J Sci Comput
DOI 10.1007/s10915-013-9684-1

Multigrid Methods for the Stokes Equations using Distributive Gauss–Seidel Relaxations based on the Least Squares Commutator

Ming Wang · **Long Chen**

Received: 25 April 2012 / Revised: 25 December 2012 / Accepted: 6 January 2013

Abstract A distributive Gauss–Seidel relaxation based on the least squares commutator is devised for the saddle-point systems arising from the discretized Stokes equations. Based on that, an efficient multigrid method is developed for finite element discretizations of the Stokes equations on both structured grids and unstructured grids. On rectangular grids, an auxiliary space multigrid method using one multigrid cycle for the Marker and Cell scheme as auxiliary space correction and least squares commutator distributive Gauss–Seidel relaxation as a smoother is shown to be very efficient and outperforms the popular block preconditioned Krylov subspace methods.

Keywords Multigrid · Stokes · Finite element

1 Introduction

How to effectively solve large scale algebraic systems arising from the discretization of partial differential equations is a fundamental question in scientific and engineering computing. For the positive definite linear systems corresponding to elliptic boundary value problems, multigrid (MG) methods are proven to be one of the most efficient algorithms [9,10,29,56]. However, it is much more challenging for saddle-point systems [5]. In this paper, we consider multigrid methods for solving the linear saddle-point algebraic system arising from finite element methods (FEM) discretization of the stationary Stokes equations

M. Wang
LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, China
e-mail: wangming.pku@gmail.com

L. Chen (✉)
Department of Mathematics, University of California at Irvine, Irvine, CA 92697, USA
e-mail: chenlong@math.uci.edu

Published online: 05 February 2013

$$\begin{cases} -\Delta \boldsymbol{u} + \operatorname{grad} p = \boldsymbol{f} & \text{in } \Omega, \\ -\operatorname{div} \boldsymbol{u} = 0 & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{g}_D & \text{on } \partial\Omega, \end{cases} \tag{1.1}$$

where $\boldsymbol{u}$ is the velocity field, p represents pressure, and $\boldsymbol{f}$ is the external force field. For the ease of exposition, the presentation is devoted to domains Ω in $\mathbb{R}^2$ and Dirichlet boundary condition. Our methods can be easily generalized to domains in three dimensions and other boundary conditions.

Among various existing solvers for the saddle point systems, one may distinguish between Krylov iterative methods with block preconditioners and multigrid methods. For various preconditioning techniques, we refer to [5,7,8,32] and the references therein. In this paper, we focus on multigrid methods.

Several branches of efficient smoothers have been developed for the Stokes equations. They can be roughly classified into two categories: coupled and decoupled smoothers [41]. Coupled smoothers [44,51], also known as Vanka smoothers, are solving a small saddle point system at a grid point or an appropriate patch. Decoupled smoothers, i.e., equation-wise relaxation, have an advantage in their efficiency especially when line-wise smoothers are needed [41]. The first decoupled smoother is the DGS smoother introduced in [11]. Later on, it was generalized to the incomplete LU factorization (ILU) smoother for a transformed system [54,55] and was shown numerically effective [19,54]. Recently, DGS relaxation has also been designed for the linear elasticity [59] and poroelastic system [24,25,53]. Some other effective decoupled smoothers can be found in [4,6,48] and will be recalled in later sections for comparison.

Among the above mentioned decoupled smoothers, DGS-type smoothers (including the ILU smoother based on the transformed system) seem to be more efficient when applicable. This type of smoother, however, is only known for the Marker and Cell (MAC) scheme discretization and mini finite element on rectangular grids [19,54]. Generalization to other stable pairs of finite element methods seems difficult; see [29, p. 248]. A recent attempt based on high regularity of the Laplacian can be found in [3].

