INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 0000; **00**:1–17
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

Efficient Augmented Lagrangian-type Preconditioning for the Oseen Problem using Grad-Div Stabilization

Timo Heister[1*], Gerd Rapin[2]

[1]*Department of Mathematics, Texas A&M University, College Station, TX 77843-3368*
(heister@math.tamu.edu)
[2]*Letter box 1697, Volkswagen AG, D-38436 Wolfsburg*
(grapin@math.uni-goettingen.de)

SUMMARY

Efficient preconditioning for Oseen-type problems is an active research topic. We present a novel approach leveraging stabilization for inf-sup stable discretizations. The Grad-Div stabilization shares the algebraic properties with an augmented Lagrangian-type term. Both simplify the approximation of the Schur complement, especially in the convection dominated case. We exploit this for the construction of the preconditioner. Solving the discretized Oseen problem with an iterative Krylov-type method shows that the outer iteration numbers are retained independent of mesh size, viscosity, and finite element order. Thus, the preconditioner is very competitive. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Preconditioning; Oseen problem; Grad-Div stabilization; Navier-Stokes; Schur complement; augmented Lagrangian

1. INTRODUCTION

The simulation of incompressible flow problems plays a vital role in various scientific areas. Incompressible flows are governed by the incompressible Navier-Stokes equations and are given by the following set of equations: One has to find a velocity field $\boldsymbol{u}$ and pressure field p satisfying

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} - \nabla \cdot (\nu \nabla \boldsymbol{u}) + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad & \text{in } (0,T] \times \Omega, \\ \nabla \cdot \boldsymbol{u} = 0 \qquad & \text{in } [0,T] \times \Omega. \end{aligned} \tag{1}$$

Here, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded polyhedral domain, $\nu = \nu(x) > 0$ is the scalar viscosity that may depend on $x \in \Omega$, and $\boldsymbol{f} \in [L^2(\Omega)]^d$ is a given source term.

After time discretization and linearization one is left with solving a sequence of stationary Oseen type problems

$$\begin{aligned} -\nabla \cdot (\nu \nabla \boldsymbol{u}) + (\boldsymbol{b} \cdot \nabla)\boldsymbol{u} + c\boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad & \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} = 0 \qquad & \text{in } \Omega \\ \boldsymbol{u} = 0 \qquad & \text{on } \partial\Omega. \end{aligned} \tag{2}$$

For the ease of presentation we assume homogeneous Dirichlet boundary conditions for the velocity. $\boldsymbol{b} \in [L^2(\Omega) \cap W^{1,\infty}(\Omega)]^d$ is a vector field describing the convection acting on the velocity and is

[*] Correspondence to: T. Heister, Department of Mathematics, Texas A&M University, College Station, TX 77843-3368. E-mail: heister@math.tamu.edu

typically given by the velocity stemming from a fixed-point linearization scheme or an extrapolation from the last time step. The constant reaction coefficient $c \geq 0$ enters the system due to the time discretization and is proportional to the inverse of the time-step size. The case $c = 0$ appears after linearizing stationary Navier-Stokes problems. For $\boldsymbol{b} = 0$ and $c = 0$ the system reduces to a Stokes problem.

In applications the viscosity $\nu(x) > 0$ might vary in the computational domain. For example, simulating turbulent flows requires a turbulence model which is mostly modeled as an artificial, additional viscosity term (e.g. in VMS methods, see [1]). We will limit the remainder of this presentation to constant viscosity. Extending our results to varying viscosity would be preferable but this is a topic of further research. Note that the sum of physical and turbulent viscosity in turbulent flows is small compared to the other terms. Therefore, the influence of the varying turbulence is small from the preconditioner point of view and taking an average viscosity is possible.

Solving Oseen-type systems has a long history and there are dozens of different solution approaches ranging from Uzawa type methods [2] to projection methods [3], special multi-grid methods [4] or block saddle point preconditioners [5, 6, 7]. A good overview for solvers for saddle point problems is given in [8].

Here, we consider an iterative Krylov method to solve the arising linear system with a block saddle point preconditioner. Since the linear system is badly conditioned preconditioning is mandatory. The biggest challenge is finding a preconditioner that performs equally well for different mesh sizes h and different coefficient ranges of c, ν, and $\boldsymbol{b}$. In the case of the Stokes problem (for $\mathbf{b} = 0$) the preconditioning is much simpler, cf. [9]. The most challenging configuration is $\nu \ll 1$, $c = 0$, and $\boldsymbol{b} \neq 0$, cf. [5, 6, 7].

Recently, it has been shown that *augmented Lagrangian* (AL) methods work quite promising in this case. *Augmented Lagrangian* approaches are well known for many years (see [10]) and are used in various applications, cf. [11] and references therein. In [12] Benzi and Olshanskii present an *augmented Lagrangian*-based preconditioner for the Oseen problem that shows impressive results for various h and ν. Due to difficulties in solving the augmented velocity block, a *modified augmented Lagrangian* formulation is presented in [13] and analyzed further in [14]. In the original approach [12] the velocity block is augmented with an algebraic term possessing a large kernel. This gives rise to an efficient preconditioner for the Schur complement but complicates the solve of the velocity block. Moreover, assembling the augmentation term is quite expensive, since a product of sparse matrices has to be computed. The modified version in [13] simplifies the solution of the velocity block by only applying the augmentation to the upper right blocks. Unfortunately at the same time this spoils the quality of the Schur complement approximation and leads to larger number of iterations.

In this paper we explain that one can consider Grad-Div stabilization as a different discretization of the *augmented Lagrangian* term. This motivates the construction of a preconditioner where we replace the augmented Lagrangian term with Grad-Div stabilization. This approach is preferable, because this removes the difficulty of applying the augmentation. Moreover, the numerical experiments in this paper show clearly that the number of iterations for the saddle point problem stays independent of the problem.

One fundamental aspect in numerical discretization of partial differential equations like the Oseen problem is stabilization. Solutions may contain spurious oscillations. This non physical behavior appears for example for convection dominated flows, i.e. $\nu \ll \|\boldsymbol{b}\|$, and can completely spoil the solution. Stabilization methods typically add additional terms to penalize these oscillations. These terms have an impact on the algebraic properties of the discretized problem. Taking the effect of stabilization on preconditioning into account is challenging and therefore often neglected. Some exceptions are [13, 15, 16]. A typical excuse is the possibility to explicitly treat the stabilization terms in order to remove them from the linear system. In contrast, we try to take advantage of existing Grad-Div stabilization, cf. [17, 18], in the preconditioner.

We show that Grad-Div stabilization can be seen as a sum of an augmented Lagrangian term, that can be used for preconditioning and does not influence the solution, and a projection-type stabilization term. The similarity between augmentation and Grad-Div stabilization has already

DOI: 10.1002/fld

been described in [13], is used for Stokes problems in [18] in the context of Uzawa methods, and is analyzed in [19] for Oseen problems.

We describe the underlying discretization and stabilization in Section 2, recall typical block-triangular preconditioning for the linear system in Section 3, and present the new preconditioner in Section 4. Based on the analysis of the augmented Lagrangian preconditioner in [12], we discuss some advantages and disadvantages in Section 6. Numerical results in Section 5 show the efficiency of the preconditioner.

2. DISCRETIZATION USING INF-SUP STABLE FINITE ELEMENTS AND GRAD-DIV STABILIZATION

A traditional equal-order Galerkin finite element discretization of the Oseen problem (2) may suffer from the violation of the discrete inf-sup (or Babuška-Brezzi) condition and results in spurious oscillations in the solution. One remedy is to impose pressure stabilization, like Pressure Stabilized Petrov Galerkin (PSPG) methods [20], or Local Projection Stabilization (LPS) [21]. Alternatively one can switch to inf-sup stable elements. A common family of inf-sup stable elements are the Taylor-Hood elements. We will concentrate on using those throughout this paper.

