

MULTIGRID PRECONDITIONING OF THE NON-REGULARIZED AUGMENTED BINGHAM FLUID PROBLEM*

ALEXIS AOSPORIDIS[†], PANAYOT S. VASSILEVSKI[‡], AND ALESSANDRO VENEZIANI[†]

Abstract. In the numerical solution of visco-plastic fluids, one of the hard problems is the effective detection of *rigid* or *plug* regions. These occur when the strain-rate tensor vanishes and consequently the equations for the fluid region become singular. In order to manage this lack of regularity, different approaches are possible. *Regularization procedures* replace the plug regions with high-viscosity fluid regions, featuring a regularization parameter $\varepsilon > 0$. In Aposporidis et al. [Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2434–2446], an augmented formulation for Bingham fluids was introduced to improve the regularity properties of the problem. Results presented there show that the augmented formulation is more effective for numerical purposes and it works also in the non-regularized case ($\varepsilon = 0$) without a significant degradation of the non-linear solver’s performance. However, when solving high-dimensional Bingham problems, the augmented formulation leads to more challenging linear systems. In this paper we develop a strategy for preconditioning large non-regularized augmented Bingham systems. We use the regularized problem as a preconditioner for the non-regularized case. Then, we resort to a nonlinear geometric multilevel preconditioner to accelerate the convergence of the flexible Krylov linear solver for the regularized Bingham preconditioner. Results presented here demonstrate the effectiveness of the strategy also in realistic (non-academic) test cases.

Key words. multigrid, multilevel flexible GMRES, Bingham flow, mixed finite elements

AMS subject classifications. 65F10, 65N30, 65N55

1. Introduction. Many fluids of industrial, geophysical, and medical interest exhibit a shear-dependent viscosity. In particular, visco-plastic materials show properties of a rigid continuum as long as the applied stress remains below a certain threshold and become incompressible fluids if this critical value is exceeded [11]. A common example of a visco-plastic material is the Bingham fluid [10, 38]. If \mathbf{u} denotes the velocity field of an incompressible fluid in the domain Ω and p is the pressure, we denote by $D\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ the strain rate tensor and consider its Frobenius norm $|D\mathbf{u}| = \sqrt{\text{tr}(D\mathbf{u}^T D\mathbf{u})}$. In Bingham fluids, setting

$$(1.1) \quad \boldsymbol{\tau} = 2\mu D\mathbf{u} + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|}, \quad \text{when } |\boldsymbol{\tau}| \geq \tau_s,$$

we solve the system

$$(1.2) \quad \rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - \nabla \cdot \boldsymbol{\tau} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0$$

Here, $\mu > 0$ (*plastic viscosity*), $\rho > 0$ (*fluid density*), and $\tau_s \geq 0$ (*yield stress*) are assumed to be constant. When $|\boldsymbol{\tau}| \leq \tau_s$, we set

$$D\mathbf{u} = \mathbf{0}.$$

*Received November 8, 2012. Accepted November 24, 2013. Published online on March 17, 2014. Recommended by V. Simoncini. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

[†]Department of Mathematics and Computer Science, Emory University, Atlanta, GA (aapospo@gmail.com, avenez2@emory.edu).

[‡]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Avenue, Mail Stop L-561, Livermore, CA 94550 (vassilevski1@llnl.gov).

The region of Ω where the latter equation holds is called *rigid* or *plug* region, as opposed to the *fluid* region, where (1.2) is assumed to hold. The constitutive relation therefore reads

$$(1.3) \quad D\mathbf{u} = \begin{cases} \mathbf{0} & \text{if } |\boldsymbol{\tau}| \leq \tau_s \text{ (plug region),} \\ \left(1 - \frac{\tau_s}{|\boldsymbol{\tau}|}\right) \frac{\boldsymbol{\tau}}{2\mu} & \text{if } |\boldsymbol{\tau}| > \tau_s \text{ (fluid region).} \end{cases}$$

Equations (1.1), (1.2) can be viewed as a generalization of the Navier-Stokes equations with shear-dependent viscosity $\hat{\mu} = 2\mu + \frac{\tau_s}{|D\mathbf{u}|}$ in the fluid region, reducing to the classical Navier-Stokes equations with constant viscosity if $\tau_s = 0$. A major difficulty associated with solving the Bingham equations is that the flow and plug regions are unknown *a priori*. Notice that $\hat{\mu}$ is singular in the plug region where $|D\mathbf{u}|$ vanishes. These difficulties can be addressed by *regularizing* $\hat{\mu}$. The most common types of regularization are the Bercovier-Engelmann regularization [9], in which $|D\mathbf{u}|$ is replaced by $|D\mathbf{u}|_\varepsilon = \sqrt{|D\mathbf{u}|^2 + \varepsilon^2}$, and the Papanastasiou variant [32]. In practice, regularization techniques replace the plug region by a high viscosity flow region. This clearly improves the regularity of the problem and eventually the performance of the nonlinear solvers even though it affects the accuracy. For this reason, other methods based on a different formulation have been proposed. Among them, we mention here the method introduced by Duvaut and Lions [19, 20]. The latter approach is based on a variational inequality and Uzawa-like iterative methods, whose convergence may be slow.

In this paper, we consider the augmented formulation of the Bingham fluid, further referred to as the ABF problem, introduced in [3]. An auxiliary symmetric tensor $W = \frac{D\mathbf{u}}{|D\mathbf{u}|}$ is defined and (1.2), (1.3) are reformulated as

$$(1.4) \quad \begin{aligned} \rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - \nabla \cdot (2\mu D\mathbf{u} + \tau_s W) + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \\ D\mathbf{u} - |D\mathbf{u}|W &= \mathbf{0}. \end{aligned}$$

Note that this formulation contains no division by $|D\mathbf{u}|$, so the overall regularity of the problem including rigid regions is improved. In this respect, (1.4) is more regular than the primitive formulation (1.1), (1.2). The idea of circumventing a singularity by adding an unknown was inspired by [12], where a similar approach has been successfully applied to a total-variation based image processing problem; see also [43]. Also in the case of ABF, we may think of a regularized version by replacing $|D\mathbf{u}|$ with $|D\mathbf{u}|_\varepsilon$. In [3] in particular, the augmented steady Stokes Bingham problem, when in (1.4) the Lagrangian time derivative is dropped, was considered. An analysis of well-posedness of the regularized augmented problem was carried out. Moreover, numerical results indicate that the iterative nonlinear solver for the augmented formulation—when either Picard or Newton-like linearizations are carried out—converges within a small number of iterations. It is also robust with respect to both mesh size and ε and predicts flow and plug regions accurately. In particular, the augmented formulation works also in the non-regularized case ($\varepsilon = 0$). In addition, it shows superior convergence properties when compared to the Uzawa-like method by Duvaut and Lions mentioned above. However, the results presented in [3] refer to academic test cases where the size of the linear systems makes it affordable using direct methods. As a matter of fact, ABF for real (large) practical problems has the drawback of introducing more challenging linear systems. The specific purpose of this paper is to address an effective way for solving ABF for practical applications expected to be large. More precisely, we introduce an efficient and robust preconditioner to accelerate the convergence of a Krylov subspace method. The preconditioning procedure is based on two ingredients.

- (i) The *regularized* Bingham problem is used as a preconditioner for solving the *non-regularized* problem. In this respect, the regularization parameter ε serves as a control parameter driving the performance of the preconditioner rather than as a perturbation of the problem.
- (ii) The regularized problem is then approximately solved using a multilevel technique. In particular, we introduce a geometric multilevel preconditioner where the smoothing is performed by a flexible GMRES (FGMRES) scheme preconditioned by an overlapping additive Schwarz domain decomposition method. The multigrid iterations are based on recursive cycles performed again within a flexible FGMRES scheme on different grids. The overall scheme gives rise to a nonlinear method sometimes referred to as the nonlinear *algebraic multilevel iteration* (AMLI [42]).

At the coarsest level, we use a direct solver.

Since we are concerned with problems of real interest, here we consider the unsteady version of the problem including the nonlinear convective term. Numerical results exhibit robustness and scalability of the solver with respect to the mesh size, demonstrating that the proposed method can be used for the accurate simulation of non-regularized Bingham fluids. A complete convergence analysis of AMLI-preconditioned ABF is fairly complex (stemming from the fact that this is an indefinite saddle-point problem) and is left to future work. For the symmetric positive definite case, several analyses are available; cf., e.g., [26, 42].