One main contribution of this paper is to develop DGS-type smoothers for the linear system arising from finite element discretizations of the Stokes equations. The success of the DGS smoother depends on the existence of a Laplacian operator Δ_p for the discrete pressure space such that the commutator $-\Delta\operatorname{grad} + \operatorname{grad}\Delta_p$ is small. Let A, B be the finite element discretization of operators $-\Delta$ and $-\operatorname{div}$, respectively. We shall chose $A_p = (BB')^{-1}BAB'$ as the approximation of operator $-\Delta_p$. Then the commutator can be expressed as $AB' - B'A_p = (I - B'(BB')^{-1}B)AB'$. Observe that $P = I - B'(BB')^{-1}B$ is a projection operator orthogonal to the range of B' and therefore $AB' - B'A_p = P(AB' - B'X)$ for any operator X. As a result, we minimize the Frobenius norm of the commutator in the least squares sense and thus call the corresponding DGS smoother as Least Squares Commutator DGS (LSC-DGS) smoother. In most existing smoothers, e.g., Braess-Sarazin smoother [6], SIMPLE smoother [43], and inexact Uzawa methods [4,6,60] etc, a scalar parameter which could depend on the eigenvalues of matrices under consideration, should be determined prior to the iteration. In contrast, our LSC-DGS smoother is parameter-free and can be implemented on the algebraic level. Numerically LSC-DGS works very well for all examples tested in this work.

Although we found the LSC approximation A_p independently, it has been developed in [20] (see also [21,22]), and wherein $BB'A_p^{-1}$ is further used as an approximation to the Schur complement $BA^{-1}B'$, resulting in the so-called BFBt preconditioner. In this paper, the LSC approximation A_p is used to construct an efficient smoother.

Another main ingredient of multigrid methods is the coarse grid correction. In addition to standard geometric multigrid methods, we can use MAC scheme as a 'coarse grid correction'. More precisely, in one $\mathcal{V}$-cycle, we first perform a pre-smoothing with an effective smoother for the Stokes equations, then call one multigrid $\mathcal{F}$-cycle for the MAC scheme [11,31], and finally complement with a post-smoothing. Notice that MAC is not always a subspace of the discretization on the fine grid, e.g., discretization using a continuous pressure space. From this point of view, our method is in the spirit of the auxiliary space method [57] and thus named ASMG. It is also similar to defect correction multigrid methods [2], and the double discretization scheme [10].

Numerical experiments are provided to show that standard geometric multigrid methods with a few LSC-DGS relaxations converge uniformly with respect to the grid size h. It is much more efficient and robust than inexact Uzawa smoothers [4,6,60], especially on the unstructured grids. On rectangular grids, the proposed auxiliary space multigrid solver works best and attains the solution in just a few normalized work units for several popular finite element discretizations. The combination of LSC-DGS smoothing and ASMG (ASMG/LSC-DGS) is more efficient and robust than the other methods tested. In particular, ASMG/LSC-DGS outperforms the popular preconditioned Krylov spaces methods being two to three times faster [19,26]. Compared with the work in [19], where traditional multigrid method with DGS or ILU smoother is only applicable for MAC scheme and mini element discretization, the present multigrid solver with the LSC-DGS smoother fills this gap for other finite element discretizations.

The rest of this paper is organized as follows. In Sect. 2, we review the DGS smoother and multigrid methods for the MAC scheme. In Sect. 3, we collect notation of the discrete setting and discuss DGS-type smoothers for finite element discretization of the Stokes equations. In Sect. 4, we describe our auxiliary space multigrid algorithm. In Sect. 5, we first give a comparative study of the operation cost for the multigrid methods and preconditioned Krylov subspace methods and then present some numerical experiments to show the efficiency of our method. We also give conclusions and point to possible areas of future research in the last section.

2 DGS Smoother and Multigrid for the MAC Scheme

In this section, we recall the well-known MAC scheme [31] and the multigrid method using an effective DGS smoother [11].