The weak formulation of (2) reads: Find $(\boldsymbol{u}, p) \in \boldsymbol{V} \times Q := [H_0^1(\Omega)]^d \times L_*^2(\Omega)$, with $L_*^2(\Omega) := \{v \in L^2(\Omega) \mid \int_\Omega v \, dx = 0\}$, such that

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) &= (\boldsymbol{f},\boldsymbol{v}) \qquad \forall \boldsymbol{v} \in \boldsymbol{V} && (3)\\ b(\boldsymbol{u},q) &= 0 \qquad \forall q \in Q && (4)\end{aligned}$$

with the bilinear forms

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) &:= (\nu\nabla\boldsymbol{u},\nabla\boldsymbol{v}) + ((\boldsymbol{b}\cdot\nabla)\boldsymbol{u} + c\boldsymbol{u},\boldsymbol{v}),\\ b(\boldsymbol{v},p) &:= -(\nabla\cdot\boldsymbol{v},p).\end{aligned}$$

For the discretization we consider quadrilateral or hexahedral meshes $\mathcal{T}_h = \{K\}$ in our numerical examples. The finite element space Q_k, $k \in \mathbb{N}$, is given by

$$Q_k := \{v \in C(\Omega) \mid v|_K \circ F_K \in \mathbb{Q}_k,\ K \in \mathcal{T}_h\}$$

where $\mathbb{Q}_k$ is made of tensor-product polynomials up to order k on the reference cell $\hat{K}$ and $F_K : \hat{K} \to K$ denotes the mapping from the reference cell to the cell K. We denote the discrete spaces with $\boldsymbol{V}_h = [Q_{k+1}]^d \cap \boldsymbol{V}$ and $Q_h = Q_k \cap Q$. Note that using simplicial meshes is also possible but not considered in this paper.

The chosen discrete Taylor-Hood pair $\boldsymbol{V}_h \times Q_h$ satisfies the discrete inf-sup condition

$$\exists C > 0 \mid \quad \inf_{q \in Q_h \setminus \{0\}} \sup_{\boldsymbol{v}_h \in \boldsymbol{V}_h \setminus \{0\}} \frac{b(\boldsymbol{v}_h, q_h)}{\|\boldsymbol{v}_h\|_{\boldsymbol{V}} \|q_h\|_Q} \geq C.$$

The proof and other stable pairs can be found in [22].

Additional stabilization is useful and often needed even for inf-sup stable elements. We consider Grad-Div stabilization (see [23]), which results in a modified bilinear form

$$\tilde{a}(\boldsymbol{u},\boldsymbol{v}) := a(\boldsymbol{u},\boldsymbol{v}) + (\gamma\nabla\cdot\boldsymbol{u},\nabla\cdot\boldsymbol{v}).$$

Thus, we end up with the following discrete system: Find $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h$ such that

$$\begin{aligned} \tilde{a}(\boldsymbol{u}_h,\boldsymbol{v}_h) + b(\boldsymbol{v}_h,p_h) &= (\boldsymbol{f},\boldsymbol{v}_h) \qquad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h && (5)\\ b(\boldsymbol{u}_h,q_h) &= 0 \qquad \forall q_h \in Q_h. \end{aligned}$$

The parameter $\gamma \geq 0$ is assumed to be constant per element or a global constant. The stabilization improves the numerical accuracy of the solution and helps reducing spurious oscillations for convection dominated flow.

DOI: 10.1002/fld

We refer to [23] for a detailed discussion of Grad-Div stabilization. There, the parameter choice is discussed in detail. In general the optimal value for γ depends on the solution on the element K. For sufficiently smooth solutions the following formula for inf-sup stable elements can be derived:

$$\gamma_K \sim \max\left\{ \frac{|p|_{H^k(K)}}{|\boldsymbol{u}|_{H^{k+1}(K)}} - \nu, 0 \right\}. \tag{6}$$

In practice, the unknown continuous solution $(\boldsymbol{u}, p)$ in (6) is replaced by their discrete counterpart $(\boldsymbol{u}_h, p_h)$. Evaluating the norms of $\boldsymbol{u}_h$ and p_h is costly, since higher order derivatives are involved. Therefore a common assumption is $|p|_{H^k(K)} \approx |\boldsymbol{u}|_{H^{k+1}(K)}$ and $\nu \ll 1$, which results in a constant $\gamma \sim 1$. Unfortunately, this assumption is not valid in general. For example, in the case of a laminar channel flow (also known as Poiseuille flow) $|p|_{H^k(K)}$ is proportional to $\nu |\boldsymbol{u}|_{H^{k+1}(K)}$ and therefore[†] $\gamma_K \sim \nu$.

The solver and preconditioner concept in this paper works identically with additional, symmetric stabilization terms for the velocity, like local projection stabilization, see [21, 24, 25] and references therein. We omit the definition and the terms for clarity of presentation. The addition of non-symmetric stabilization like Streamline Upwind/Petrov Galerkin (SUPG) methods (see the overview in [26] and [20, 27, 28]) is non-trivial, as it spoils the saddle point structure of our problem. Here, we recommend to shift the asymmetric terms to the right-hand side.

Assembling the discretized equations (5) results in a linear system of the form

$$Mx = G \tag{7}$$

with unknowns $x = (U, P)^T$, right-hand side $G = (F, 0)^T$ and block matrix

$$M = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}.$$

U and P are the coefficient vectors that belong to the degrees of freedom of the velocity $\boldsymbol{u}_h$ and pressure p_h. The blocks A and B correspond to the bilinear forms $\tilde{a}$ and b, respectively. The matrix A is positive but non-symmetric for $\boldsymbol{b} \neq 0$. The linear system is sparse as it is typical for linear systems stemming from finite element discretizations. Typically, the sparsity pattern and the size of the linear system require the application of iterative methods. Naturally, iterative Krylov methods, like GMRES [29], are used. Because of the bad condition of (7) preconditioning is mandatory. We describe a prominent way of dealing with the preconditioning in the next section.

3. BLOCK-TRIANGULAR PRECONDITIONING

Instead of solving the system $Mx = G$ directly, we apply right preconditioning as described in [29] with an operator $\mathcal{P}^{-1}$ and calculate the solution $x = \mathcal{P}^{-1} y$ from the auxiliary variable y, which is given as the solution of

$$M\mathcal{P}^{-1} y = G.$$

In general, $\mathcal{P}^{-1}$ is not given by a matrix, since building this inverse is typically not appropriate. Here, we define $\mathcal{P}^{-1}$ as an implicitly defined operator given in a block-triangular way (see [8]):

$$\mathcal{P}^{-1} := \begin{pmatrix} \widetilde{A} & B^T \\ 0 & \widetilde{S} \end{pmatrix}^{-1} = \begin{pmatrix} \widetilde{A}^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \widetilde{S}^{-1} \end{pmatrix}, \tag{8}$$

where $\widetilde{S}$ is an approximation of the Schur complement

$$S = -BA^{-1}B^T \tag{9}$$

[†] We assume the proportionality constant to be different than one.

DOI: 10.1002/fld

and $\widetilde{A}$ of the velocity block A, respectively. Thus, applying the preconditioner involves one solve for $\widetilde{A}$, one solve for $\widetilde{S}$, and one matrix-vector multiplication with B^T.

Now the goal is to define good and computationally cheap approximations for $\widetilde{A}$ and $\widetilde{S}$. Exact solves would result in at most two outer GMRES iterations, see [8]. Our construction of approximations of the Schur complement is based on [9]. The idea is to separately look at the main building blocks of A. The matrix A can be written as

$$A = \nu L_u + cM_u + N_u + R_u,$$

where L_u represents the diffusion term and M_u is the mass matrix of the velocity space. N_u represents the convective term and R_u the Grad-Div term.

Assuming that the diffusion part is dominant the Schur complement can be approximated by

$$S^{-1} \approx -\left[B(\nu L_u)^{-1}B^T\right]^{-1} \approx -\nu M_p^{-1}, \tag{10}$$

where M_p is the mass matrix in the pressure space. The approximation can be motivated by assuming that the continuous operators commute, see [30] for details. Note, that we assume ν to be constant. Similarly, for a dominating reaction the Schur complement can be approximated by

$$S^{-1} \approx -\left[B(cM_u)^{-1}B^T\right]^{-1} \approx -cL_p^{-1}, \tag{11}$$

where L_p is the stiffness matrix of the pressure Poisson problem with Neumann boundary conditions. It is not known, how to treat the convective term in a similar way. With (10) and (11) one has good preconditioners for the Schur complement in case of dominating diffusion and reaction, respectively. To automatically switch between these two preconditioners,

$$\widetilde{S}^{-1} = -\nu M_p^{-1} - cL_p^{-1}, \tag{12}$$

is suggested in [30, 9], which works remarkably well as long the problem is not convection dominated.

