for the sub-blocks $F_{\gamma,11}$, $F_{\gamma,22}$ shows that the modified AL preconditioner is superior to the ideal AL preconditioner in terms of overall CPU time. Details are described in Section 5.

For the case of constant viscosity, attempts to determine the optimal γ are found in [7]. Although some theory has been derived in [7], the optimal value turns out to be problem dependent and expensive to calculate. Therefore, $\gamma = 1$ has been used in the numerical tests in many studies, for example [6, 15]. In [16] the good properties of that choice are justified. For non-Newtonian flows, we studied the effect of γ on the behaviour of the solver numerically. Results, not included here, show that a minimal number of iterations is obtained by choosing the value of γ to be 1. Therefore, for all numerical experiments in this paper, $\gamma = 1$.

Up to the knowledge of the authors, so far variable viscosity in considered only for Stokes-type problems, see [14]. For comparison reasons we include here the approximate formulations of (3) and (5) that lead to Stokes-type problem.

4 The variable viscosity Stokes-type problem

As given in Section 2, at each nonlinear step the updates $(\delta\mathbf{u}_k, \delta p_k)$ are computed by solving the linear problem (3) via Newton method or the problem (5) via Picard method. At the $(k+1)$ step, the velocity and the pressure are corrected as $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}_k$ and $p_{k+1} = p_k + \delta p_k$. The above process continues until convergence. Since $\mathbf{u}_{k+1}$ and p_{k+1} are approximate solutions, when computing the updates $(\delta\mathbf{u}_k, \delta p_k)$, we can even drop the linearization terms from the convection terms in (3) and (5). Then, the linear problem in Newton method reads as follows:
Find $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\begin{aligned}\int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_k))\mathbf{D}\delta\mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega + \int_\Omega 2\nu'(D_{\mathrm{II}}(\mathbf{u}_k))[\mathbf{D}\mathbf{u}_k : \mathbf{D}\delta\mathbf{u}_k][\mathbf{D}\mathbf{u}_k : \mathbf{D}\mathbf{v}] d\Omega& \\ - \int_\Omega \delta p_k \ (\nabla\cdot\mathbf{v}) d\Omega = R_k& \\ \int_\Omega q \ (\nabla\cdot\delta\mathbf{u}_k) d\Omega = P_k.&\end{aligned} \tag{17}$$

The linear problem for the Picard method reads as follows:
Find $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\begin{aligned}\int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_k))\mathbf{D}\delta\mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega - \int_\Omega \delta p_k \ (\nabla\cdot\mathbf{v}) d\Omega = R_k& \\ \int_\Omega q \ (\nabla\cdot\delta\mathbf{u}_k) d\Omega = P_k,&\end{aligned} \tag{18}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and $q \in L_2(\Omega)$. The residuals R_k, P_k are the same as given in (4), i.e.,

$$\begin{aligned}R_k = \int_\Omega \mathbf{f}\cdot\mathbf{v} d\Omega - \int_\Omega 2\nu(D_{\mathrm{II}}(\mathbf{u}_k))\mathbf{D}\mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega - \int_\Omega (\mathbf{u}_k\cdot\nabla\mathbf{u}_k)\cdot\mathbf{v} d\Omega + \int_\Omega p_k \nabla\cdot\mathbf{v} d\Omega& \\ P_k = -\int_\Omega q \ (\nabla\cdot\mathbf{u}_k) d\Omega.&\end{aligned}$$

In this way, we see that the above iterative procedure involves the convection term in the right-hand side vector only. If the norm of the residuals R_k and P_k is smaller than the stopping tolerance, we can guarantee that the corresponding solutions satisfy the weak formulation (2).

After discretization with stable FEM pairs, the linear systems arising in (17) and (18) are of the form

$$\begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_h \\ \delta \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{A}\mathbf{x} = \mathbf{b}, \tag{19}$$

where the pivot block A is symmetric positive definite (spd) and of the form $A = A_\nu + \delta_1 \widetilde{A}_\nu$. The terms A_ν and $\widetilde{A}_\nu$ are the same as in (7). Newton method corresponds to $\delta_1 = 1$ and Picard method – $\delta_1 = 0$. Although the pivot block A is spd, the coefficient matrix $\mathcal{A}$ is symmetric but indefinite.

4.1 Preconditioning the variable viscosity Stokes-type problem

We refer to the problems (17) and (18) or their representations in matrix form (19) as the variable viscosity Stokes-type problems. Compared to the Oseen-type problem, the benefits of solving the Stokes-type problem are that due to the absence of the convection term, the coefficient matrix $\mathcal{A}$ is symmetric and some numerically cheaper Krylov subspace method can be used, such as the minimal residual method (`MINRES`). Besides, efficient preconditioners of the Schur complement for the Stokes-type problems are easier to construct. We test two preconditioners for the Stokes-type problem–the block lower-triangular preconditioner and the SIMPLER preconditioner.

The block lower-triangular preconditioner is of the form $\mathcal{P}_L = \begin{bmatrix} \widetilde{A} & O \\ B & \widetilde{S} \end{bmatrix}$, where $\widetilde{A}$ denotes an approximation of the pivot block A, and the term $\widetilde{S}$ denotes an approximation of the exact Schur complement $S = -BA^{-1}B^T$. The Stokes problem arising in the incompressible Newtonian flows has been studied rather well and efficient approximations of S are well-known. For example, the pressure mass matrix M is a very efficient and numerically cheap such approximation, see [22]. For the variable viscosity Stokes-type problem (19), in [14] a scaled pressure mass matrix M_ν is originally proposed as an approximation of the Schur complement. The definition of M_ν is presented above, and as analysed in [14] the scaled pressure mass matrix M_ν leads to a much better approximation of the Schur complement than the pressure mass matrix. In this work $\widetilde{S}$ is taken to be the diagonal of M_ν, i.e., $\widetilde{S} = -\text{diag}(M_\nu)$.