The paper is organized as follows. In Section 2 the problem setting is given including the linearization and discretization. The first part of Section 3 introduces the preconditioner based on the regularized Bingham problem. In the second part we describe the multilevel algorithm which is used to solve the regularized problem. Numerical results on two benchmark problems in two and three dimensions for several values of the mesh size as well as a test case on a more complex geometry are presented in Section 4. Conclusions are drawn in Section 5.

2. Problem setting. We denote by $H^s(\Omega)$ the Sobolev space of functions with s distributional derivatives with summable squares (H_0^1 denotes the set of H^1 functions with null trace on the boundary). In addition, $L^2(0, T; H^s)$ denotes the vector space of functions whose H^s norm for the spatial dependence is square summable in the time interval $(0, T)$. We use \mathbf{H}_0^1 for vector functions with components in H_0^1 and \mathcal{L}^2 for tensor functions with components in L^2 . If we assume for simplicity that the boundary conditions prescribe $\mathbf{u} = 0$ on $\partial\Omega$ (for $t > 0$), the weak regularized ABF problem reads: for $\mathbf{f} \in L^2(0, T; L^2(\Omega))$, find $\mathbf{u} \in L^2(0, T; \mathbf{H}_0^1(\Omega))$, $p \in L^2(0, T; L_0^2(\Omega))$, and $W \in L^2(0, T; \mathcal{L}^\infty(\Omega))$ such that

$$\begin{aligned} \rho \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \mathbf{v} + \rho \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \mathbf{v} + \mu \int_{\Omega} D\mathbf{u} D\mathbf{v} - \int_{\Omega} p \nabla \cdot \mathbf{v} + \tau_s \int_{\Omega} \nabla \cdot W \mathbf{v} &= \int_{\Omega} \mathbf{f} \mathbf{v} \\ - \int_{\Omega} q \nabla \cdot \mathbf{u} &= 0 \\ \int_{\Omega} Z : \nabla \mathbf{u} - \int_{\Omega} |D\mathbf{u}|_{\varepsilon} W : Z &= 0, \end{aligned}$$

with $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ a given initial condition in $L^2(\Omega)$, for all $\mathbf{v} \in \mathbf{H}_0^1(\Omega)$, $q \in L_0^2(\Omega)$, and $Z \in \mathcal{L}^2(\Omega)$. We placed the pressure in the null-average square-summable function space $L_0^2(\Omega)$.

For the sake of a numerical solution, we need to discretize the problem. As for the time-discretization, we rely on a classical backward Euler method. The reason for this choice is to address a simple time-advancing method with good stability properties (stability may be significantly affected by a completely explicit scheme). Different, more accurate (either implicit or semi-implicit) time-advancing schemes may be considered as well with similar procedures.

For the space discretization, we resort to finite elements. Again, different discretization techniques may be considered such as finite difference schemes on staggered grids [24, 31] and finite volume discretizations [35]. Hereafter, we denote by $\mathbf{V}_h, Q_h, \mathcal{Z}_h$ the finite-dimensional subspaces selected for \mathbf{u}, p , and W , respectively. In [3] it has been proven that for $\varepsilon > 0$, no *inf-sup* constraint needs to be fulfilled in the selection of a finite-dimensional space of W for the well-posedness of the discrete problem. In other words, if velocity and pressure are discretized in *inf-sup* compatible spaces, the non-singularity of the discrete problem is guaranteed and is not affected by the choice of the discretization space of W .

When using a Picard linearization for both the convective term and the nonlinearity induced by the rheology at Picard step $k \geq 1$, the discrete problem reads: for $n = 0, 1, \dots, N$, find $\mathbf{u}_h^{n+1} \in \mathbf{V}_h, p_h^{n+1} \in Q_h$, and tensors $W_h^{n+1} \in \mathcal{Z}_h$ such that

$$\begin{aligned} \frac{1}{\Delta t} \rho \int_{\Omega} \mathbf{u}_h^{n+1,k} \mathbf{v}_h + \rho \int_{\Omega} (\mathbf{u}_h^{n+1,k-1} \cdot \nabla) \mathbf{u}_h^{n+1,k} \mathbf{v}_h + \mu \int_{\Omega} D\mathbf{u}_h^{n+1,k} \nabla \mathbf{v}_h \\ - \int_{\Omega} p_h^{n+1,k} \nabla \cdot \mathbf{v}_h + \tau_s \int_{\Omega} W_h^{n+1,k} : \nabla \mathbf{v}_h = \frac{1}{\Delta t} \rho \int_{\Omega} \mathbf{u}_h^n \mathbf{v} + \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v}_h \\ - \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h^{n+1,k} + \alpha \int_{\Omega} p^{n+1,k} q_h = 0 \\ \int_{\Omega} \nabla \mathbf{u}_h^{n+1,k} : Z_h - \int_{\Omega} |D\mathbf{u}_h^{n+1,k-1}|_{\varepsilon} W_h^{n+1,k} : Z_h = 0 \end{aligned}$$

for all $\mathbf{v}_h \in \mathbf{V}_h, q_h \in Q_h$, and $Z_h \in \mathcal{Z}_h$. Here $n, n+1$ refer to the time step, Δt is the time step size, and $k, k-1$ refer to the Picard iteration. The index h indicates the size of the space discretization mesh. The mass conservation equation features a mass-pressure stabilizing term that determines implicitly a value for the pressure. The parameter α will be taken as small as 10^{-10} (as done, for instance, in [25]). Other methods can be pursued similarly to manage the rank deficiency of fully Dirichlet problems.

The matrix formulation of the problem reads $\mathcal{A}_{\varepsilon} \mathbf{w} = \mathbf{b}$ (we drop the time index for the ease of notation) with

$$(2.1) \quad \mathcal{A}_{\varepsilon}(\mathbf{u}^{(k-1)}) = \begin{bmatrix} A(\mathbf{u}^{(k-1)}) & B^T & C^T \\ B & -\alpha Q & 0 \\ C & 0 & -N_{\varepsilon}(\mathbf{u}^{(k-1)}) \end{bmatrix},$$

$$\mathbf{w} = \mathbf{w}^{(k)} = \begin{bmatrix} \mathbf{u}^{(k)} \\ p^{(k)} \\ W^{(k)} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{f} + M\mathbf{u}^n \\ 0 \\ 0 \end{bmatrix},$$

for $k = 1, 2, \dots$.

We denote by \mathbf{v}_j, q_j , and Z_j the generic test basis functions for the three unknowns. We have

$$\begin{aligned} M_{ij} &\equiv \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{v}_i \mathbf{v}_j, & A_{ij}(\mathbf{u}_h^{(k-1)}) &\equiv M_{ij} + \rho \int_{\Omega} (\mathbf{u}_h^{(k-1)} \cdot \nabla) \mathbf{v}_i \mathbf{v}_j + \mu \int_{\Omega} D\mathbf{v}_i \nabla \mathbf{v}_j, \\ B_{ij} &\equiv \int_{\Omega} \nabla \cdot \mathbf{v}_i q_j, & Q_{ij} &\equiv \int_{\Omega} q_i q_j, \\ C_{ij} &\equiv \int_{\Omega} Z_i : \nabla \mathbf{v}_j, & N_{\varepsilon,ij}(\mathbf{u}^{(k-1)}) &\equiv \int_{\Omega} |D\mathbf{u}_h^{(k-1)}|_{\varepsilon} Z_i : Z_j. \end{aligned}$$

Notice that the matrix $N_{\varepsilon}(\mathbf{u}^{(k-1)})$ is symmetric positive definite for $\varepsilon > 0$ (and semidefinite for $\varepsilon = 0$). The matrix N_{ε} is written componentwise (3 blocks in the 2D case and 6 blocks in the 3D case since the tensor is symmetric).

This system features a twofold saddle-point structure. Its efficient solution with realistic geometries and a large number of degrees of freedom is not trivial. It can be obtained by an approximate factorization of \mathcal{A}_ε by splitting or segregating the computation of velocity, pressure, and the tensor W . Another line of investigation is to design an efficient ad-hoc preconditioner by taking advantage of the structure of the matrix.