We will compare our results with the well-known pressure-convection-diffusion preconditioner (short: PCD) for the Schur complement, see [7]. Here, the Schur complement is approximated by:

$$S_{PCD}^{-1} = -M_p^{-1}F_pL_p^{-1},$$

where M_p and L_p are defined as before and F_p is a convection-diffusion operator defined on the pressure space (that can also contain a reaction term if the original system did).

4. A PRECONDITIONER UTILIZING GRAD-DIV STABILIZATION

Similar to approximations for diffusion and reaction in the last section, we will look at the term in the A block which stems from the Grad-Div stabilization. We enhance (12) to account for Grad-Div stabilization in A. The resulting Schur complement approximation is given as

$$S_{GD}^{-1} = -(\nu + \gamma)M_p^{-1} - cL_p^{-1}. \tag{13}$$

We call applying the block-triangular preconditioner (8) together with this Schur complement approximation the *Grad-Div preconditioner* (or short: GD). Note that the matrix A contains the Grad-Div stabilization with coefficient $\gamma > 0$. For $\gamma = 0$, the method reduces to the one described in Section 3.

We now proceed to motivate and analyze the choice in (13). Let $\pi : Q \to Q_h$ be the orthogonal L^2-projector, i.e.

$$(p - \pi p, q) = 0 \quad \forall q \in Q_h.$$

DOI: 10.1002/fld

For $\boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{V}_h$ the Grad-Div term can be split into the following sum:

$$(\nabla \cdot \boldsymbol{u}_h, \nabla \cdot \boldsymbol{v}_h) = (\pi(\nabla \cdot \boldsymbol{u}_h), \nabla \cdot \boldsymbol{v}_h) + ((I - \pi)(\nabla \cdot \boldsymbol{u}_h), \nabla \cdot \boldsymbol{v}_h). \qquad (14)$$

Using the fluctuation operator $\kappa := I - \pi$ and the projection property of π we obtain

$$(\nabla \cdot \boldsymbol{u}_h, \nabla \cdot \boldsymbol{v}_h) = (\pi(\nabla \cdot \boldsymbol{u}_h), \pi(\nabla \cdot \boldsymbol{v}_h)) + (\kappa(\nabla \cdot \boldsymbol{u}_h), \kappa(\nabla \cdot \boldsymbol{v}_h)). \qquad (15)$$

We call the first part *algebraic term* and the second part *stabilizing term*. The following lemma shows that the algebraic term can be written as a product of known matrices.

Lemma 1
The discretized algebraic term of the Grad-Div stabilization is given by

$$(\pi(\nabla \cdot \phi_j), \pi(\nabla \cdot \phi_i)) = (B^T M_p^{-1} B)_{ij} \qquad \forall i, j \in \{1, \ldots, n\}$$

using basis $\{\phi_i\}_{i=1}^n$ of $\boldsymbol{V}_h$ and $\{\psi_j\}_{j=1}^m$ of Q_h. The matrices B and M_p are defined by $(B)_{ij} := (\psi_i, \nabla \cdot \phi_j)$ and $(M_p)_{ij} := (\psi_i, \psi_j)$.

Proof: We first define a matrix representation P of the projection π with

$$\pi(\nabla \cdot \phi_j) = \sum_{i=1}^m P_{ij} \psi_i \quad \forall j = 1, \ldots, n.$$

Plugging in the definition of π and P into B gives after some rearrangement:

$$\begin{aligned}(B)_{ij} = (\psi_i, \nabla \cdot \phi_j) &= (\pi(\nabla \cdot \phi_j), \psi_i) = (\sum_{k=1}^m P_{kj} \psi_k, \psi_i) \\ &= \sum_{k=1}^m P_{kj} (\psi_k, \psi_i) = \sum_{k=1}^m (\psi_i, \psi_k) P_{kj} = (M_p P)_{ij}.\end{aligned}$$

The augmentation term $B^T M_p^{-1} B$ can now be written as

$$\begin{aligned}(B^T M_p^{-1} B)_{ij} &= (B^T M_p^{-1} M_p P)_{ij} = (B^T P)_{ij} = \sum_{k=1}^m (B^T)_{ik} P_{kj} = \sum_{k=1}^m (P_{kj} (\psi_k, \nabla \cdot \phi_i)) \\ &= \left(\sum_{k=1}^m P_{kj} \psi_k, \nabla \cdot \phi_i \right) = (\pi(\nabla \cdot \phi_j), \nabla \cdot \phi_i) = (\pi(\nabla \cdot \phi_j), \pi(\nabla \cdot \phi_i)),\end{aligned}$$

which shows the proposition. □

Adding the term $B^T M_p^{-1} B$ to the system block A is known as the *augmented Lagrangian* approach, see [12], and does not change the solution due to $B^T M_p^{-1} B U = 0$ for a solution (U, P) of the linear system (7). Nevertheless, it modifies the algebraic properties of the velocity-velocity block.

The second term $(\kappa(\nabla \cdot \boldsymbol{u}_h), \kappa(\nabla \cdot \boldsymbol{v}_h))$ contains the stabilization for which Grad-Div is used. The difference between discretized Grad-Div stabilization (R_u) and augmentation $B^T M_p^{-1} B$ can be written as

$$(R_u - B^T M_p^{-1} B)_{ij} = (\kappa(\nabla \cdot \phi_j), \kappa(\nabla \cdot \phi_i)).$$

The stabilizing term in the Grad-Div stabilization vanishes for $h \to 0$ and thus in the limit only the algebraic augmentation remains:

Lemma 2
Let $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h$ be a solution of the stabilized linear system (5) with corresponding degrees of freedom (U, P). Then, we obtain for Taylor-Hood elements Q_{k+1}/Q_k, $k \geq 1$, and a sufficiently

DOI: 10.1002/fld

smooth solution $(\boldsymbol{u}, p)$ of the continuous Oseen problem (4), i.e. $\boldsymbol{u} \in [H^{k+1}(\Omega)]^d$ and $p \in H^k(\Omega)$,

$$\| \left(R_u - B^T M_p^{-1} B\right) U\|_{\mathbb{R}^n} \leq C h^{k+(d-2)/2} \left(\|\boldsymbol{u}\|_{k+1}^2 + \|p\|_k^2\right)^{1/2}.$$

Proof:

Assume $\boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{V}_h$. Let $\|\cdot\|_{\mathbb{R}^n}$ denote the Euclidean and $\|\cdot\|_0$ the L^2 norm. Using the basis representations

$$\boldsymbol{u}_h = \sum_{i=1}^n U_i \phi_i, \qquad \boldsymbol{v}_h = \sum_{i=1}^n V_i \phi_i$$

for $\boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{V}_h$, we obtain the assertion

$$\begin{aligned}
&\| \left(R_u - B^T M_p^{-1} B\right) U\|_{\mathbb{R}^n} \\
&\quad = \sup_{\|V\|_{\mathbb{R}^n}=1} V^T \left(R_u - B^T M_p^{-1} B\right) U = \sup_{\|V\|_{\mathbb{R}^n}=1} (\kappa(\nabla \cdot \boldsymbol{v}_h), \kappa(\nabla \cdot \boldsymbol{u}_h)) \\
&\quad \leq \sup_{\|V\|_{\mathbb{R}^n}=1} \|\kappa(\nabla \cdot \boldsymbol{v}_h)\|_0 \|\kappa(\nabla \cdot \boldsymbol{u}_h)\|_0 \\
&\quad \leq \|\kappa\|^2 \sup_{\|V\|_{\mathbb{R}^n}=1} \|\nabla \cdot \boldsymbol{v}_h\|_0 \|\nabla \cdot (\boldsymbol{u}_h - \boldsymbol{u})\|_0 \\
&\quad \leq \underbrace{\|\kappa\|^2}_{=:T_1} \underbrace{\sup_{\|V\|_{\mathbb{R}^n}=1} \|\nabla \cdot \boldsymbol{v}_h\|_0}_{=:T_2} \underbrace{\|\nabla \cdot (\boldsymbol{u}_h - \boldsymbol{u})\|_0}_{=:T_3}.
\end{aligned}$$