2.1 MAC Scheme

The MAC scheme (see Harlow and Welch [31]) is to discretize the Stokes equations (1.1) on staggered grids. The three unknowns u, v and p are defined at different positions on the grid, as Fig. 1 shows: the discrete values of pressure p are defined at cell centers (•), and the discrete values of velocity u and v are located at the grid cell faces (× and ∘). Some values outside of the boundary must be taken care of by extrapolation, which should be at least linear, in order not to spoil the whole approximation order [18]. To keep the notation simple, the mesh size parameter h is skipped in u, v, p.

The Stokes equations are discretized using the nearest-neighbor central differences. More precisely, the x- and y-momentum equations are discretized at centers of vertical edges and centers of horizontal edges, respectively, with the standard five-point centered approximation to $-\Delta$ and central difference approximation to the grad operator. The discretization of the

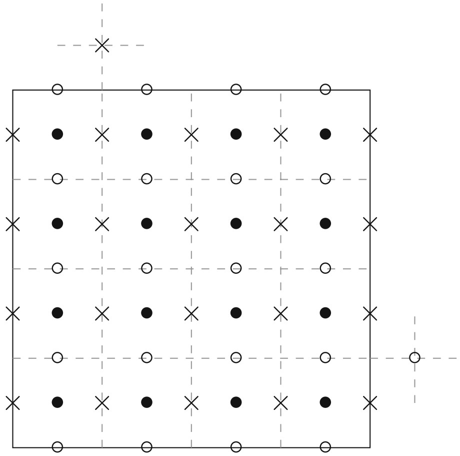


Fig. 1 Staggered grid location of unknowns for the MAC scheme. The discrete pressure p is defined at cell centers (*filled circle*). The discrete velocity u and v are defined at vertical edges centers (*times*) and horizontal edges centers (*open circle*), respectively

continuity equation div $\boldsymbol{u} = 0$ is defined at cell centers, with the central difference approximation to the div operator. All together, the discrete approximation of the matrix-vector form of (1.1) reads as

$$\mathcal{L}\boldsymbol{x} = \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} := \boldsymbol{b}, \tag{2.1}$$

where $\boldsymbol{x} = (\boldsymbol{u}, p)^t$ denotes the grid function, and A, B and B' are discrete approximations of operators $-\Delta$, $-\operatorname{div}$, and grad, respectively. An important feature of the MAC scheme is that discrete operators mimic differential operators. For example, BB' will be the standard five-point centered approximation to $-\Delta_p$ for the pressure, with Neumann boundary conditions [17,27]. Analysis and convergence of the MAC scheme can be found in, e.g., [18,30,37,39].

2.2 DGS Smoother

The standard relaxations, e.g., the Gauss–Seidel relaxation, are not applicable to the system (2.1), since $\mathcal{L}$ is not diagonally dominant, and especially the zero block in the diagonal hampers the relaxation. The idea of the distributive relaxation is to transform the principle operators to the main diagonal and apply the equation-wise decoupled relaxation.

For the Stokes equations, multiplying $\mathcal{L}$ with a right-side operator $\mathcal{M}$ given by

$$\mathcal{M} = \begin{pmatrix} I & B' \\ 0 & -BB' \end{pmatrix}, \tag{2.2}$$

yields

$$\mathcal{L}\mathcal{M} = \begin{pmatrix} A & W \\ B & BB' \end{pmatrix} \approx \begin{pmatrix} A & 0 \\ B & BB' \end{pmatrix} := \widetilde{\mathcal{L}\mathcal{M}}, \quad \text{with } W = AB' - B'BB',$$

in the block lower-triangular form, which is well-suited for the standard relaxations. By "≈" here we mean that the commutator W is zero in the interior of Ω (i.e., when applied to p vanishing in a certain neighborhood of $\partial\Omega$ [17,54]) and is of low rank. Thus it may be omitted in order to design relaxation methods.