In [14] at each preconditioning step the sub-systems with the pivot block A are solved by a direct method, which is clearly not suitable for large scale simulations in terms of CPU time and memory requirements. Here we suggest a computational improvement, based on the strategy of constructing the modified AL preconditioner, as described in (16), namely, in the two dimensional case, the pivot block $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is approximated by

$\widetilde{A} = \begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{A}_{22} \end{bmatrix}$. The terms $\widetilde{A}_{11}$ and $\widetilde{A}_{22}$ denote approximations of A_{11} and A_{22} obtained by an inner iterative solution method with a proper stopping tolerance. In summary, the block lower-triangular preconditioner for the Stokes-type problem with variable viscosity is

$$\mathcal{P}_{Stokes} = \begin{bmatrix} \widetilde{A} & O \\ B & \widetilde{S} \end{bmatrix} = \begin{bmatrix} \widetilde{A}_{11} & O & O \\ A_{21} & \widetilde{A}_{22} & O \\ B_1 & B_2 & -\mathrm{diag}(M_\nu) \end{bmatrix}. \tag{20}$$

SIMPLE (semi-implicit pressure linked equation) is used by Patanker [27] as an iterative method to solve the Navier-Stokes problem. The scheme belongs to the class of basic iterative methods and exhibits slow convergence. In [18, 33] SIMPLE and its variant SIMPLER are used as preconditioners in a Krylov subspace method to solve the incompressible Navier-Stokes equations, achieving in this way, a much faster convergence. SIMPLE and SIMPLER rely on an approximate block-factorization of saddle point matrices and due to their simplicity, remain attractive preconditioning techniques. We briefly describe both formulations for the Stokes matrix $\mathcal{A}$ in (19).

The SIMPLE preconditioner $\mathcal{P}_{SIMPLE}$ reads:

$$\mathcal{P}_{SIMPLE} = \begin{bmatrix} A & O \\ B & \widetilde{S} \end{bmatrix} \begin{bmatrix} I_1 & D^{-1}B^T \\ O & I_2 \end{bmatrix},$$

where D is the diagonal of the block A and $\widetilde{S} = -BD^{-1}B^T$. Solutions of systems with $\mathcal{P}_{SIMPLE}$ are straightforward, see Algorithm 4.1.

Algorithm 4.1 (Algorithm SIMPLE) *Given* $\mathbf{y} = [\boldsymbol{y}_u; \boldsymbol{y}_p]$, $\mathbf{x} = \mathcal{P}_{SIMPLE}^{-1}\mathbf{y}$ *is found within the following steps.*

```
Step 1: Solve $A\boldsymbol{x}_u^\star = \boldsymbol{y}_u$
Step 2: Solve $\widetilde{S}\boldsymbol{x}_p = \boldsymbol{y}_p - B\boldsymbol{x}_u^\star$
Step 3: Compute $\boldsymbol{x}_u = \boldsymbol{x}_u^\star - D^{-1}B^T\boldsymbol{x}_p$
Step 4: Set $\boldsymbol{x} = [\boldsymbol{x}_u; \boldsymbol{x}_p]$
```

SIMPLER differs slightly from SIMPLE. It includes a pressure prediction step, see Algorithm 4.2.

Algorithm 4.2 (Algorithm SIMPLER) *Given* $\mathbf{y} = [\boldsymbol{y}_u; \boldsymbol{y}_p]$, $\mathbf{x} = \mathcal{P}_{SIMPLER}^{-1}\mathbf{y}$ *is found within the following steps.*

```
Step 0: Solve $\widetilde{S}\boldsymbol{x}_p^\star = \boldsymbol{y}_p - BD^{-1}\boldsymbol{y}_u$
Step 1: Solve $A\boldsymbol{x}_u^\star = \boldsymbol{y}_u - B^T\boldsymbol{x}_p^\star$
```

Step 2: Solve $\widetilde{S}\delta\boldsymbol{x}_p = \boldsymbol{y}_p - B\boldsymbol{x}_u^\star$

Step 3: Update $\boldsymbol{x}_p = \boldsymbol{x}_p^\star + \delta\boldsymbol{x}_p$ *and* $\boldsymbol{x}_u = \boldsymbol{x}_u^\star - D^{-1}B^T\delta\boldsymbol{x}_p$

Step 4: Set $\boldsymbol{x} = [\boldsymbol{x}_u; \boldsymbol{x}_p]$

We see, that when applying $\mathcal{P}_{SIMPLER}$, two solutions with $\widetilde{S}$ and one solution with A are required. Based on earlier experience, we modify $\mathcal{P}_{SIMPLER}$ as follows:

(i) We choose $\widetilde{S} = -\text{diag}(M_\nu)$. The choice is motivated first by the observation that M_ν is a good approximation of $BA^{-1}B^T$ and second, that solutions with a diagonal matrix are cheap and trivially implemented.

(ii) In Step 1, instead of solving systems with A, we approximate A as $\widetilde{A} = \begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{A}_{22} \end{bmatrix}$, where $\widetilde{A}_{11}$ and $\widetilde{A}_{22}$ indicate that we use an inner iterative solver with a proper stopping tolerance for the blocks A_{11} and A_{22}.

In this way the computational complexities of applying $\mathcal{P}_{SIMPLER}$ and $\mathcal{P}_{Stokes}$ become nearly the same.

5 Numerical illustrations

We choose as a benchmark the well-known two-dimensional lid-driven cavity problem, equipped with the boundary conditions $u_1 = u_2 = 0$ for $x = 0, x = 1$ and $y = 0$; $u_1 = 1, u_2 = 0$ for $y = 1$. The problem is discretized using a uniform Cartesian mesh and the Q2-Q1 finite element pair. In this paper we consider the regularized Bingham model, i.e., $\nu(D_{\text{II}}(\mathbf{u})) = \nu_0 + \tau(D_{\text{II}}(\mathbf{u}) + \varepsilon^2)^{-\frac{1}{2}}$. We fix $\nu_0 = 1$ and vary the regularization parameter ε and the coefficient τ, as $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, and $\tau = 1,\ 2.5$.

As already mentioned, two nonlinear solution methods are utilized - Picard and Newton. The nonlinear iterations are terminated when the norm of the relative residual $\|[R_k; P_k]\|/\|[R_0; P_0]\|$ is decreased by six orders of magnitude, where (R_k, P_k) is defined in (4).