We have $T_1 \leq C_1$, because the fluctuation operator κ is continuous. The inverse inequality gives

$$T_2 = \sup_{\boldsymbol{v}_h} \frac{\|\nabla \cdot \boldsymbol{v}_h\|_0}{\|V\|_{\mathbb{R}^n}} \leq C' \sup_{\boldsymbol{v}_h} \frac{h^{-1}\|\boldsymbol{v}_h\|_0}{\|V\|_{\mathbb{R}^n}}$$

and [31], Theorem 3.43 gives

$$\|V\|_{\mathbb{R}^n} \geq C h^{-d/2} \|\boldsymbol{v}_h\|_0$$

with $C > 0$ independent of h. From this it follows (because of $d \geq 2$):

$$T_2 \leq \frac{C'}{C} \sup_{v_h} h^{-1+d/2} \frac{\|v_h\|_0}{\|v_h\|_0} = C_2 h^{-1+d/2}.$$

The a priori error estimation for the Grad-Div stabilized Oseen problem in [32] (Corollary 3.3) gives for sufficiently smooth solutions $\boldsymbol{u}, p$:

$$T_3^2 = \|\nabla \cdot (\boldsymbol{u}_h - \boldsymbol{u})\|_0^2 \leq C_3 h^{2k} \|\boldsymbol{u}\|_{k+1}^2 + C_4 h^{2k} \|p\|_k^2.$$

Putting the terms together finally gives

$$\| \left(R - B^T M_p^{-1} B\right) U\|_{\mathbb{R}^n} \leq C h^{k+(d-2)/2} \left(\|\boldsymbol{u}\|_{k+1}^2 + \|p\|_k^2\right)^{1/2} \to 0 \quad \text{for } h \to 0.$$

□

The lemma explains why the preconditioner works and behaves very similar to the augmented Lagrangian approach. The results in [13], which show h and ν independent iteration numbers, can therefore expected to be achieved here, too. This is confirmed in Section 5.

One can interpret the stabilizing effect of Grad-Div as adding a penalty term for the fluctuations of the divergence given by the projection π. The term closely resembles projection based stabilization, though it is not a local projection and thus can not be assembled easily.

DOI: 10.1002/fld

The Schur complement for the augmented matrix A with the algebraic term $B^T M_p^{-1} B$ can be simplified to

$$\left[B(A+\gamma B^T M_p^{-1} B)^{-1} B^T\right]^{-1} = \left(BA^{-1}B^T\right)^{-1} + \gamma M_p^{-1}. \tag{16}$$

Therefore, in [12] the authors propose to approximate the Schur complement by

$$\widetilde{S}^{-1} = -(\nu+\gamma) M_p^{-1}.$$

Note, that in contrast to the approximation for the diffusion term, (16) is exact if B is assumed to have full rank. This only accounts for the algebraic component in the Grad-Div stabilization.

We propose an extension for instationary problems, which is motivated as already explained in Section 3. It is not strictly necessary for $c \neq 0$ but accelerates the solution process especially for large c. With this we arrive at (13). One can decide on a case by case basis to not implement the last part. Obviously, the coefficient c in there automatically reduces the influence of that term for large time steps and stationary problems.

In short we can use the same approximation for the Schur complement as in the AL approach, but we do not need to add the augmentation to the A block. This gives a huge advantage over the AL approach, since the solution of the augmented

$$A_\gamma = A + \gamma B^T M_p^{-1} B$$

is very costly. On the one hand assembling A_γ is extremely expensive. In [13] the authors present various tricks like lumping the mass matrix and moving to a cheaper approximated AL formulation where only part of the augmentation is applied. The problem is that the product $B^T B$ possesses much more non-zero entries than A itself. Building a product of sparse matrices is also computationally expensive as one can not easily generate a correct sparsity pattern beforehand. Of course, one can avoid the assembling of A_γ and only supply it as an operator. But then one can not apply preconditioners like algebraic multi-grid or ILU decompositions. On the other hand the iterative solution of A_γ becomes difficult due to the large kernel of $B^T M_p^{-1} B$.

All these problems are not present in our case. The matrix A which already contains the augmentation through Grad-Div stabilization is easy to assemble. One can directly use various kind of solvers. This still leaves us with the fact, that solving the velocity block A with Grad-Div stabilization is more expensive than solving it without. As an approximation needs to be done for every outer iteration step, this can be expensive. Nevertheless, it reduces the number of outer iterations drastically, as can be seen in the next Section.

Replacing the augmentation in $\tilde{A}_\gamma$ by Grad-Div stabilization was proposed and tested in [19] and is also suggested in a comment in [13].

5. NUMERICAL RESULTS

For the numerical tests we have used the finite element library `deal.II`, see [33, 34]. The computations have been performed on unstructured quadrilateral meshes, see Figure 1 for an example. We construct a series of those meshes for the parameter studies. Unstructured meshes are more realistic and naturally arise when dealing with complex geometries. Moreover, super convergence effects are avoided and more realistic error bounds are achieved.

For the outer iteration a flexible GMRES method (see [35, 29]) is used. Standard Krylov methods can not be applied, since we use iterative solvers within the preconditioner. Therefore, the preconditioner can not be considered as constant during the outer iterations. The outer iteration loop is stopped when the residual is dropped by a factor of 10^{-10} relative to the starting residual. This stopping criterion is more strict compared to other papers analyzing preconditioners and leads to higher iteration numbers. We have chosen this convergence criterion, since it allows us to see trends easier due to the higher number of iterations. Moreover, a softer criterion can be misleading, since the errors of the solution are very often dominated by the iterative process and not by the

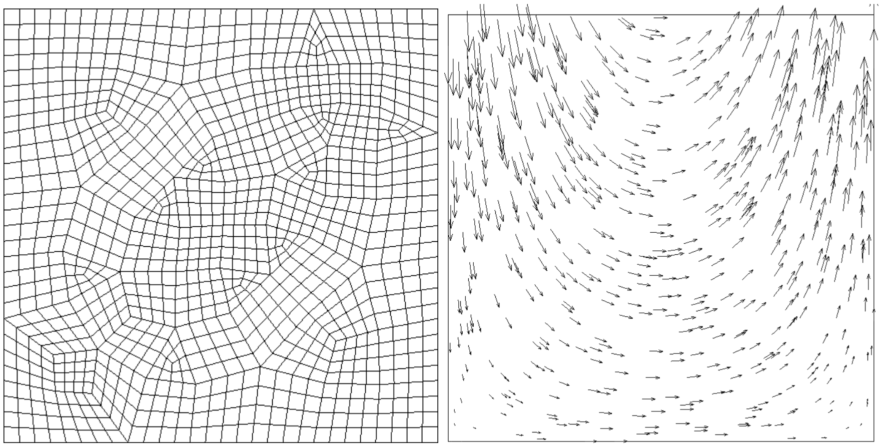

Figure 1. Example for an unstructured mesh (left). A series of those meshes is used for the computations to avoid super convergence effects. The solution of Problem 1 is shown on the right-hand side.

approximation properties of the mesh and the finite element space. This is especially true for higher order elements.

The inner blocks are solved using the direct solver UMFPACK, see [36]. If we use a Krylov method for the inner solve, it is stated.

5.1. The Oseen problem

Problem 1 is defined in the domain $\Omega = (0,1)^2$, see [37], example 1. The right-hand side $\boldsymbol{f}$ is calculated from the smooth reference solution

$$\begin{aligned} \boldsymbol{u} &= \left(\sin(\pi x), -\pi y \cos(\pi x)\right)^T, \\ p &= \sin(\pi x)\cos(\pi y) \end{aligned}$$

with convection vector $\boldsymbol{b} = \boldsymbol{u}$ or $\boldsymbol{b} = 0$, see Figure 1 for an illustration of the solution. Note that ν and c can be chosen arbitrarily. Thus, we can test coefficient choices including the Stokes problem. The numerical error can be calculated as the difference between the discrete solution and the reference solution. The case $\boldsymbol{b} = \boldsymbol{u}$ is more complicated than examples often chosen for Oseen problems and resembles a linearization step in a Navier-Stokes problem. The smooth solution enables higher order elements to achieve better convergence rates.