Now the transformed operator $\widetilde{\mathcal{L}\mathcal{M}}$ is diagonally dominant and thus can be easily solved or relaxed. Suppose $\widetilde{\mathcal{L}\mathcal{M}}$ is further approximated by

$$\mathcal{S} = \begin{pmatrix} \hat{A} & 0 \\ B & \hat{A}_p \end{pmatrix}, \tag{2.3}$$

where $\hat{A}$ and $\hat{A}_p$ are easily invertible approximations of A and $A_p := BB'$, respectively. The matrix $\mathcal{M}\mathcal{S}^{-1}$ defines an iterative method for the original system (2.1):

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mathcal{M}\mathcal{S}^{-1}(\boldsymbol{b} - \mathcal{L}\boldsymbol{x}^k). \tag{2.4}$$

One iteration of (2.4) can be performed by the following algorithm.

```
Algorithm 2.1. $[\boldsymbol{u}^{k+1}, p^{k+1}] \leftarrow \texttt{DGS}(\boldsymbol{u}^k, p^k)$
  1. Relax momentum equations
         $\boldsymbol{u}^{k+\frac{1}{2}} = \boldsymbol{u}^k + \hat{A}^{-1}(\boldsymbol{f} - A\boldsymbol{u}^k - B'p^k)$,
  2. Relax transformed continuity equations
         $\delta q = \hat{A}_p^{-1}(0 - B\boldsymbol{u}^{k+\frac{1}{2}})$.
  3. Transform the correction back to the original variables
         $\boldsymbol{u}^{k+1} = \boldsymbol{u}^{k+\frac{1}{2}} + B'\delta q$,
         $p^{k+1} = p^k - BB'\delta q$.
```

The so-called DGS smoother introduced by Brandt and Dinar [10,11] is derived from consecutive Gauss–Seidel relaxation for the operator $\widetilde{\mathcal{L}\mathcal{M}}$, i.e., $\hat{A}$ and $\hat{A}_p$ are taken as the lower or upper triangular parts of the matrix of A and BB', respectively. The name distributive relaxation comes from the fact that the approximated correction $\mathcal{S}^{-1}(\boldsymbol{b} - \mathcal{L}\boldsymbol{x}^k)$ in (2.4) is distributed over the entries of $\boldsymbol{x}^{k+1}$ through the distributive matrix $\mathcal{M}$.

Instead of replacing $\mathcal{L}\mathcal{M}$ by a block triangular operator $\mathcal{S}$ in (2.3), the ILU smoother introduced by Wittum [54] applies an ILU-factorization to the operator $\mathcal{L}\mathcal{M}$, which results in a better performance.

Remark 2.1 One can choose $\hat{A}^{-1}$ and $\hat{A}_p^{-1}$ as one $\mathcal{V}$-cycle for the corresponding discrete Laplacian. With such a choice, the matrix $\mathcal{M}\mathcal{S}^{-1}$ might be used as a block preconditioner for solving system (2.1) with Krylov subspace methods. But this idea is not explored in this paper.

2.3 Multigrid Methods

We use the standard geometric multigrid method. Starting from a rectangular grid with a few elements, we consecutively refine each rectangle into four equal-size small rectangles to get a finer grid. Starting from the finest grid, we perform the DGS smoothing and restrict the residual equation to the coarser level. We solve the coarsest grid problem by a direct solver. While we have already discussed the smoother, now we move to the transfer operators.

2.3.1 Prolongation

Piecewise constant (first-order) interpolation is used for the p variable, and bilinear interpolation of neighboring coarse-grid unknowns in the staggered grid is utilized for the prolongation of velocity u and v.

2.3.2 Restriction

The restriction is the transpose of the prolongation with a suitable scaling. More precisely, at u- and v-grid points, we consider six points restrictions, and at p-grid points, a four-point cell-centered restriction. In stencil notation, the restriction operators are ($*$ indicates the position of the coarse-grid point)

$$R^u_{h,2h} = \frac{1}{8}\begin{pmatrix} 1 & 2 & 1 \\ & * & \\ 1 & 2 & 1 \end{pmatrix}, \quad R^v_{h,2h} = \frac{1}{8}\begin{pmatrix} 1 & & 1 \\ 2 & * & 2 \\ 1 & & 1 \end{pmatrix}, \quad R^p_{h,2h} = \frac{1}{4}\begin{pmatrix} 1 & & 1 \\ & * & \\ 1 & & 1 \end{pmatrix}.$$

For the isotropic ($h_x = h_y$) discretization, simple transfer operators are sufficient to obtain optimal rates, but not sufficient in the anisotropic case. The influence of various grid transfer operators is studied in [40].