Solutions with the linearized problems, i.e., $\mathcal{F}_\gamma$ in (10) and $\mathcal{A}$ in (19), are solved by a preconditioned iterative method, in this case the generalized conjugate residual method (GCR) [1] as it allows the use of variable preconditioners. The stopping tolerance for GCR is also relative and is denoted by $\epsilon_{\texttt{GCR}}$.

The preconditioner for the Oseen-type problem is $\widetilde{\mathcal{P}}_\gamma$, defined in (11). Here, $\gamma = 1$ and W is chosen as $W = \text{diag}(M_\nu)$ or $W = \text{diag}(M)$. The preconditioner for the Stokes-type problem is either $\mathcal{P}_{Stokes}$ in (20) or $\mathcal{P}_{SIMPLER}$ in Algorithm SIMPLER. When applying those preconditioners, we need to solve systems with the sub-blocks $F_{\gamma,11}$, $F_{\gamma,22}$ and A_{11} and A_{22}, respectively. This is done either by a direct method or by an algebraic multigrid method, namely, agmg (see [23, 21, 24]). For agmg, the relative stopping tolerance is denoted by $\epsilon_{\texttt{agmg}}$.

The implementation of agmg is in Fortran and Matlab interface is provided. Therefore, its performance in terms of CPU time is comparable with that of the sparse direct solver in Matlab - the 'backslash' operator. For nonsymmetric matrices agmg uses the GCR method accelerated by an algebraic multigrid preconditioner, based on aggregation techniques.

The presented GCR iterations are averaged over the total number of nonlinear iterations. All the results in this paper are carried out in Matlab 7.13 (R2011b), and performed on a Linux-64 platform with 4 Intel(R) Core i5 CPUs, 660@3.33GHz. The reported execution time is in seconds. Whenever agmg is used, the setup time is included in the reported time figures.

Tables 1-6 present results for the Oseen-type problem and Tables 7-9 for the Stokes-type problem.

In Tables 1-2 we choose $\epsilon_{\texttt{agmg}} = \epsilon_{\texttt{GCR}} = 10^{-6}$. Such a small tolerance is not necessary in practice, but here it is used just for the analysis reason. Table 1 shows the performance of the modified AL preconditioner $\widetilde{\mathcal{P}}_\gamma$ for the two choices of W. The sub-blocks $F_{\gamma,11}$, $F_{\gamma,22}$ in $\widetilde{\mathcal{P}}_\gamma$ are solved by agmg. We see from Table 1 that for the choice of $W = \text{diag}(M_\nu)$ the linear GCR iterations are very robust with respect to h, ε and τ. Therefore, in the rest of the experiments we fix $W = \text{diag}(M_\nu)$.

Table 2 shows a comparison between the ideal and the modified AL preconditions, i.e., $\mathcal{P}_\gamma$ and $\widetilde{\mathcal{P}}_\gamma$. The sub-blocks $F_{\gamma,11}$, $F_{\gamma,22}$ in $\widetilde{\mathcal{P}}_\gamma$ and F_γ in $\mathcal{P}_\gamma$ are solved by agmg. We see from Table 2 that using $\widetilde{\mathcal{P}}_\gamma$ leads to a slight increase in the number of linear iterations, however, the gain in total solution time is substantial. We note also that the performance of agmg, applied to F_γ turns out to be rather sensitive to the regularized parameter ε in terms of the iteration number. Its performance when applied to $F_{\gamma,11}$ and $F_{\gamma,22}$ is mainly independent of ε. This is illustrated in Table 3. As seen, the reported CPU time for the agmg solver consists of two parts, i.e., the setup time and the solution time. The sub-blocks $F_{\gamma,11}$ and F_γ are obtained from the last Picard iteration.

Next, we present experiments with $\epsilon_{\texttt{agmg}} = \epsilon_{\texttt{GCR}} = 10^{-2}$ in Tables 4 and 5. Comparison between the results with $h = 1/64$ in Tables 2 and 4 shows that in this case almost no increase in the number of nonlinear iterations is observed, however, the total execution time is reduced by a factor 10. Results in Tables 4 and 5 show that the modified AL preconditioner $\widetilde{\mathcal{P}}_\gamma$ is robust with respect to the parameters τ, ε, the mesh size and different linearization methods in terms of the linear GCR iteration number. The nonlinear iterations and the total solution time are reduced by a factor around 2 when using Newton linearization method.

Table 6 illustrates the performance of agmg itself. We set the parameter $\varepsilon = 10^{-4}$ to simulate a more difficult scenario. In the regularized Bingham model the extreme values of the viscosity are $\nu_{\min} = O(1)$ and $\nu_{\max} = O(\varepsilon^{-1})$. A smaller value of ε results in a larger variation in viscosity. We see that the agmg solver is fully independent of the mesh size, the parameter τ and the different linearization methods. Also, we compare it with the 'backslash' direct sparse solver in Matlab. For the problem sizes we test, agmg already shows its superiority, and this superiority will be increasingly stronger for larger problem sizes.

Next, in Tables 7-9 we illustrate the overall performance of the nonlinear solver when the original discrete problem is written in Stokes form. The efficiency of $\mathcal{P}_{\text{Stokes}}$ and $\mathcal{P}_{SIMPLER}$ with $\epsilon_{\texttt{agmg}} = \epsilon_{\texttt{GCR}} = 10^{-2}$ are shown in Tables 7 and 8.

As seen, both preconditioners are quite robust with respect to the mesh size, the parameters ε, τ in the Bingham model, and the different linearization methods in terms of the linear iterations. The total solution time by using $\mathcal{P}_{\text{Stokes}}$ preconditioner is slightly smaller than $\mathcal{P}_{SIMPLER}$ in both Picard and Newton linearizations. The same as in Oseen-type problem, Newton method reduces the nonlinear iterations and the overall computational time by a factor 2, compared to Picard method.

The efficiency of `agmg` for A_{11} is presented in Table 9. In the Stokes-type problem, since the sub-blocks A_{11} and A_{22} are `spd`, `agmg` uses the conjugate gradient (`CG`) Krylov subspace method accelerated by the multigrid preconditioner. Here, the `agmg` solver is also fully independent of the mesh size, the parameter τ and the different linearization methods. The superiority to the direct method is exhibited too.