This can be done in several ways. As a matter of fact, there are two different ways to recognize the saddle-point structure of (2.1). Letting

$$\mathcal{B} = \begin{bmatrix} B \\ C \end{bmatrix} \quad \text{and} \quad \mathcal{N}_\varepsilon = \begin{bmatrix} -\alpha Q & 0 \\ 0 & -N_\varepsilon(\mathbf{u}^{k-1}) \end{bmatrix}$$

gives a saddle-point problem of the form

$$\mathcal{A}_\varepsilon = \begin{bmatrix} A & \mathcal{B}^T \\ \mathcal{B} & \mathcal{N}_\varepsilon \end{bmatrix}$$

with a positive definite (1,1)-block, which is also symmetric in the case of the Stokes type problem.

On the other hand, one may define

$$\mathcal{S} = \begin{bmatrix} A & B^T \\ B & -\alpha Q \end{bmatrix} \quad \text{and} \quad \mathcal{C} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

In this case, the problem becomes

$$\begin{bmatrix} \mathcal{S} & \mathcal{C}^T \\ \mathcal{C} & -N_\varepsilon(\mathbf{u}^{k-1}) \end{bmatrix}$$

and the (1,1)-block of the saddle-point problem is indefinite and represents in turn itself a saddle-point problem. Many preconditioners have been suggested for saddle-point problems either when the matrix (1,1)-block of the system is s.p.d (symmetric positive definite) or its symmetric part is s.p.d. A broad spectrum of preconditioners relies on inexact factorizations of the system and approximations of the Schur complement such as the least square commutator preconditioner or the pressure convection diffusion preconditioner [21, 22]. Other preconditioning techniques for saddle-point problems include augmented Lagrangian preconditioners [7, 8] or preconditioners based on a dimensional splitting [5, 6].

Here, we do not follow these strategies based on the algebraic structure of the problem, but we rely upon a model-based approach. As pointed out in the introduction, we target preconditioning the non-regularized matrix \mathcal{A}_0 by using the regularized matrix \mathcal{A}_ε so that the role of ε turns from controlling the accuracy of the solution to driving the effectiveness of the preconditioner. The final solution will correspond to the non-regularized problem, and this guarantees that the accuracy is affected only by the numerical discretization. The regularized problem needs in turn to be solved effectively. Here, we use a geometric multigrid method. In the following sections we provide an accurate description of the method and its numerical assessment. A brief comparison with a simple, block diagonal preconditioner is provided in Section 4.4. Other methods for ABF may be pursued and will be investigated elsewhere.

3. The multilevel regularization-based preconditioner.

3.1. Spectral investigation of the regularized versus the non-regularized problem.

To support the idea of using the regularized Bingham problem to precondition the non-regularized one, a preliminary spectral analysis on a small-size problem is performed. In Figure 3.1 (left) we report the eigenvalues of the non-regularized Bingham matrix \mathcal{A}_0 for the

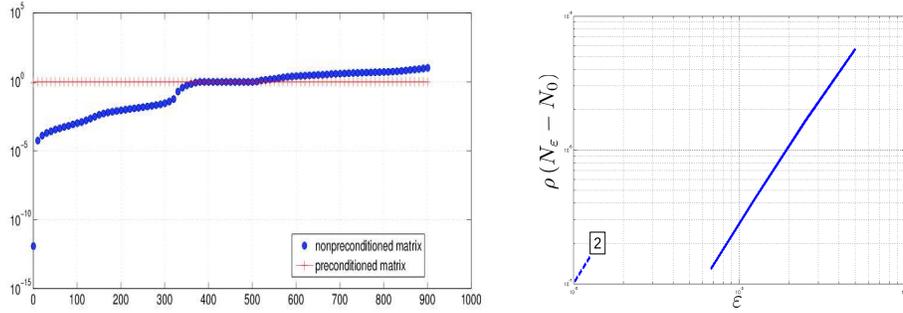


FIG. 3.1. *Left: absolute values of the eigenvalues of the discrete linearized Bingham matrix \mathcal{A} (blue) and eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}_0$ (red) in the analytical test case, where \mathcal{A}_ε is the regularized Bingham matrix with $\varepsilon = 10^{-2}$. The clustering of the eigenvalues of the preconditioned matrix around 1 indicates that—even with a relatively large value of the regularizing parameter—the regularized problem offers the potential for preconditioning the non-regularized one. Right: spectral radius of $N_\varepsilon - N_0$ for several values of ε (logarithmic scale). The reference dash-dotted line has slope 2.*

case of the steady Stokes type equations computed for the flow between parallel plates (see Section 4.3) with $h = 1/16$ on a 2D unit square. This size allows to perform this analysis in Matlab. In the same panel we also display the eigenvalues of \mathcal{A} when preconditioned by the regularized problem \mathcal{A}_ε , i.e., the eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}_0$ with $\varepsilon = 10^{-2}$. Clustering of the eigenvalues around $\lambda = 1$ is evident, and this suggests that the matrix corresponding to the regularized problem may actually be a good preconditioner for the non-regularized one.

For assessing the impact of the regularization parameter on the non-regularized problem, we consider the following factorization of \mathcal{A}_ε with $\varepsilon > 0$ (we omit the dependence of the matrix N_ε on the velocity field for the sake of readability)

$$\mathcal{A}_\varepsilon = \begin{bmatrix} \mathcal{S} & \mathcal{C}^T \\ \mathcal{C} & -N_\varepsilon \end{bmatrix} = \begin{bmatrix} \mathcal{S} & 0 \\ \mathcal{C} & -N_\varepsilon - \mathcal{C}\mathcal{S}^{-1}\mathcal{C}^T \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}\mathcal{C}^T \\ 0 & I \end{bmatrix}.$$

From this factorization it follows that \mathcal{A}_ε (and in particular \mathcal{A}_0) is nonsingular if and only if $N_\varepsilon + \mathcal{C}\mathcal{S}^{-1}\mathcal{C}^T$ ($N_0 + \mathcal{C}\mathcal{S}^{-1}\mathcal{C}^T$) is nonsingular.

In addition, we investigate the matrix $N_\varepsilon - N_0$, whose entries read $\int_\Omega g(\varepsilon)Z_i : Z_j$ with

$$g(\varepsilon) \equiv \sqrt{D\mathbf{u}^2 + \varepsilon^2} - |D\mathbf{u}| = \frac{\varepsilon^2}{\sqrt{D\mathbf{u}^2 + \varepsilon^2} + |D\mathbf{u}|}.$$

By direct inspection, it is readily seen that $g(\varepsilon) > 0$ for any $\varepsilon > 0$, $g(0) = 0$ and $\frac{\partial g}{\partial \varepsilon}(0) = 0$ for $|D\mathbf{u}| \neq 0$, while in the rigid region $g(\varepsilon) = \varepsilon$. We conclude therefore that $N_\varepsilon - N_0$ is s.p.d. for $\varepsilon > 0$, and for $\varepsilon \rightarrow 0$ the spectral radius of $N_\varepsilon - N_0$ vanishes with ε^2 in absence of rigid regions and with ε when rigid regions are present.

PROPOSITION 3.1. *For $\varepsilon \rightarrow 0$, the eigenvalues of the preconditioned matrix $\mathcal{A}_\varepsilon^{-1}\mathcal{A}_0$ cluster around 1. In absence of rigid regions ($|D\mathbf{u}| \neq 0$), the distance of the eigenvalues from 1 scales with ε^2 . When rigid regions are present, the distance scales with ε .*

Proof. For $\Sigma_\varepsilon \equiv N_\varepsilon + \mathcal{C}\mathcal{S}^{-1}\mathcal{C}^T$, notice that

$$\mathcal{A}_\varepsilon^{-1} = \begin{bmatrix} I & -\mathcal{S}^{-1}\mathcal{C}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{S}^{-1} & 0 \\ \Sigma_\varepsilon^{-1}\mathcal{C}\mathcal{S}^{-1} & -\Sigma_\varepsilon^{-1} \end{bmatrix},$$

and by a direct computation we get

$$(3.1) \quad \mathcal{A}_\varepsilon^{-1} \mathcal{A}_0 = \begin{bmatrix} I & \mathcal{S}^{-1} \mathcal{C}^T (I - \Sigma_\varepsilon^{-1} \Sigma_0) \\ 0 & \Sigma_\varepsilon^{-1} \Sigma_0 \end{bmatrix}.$$