Problem 2 is a modified Green-Taylor vortex for a fixed time step and without an exponential decaying term:

$$\begin{aligned} \boldsymbol{u} &= \left(-\cos(\omega\pi x)\sin(\omega\pi y), \sin(\omega\pi x)\cos(\omega\pi y)\right)^T, \\ p &= -\frac{1}{4}\cos(2\omega\pi x) - \frac{1}{4}\cos(2\omega\pi y). \end{aligned}$$

Considering $\Omega = (0,1)^2$ the constant ω determines the number of vortices in x- and y-direction and is set to $\omega = 4$. $\boldsymbol{b}$ and the right-hand side $\boldsymbol{f}$ are defined in the same way as in Problem 1. This results in a more complex structure compared to *Problem 1*. Care needs to be taken to not get over-stabilization as can be seen from plots in Figure 2.

Figure 2 shows the influence of the stabilization on the quality of the solution and the number of iteration steps of the solver. We consider *Problem 1* and *Problem 2* with $\boldsymbol{b} = \boldsymbol{u}$. A similar value for the optimization of both would be desirable. The four plots in Figure 2 show different configurations. For large viscosities (upper left) stabilization does not improve the solution. Only for $\gamma > 1$ we see a slight influence. For smaller viscosity $\nu = 10^{-3}$ there is a clear minimum around $\gamma = 0.3$ for problem 1. We can observe this behavior with and without reaction term (upper right and lower right). The fine structures in *Problem 2* on the other hand are already damped too much with a γ of that size as can be seen in the lower left. The optimum moves to $\gamma = 0.03$ there.

DOI: 10.1002/fld

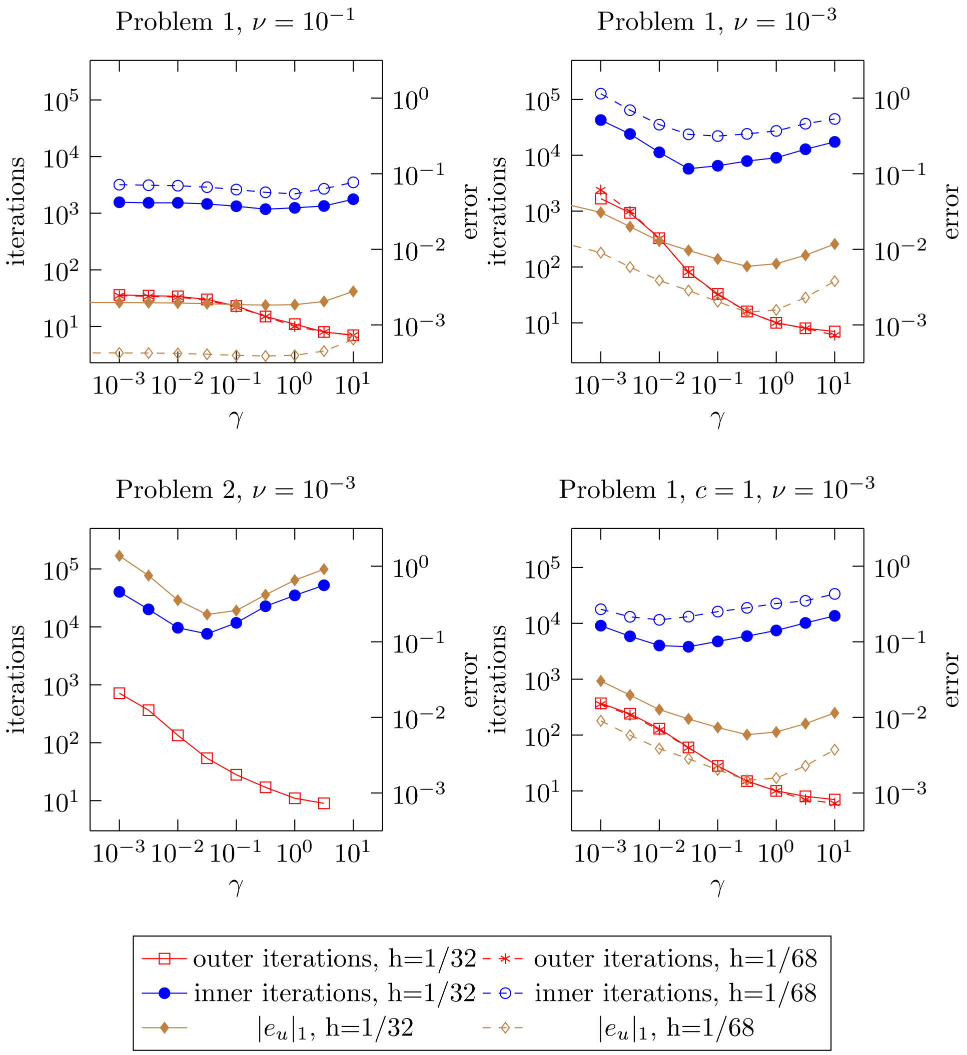

Figure 2. Analysis of solver performance and influence of the stabilization on the error with respect to the parameter choice γ. The outer iterations are given by the number of needed iterations for the preconditioned block system. The inner iterations show the total number of iterations for the A block summed up over all outer iterations. Note, that the number of inner iterations is heavily dependent on the preconditioner chosen for A (here, ILU(0)) and the result should only be considered quantitatively (Problem 1 and Problem 2 with $\boldsymbol{b} = \boldsymbol{u}$, unstructured mesh). The error is given in the H^1 semi-norm of the difference between the reference solution and calculated finite element solution.

The number of outer iterations drops for larger γ. The number of outer iterations needed for the optimal γ from the stabilization point of view is around 10 to 30. When using an iterative solver the difficulty of solving for A increases with larger γ. The solution time does not depend on γ for a direct solver of the A block. For an iterative solver it depends heavily on the iterative algorithm used for the A block. For an estimate of the total cost we also plot the number of total inner iterations needed to solve the whole system. In our applications the A block is solved using the GMRES method with ILU(0) preconditioning and diagonal strengthening. While the number of inner iterations increases for larger γ, the number of outer iterations decreases. Fortunately, the optimal choices of γ lie close to the optimal choices with respect to the stabilization. Choosing a worse preconditioner for A, like SOR, the optimum would shift to the right. Thus, the results for the sum of the inner iterations has to be taken with cautiousness as it depends on the preconditioner used. We still include the total number of inner iterations, because it shows, that it is possible to choose γ in such a way that the

			γ=1.0			γ=0.31			γ=0.1		
	h:	ν=	10^{-1}	10^{-3}	10^{-5}	10^{-1}	10^{-3}	10^{-5}	10^{-1}	10^{-3}	10^{-5}
Q2Q1	1/16		13	13	13	19	19	20	28	38	38
	1/32		13	12	12	19	19	19	28	38	38
	1/64		13	12	12	18	19	19	27	37	37
Q3Q2	1/16		13	13	13	19	20	20	29	38	38
	1/32		13	12	13	19	19	19	27	37	38
	1/64		13	12	12	18	19	19	27	36	37
Q4Q3	1/16		13	13	13	19	20	20	28	37	38
	1/32		13	12	13	19	19	19	27	37	37
	1/64		13	12	13	18	19	19	27	36	36

Table I. Number of outer FGMRES iterations for different problem sizes with different order of finite element spaces (Problem 1, regular mesh, stopping criterion: relative residual of 10^{-10}). The number of iterations is clearly independent of mesh size h and element order. Independence of the viscosity is achieved for $\gamma = 1$ and the optimal value $\gamma = 0.31$.

	$\#\{M_{ij} \neq 0\}$	$\#\{\lvert M_{ij}\rvert > 10^{-15}\}$
Galerkin	50884	15642
with Grad-Div	72250	27966
with augmentation	231704	47755

Table II. Number of non-zero elements in the system matrix M (Problem 1, structured mesh, h=1/16, Q_2-Q_1, 2178+289=2467 unknowns). The second column represents the number of elements in the matrix with an absolute value bigger than 10^{-15}. The rows represent the base matrix without stabilization, the matrix with Grad-Div stabilization, and the matrix with the augmented Lagrangian term.