The comparison between the behaviour of the numerical solution methods for the Oseen and Stokes formulations (Table 4-5 and 7-8) shows that in both cases the nonlinear iterations are the same. The explanation for this effect is that for the considered non-Newtonian flow with the Bingham model, the diffusion is dominant and for the linearization the convection term can be moved to the right-hand side vector.

For the two problems, the average linear iterations are almost the same but the overall computational time for the Stokes-type problem is only half of that for the Oseen-type problem in both Newton and Picard methods. Further, the sub-blocks A_{11} and A_{22} are `spd` and `agmg` uses the `CG` method, which is numerically cheaper than the `GCR` method used for $F_{\gamma,11}$ and $F_{\gamma,22}$. Also, the blocks A_{11} and A_{22} are sparser than $F_{\gamma,11}$ and $F_{\gamma,22}$, and the sparsity is another reason making `agmg` to work more efficiently for the Stokes formulation. Comparing the results in Table 6 and 9, we see that the overall computational time of `agmg` for the Stokes-type problem is reduced about three times, compared to that for the Oseen-type problem.

Finally, we include some plots of the numerically computed solution. Determining the rigid regions of the viscoplastic flow, formally regions where $D_{\text{II}}(\mathbf{u}) = 0$, is the most challenging task from modeling point of view. However, when a regularized model is used the condition $D_{\text{II}}(\mathbf{u}) = 0$ does not hold exactly. In practice one needs to choose the regularization parameter ε as small as possible. On the other hand, small values of the regularization parameter ε lead to more computational work, see the nonlinear iterations and the total solution time in the previous tables. To give an insight regarding reasonable values of ε which can well predict the rigid regions, Figures 1-2 show the computed isolines of $(D_{\text{II}}(\mathbf{u}))^{\frac{1}{2}}$ for $\varepsilon \in \{10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}\}$. These figures appear to be nearly identical with those in [10, 14] and we see that for $\varepsilon \approx 10^{-4}$ the computed results give a fairly good prediction of the rigid regions. The relatively large values, i.e., $\varepsilon \geq 10^{-3}$ are not enough to recover the viscoplastic properties.

6 Conclusions and future work

In this paper we consider fast and reliable numerical solution methods for the incompressible non-Newtonian Navier-Stokes equations. Among the several non-Newtonian fluid models here we limit ourselves to the regularized Bingham model. When linearizing the governing nonlinear equations, Oseen- or Stokes-type problems arise. In both cases, the coefficient matrices are of a two-by-two block form. Numerically and computationally efficient preconditioners for the so-arising systems are the main concern in this paper. Various preconditioners are analysed, namely, the modified augmented Lagrangian preconditioner for the Oseen formulation and the block lower-triangular and the SIMPLER preconditioners for the Stokes formulation. Numerical experiments show that all the preconditioners are fully independent of the mesh size, and are rather robust with respect to the parameters in the Bingham model.

Due to their algebraic construction, the tested preconditioning techniques are straightforwardly applicable for other non-Newtonian fluid models. A detailed study on their performance in other applications is subject to a future research. Besides, how to further accelerate the convergence of the nonlinear iterations needs separate attention. One possible way is to combine the Newton and Picard methods, where a few Picard iterations can be performed to obtain a better initial guess for Newton method.

References

[1] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press: Cambridge, 1994.

[2] Axelsson O, Blaheta R. Preconditioning of matrices partitioned in two-by-two block form: Eigenvalue estimates and Schwarz DD for mixed FEM. *Numerical Linear Algebra with Applications* 2010; **17**(5):787–810. DOI: 10.1002/nla.728.

[3] Axelsson O, Neytcheva M. A general approach to analyse preconditioners for two-by-two block matrices. *Numerical Linear Algebra with Applications* 2011; Article first published online: 14 DEC 2011. DOI: 10.1002/nla.830.

[4] Axelsson O, Neytcheva M. Eigenvalue estimates for preconditioned saddle point matrices. *Numerical Linear Algebra with Applications* 2006; **13**(4):339-360. DOI: 10.1002/nla.469.

[5] Benzi M. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics* 2002; **182**(2):418-477. DOI: 10.1006/jcph.2002.7176.

[6] Benzi M, Olshanskii MA. An augmented Lagrangian-based approach to the Oseen problem. *SIAM Journal on Scientific Computing* 2006; **28**(6):2095-2113. DOI: 10.1137/050646421.

[7] Benzi M, Olshanskii MA, Wang Z. Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods In Fluids* 2011; **66**(4):486-508. DOI: 10.1002/fld.2267.

[8] Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1-137. DOI: 10.1017/S0962492904000212.

[9] de Niet AC, Wubs FW. Two preconditioners for saddle point problems in fluid flows. *International Journal for Numerical Methods in Fluids* 2007; **54**(4):355-377. DOI: 10.1002/fld.1401.

[10] Dean EJ, Glowinski R, Guidoboni G. On the numerical simulation of Bingham viscoplastic flow: old and new results. *Journal of Non-Newtonian Fluid Mechanics* 2007; **142**:36-62. DOI: 10.1016/j.jnnfm.2006.09.002.

[11] Elman HC, Silvester DJ. Fast nonsymmetric iterations and preconditioning for the Navier-Stikes equations. *SIAM Journal on Scientific Computing* 1996; **17**(1):33-46. DOI: 10.1137/0917004.

[12] Elman HC, Silvester DJ, Wathen AJ. *Finite Element and Fast Iterative Solvers: With Application in Incompressible Fluid Dynamics*. Oxford Series in Numerical Mathematics and Scientific Computation. Oxford University Press: Oxford, 2005.

[13] Elman HC, Silvester DJ, Wathen AJ. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numerische. Mathematik* 2002; **90**:665-688. DOI: 10.1007/S002110100300.

[14] Grinevich PP, Olshanskii MA. An iterative method for the Stokes-type problem with variable viscosity. *SIAM Journal on Scientific Computing* 2009; **31**(5):3959-3978. DOI: 10.1137/08744803.