Since \mathcal{S} represents the (Newtonian) Stokes (or linearized Navier-Stokes) part of the linear system, the inverse \mathcal{S}^{-1} is well-defined provided either \mathbf{u} and p are discretized in *inf-sup* compatible spaces or $\alpha > 0$ (even though the matrix is indefinite). We can see from (3.1) that the eigenvalues of the preconditioned matrix $\mathcal{A}_\varepsilon^{-1} \mathcal{A}_0$ cluster around one for $\Sigma_\varepsilon^{-1} \Sigma_0 \rightarrow I$. More precisely, since N_ε is s.p.d.,

$$\begin{aligned} \Sigma_\varepsilon^{-1} \Sigma_0 &= (N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_0 + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T) \\ &= (N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T + N_0 - N_\varepsilon) \\ &= I - (N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_\varepsilon - N_0). \end{aligned}$$

Let us denote by λ the eigenvalues of $(N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_\varepsilon - N_0)$, i.e.,

$$(N_\varepsilon + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_\varepsilon - N_0) \mathbf{x} = \lambda \mathbf{x}.$$

The deviation of the eigenvalues of $\mathcal{A}_\varepsilon^{-1} \mathcal{A}_0$ from 1 is given by λ . Then, we have

$$(1 - \lambda) N_\varepsilon \mathbf{x} = (N_0 + \lambda \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T) \mathbf{x}.$$

Since $N_0 + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T$ is nonsingular, $\lambda \neq 1$. In addition, we notice that if μ is a generic eigenvalue of

$$(N_0 + \mathcal{C} \mathcal{S}^{-1} \mathcal{C}^T)^{-1} (N_\varepsilon - N_0),$$

then $\mu = \lambda / (1 - \lambda)$. From the preliminary analysis of the matrix $N_\varepsilon - N_0$, we notice that μ scales with ε^2 for $|D\mathbf{u}| \neq 0$ and with ε when $|D\mathbf{u}| = 0$. Since $\lambda = \mu / (1 + \mu)$, the eigenvalue λ scales in the same way and the proposition is proved. \square

From the previous proposition it is promptly verified that for ε small enough, the eigenvalues of the symmetric part of $\mathcal{A}_\varepsilon^{-1} \mathcal{A}_0$ approaches 1, so they are positive. From [21, Proposition 4.3], the GMRES method converges, the convergence being faster when ε tends to 0. In Figure 3.1 (right), the spectral radius of $N_\varepsilon - N_0$ is displayed for several values of ε in the case of a flow between two parallel plates. For the same test case, Figure 3.2 shows the residual for the first 30 iterations of GMRES when solving the preconditioned system for different values of ε . To provide this proof of concept, the inverse of \mathcal{A}_ε is computed by the backslash Matlab command. As expected, the smaller ε , the faster the GMRES iterations reach any given tolerance. The combination of regularized and non-regularized models presented here specifically for the solution of Bingham fluids is a novel contribution of the present paper. It is worth mentioning that, however, the adoption of regularized problems to precondition Stokes-like systems was advocated by O. Axelsson [4].

For small problems, when the matrix \mathcal{A}_ε is easily solved and its spectral properties do not affect the overall performances of the preconditioned solver, small values of the parameter guarantee faster convergence. Unfortunately, in real applications, we need to resort to different solvers for the preconditioner.

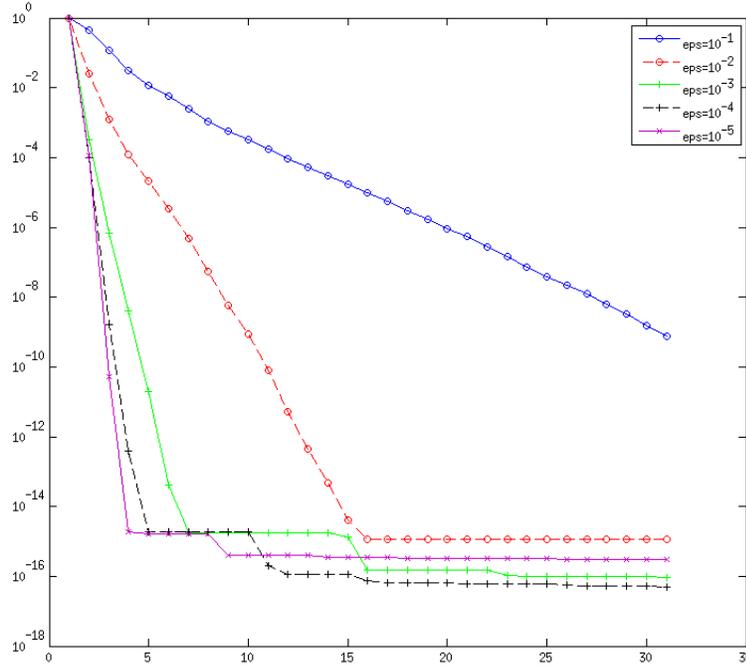


FIG. 3.2. Residual of GMRES for the first 30 iterations when solving the non-regularized problem preconditioned by the regularized one with different values of ε .

3.2. Geometric multigrid approximation of the regularized problem. Even though the preconditioned solver converges, the solution of the regularized preconditioner may be expensive especially for really large problems. For this reason, we need now to devise an efficient solver for the regularized Bingham problem. In particular, we propose here to resort to a *geometrical multigrid technique*. As a matter of fact, multigrid (MG) methods have experienced an increasing popularity for a large range of problems due to their potential for optimality [42] including the solution of indefinite problems; see, e.g., [36, 44] in the context of constrained optimization problems and fluid-structure interaction, respectively.

To define our approximate solver, we first introduce some notation. Consider a sequence of L regular finite element meshes \mathcal{T}_k with $k = 1, 2, \dots, L$, such that \mathcal{T}_L is the finest grid, where we solve the problem. Each finer mesh at level k is assumed to be a refinement of the coarser level $k - 1$. Correspondingly, we associate the matrices $\mathcal{A}_{\varepsilon,k}$ obtained by discretizing the regularized ABF problem (either steady or unsteady). In the sequel, when there is no ambiguity, we drop the index ε for notational convenience. Throughout this section we refer to the regularized Bingham problem.

Let further $\{\Pi_k\}_{k=1}^{L-1}$ be the natural prolongation (by interpolation) matrices relating the system matrix \mathcal{A}_k to its coarser counterpart $\mathcal{A}_{k-1} = \Pi_k^T \mathcal{A}_k \Pi_k$, and let \mathbf{y} be a generic input vector, \mathbf{x} the corresponding output vector, and σ, ν, ℓ , and tol be given parameters, where ℓ represents the current level and tol the given tolerance.

In Algorithm 1, we introduce our recursive solver of the regularized ABF problem. The method “MLPrecond” recursively calls itself σ times as a preconditioner inside an FGMRES scheme [34]. For $\sigma = 2$, this method provides a variant of the classical W-cycle multigrid.

3.2.1. Interpolation and restriction. Starting with a mesh that is sufficiently coarse to allow a fast solution of the discrete system (e.g., by a direct solver that we have denoted with “\”), we refine the mesh uniformly L times. With each mesh, we associate a corresponding triple of finite element spaces, $\mathbf{V}_{h,k}, Q_{h,k}, \mathcal{Z}_{h,k}$, $k = 1, \dots, L$. By construction, the coarse level spaces are subspaces of the next fine level spaces. This defines natural embeddings $\{\Pi_k^u\}_{k=1}^L$, $\{\Pi_k^p\}_{k=1}^L$, and $\{\Pi_k^W\}_{k=1}^L$ which transfer (interpolate) the degrees of freedom of \mathbf{u} , p , and W , respectively, from the coarse mesh (level $k - 1$) to the fine mesh (level k). The (monolithic) interpolation operator is given by

$$\Pi_k = \begin{bmatrix} \Pi_k^u & 0 & 0 \\ 0 & \Pi_k^p & 0 \\ 0 & 0 & \Pi_k^W \end{bmatrix}.$$

As previously pointed out, the matrix \mathcal{A}_L is assembled on the finest mesh, and the coarse ones are obtained via the Galerkin condition $\mathcal{A}_{k-1} = \Pi_k^T \mathcal{A}_k \Pi_k$ for $k = 1, \dots, L$.