	$\gamma = 1$			$\gamma = 0.1$			$\gamma = 0$			PCD ($\gamma = 0$)		
ν	it	factor/s	solv/s	it	factor/s	solv/s	it	factor/s	solv/s	it	factor/s	solv/s
10^{-1}	5	4.4	2.9	13	4.4	7.0	21	2.5	6.8	20	2.8	7.2
10^{-2}	5	4.4	2.9	21	4.4	10.7	229	2.5	80.9	33	2.8	11.6
10^{-3}	5	4.4	2.6	23	4.5	10.2	-	-	-	390	3.0	138.8
10^{-5}	5	4.5	2.7	23	4.4	11.1	-	-	-	-	-	-

Table III. Timings for Problem 1 on an irregular mesh with 19280 cells (stopping criterion: relative residual of 10^{-6}, sub-problems are solved with a direct solver). The number of iterations (it) and the seconds to setup (factor) and solve (solv) the system are given for different γ and different viscosities. For comparison the computational times using PCD are also given. A dash denotes no convergence in 500 iterations.

number of iterations and the error of the solution are small at the same time. All in all the number of outer iterations does not crucially depend on the choice of γ. It is reasonable to calculate solutions without the optimal parameter at hand.

Table I shows that the number of outer iterations does not depend on the element order or the mesh size. For sufficiently large γ the number of outer iterations is also independent of ν. The optimal value of $\gamma = 0.31$ results in a slightly larger number of iterations. For smaller γ there is a slight dependency on ν. This is (not surprisingly) comparable to the augmented Lagrangian approach.

In Table II one can see statistics about the number of non-zero elements comparing the Galerkin system matrix, the system with added Grad-Div stabilization, and the augmented Lagrangian formulation. The augmentation decreases the sparsity of the system by a huge margin. Note that the Grad-Div stabilization only adds new entries, where the components of the velocity couple. If we consider for example a diffusion term given by the symmetric deformation tensor $\mathbb{D}$, Grad-Div stabilization would add no additional entries.

DOI: 10.1002/fld

	ν	c	$\|\|b\|\|$	γ	solver	h 1/8	 1/32	 1/128
Stokes	10^{-2}	0	0	0	PCD	16	18	17
				1	GD	6	5	5
				0	GD	16	18	17
Stokes+reaction	10^{-2}	1	0	0	PCD	31	29	28
				1	GD	7	7	6
				0	GD	58	94	88
Oseen	10^{-2}	0	> 0	0	PCD	14	14	13
				1	GD	6	5	5
				0	GD	14	14	13
Oseen+reaction	10^{-2}	1	> 0	0	PCD	25	24	22
				1	GD	7	7	6
				0	GD	45	52	48
reaction dom.	10^{-2}	100	> 0	0	PCD	8	10	10
				1	GD	6	6	5
				0	GD	9	10	11

Table IV. Number of iterations for different parameter choices and different mesh sizes (Problem 3, regular mesh) with a relative residual of 10^{-8}. The Grad-Div preconditioner with $\gamma = 1$ is compared to PCD and the basic preconditioner ($\gamma = 0$) explained in Section 3.

In Table III we measure the runtime[‡] for Problem 1 on a irregular mesh with 19280 cells using $Q2/Q1$ elements withour a reaction term. The table is set up to give comparable values to Table III in [13]. We use the direct solver UMFPACK for the inner problems. Although the mesh is 10% finer and not regular, our timings are very competitive compared to the augmented Lagrangian preconditioner. Of course, the setup time to assemble the matrices is independent of ν and γ. For comparison we include the timings and iterations numbers for the PCD preconditioner. There, factorization of the matrix is cheaper as the A-block only contains the Galerkin terms (see also Table II). The PCD performance deteriorates for small viscosities because the number of necessary iterations increases.

So far we have only looked at the Oseen problem. In Table IV we consider different prototypes of equations. *Problem 3* is given by the polynomial solution[§]

$$\begin{aligned} \boldsymbol{u} &= (4(2y-1)(1-x)x, -4(2x-1)(1-y)y)^T, \\ p &= x^2 + y^2 - 2/3. \end{aligned}$$

We compare the number of iterations with the same block preconditioner for $\gamma = 1$ and $\gamma = 0$. The last choice coincides with the traditional way of preconditioning for example the Stokes problem (see [9]). The choice $\gamma = 1$ improves the quality of the solution in all cases. Surprisingly the Grad-Div preconditioner also helps for the pure Stokes problem. This is most likely due to the fact that the approximation of the diffusion in the Schur complement is worse than the exact approximation for the Grad-Div term. Since the splitting of the Schur complement in diffusion and algebraic term is exact (see (16)), no additional error is introduced there. The new preconditioner shows its main advantage in the convection dominated case. The original preconditioner does barely work there. The reaction term helps especially in the reaction dominated case (which represents a time dependent problem with small time step sizes).

[‡] All problems were run on an Intel core 2 duo Laptop with 2.5ghz on one core. The code is neither optimized, nor implemented to take advantage of multiple cores.

[§] We chose a solution without in- or outflow, to avoid difficulties with the PCD preconditioner that we use for comparison.

DOI: 10.1002/fld

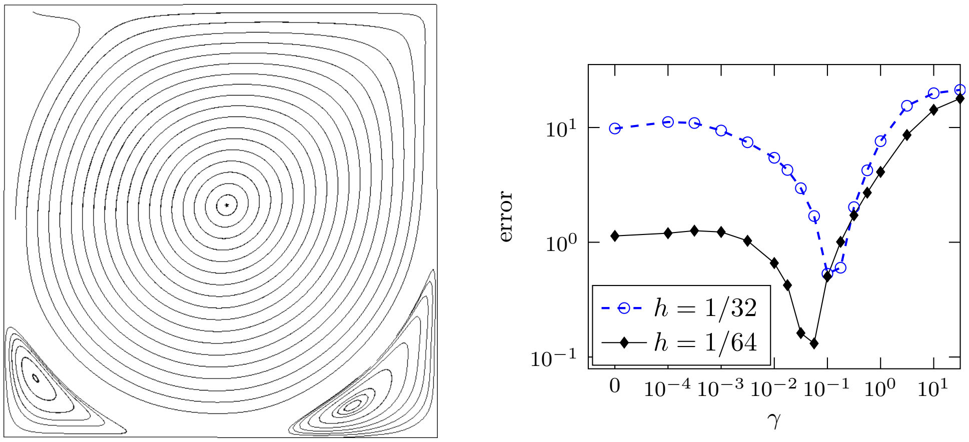

Figure 3. Lid driven cavity flow with Re=5000. Left: selected streamlines. Right: Error in minimum of the stream function (see [38]) dependent on the Grad-Div stabilization γ for two different meshes. Grad-Div stabilization decreases the error by more than one order of magnitude, which is better than a regular refinement.

5.2. Application to Driven Cavity Flow

Let us now consider a more involved application: the two dimensional lid-driven cavity flow. We consider Reynolds numbers up to $Re = 5000$, which still results in stationary solutions. The non-linear system is solved on the unit-square with right-hand side $\boldsymbol{f} = 0$ and typical boundary conditions on $\partial\Omega$:

$$\boldsymbol{u}(x, y) = (u(x, y), v(x, y)) = \begin{cases} (1, 0) & \text{for } y = 0, \text{ and } 0 < x < 1 \\ (0, 0) & \text{else,} \end{cases}$$

i.e. we describe a force to the right at the top border. See [38] for a discussion for the problem with extensive numerical reference data. A plot of a few selected streamlines is given in Figure 3 (left).

We solve the non-linear Navier-Stokes problem with a damped fixed point iteration with a simple backtracking algorithm. All calculations are done on uniform and coarse meshes, which means the boundary layers are not resolved (especially for Re 5000). This is done for two reasons: First, not being able to resolve boundary layers is realistic in most applications and results in increased difficulty for the linear solvers. Second, fully resolved flow does not require stabilization and will therefore not benefit from Grad-Div stabilization. In each iteration step one has to solve a Oseen type problem where the convection vector is given by the last iterate of the velocity.