[15] He X, Neytcheva M, Capizzano SS. On an Augmented Lagrangian-Based Preconditioning of Oseen Type Problems. *BIT Numerical Mathematics* 2011; **51**(4):865-888. DOI: 10.1007/s10543-011-0334-4.

[16] He X, Neytcheva M. Preconditioning the incompressible Navier-Stokes equations with variable viscosity. *Journal of Computational Mathematics* 2012; **30**:461-482. DOI:

[17] Hron J, Quazzi A, Turek S. A computational comparison of two FEM solvers for nonlinear incompressible flow. *Lecture Notes in Computational Science and Engineering* 2003; **35**:87–109. DOI: 10.1007/978-3-642-19014-8-5.

[18] Li C, Vuik C. Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow. *Numerical Linear Algebra with Applications* 2004; **11**(5-6):511-523. DOI: 10.1002/nla.358.

[19] Malék J, Nečas J, Rajagopal KR. Global existence of solutions for flows of fluids with pressure and shear dependent viscosities. *Applied Mathematics Letters* 2002; **15**(8):961-967.

[20] Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing* 2000; **21**(6):1969-1972. DOI: 10.1137/S1064827599355153.

[21] Napov A, Notay Y, An algebraic multigrid method with guaranteed convergence rate, textitSIAM Journal on Scientific Computing 2012; **34**(2):A1079-A1109. DOI: 10.1137/100818509.

[22] Neytcheva M, Do-Quang M, He X. Element-by-element Schur complement approximations for general nonsymmetric matrices of two-by-two block form. *Lecture Notes in Computer Science* 2009; **5910**:108-115.

[23] Notay Y. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis* 2010; **37**:123-146.

[24] Notay Y. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM Journal on Scientific Computing* 2012; **34**(4):A2288-A2316. DOI: 10.1137/110835347.

[25] Notay Y. A new analysis of block preconditioners for saddle point problems. *Report GANMN 13-01*, Universite Libre de Bruxelles, Brussels, Belgium, 2013.

[26] Olshanskii MA, Vassilevski YV. Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM Journal on Scientific Computing* 2007; **29**(6):2686-2704. DOI: 10.1137/0706679776.

[27] S.V. Patankar. *Numerical heat transfer and fluid flow.* McGraw-Hill: New York, 1980.

[28] ur Rehman M, Geenen T, Vuik C, Segal G, MacLachlan SP. On iterative methods for the incompressible Stokes problem. *International Journal for Numerical Methods in Fluids* 2011; **65**(10):1180-1200. DOI: 10.1002/fld.2235.

[29] T. Rusten and R. Winther. A preconditioned iterative method for saddle point problems. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**(3):887-904. DOI: 10.1137/0613054.

[30] Saad Y. *Iterative Methods for Sparse Linear Systems* (2nd edn). SIAM: Philadelphia, PA, 2003.

[31] D. Schaeffer. Instability in the evolution equations describing incompressible granular flow. *Journal of Dynamics and Differential Equations* 1987; **66**:19-50.

[32] Segal A, ur Rehman M, Vuik C. Preconditioners for the incompressible Navier-Stokes equations. *Numerical Mathematics: Theory, Methods and Applications* 2010; **3**(3):245-275. DOI: 10.4208/nmtma.2010.33.1.

[33] Vuik C, Saghir A, Boerstoel GP. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *International Journal for Numerical Methods in Fluids* 2000; **33**:1027-1040.

Table 1: Oseen formulation with Picard method, comparison between two choices of W in $\widetilde{\mathcal{P}}_\gamma$.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-2}	10^{-3}	10^{-4}
$h = 1/32$								
Picard iter.	17	74	161	197	30	105	198	193
`GCR` iter. with $W = \mathrm{diag}(M_\nu)$	22	22	24	26	24	25	27	35
`GCR` iter. with $W = \mathrm{diag}(M)$	46	124	338	> 500	64	167	418	> 500
$h = 1/64$								
Picard iter.	16	86	230	264	28	135	299	543
`GCR` iter. with $W = \mathrm{diag}(M_\nu)$	21	21	23	27	22	22	25	27
`GCR` iter. with $W = \mathrm{diag}(M)$	46	122	351	> 500	67	173	> 500	> 500

Table 2: Oseen formulation with Picard method, comparison between $\mathcal{P}_\gamma$ and $\widetilde{\mathcal{P}}_\gamma$, $h = 1/64$.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-2}	10^{-3}	10^{-4}
$\mathcal{P}_\gamma$								
Picard iter.	16	86	230	264	28	135	299	543
`GCR` iter.	18	19	19	19	18	19	19	16
Total solution time	81.63	452.20	1740.06	2405.62	115.29	714.34	2232.95	5656.66
$\widetilde{\mathcal{P}}_\gamma$								
Picard iter.	16	86	230	264	28	135	299	543
`GCR` iter.	21	21	23	27	22	22	25	27
Total solution time	20.41	120.91	371.06	680.08	41.36	211.75	576.97	1720.52

Table 3: Oseen formulation with Picard method, comparison of the `agmg` solver for systems with $F_{\gamma,11}$ and F_γ. $\epsilon_{\texttt{agmg}} = 10^{-6}$, $h = 1/64$, random right-hand side vector.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-2}	10^{-3}	10^{-4}
$F_{\gamma,11}$								
Setup time:	3.80e-2	4.00e-2	4.00e-2	4.00e-2	4.00e-2	3.80e-2	3.90e-2	3.80e-2
Solution time:	2.80e-2	2.80e-2	3.00e-2	4.40e-2	2.70e-2	2.80e-2	3.40e-2	4.50e-2
Total time:	6.60e-2	6.80e-2	7.00e-2	8.40e-2	6.70e-2	6.60e-2	7.30e-2	8.30e-2
`agmg` iter.:	11	11	11	16	11	11	13	17
F_γ								
Setup time:	1.99e-1	1.95e-1	2.11e-1	2.13e-1	1.93e-1	2.01e-1	2.01e-1	2.03e-1
Solution time:	2.27e-1	2.47e-1	3.69e-1	4.50e-1	2.02e-1	2.34e-1	3.51e-1	5.95e-1
Total time:	4.26e-1	4.42e-1	5.80e-1	6.63e-1	3.95e-1	4.35e-1	5.52e-1	7.98e-1
`agmg` iter.:	22	24	34	40	20	23	33	48