3.2.2. Smoothing. Several types of smoothers may be considered. In the case of symmetric positive definite problems, stationary iterations are typically the method of choice. If the problem is symmetric indefinite, a feasible approach is to perform a few iterations of a preconditioned Krylov subspace method with a simple (for example Jacobi or Gauss-Seidel) preconditioner. Since (2.1) is indefinite, we use the additive Schwarz method as a preconditioner in a GMRES solver. We have to use GMRES since the Schwarz preconditioner is generally also indefinite. Using the variational iterative method as a smoother in a MG cycle, strictly speaking, leads to a (mildly) non-linear mapping of the overall MG cycle. For the sake of simplicity, we will omit the index k indicating the level of discretization for the remainder of this section. Given the discretized domain Ω on any given level, we subdivide the domain into m overlapping subsets $\{\Omega_i\}_{i=1}^m$. Then we set up linear mappings $\{I_i^u\}_{i=1}^m$, $\{I_i^p\}_{i=1}^m$, and $\{I_i^W\}_{i=1}^m$ restricting the degrees of freedom of \mathbf{u} , p , and W , respectively, to the local domain Ω_i . The i th subdomain local matrix is then given by

$$\mathcal{A}_i = I_i \mathcal{A} I_i^T \quad \text{with} \quad I_i = \begin{bmatrix} I_i^u & 0 & 0 \\ 0 & I_i^p & 0 \\ 0 & 0 & I_i^W \end{bmatrix},$$

and the inverse of the global matrix is approximated by the additive Schwarz preconditioner defined by the formula

$$\mathcal{A}^{-1} \approx \sum_{i=1}^m I_i^T \mathcal{A}_i^{-1} I_i.$$

The size of the subdomains should be chosen sufficiently small so that the inverse of the local matrices \mathcal{A}_i can be easily computed.

This smoothing technique can be viewed as an additive extension of the Vanka-smoother [39, 41] based on a block Gauss-Seidel iteration where each block is identified by the patch of each element [28, 37]. The latter has been specifically proposed for the finite difference solution of the incompressible (Newtonian) Navier-Stokes equations.

4. Numerical results.

4.1. Implementation details. Numerical tests presented hereafter have been obtained with the finite element library MFEM [30]. In particular, we use $\mathbb{P}^2/\mathbb{P}^1$ finite elements for velocity and pressure, respectively, while the auxiliary variable W is discretized with \mathbb{P}^1 finite elements.

As a solver on the coarsest grid we use a direct solver within the C library SUITESPARSE. More precisely, when solving the Stokes type equations, we resort to an LDL^T -type factorization using the LDL package described in [15]; see [40]. For the Navier-Stokes problem, the LU factorization is computed by the UMFPACK package [13, 14, 17, 18]). Before computing all factorizations, we apply a fill-in reducing reordering provided by AMD [1, 2, 16].

To set up the smoother on each level (except for the coarsest), we first generate an adjacency matrix $S = [s_{ij}]$ (with $s_{ij} = 1$ if element i and j share a common face in three dimensions or a common edge in two dimensions and $s_{ij} = 0$ otherwise). We then apply a graph partitioner in METIS [29] on S . This procedure results in a partitioning of the mesh in which the overlap consists of one layer of elements at the interface. Extra layers of overlap may be included as well. The solves on each subdomain are again done by the direct solvers provided in SUITESPARSE. Table 4.1 shows the different meshes we use for our experiments and the number of multigrid levels used for each mesh. The numbers of subdomains and of overlapping nodes are shown as well. In particular, the number of subdomains on each level has a strong influence on the performance of our preconditioner. The trade-off is between the size of the local system (not too large) and the overall efficacy of the smoother. This is achieved by increasing the number of subdomains by a factor of 4 in 2D and a factor of 6 in 3D for each additional multigrid level as shown in the table. The size of the discrete system for tests running on a unit square and on a unit cube is also shown.

Initialization of the Picard iteration is set to be $\mathbf{u} = \mathbf{u}^0$, $p \equiv 0$, $W \equiv 0$, where \mathbf{u}^0 is the solution of $-\mu\Delta\mathbf{u}^0 = \mathbf{f}$ solved with preconditioned CG iterations. Then, we continue the nonlinear iterations until

$$\frac{\|\mathbf{r}\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-2},$$

where \mathbf{r} (\mathbf{r}_0) is the current (initial) residual. The absolute tolerance is set to $5 \cdot 10^{-6}$. In this way the linear solver is accurate enough to guarantee nonlinear convergence in all our test cases. Since the initial guess already yields a relatively small initial residual, this stopping criterion is sufficient to achieve a good approximation of plug and fluid regions as we will see later. The linear system is solved by FGMRES with our geometric nonlinear AMLI multigrid preconditioner. The solution is considered converged if the quotient of current and initial residual drops below 10^{-6} in the L^2 -norm. All tables display the number of linear iterations needed for the convergence of the first nonlinear iteration. To summarize, the overall procedure is presented in Algorithm 2.

4.2. Choosing the regularization parameter. In Section 3.1 we stated that the performance of the preconditioner \mathcal{A}_ε improves as ε decreases provided that the inverse $\mathcal{A}_\varepsilon^{-1}$ is computed exactly. However, the reduction of the regularization parameter in general deteriorates the conditioning properties of the matrix, and this may impair the quality and effectiveness of the preconditioner in presence of plug regions. In this respect, finding the optimal value of ε involves finding the right trade-off between conditioning of the regularized problem (which improves when ε is large so that the regularization is stronger) and its consistency as preconditioner of the non-regularized one (improves when ε is small). In our experiments we empirically found that the optimal choice is $\varepsilon = 10^{-2}$.

It is worth noticing that the domain decomposition used in our experiments is based solely on the mesh and not on the solution. If a subdomain is entirely contained in a plug region, we may experience some performance degradation. As a matter of fact, some of the (local) linear systems representing a subdomain are extremely ill-conditioned for small values of ε , and the local (direct) solves on these subdomains may be very inaccurate resulting in a

Algorithm 2: Sketch of the preconditioned scheme.

```

begin
  pre-processing (mesh generation, etc.) ;
  solve  $-\mu\Delta\mathbf{u}^0 = \mathbf{f}$  ;
   $\mathbf{x} = [\mathbf{u}^0; 0, 0]$ ; // initial guess computation ;
  while relative residual  $> 10^{-2}$  do
    //Picard outer loop ;
    assemble matrix  $\mathcal{A}_L$  and right-hand side with the current guess  $\mathbf{x}$ 
    while relative residual  $> 10^{-6}$  do
      //linear inner loop ;
      FGMRES( $\mathcal{A}_L, \mathbf{x}, \mathbf{r}, \dots, \text{MLPrecond}$ )
    end
  end
  post-processing ;
end

```

TABLE 4.1

Number of multigrid levels, number of subdomains, size of each subdomain, size of overlap, and the size of the linear system to be solved in two and three dimensions.

Experiments on unit square					
mesh	# levels	# subd.	#overlap. nod.	size overl.	size lin. syst.
$h = 1/8$	2	3	34-40	27	902
$h = 1/16$	3	9	42-50	126	3,334
$h = 1/32$	4	27	56-72	498	12,806
$h = 1/64$	5	81	68-85	1,839	50,182
$h = 1/128$	6	243	89-110	6,751	198,662
$h = 1/256$	7	729	114-143	24,343	790,534

Experiments on unit cube					
mesh	# levels	# subd.	size subd.	size overl.	size lin. syst.
$h = 1/4$	2	12	24-31	97	3,062
$h = 1/8$	3	72	27-49	669	19,842
$h = 1/16$	4	432	34-55	4,697	142,202
$h = 1/32$	5	2,392	41-70	34,925	1,075,434

failure of the smoother. Larger values of ε yield an improved conditioning of the systems on these subdomains. A future development of the method would include an adaptive domain decomposition approach to avoid these troublesome situations.

We also noticed that the condition number of the regularized block N_ε in (2.1) grows mildly as $\varepsilon \rightarrow 0$ except when between 10^{-2} and 10^{-3} where the increase is more evident; see Figure 4.1. This effect is independent of the mesh size and it provides an additional *a posteriori* motivation of our choice.