An important quantity of interest is the minimum of the stream function, see [38] for reference values. For meshes with $h = 1/128$ we could reproduce the results for Re 5000 without stabilization but the quality of the minimum of the stream function and the cuts of the velocity increases significantly with Grad-Div stabilization for coarser meshes. Figure 3, right shows the error of the minimum of the stream function in percent depending on the mesh size and Grad-Div parameter. The optimal γ is at 10^{-1} and is slightly mesh size dependent and tends to zero for finer meshes. The advantage of using Grad-Div stabilization for $h = 1/32$ is fairly obvious and one can gain one order of magnitude in the quality, which is more than a regular refinement. For smaller Reynolds numbers the importance of Grad-Div decreases. While it is still an half an order of magnitude for Re=1000 (optimal value at $\gamma = 0.1$), the effect vanishes for For Re=100. The velocity profiles also improve, which can be seen in Figure 4. We plot the second component of the velocity on a horizontal cut in the middle through the domain.

The non-linear iteration is done until the residual is smaller than 10^{-9} and each linear problem is solved with a relative residual of 10^{-2} with respect to the starting residual (which is the same as the non-linear residual). We compare the average number iterations for different choices of Grad-Div stabilization in Table V. For comparison we also included the numbers for the PCD preconditioner,

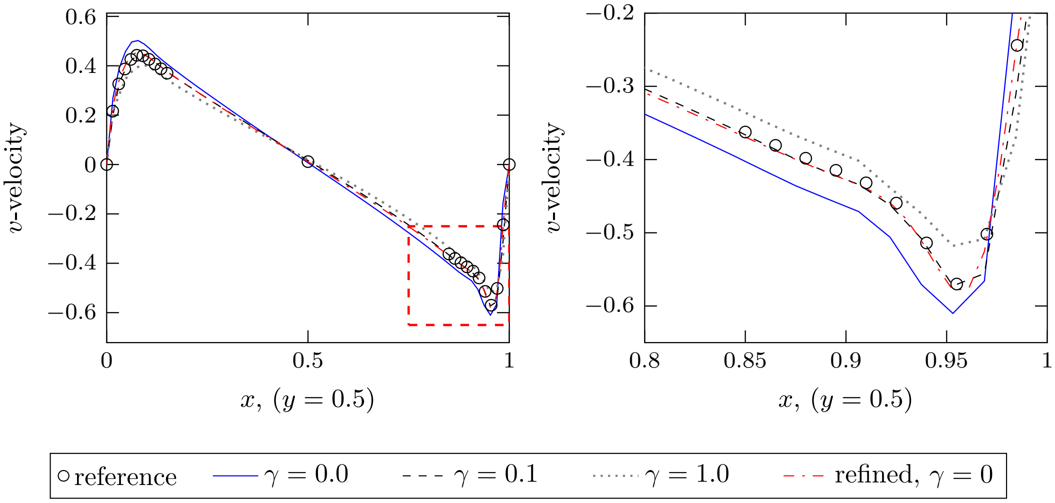

Figure 4. v-component of the velocity on a horizontal cut in the middle of the cavity with Re=5000 on a regular mesh with $h = 1/32$ and different values for Grad-Div stabilization in comparison with reference values from [38] and a comparison on a finer mesh with $h = 1/64$ and no Grad-Div stabilization. Right: zoom of left picture.

			$h = 1/32$			$h = 1/64$	
viscosity	Grad-Div	PCD	GD	#nonlinear	PCD	GD	#nonlinear
$\nu = 10^{-2}$	$\gamma = 0$	13	18	15	13	18	15
	$\gamma = 0.1$	17	**4**	15	16	**5**	15
$\nu = 10^{-3}$	$\gamma = 0$	44	342	34	42	511	29
	$\gamma = 0.1$	91	**6**	31	109	**8**	29
$\nu = 2 \cdot 10^{-4}$	$\gamma = 0$	4822	-	104	1031	-	49
	$\gamma = 0.1$	1064	**7**	40	1249	**8**	43

Table V. Number of non-linear iterations and average number of linear iterations per non-linear step for the PCD and the Grad-Div preconditioner in comparison for the cavity with $Re = 100$, $Re = 1000$ and $Re = 5000$ on a regular mesh. The optimal γ from the error point of view is selected.

see [7]. The two methods are comparable for small Reynolds numbers, where the Grad-Div based preconditioner also works without Grad-Div stabilization. We again see the excellent scaling with the viscosity as in the earlier tests. This is in contrast to the PCD preconditioner, which fails for small viscosities. One can find similar behavior in e.g. [12].

6. REVIEW OF THE PRECONDITIONER

We now discuss the advantages and disadvantages of the Grad-Div preconditioner.

Let us start with the advantages: The number of iteration steps in the solver is independent of h and ν and gives small iteration counts comparable to the augmented Lagrangian approach, see the results in Section 5. This is an excellent behavior compared to other preconditioners.

The usage of Grad-Div stabilization improves the accuracy of the discretization scheme, cf. [23]. The preconditioner has a wide range of applicability. It can be used for Stokes, Oseen or Navier-Stokes problems, in transient and stationary cases and in different ranges of viscosity. See Table IV for an example.

Additionally, the Grad-Div preconditioner is easy to apply. Assembling the linear system is straight-forward. There are no complicated, additional matrices to be assembled for the preconditioner. In contrast, the matrix A_γ in the augmented Lagrangian is much harder to handle, since it is not immediately available as a matrix. One can either implement it as an operator, which restricts the choice of the preconditioner (which is a huge disadvantage, but some work to remedy exists in [13]) or one has to simplify the mass matrix M_p in the term $B^T M_p^{-1} B$ by lumping for example. Multiplying two sparse matrices is an expensive process and the resulting matrix is more dense than the Grad-Div preconditioned matrix, which only contains additional couplings between the velocity components. In many finite element packages the integration of augmented Lagrangian-type preconditioners is not simple. Often the degrees of freedom on the boundary are treated in the same way as inner degrees of freedom and they get eliminated before or after writing them to the global matrix. This can result in either loosing the symmetry in the B, B^T blocks or in non-zero entries in the second block of the right-hand side. Implementing the Grad-Div preconditioner does not pose any of these difficulties.

Our proposed approach also possesses some disadvantages. The Grad-Div stabilization is needed for the Grad-Div preconditioner even when stabilization is not necessary. This adds additional coupling to the A block and slows down the assembly process. Fortunately, most of the real life problems need additional stabilization, see [23]. If the traditional Laplacian in the Navier-Stokes problem is replaced by symmetrized deformation tensors $\mathbb{D}$, as it is regularly done in turbulent flow simulations, the Grad-Div stabilization does not produce additional non-zero entries in the A-block. Even though assembly of the Grad-Div stabilization is simple, approximating the velocity block in the block preconditioner becomes more expensive. As already known from [12], increasing γ can be seen as a parameter to shift the difficulty from the outer problem to the inner velocity block.

Choosing the stabilization coefficient can not be solely done from the stabilization point of view. The preconditioner performance has to be taken into account, too. Different choices of γ influence the quality of the solution. As for most of the preconditioning techniques preconditioning and stabilization can not be treated independently. As with most stabilization methods, the optimal Grad-Div parameter from the stabilization point of view is problem dependent. From the preconditioner side, parameters around 0.1 seem to be optimal in most cases and the performance seems to be robust regarding the parameter choice.

A general problem of the proposed class of preconditioners is the assembling of new matrices, which are not part of the primal problem. For the approximate Schur complement $\tilde{S}$, defined in (13), one has to assemble and store the matrices M_p and L_p.

Summarized, we think that the advantages clearly outbalance the disadvantages.

7. CONCLUSIONS AND OUTLOOK

We presented a new preconditioner that shows to be competitive to state of the art solution strategies. It is especially useful in the case where Grad-Div stabilization is already employed. It is helpful to use Grad-Div stabilization in all kind of problems with different parameters and the preconditioner helps in any of those cases.

The preconditioner gives h, ν, and element order independent iteration numbers, as long as the Grad-Div parameter is in a sensible range. It is possible to satisfy good accuracy and fast performance of the solver, because the parameter is not crucially sensitive and is in the same order of magnitude. Especially the ν independence can not be found in the typical preconditioners used today.