Table 4: Oseen formulation with Picard method, $\widetilde{\mathcal{P}}_\gamma$ as a preconditioner.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$h = 1/32$								
Picard iter.	17	73	160	197	30	105	198	193
GCR iter.	6	6	7	8	7	6	7	7
Total solution time	1.33	5.30	12.22	18.28	2.51	7.96	15.76	19.06
$h = 1/64$								
Picard iter.	16	86	229	264	28	134	299	542
GCR iter.	6	5	6	7	6	6	6	6
Total solution time	5.78	24.59	71.49	96.93	9.16	42.19	95.63	194.27
$h = 1/128$								
Picard iter.	14	94	297	548	26	157	426	580
GCR iter.	6	5	5	6	5	5	5	6
Total solution time	19.68	110.56	347.31	733.54	33.41	188.42	524.38	851.94
$h = 1/256$								
Picard iter.	12	101	365	844	23	171	554	1106
GCR iter.	6	4	5	5	5	5	5	6
Total solution time	88.67	527.58	1952.55	4679.06	147.61	978.13	3106.49	6814.22

Table 5: Oseen formulation with Newton method, $\widetilde{\mathcal{P}}_\gamma$ as a preconditioner.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$h = 1/32$								
Newton iter.	11	39	82	101	18	55	101	99
GCR iter.	6	6	7	8	8	6	7	8
Total solution time	1.04	3.42	7.68	10.03	1.96	4.99	9.53	11.52
$h = 1/64$								
Newton iter.	10	45	117	134	16	69	151	272
GCR iter.	6	6	6	7	6	6	7	6
Total solution time	4.94	18.29	48.16	59.67	7.92	30.48	64.60	117.26
$h = 1/128$								
Newton iter.	9	50	151	276	15	81	216	291
GCR iter.	6	5	6	6	5	6	6	6
Total solution time	12.58	65.53	215.11	421.35	21.67	111.06	316.21	496.87
$h = 1/256$								
Newton iter.	8	53	185	427	14	88	280	555
GCR iter.	6	5	5	6	6	5	6	6
Total solution time	50.73	300.98	1045.20	2675.65	90.17	616.86	2044.52	4423.35

Table 6: Oseen formulation, agmg performance for $F_{\gamma,11}$, $\epsilon_{\texttt{agmg}} = 10^{-6}$ and random right-hand side vectors.

	Picard linearization		Newton's linearization	
	$\tau = 1$	$\tau = 2.5$	$\tau = 1$	$\tau = 2.5$
$h = 1/32$				
Setup time:	1.00e-2	1.00e-2	1.20e-1	1.10e-2
Solution time:	1.00e-2	1.00e-2	1.10e-2	1.10e-2
Total time:	2.00e-2	2.00e-2	2.30e-2	2.20e-2
agmg iter.:	17	17	16	16
Direct solver time:	1.88e-2	1.90e-2	1.89e-2	1.82e-2
$h = 1/64$				
Setup time:	3.80e-2	4.00e-2	6.30e-2	5.60e-2
Solution time:	5.10e-2	4.90e-2	5.50e-2	6.70e-2
Total time:	8.90e-2	8.90e-2	1.18e-1	1.23e-1
agmg iter.:	17	16	19	19
Direct solver time:	1.35e-1	1.38e-1	1.14e-1	1.19e-1
$h = 1/128$				
Setup time:	1.85e-1	1.86e-1	1.90e-1	1.88e-1
Solution time:	2.13e-1	2.46e-1	2.45e-1	2.70e-1
Total time:	3.98e-1	4.32e-1	4.35e-1	4.58e-1
agmg iter.:	18	21	19	23
Direct solver time:	7.47e-1	7.47e-1	7.43e-1	7.47e-1
$h = 1/256$				
Setup time:	8.43e-1	8.51e-1	8.36e-1	1.01
Solution time:	6.49e-1	7.72e-1	7.77e-1	1.22
Total time:	1.49	1.62	1.61	2.23
agmg iter.:	13	15	15	17
Direct solver time:	4.85	4.86	4.89	5.01

Table 7: Stokes formulation with Picard method.