4.3. Flow between two parallel plates. This test case is one of the few examples in which the analytical solution is known for the steady (Navier) Stokes type Bingham problem. The domain is a unit square where the coarsest mesh adopted has mesh size $h = 1/4$. In 3D extensions of this case, running on a unit cube, the coarsest level features $h = 1/2$. The test

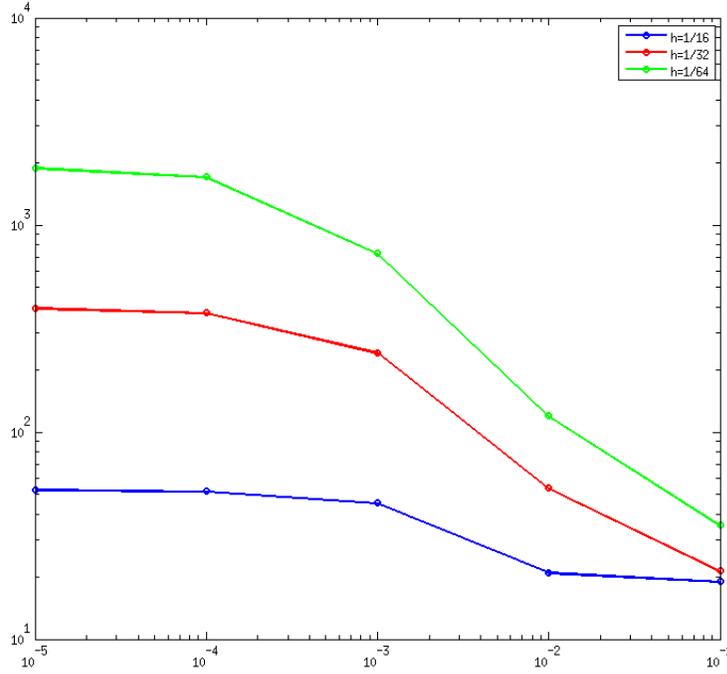


FIG. 4.1. Condition number of the block N_ϵ for different values of ϵ (on the horizontal axis) and h .

case describes a flow between two parallel plates and its solution is given by

$$(4.1) \quad u_1 = \begin{cases} \frac{1}{8\mu}[(1 - 2\tau_s)^2 + (1 - 2\tau_s - 2y)^2] & \text{if } 0 \leq y < \frac{1}{2} - \tau_s, \\ \frac{1}{8\mu}(1 - 2\tau_s)^2, & \text{if } \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s, \\ \frac{1}{8\mu}[(1 - 2\tau_s)^2 - (2y - 2\tau_s - 1)^2] & \text{if } \frac{1}{2} + \tau_s < y \leq 1, \end{cases}$$

with $u_2 \equiv u_3 \equiv 0$ and $p = -x$. The strain rate vanishes in the plug region

$$\{(x, y, z) \mid \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s\}.$$

In our experiment we impose Dirichlet boundary conditions on the unit square and cube according to (4.1) with $\tau_s = 0.3$ and $\mu = 1$.

To precondition the flexible GMRES iterations, we use the algorithm from Section 3 with two smoothing steps ($\nu = 2$) in 2D and four smoothings ($\nu = 4$) in 3D as well as two iterations of FGMRES on each multigrid level ($\sigma = 2$). Table 4.2 displays the number of flexible GMRES iterations needed for convergence for the first Picard step, the total number of nonlinear iterations needed for convergence, as well as the CPU time needed for solving the linearized system. In the three-dimensional case the number of linear iterations slightly increases with the size of the mesh. However, the parameters for the preconditioner were chosen to minimize the CPU time as opposed to the iteration count. By properly tuning the number of smoothings or the number of subdomains, we get an even more evident mesh independence.

TABLE 4.2

The flow between two parallel plates, an analytical test case in two and three dimensions: number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds), and the total number of nonlinear (Picard) iterations. The unit of time is second.

Two dimensional experiments					
mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
$h = 1/8$	10	0.01	0.01	0.01	6
$h = 1/16$	13	0.08	0.02	0.03	6
$h = 1/32$	14	0.50	0.05	0.09	6
$h = 1/64$	14	2.19	0.18	0.40	6
$h = 1/128$	14	9.60	0.86	1.72	7
$h = 1/256$	12	37.93	5.44	7.25	7

Three dimensional experiments					
mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
$h = 1/4$	6	0.08	0.01	0.07	4
$h = 1/8$	8	1.51	0.17	0.75	5
$h = 1/16$	16	27.89	1.81	6.74	6
$h = 1/32$	11	258.53	25.53	90.46	6

Note that for the Stokes type problem, only the matrix N_ε needs to be updated before each nonlinear iteration. In this respect, the timings provided in Table 4.2 for setting up the preconditioner are divided into initial setup time (this includes setting up the interpolations between the different levels, determining the subdivision of the domains, and setting up the restriction operators for each subdomain) and updating time (this includes the updating of N_ε in the preconditioner, computing a factorization of the local matrices on each subdomain, and computing a factorization for the direct solver on the coarsest level). All experiments are performed with a serial code. Timings for the Stokes type Bingham problem are obtained on a personal laptop with an Intel Core i7 processor, 2.6 GHz, and 8 GB of memory. Due to a higher memory requirement, experiments involving the Navier-Stokes type Bingham problems are run on a Sun Microsystems SunFire X4600 with 20 AMD Opteron(tm) cores and 32 GB of memory.

In Table 4.3, we examine the effect of the regularization parameter ε on the performance of the preconditioner. The table displays the number of linear iterations needed for solving the two-dimensional Bingham problem for different mesh sizes and different values for ε (used as a parameter in the preconditioner). It emphasizes the profound influence of the choice of this parameter: while $\varepsilon = 10^{-2}$ yields mesh independent convergence, for other choices of the parameter, the number of iterations increases—at times significantly—as the mesh size decreases. Choosing $\varepsilon = 10^{-2}$ was the optimal value in all our experiments, however, the effect was the strongest in this test case.

Figure 4.2 shows plug and fluid regions as well as the streamlines for this flow in two dimensions for different values of τ_s . The colors in the figure represent values of $|Du|$ so that the plug regions are approximately the parts of the domain in which $|Du|$ is small.

4.4. The lid-driven cavity. This is a standard benchmark problem for CFD codes. The lid is moved at a velocity of magnitude 1 in the x -direction, i.e.,

$$\mathbf{u} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{if } y = 1,$$

TABLE 4.3

Number of linear iterations needed for solving the flow between parallel plates when preconditioning the problem with different values of ε and for different mesh sizes.

	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 0$
$h = 1/8$	25	10	9	9	9	9
$h = 1/16$	26	13	13	13	13	13
$h = 1/32$	27	14	15	15	15	15
$h = 1/64$	23	14	16	17	18	18
$h = 1/128$	19	14	21	24	24	24
$h = 1/256$	17	12	29	58	59	50

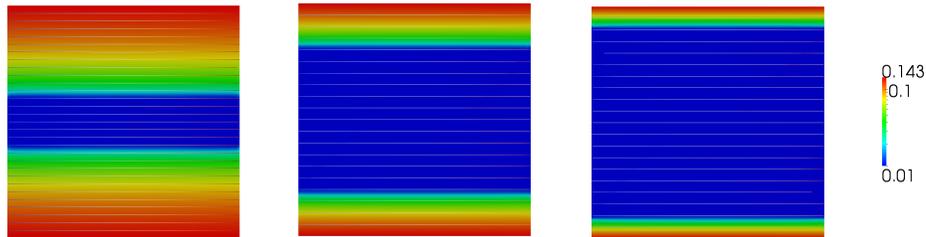


FIG. 4.2. Plug regions (blue) and streamlines for the flow between parallel plates for different $\tau_s = 0.1$ (left), $\tau_s = 0.3$ (center), and $\tau_s = 0.4$ (right). Regions in red indicate higher values of $|D\mathbf{u}|$.

and we impose homogeneous Dirichlet boundary conditions elsewhere. Again we set $\tau_s = 2$ and $\mu = 1$. For preconditioning we use the multilevel algorithm with two smoothings in 2D and four smoothings in 3D as well as two inner GMRES iterations on each level. Table 4.4 shows the numerical results for this experiment. Note that in the case $\tau_s = 10$, we use four smoothings in both 2D and 3D in order to maintain mesh independent convergence. Plug and fluid regions for this flow are shown in Figure 4.3. We simulated the same case considered in [24]. Visual comparison of our results with the one in there [24, Figure 5.4] confirms again what was already found in [3], i.e., that ABF is able of correctly detecting the plug regions.