Some things will be very interesting to look at but do not fit into the scope of this paper. Adapting the preconditioner to variable viscosity and varying Grad-Div parameter γ should be possible. It simply results in modifications in the Schur complement approximation. The diffusive part no longer reads

$$-(\nu+\gamma)M_p^{-1},$$

because $\nu+\gamma$ is no longer a constant. Small variations in ν (for example with turbulence models) do not make a difference for the preconditioner, because ν is still quite small compared to the reaction term for example. Choosing an average for $\nu+\gamma$ is a practical solution. When dealing with large jumps from one cell to another one has to include it this in the Schur complement. In [39] the different ways of how to do that are explained. The best was is to move the coefficients into the assembly of a modified mass matrix:

$$(M_p^{\nu,\gamma})_{ij} = \int_\Omega (\nu(x) + \gamma(x))^{-1} \phi_i \phi_j \ \mathrm{d}\Omega.$$

Alternatively one can scale the mass matrix with an approximation for each degree of freedom.

Another interesting aspect is to combine the augmented Lagrangian approach with the Grad-Div stabilization. This is straight-forward and could help in the case where the Grad-Div parameter has to be chosen very small for accuracy reasons. Let γ be the Grad-Div parameter. One can then additionally augment the system with $\gamma' B^T M_p^{-1} B$. The diffusive part of the Schur complement is then chosen as

$$-(\nu + \gamma + \gamma') M_p^{-1},$$

and one can change γ', so that $\gamma + \gamma'$ is optimal for preconditioning.

ACKNOWLEDGEMENTS

The majority of the research was done, while both authors worked at University of Göttingen, Germany. Timo Heister was partially supported by the German Research Foundation (DFG) through Research Training Group GK 1023. This publication is based in part on work supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

We would like to thank Maxim Olshanskii for the helpful discussion.

REFERENCES

1. Röhe L, Lube G. Analysis of a variational multiscale method for Large-Eddy simulation and its application to homogeneous isotropic turbulence. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(37-40):2331 – 2342.
2. Bramble J, Pasciak J, Vasiliev A. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM J. Numer. Anal.* 1997; **34**:1072–1092.
3. Guermond JL, Minev P, Shen J. An Overview of Projection methods for incompressible flows. *Comp. Meth. Appl. Mech. Engrg.* 2006; **195**:6011–6045.
4. John V. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations. *Int. J. Numer. Methods Fluids* 2002; **40**(6):775–798.
5. de Niet AC, Wubs FW. Two Preconditioners for Saddle Point Problems in Fluid Flows. *Int. J. Numer. Methods Fluids* 2006; .
6. Elman HC. Preconditioning for the Steady-State Navier–Stokes Equations with Low Viscosity. *SIAM Journal on Scientific Computing* 1999; **20**(4):1299–1316.
7. 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**(4):665–688.
8. Benzi M, Golub GH, Liesen J. Numerical Solution of Saddle Point Problems. *Acta Numerica* 2005; **14**:1–137.
9. Turek S. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.
10. Fortin M, Glowinski R. Augmented lagrangian methods: Applications to the numerical solution of boundary value problems. *Stud. Math. Appl* 1983; **15**.
11. Glowinski R, Le Tallec P. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. Society for Industrial Mathematics, 1989.
12. Benzi M, Olshanskii MA. An Augmented Lagrangian-Based Approach to the Oseen Problem. *SIAM J. Sci. Comput* 2006; **28**:2095–2113.
13. Benzi M, Olshanskii MA, Wang Z. Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. *Int. J. Numer. Methods Fluids* 2011; **66**:486–508.
14. Benzi M, Wang Z. Analysis of augmented lagrangian-based preconditioners for the steady incompressible navier-stokes equations. *Math/CS Technical Report TR-2010-011* 2010.
15. Silvester D, Wathen A. Fast Iterative Solution of Stabilised Stokes Systems Part II: Using General Block Preconditioners. *SIAM J. Numer. Anal.* 1994; **31**(5):1352–1367.
16. Elman H, Howle V, Shadid J, Silvester D, Tuminaro R. Least squares preconditioners for stabilized discretizations of the navier-stokes equations. *SIAM J. Sci. Comput* 2007; **30**(1):290–311.
17. Franca L, Frey S. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**(2-3):209–233.
18. Olshanskii MA, Reusken A. Grad-Div Stablilization for Stokes Equations. *Math. Comput.* 2004; **73**(248):1699–1718.
19. Heister T. Vorkonditionierungsstrategien für das stabilisierte Oseen-Problem. Master's Thesis, Institut für Numerische und Angewandte Mathematik, Georg-August-Universität zu Göttingen 2008.
20. Hughes TJ, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. circumventing the babuska-brezzi condition: a stable petrov-galerkin formulation of the stokes problem

DOI: 10.1002/fld

accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**(1):85 – 99.

21. Braack M, Burman E. Local Projection Stabilization for the Oseen Problem and its Interpretation as a Variational Multiscale Method. *SIAM J. Numer. Anal.* 2006; **43**(6):2544–2566.
22. Girault V, Raviart PA. *Finite Element Methods for the Navier–Stokes Equations*. Springer–Verlag: New York, 1986.
23. Olshanskii M, Lube G, Heister T, Löwe J. Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(49-52):3975 – 3988.
24. Lube G, Rapin G, Löwe J. Local Projection Stabilizations for Incompressible Flows: Equal-Order vs. Inf-Sup Stable Interpolation. *ETNA* 2008; **32**:106–122.
25. Matthies G, Skrzypacz P, Tobiska L. A unified convergence analysis for local projection stabilisations applied to the Oseen problem. *Mathematical Modelling and Numerical Analysis* 2007; **41**(4):713–742.
26. Roos H, Stynes M, Tobiska L. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*. Springer Verlag, 2008.
27. Johnson C, Saranen J. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Mathematics of Computation* 1986; **47**(175):1–18.
28. Matthies G, Lube G, Röhe L. Some remarks on residual-based stabilisation of inf-sup stable discretisations of the generalised Oseen problem. *Computational Methods in Applied Mathematics* 2009; **9**(4):368–390.
29. Saad Y. *Iterative Methods for Sparse Linear Systems*. Second edn., Society for Industrial and Applied Mathematics: Philadelphia, PA, 2003.
30. Cahouet J, Chabard J. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. Numer. Methods Fluids* 1988; **8**(8):869–895.
31. Knabner P, Angermann L. *Numerik partieller Differentialgleichungen*. Springer–Verlag: Berlin, Heidelberg, New York, 2000.
32. Lube G, Rapin G. Residual-Based Stabilized Higher-Order FEM for a Generalized Oseen Problem. *Math. Models & Meth. in Applied Sciences 16* 2006; :1–18.
33. Bangerth W, Kanschat G. deal.II *Differential Equations Analysis Library, Technical Reference* 2011. http://www.dealii.org.
34. Bangerth W, Hartmann R, Kanschat G. deal.II — a General Purpose Object Oriented Finite Element Library. *ACM Transactions on Mathematical Software* Aug 2007; **33**(4):27.
35. Saad Y. A Flexible Inner-Outer Preconditioned GMRES Algorithm. *Technical Report 91-279*, Minnesota Supercomputer Institute, University of Minnesota 1991.
36. Davis TA. Algorithm 832: UMFPACK V4.3 - An Unsymmetric-Pattern Multifrontal Method. *ACM Trans. Math. Softw* 2004; **30**(2):196–199.
37. Gelhard T, Lube G, Olshanskii MA, Starcke J. Stabilized Finite Element Schemes with LBB-Stable Elements for Incompressible Flows. *J. Comput. Appl. Math.* 2005; **177**(2):243–267.
38. Erturk E, Corke T, Gökçöl C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int. J. Numer. Methods Fluids* 2005; **48**(7):747–774.
39. ur Rehman M, Geenen T, Vuik C, Segal G, MacLachlan S. On iterative methods for the incompressible Stokes problem. *Int. J. Numer. Methods Fluids* 2011; **65**:1180–1200.

DOI: 10.1002/fld