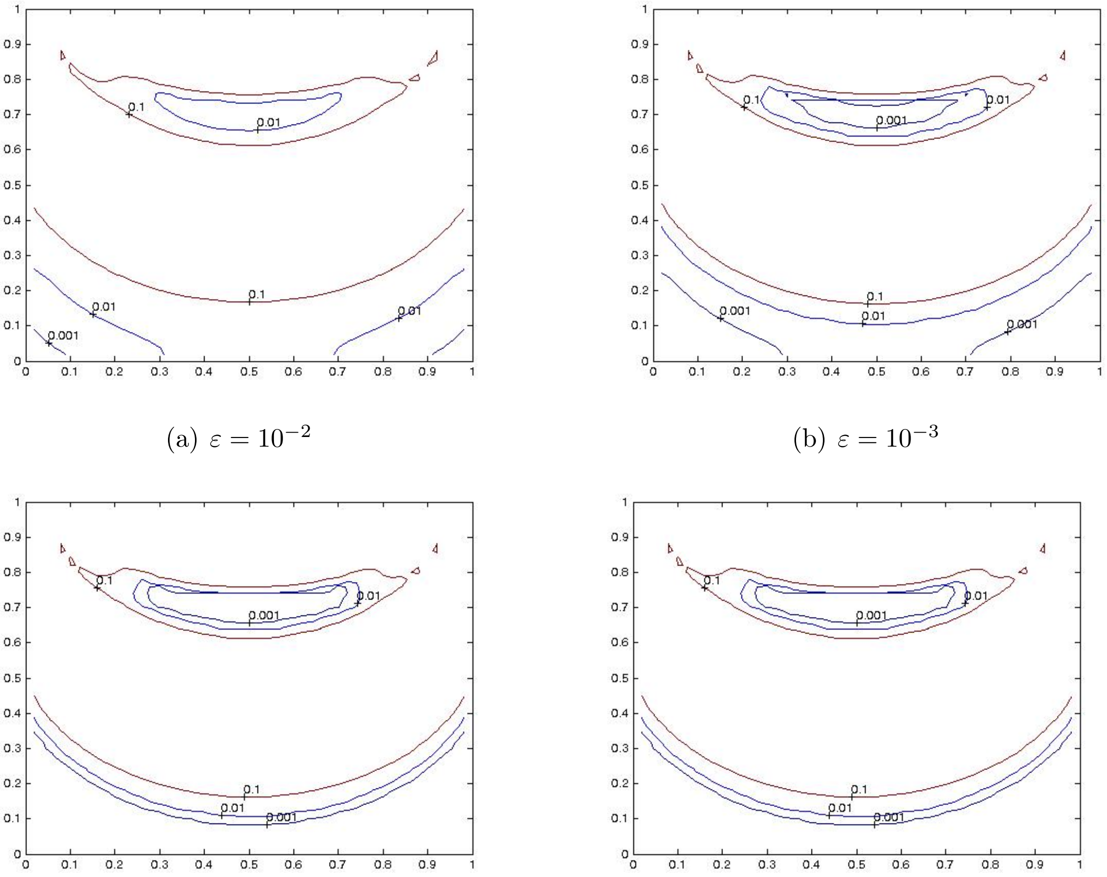
	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$h = 1/32$								
Picard iter.	17	73	161	197	30	105	198	194
GCR iter. with $\mathcal{P}_{Stokes}$.	7	8	9	10	9	9	9	10
Total solution time with $\mathcal{P}_{\text{Stokes}}$	0.75	3.79	8.75	13.55	1.53	5.68	11.44	15.67
GCR iter. with $\mathcal{P}_{SIMPLER}$.	7	8	8	10	9	9	10	11
Total solution time with $\mathcal{P}_{SIMPLER}$	0.79	3.94	9.20	15.63	1.76	6.43	13.29	17.72
$h = 1/64$								
Picard iter.	16	86	229	264	28	134	299	542
GCR iter. with $\mathcal{P}_{Stokes}$	7	7	8	9	7	8	9	7
Total solution time with $\mathcal{P}_{\text{Stokes}}$	2.37	12.83	42.74	59.98	4.15	22.03	61.48	102.00
GCR iter. with $\mathcal{P}_{SIMPLER}$.	8	8	9	11	8	9	11	11
Total solution time with $\mathcal{P}_{SIMPLER}$	2.89	15.94	53.05	75.26	5.11	26.99	82.07	129.81
$h = 1/128$								
Picard iter.	14	94	297	547	26	157	428	580
GCR iter. with $\mathcal{P}_{Stokes}$	7	6	7	8	6	7	8	8
Total solution time with $\mathcal{P}_{\text{Stokes}}$	6.98	44.12	170.85	364.46	13.35	82.54	277.31	431.85
GCR iter. with $\mathcal{P}_{SIMPLER}$.	8	8	8	8	8	9	9	11
Total solution time with $\mathcal{P}_{SIMPLER}$	10.23	62.77	222.17	440.94	16.86	120.42	373.07	595.74
$h = 1/256$								
Picard iter.	12	102	364	845	23	171	553	1109
GCR iter. with $\mathcal{P}_{Stokes}$	6	6	6	7	6	7	7	8
Total solution time with $\mathcal{P}_{\text{Stokes}}$	26.85	213.35	806.54	2169.58	49.98	404.64	1437.16	3335.93
GCR iter. with $\mathcal{P}_{SIMPLER}$.	7	7	8	8	7	8	8	10
Total solution time with $\mathcal{P}_{SIMPLER}$	31.76	281.14	1124.13	2846.15	68.22	588.82	1886.11	4441.81

Table 8: Stokes formulation with Newton method.

	$\tau = 1$				$\tau = 2.5$			
ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$h = 1/32$								
Newton iter.	11	39	83	101	18	55	101	99
GCR iter. with $\mathcal{P}_{Stokes}$	7	8	9	10	9	9	9	11
Total solution time with $\mathcal{P}_{Stokes}$	0.45	1.79	3.99	6.38	0.82	2.60	5.51	8.32
GCR iter. with $\mathcal{P}_{SIMPLER}$	7	8	8	10	9	9	10	11
Total solution time with $\mathcal{P}_{SIMPLER}$	0.49	2.01	4.32	7.28	1.08	3.28	6.33	9.37
$h = 1/64$								
Newton iter.	10	45	116	134	16	69	151	273
GCR iter. with $\mathcal{P}_{Stokes}$	7	7	8	9	8	8	9	8
Total solution time with $\mathcal{P}_{Stokes}$	1.62	7.45	21.96	31.59	2.68	12.16	31.55	55.29
GCR iter. with $\mathcal{P}_{SIMPLER}$	8	8	9	11	8	9	11	11
Total solution time with $\mathcal{P}_{SIMPLER}$	1.85	8.62	25.74	40.80	3.02	14.68	41.96	66.86
$h = 1/128$								
Newton iter.	9	50	151	275	15	81	216	291
GCR iter. with $\mathcal{P}_{Stokes}$	7	6	7	8	7	7	8	8
Total solution time with $\mathcal{P}_{Stokes}$	4.56	28.25	98.16	193.62	8.94	48.99	146.48	233.63
GCR iter. with $\mathcal{P}_{SIMPLER}$	8	8	9	9	7	8	9	11
Total solution time with $\mathcal{P}_{SIMPLER}$	6.18	36.10	126.07	239.86	10.07	66.68	189.47	313.52
$h = 1/256$								
Newton iter.	8	53	185	424	14	88	279	555
GCR iter. with $\mathcal{P}_{Stokes}$	7	6	7	7	7	7	7	8
Total solution time with $\mathcal{P}_{Stokes}$	20.35	131.73	486.04	1151.59	33.65	237.81	791.87	1761.28
GCR iter. with $\mathcal{P}_{SIMPLER}$	8	8	8	8	8	9	9	10
Total solution time with $\mathcal{P}_{SIMPLER}$	22.72	168.89	640.19	1544.96	46.20	329.62	1029.81	2377.25

Table 9: Stokes formulation, `agmg` performance for A_{11}, $\epsilon_{\texttt{agmg}} = 10^{-6}$ and random right-hand side vectors.