For the sake of comparison, we provide some numerical results for this problem if a different, simpler preconditioner is used. One may use a preconditioner of the form

$$\mathcal{P} = \begin{bmatrix} A & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & N_\varepsilon \end{bmatrix},$$

where A , Q , and N_ε are as in (2.1). This preconditioner is symmetric positive definite. Hence, we may use MINRES iterations as the outer Krylov solver. This preconditioner was tested on the two-dimensional problem. Table 4.5 shows the number of iterations required for solving the linear systems. Solutions of systems involving \mathcal{P}^{-1} are computed exactly by a direct solver on a mesh up to size $h = 1/64$. The numerical results in the table demonstrate that even though this preconditioner may be very easy to apply, the overall number of iterations needed to reach the given tolerance is very high, and the number of iterations increases significantly with the mesh size so that mesh independent convergence is lost.

4.5. The steady Navier-Stokes type problem. We now apply the lid-driven cavity test case to the steady Navier-Stokes type problem. Here all specifications are the same as in the

TABLE 4.4

The lid-driven cavity flow with $\tau_s = 2$, $\tau_s = 5$, and $\tau_s = 10$ in two and three dimensions (Stokes): number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds), and the total number of nonlinear (Picard) iterations. An asterisk * indicates that four smoothings were used in this experiment. The unit of time is second.

		Two dimensional experiments				
$\tau_s = 2$	mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
	$h = 1/8$	8	0.01	0.01	0.01	4
	$h = 1/16$	11	0.07	0.01	0.02	5
	$h = 1/32$	12	0.42	0.04	0.10	5
	$h = 1/64$	12	1.87	0.17	0.40	5
	$h = 1/128$	12	8.22	0.80	1.71	5
	$h = 1/256$	11	34.91	5.44	7.29	4
		Three dimensional experiments				
	mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
	$h = 1/4$	3	0.04	0.01	0.06	3
	$h = 1/8$	5	0.94	0.17	0.74	4
	$h = 1/16$	8	13.94	1.80	6.63	5
	$h = 1/32$	5	167.21	25.48	89.12	5
		Two dimensional experiments				
$\tau_s = 5$	mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
	$h = 1/8$	9	0.01	0.01	0.02	5
	$h = 1/16$	12	0.10	0.01	0.02	6
	$h = 1/32$	14	0.53	0.05	0.10	6
	$h = 1/64$	15	2.45	0.18	0.42	6
	$h = 1/128$	16	11.26	0.84	1.84	6
	$h = 1/256$	15	47.17	5.40	7.96	6
		Three dimensional experiments				
	mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
	$h = 1/4$	3	0.04	0.01	0.06	3
	$h = 1/8$	5	0.95	0.15	0.76	4
	$h = 1/16$	9	15.87	1.80	6.82	5
	$h = 1/32$	5	161.23	25.51	88.15	5
		Two dimensional experiments				
$\tau_s = 10$	mesh	# lin. its*	CPU time	setup	updating	# nonlin. its.
	$h = 1/8$	5	0.02	0.01	0.01	6
	$h = 1/16$	9	0.12	0.01	0.02	7
	$h = 1/32$	10	0.64	0.03	0.10	7
	$h = 1/64$	11	3.19	0.18	0.43	7
	$h = 1/128$	11	14.06	0.84	1.85	7
	$h = 1/256$	10	56.96	4.42	7.98	7
		Three dimensional experiments				
	mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
	$h = 1/4$	4	0.05	0.01	0.06	4
	$h = 1/8$	5	0.95	0.16	0.76	5
	$h = 1/16$	12	21.07	1.79	6.66	6
	$h = 1/32$	6	185.29	25.56	96.59	6

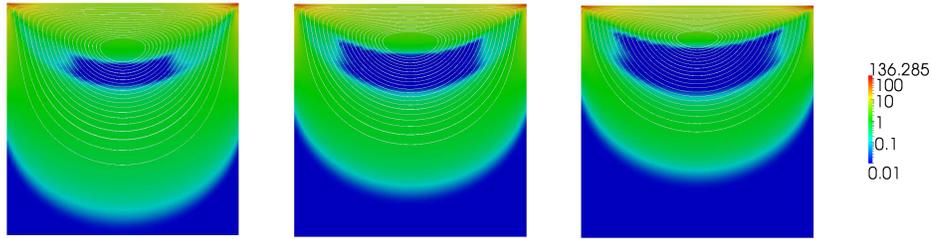


FIG. 4.3. Plug regions (blue) and streamlines in two dimensions for the lid-driven cavity problem with $\tau_s = 2$ (left), $\tau_s = 5$ (center) and $\tau_s = 10$ (right). Green and red areas represent the fluid region.

TABLE 4.5

Number of iterations needed for linear convergence in the two-dimensional lid-driven cavity when using a block diagonal preconditioner.

mesh	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
# iterations	482	584	660	748

previous subsection except that we now solve (1.4) with $\frac{\partial \mathbf{u}}{\partial t} \equiv 0$, $\rho = 1$, and we impose

$$\mathbf{u} = \begin{bmatrix} 50 \\ 0 \\ 0 \end{bmatrix} \quad \text{if } y = 1$$

corresponding to a Reynolds number $Re = 50$. We now tighten the nonlinear tolerance to 10^{-4} to ensure accurate solutions. Numerical results are shown in Table 4.6. Note the drop in the nonlinear iteration count for $h = 1/32$ in the three-dimensional case. This is due to the higher Reynolds number in this test case; the relatively high number of nonlinear iterations for the coarser meshes are due to convective instabilities. We did not see this effect for lower Reynolds numbers ($Re \approx 10$); the effect was even more evident if we increased the Reynolds number (to approximately $Re = 80$).

4.6. The unsteady Navier-Stokes type problem on a non-trivial geometry. This experiment is performed on a cylindrical domain with a sphere attached to it. It is meant as an idealized geometry that approximates a blood vessel with an aneurysm. This experiment serves as a first step in understanding the relevance of Bingham fluids in problems occurring in hemodynamics; see [23, 33]. We discretize the domain with curvilinear isoparametric (second order) finite elements; see, e.g., [27]. Using elements of higher order has the effect that the “curved” shape of the domain is captured well during the refinement process of the geometric multigrid. See Figure 4.4 for the shape of the geometry on each multigrid level.

We use multigrid preconditioning on three levels in this experiment with four smoothings and two inner FGMRES iterations. In the Bingham fluid equations, we take $\mu = 1$ and $\tau_s = 1$. We prescribe parabolic boundary conditions at the inflow, a no-slip condition at the walls, and homogeneous Neumann boundary conditions at the outflow. Time is discretized on the interval from $t = 0$ to $t = 1.5$ with a time step of $\Delta t = 0.1$. Table 4.7 shows more specifications on the preconditioner as well as the numerical results for this experiment. Figure 4.5 shows the streamlines and pressure of this flow after a steady state has been reached.

TABLE 4.6

The lid-driven cavity flow for the Navier-Stokes type problem: number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds), and the total number of Picard iterations. The unit of time is second.

Two dimensional experiments					
mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
$h = 1/8$	7	0.03	0.01	0.04	9
$h = 1/16$	10	0.41	0.01	0.19	8
$h = 1/32$	12	2.50	0.04	0.77	7
$h = 1/64$	12	11.74	0.23	3.27	6
$h = 1/128$	12	54.00	1.44	13.39	5
$h = 1/256$	11	232.63	11.87	53.32	4

Three dimensional experiments					
mesh	# lin. its	CPU time	setup	updating	# nonlin. its.
$h = 1/4$	6	0.59	0.02	0.61	12
$h = 1/8$	6	9.28	0.12	7.33	17
$h = 1/16$	8	101.76	1.77	58.29	15
$h = 1/32$	7	747.84	47.53	440.48	4

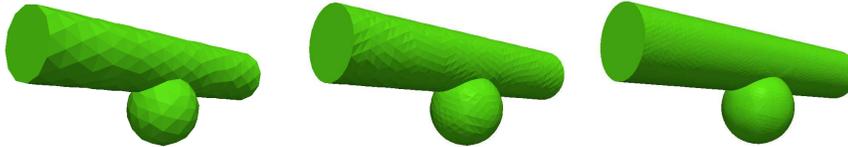


FIG. 4.4. Idealized blood vessel with aneurysm on three geometric multigrid levels. Left: The coarsest level (280 elements), center: one level of refinement (2,240 elements), right: two levels of refinement (17,920 elements).

5. Conclusion. In this paper we have introduced a new way of solving the linear systems obtained by linearizing and discretizing the non-regularized Bingham fluid flow equations in the augmented formulation. This formulation has been originally proposed in [3] together with its linearization. Here, we have focused on the effective solution of the associated linear system for real problems featuring large size. The first step is to use the regularized Bingham problem as preconditioner for the non-regularized one to take advantage of the better properties of the system without affecting the accuracy of the solution. We have proved that the regularized ABF provides a convergent preconditioner to the non-regularized one. Then, we have considered the effective solution of each preconditioned iteration. We resort to a linear solver based on a flexible Krylov subspace method. Convergence of this iterative method is accelerated by a nonlinear geometric AMLI multigrid algorithm. In using the regularized problem in this way, the regularization parameter ε drives the performance of the preconditioner rather than the accuracy of the solver. Upon a proper selection of this parameter, our numerical experiments indicate mesh independent convergence in a low number of iterations; a rigorous proof of this is left for future research. Timings are obtained here on serial machines but may be significantly improved on a parallel architecture. The application of our method to large-scale problems on parallel architectures with particular reference to problems in computational hemodynamics is a natural follow up of the present research.

TABLE 4.7
Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment.

			General information	
	fine level	intermediate level	# Picard its (per time st.):	5
# subd.:	72	12	CPU (s) (per t.s.):	53.41
size subd.:	100-140	74-98	setup (s):	0.80
size overl.:	2,791	342	updating:	72.36
			# linear its:	6

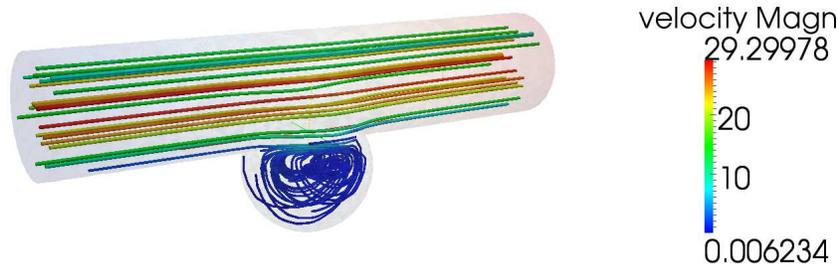


FIG. 4.5. Streamlines and pressure field of the unsteady Navier-Stokes type problem in a cylindrical domain with an attached sphere.

REFERENCES

- [1] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.
- [2] ———, *Algorithm 837: AMD, an approximate minimum degree ordering algorithm*, ACM Trans. Math. Software, 30 (2004), pp. 381–388.
- [3] A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, AND A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: analysis and numerical solution*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2434–2446.
- [4] O. AXELSSON, *Preconditioning of indefinite problems by regularization*, SIAM J. Numer. Anal., 16 (1979), pp. 58–69.
- [5] M. BENZI AND X.-P. GUO, *A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations*, Appl. Numer. Math., 61 (2011), pp. 66–76.
- [6] M. BENZI, M. NG, Q. NIU, AND Z. WANG, *A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys., 230 (2011), pp. 6185–6202.
- [7] M. BENZI AND M. A. OLSHANSKII, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput., 28 (2006), pp. 2095–2113.
- [8] M. BENZI, M. A. OLSHANSKII, AND Z. WANG, *Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations*, Internat. J. Numer. Methods Fluids, 66 (2011), pp. 486–508.
- [9] M. BERCOVIER AND M. ENGELMAN, *A finite element method for incompressible non-Newtonian flows*, J. Comput. Phys., 36 (1980), pp. 313–326.
- [10] E. C. BINGHAM, *An investigation of the laws of plastic flow*, Bull. Bur. Standards, 13 (1917), pp. 309–353.
- [11] R. BYRON-BIRD, G. DAI, AND B. J. YARUSSO, *Rheology and flow of viscoplastic materials*, Rev. Chem. Engrg., 1 (1983), pp. 1–70.
- [12] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.
- [13] T. A. DAVIS, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.
- [14] ———, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 167–195.
- [15] ———, *Algorithm 849: a concise sparse Cholesky factorization package*, ACM Trans. Math. Software, 31 (2005), pp. 587–591.
- [16] ———, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2006.

- [17] T. A. DAVIS AND I. S. DUFF, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 140–158.
- [18] ———, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Trans. Math. Software, 25 (1999), pp. 1–20.
- [19] E. J. DEAN, R. GLOWINSKI, AND G. GUIDOBONI, *On the numerical simulation of Bingham visco-plastic flow: old and new results*, J. Non-Newtonian Fluid Mech., 142 (2007), pp. 36–62.
- [20] G. DUVAUT AND J. L. LIONS, *Inequalities in Mechanics and Physics*, Springer, Berlin, 1976.
- [21] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2005.
- [22] H. C. ELMAN AND R. S. TUMINARO, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electron. Trans. Numer. Anal., 35 (2009), pp. 257–280. <http://etna.mcs.kent.edu/vol.35.2009/pp257-280.dir/>
- [23] L. FORMAGGIA, A. QUARTERONI, AND A. VENEZIANI, (eds.) *Cardiovascular Mathematics*, Springer, Milano, 2009.
- [24] P. P. GRINEVICH AND M. A. OLSHANSKII, *An iterative method for the Stokes-type problem with variable viscosity*, SIAM J. Sci. Comput., 31 (2009), pp. 3959–3978.
- [25] F. HECHT, *FreeFem++*, Laboratoire Jacques-Louis Lions, Universite Pierre et Marie Curie, Paris, 2013. <http://www.freefem.org/ff++/ftp/freefem++doc.pdf>
- [26] X. HU, P. S. VASSILEVSKI, AND J. XU, *Comparative convergence analysis of nonlinear AMLI-cycle multigrid*, SIAM J. Numer. Anal., 51 (2013), pp. 1349–1369.
- [27] T. J. R. HUGHES, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover, Mineola, NY, 2000.
- [28] V. JOHN AND L. TOBISKA, *Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations*, Internat. J. Numer. Methods Fluids, 33 (2000), pp. 453–473.
- [29] G. KARYPIS AND V. KUMAR, *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*, Manual MN55455, Department of Computer Science, University of Minnesota, Minneapolis, 2009.
- [30] MFEM, *Finite element discretization library*, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2012. <http://code.google.com/p/mfem/>
- [31] E. A. MURAVLEVA AND M. A. OLSHANSKII, *Two finite-difference schemes for the calculation of Bingham fluid flows in a cavity*, Russian J. Numer. Anal. Math. Modelling, 23 (2008), pp. 615–634.
- [32] T. C. PAPANASTASIOU, *Flow of materials with yield*, J. Rheol., 31 (1987), pp. 385–404.
- [33] A. M. ROBERTSON, A. SEQUEIRA, AND V. KAMENEVA, *Hemorheology*, in Hemodynamical Flows, G. Galdi, R. Rannacher, A. Robertson, S. Turek, eds., Oberwolfach Seminars, 37, Birkhauser, Basel, 2008, pp. 63–120.
- [34] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
- [35] M. SAHIN AND H. J. WILSON, *A semi-staggered dilation-free finite volume method for the numerical solution of viscoelastic fluid flows on all-hexahedral elements*, J. Non-Newtonian Fluid Mech., 147 (2007), pp. 79–91.
- [36] J. SCHÖBERL, R. SIMON, AND W. ZULEHNER, *A robust multigrid method for elliptic optimal control problems*, SIAM J. Numer. Anal., 49 (2011), pp. 1482–1503.
- [37] J. SCHÖBERL AND W. ZULEHNER, *On Schwarz-type smoothers for saddle point problems*, Numer. Math., 95 (2003), pp. 377–399.
- [38] S. I. SENAŠOV, *Group classification of equations of ideal plasticity with a fluidity condition of a general form*, Dinamika Sploshn. Sredy, 37 (1978), pp. 101–112.
- [39] S. TUREK, *Efficient Solvers for Incompressible Flow Problems*, Springer, Berlin, 1999.
- [40] R. J. VANDERBEI, *Symmetric quasidefinite matrices*, SIAM J. Optim., 5 (1995), pp. 100–113.
- [41] S. P. VANKA, *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.
- [42] P. S. VASSILEVSKI, *Multilevel Block Factorization Preconditioners*, Springer, New York, 2008.
- [43] C. R. VOGEL, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- [44] H. YANG AND W. ZULEHNER, *Numerical simulation of fluid-structure interaction problems on hybrid meshes with algebraic multigrid methods*, J. Comput. Appl. Math., 235 (2011), pp. 5367–5379.