	Picard linearization		Newton linearization	
	$\tau = 1$	$\tau = 2.5$	$\tau = 1$	$\tau = 2.5$
$h = 1/32$				
Setup time:	4.00e-3	3.00e-3	4.00e-3	3.00e-3
Solution time:	5.00e-3	7.00e-3	4.00e-3	6.00e-3
Total time:	9.00e-3	1.00e-2	8.00e-3	9.00e-3
`agmg` iter.:	18	18	18	20
Direct solver time:	6.76e-3	6.74e-3	5.97e-3	5.78e-3
$h = 1/64$				
Setup time:	8.00e-3	8.00e-3	8.00e-3	9.00e-3
Solution time:	1.60e-2	1.80e-2	2.10e-2	2.20e-2
Total time:	2.40e-2	2.60e-2	2.90e-2	3.10e-2
`agmg` iter.:	16	18	21	20
Direct solver time:	3.32e-2	3.23e-2	3.44e-2	3.52e-2
$h = 1/128$				
Setup time:	2.40e-2	2.70e-2	2.60e-2	2.40e-2
Solution time:	8.20e-2	1.06e-1	9.00e-2	1.23e-1
Total time:	1.06e-1	1.33e-1	1.16e-1	1.47e-1
`agmg` iter.:	19	23	21	24
Direct solver time:	1.67e-1	1.57e-1	1.54e-1	1.57e-1
$h = 1/256$				
Setup time:	9.30e-2	1.03e-1	1.00e-1	1.02e-1
Solution time:	2.83e-1	3.33e-1	3.04e-1	3.73e-1
Total time:	3.76e-1	4.36e-1	4.04e-1	4.75e-1
`agmg` iter.:	16	17	17	19
Direct solver time:	8.04e-1	8.06e-1	9.74e-1	9.46e-1



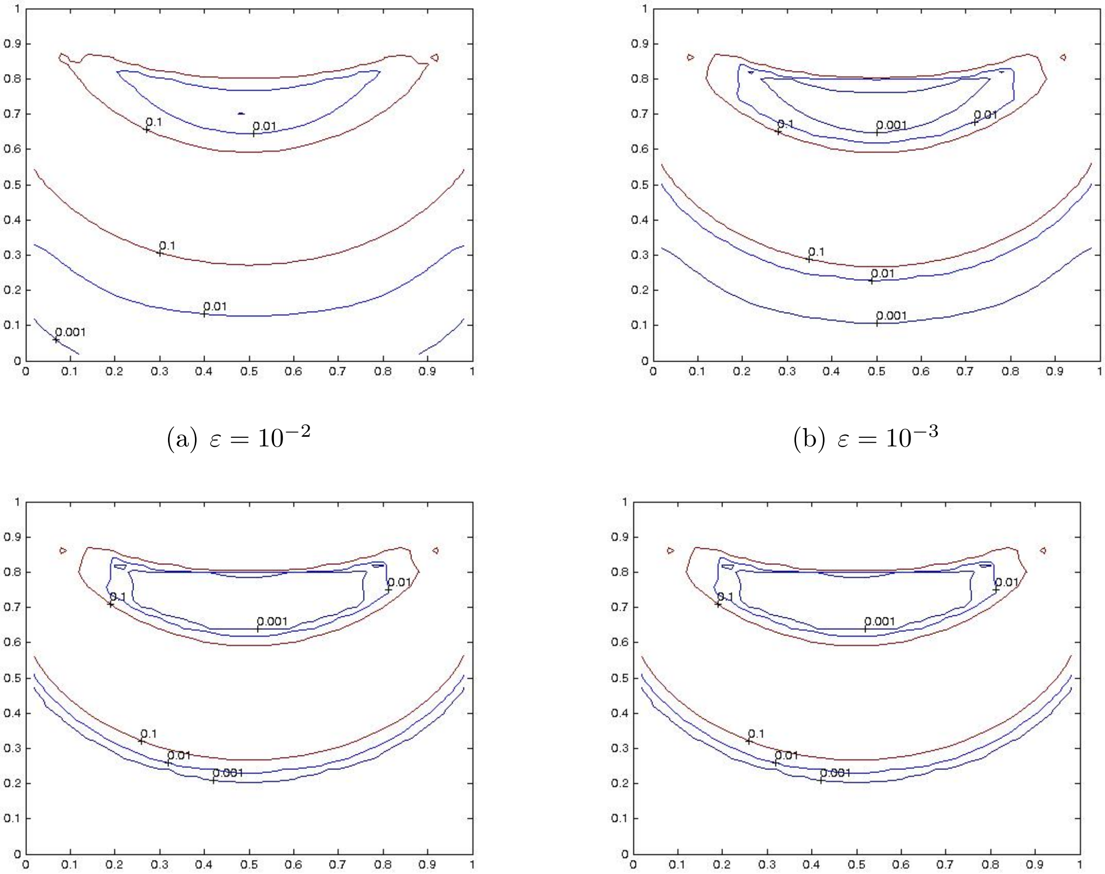
(a) $\varepsilon = 10^{-2}$

(b) $\varepsilon = 10^{-3}$

(c) $\varepsilon = 10^{-4}$

(d) $\varepsilon = 10^{-5}$

Figure 1: Computed isolines for $(D_{\mathrm{II}}(\mathbf{u}))^{\frac{1}{2}} = \{10^{-1},\ 10^{-2},\ 10^{-3}\}$ with $\tau = 1$.



(a) $\varepsilon = 10^{-2}$

(b) $\varepsilon = 10^{-3}$

(c) $\varepsilon = 10^{-4}$

(d) $\varepsilon = 10^{-5}$

Figure 2: Computed isolines for $(D_{\mathrm{II}}(\mathbf{u}))^{\frac{1}{2}} = \{10^{-1},\ 10^{-2},\ 10^{-3}\}$ with $\tau = 2.5$.