For the MAC scheme, the multigrid method with the DGS smoothing is highly efficient in the sense that only a few normalized work units are required to achieve the desired tolerance [11]. For a local mode analysis of the DGS smoothing, we refer to Niestegge and Witsch [40] and for the convergence analysis of corresponding multigrid methods, see Wittum [55].

For systems arising from finite element discretizations, it is a challenge to determine a suitable distributive operator $\mathcal{M}$. The main difficulty is to construct A_p such that the commutator is small or of low-rank. To the authors' knowledge, besides the MAC scheme, the DGS relaxation can only be applied to the mini finite element on rectangular grids [19,54] and is not available for other finite element discretizations. This paper is the first attempt to design DGS-type smoothers for other finite element discretizations of Stokes equations.

3 DGS Smoother for Finite Element Discretizations

We design several DGS-type smoothers for stable finite element discretizations of system (1.1) in this section. By choosing different distributive matrices, we will address the standard DGS, a particular DGS for continuous pressure spaces, and a new LSC-DGS smoother.

3.1 Notation

The usual weak formulation of (1.1) (assume $\boldsymbol{g}_D = 0$ for simplicity) reads as follows: find $(\boldsymbol{u}, p) \in \mathbb{V} \times \mathbb{Q} := H^1_0(\Omega)^2 \times L^2_0(\Omega)$, such that

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = (\boldsymbol{f}, \boldsymbol{v}), & \text{for all } \boldsymbol{v} \in \mathbb{V}, \\ b(\boldsymbol{u}, q) = 0, & \text{for all } q \in \mathbb{Q}, \end{cases} \tag{3.1}$$

where

$$a(\boldsymbol{u}, \boldsymbol{v}) := \int_\Omega \nabla\boldsymbol{u}\nabla\boldsymbol{v}\,\mathrm{d}x, \quad b(\boldsymbol{v}, q) := -\int_\Omega \operatorname{div}\boldsymbol{v}q\,\mathrm{d}x. \tag{3.2}$$

The space $H_0^1(\Omega)$ denotes the usual Sobolev space of Ω and $L_0^2(\Omega)$ denotes the subspace of all L^2-functions over Ω having mean value zero.

Let $\mathcal{T}_h$ be a rectangular or triangular decomposition of the domain Ω. For approximating the weak formulation (3.1) by FEM, one chooses appropriate Ladyzenskaja–Babuška–Brezzi (LBB) stable spaces $\mathbb{V}_h$ and $\mathbb{Q}_h$ consisting of piecewise polynomial functions to approximate $\mathbb{V}$ and $\mathbb{Q}$, respectively. The discrete Stokes problem reads as: find $(\boldsymbol{u}_h, p_h) \in \mathbb{V}_h \times \mathbb{Q}_h$, such that

$$\begin{cases} a(\boldsymbol{u}_h, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h) = (\boldsymbol{f}_h, \boldsymbol{v}_h), & \text{for all } \boldsymbol{v}_h \in \mathbb{V}_h, \\ b(\boldsymbol{u}_h, q_h) = 0, & \text{for all } q_h \in \mathbb{Q}_h. \end{cases} \tag{3.3}$$

To formulate (3.3) as operator equations and demonstrate the algorithm, we introduce the following operators induced by the bilinear forms:

$$\begin{array}{llll} A: & \mathbb{V}_h \mapsto \mathbb{V}_h' & \text{for } \boldsymbol{u}_h \in \mathbb{V}_h,\ \langle A\boldsymbol{u}_h, \boldsymbol{v}_h\rangle = a(\boldsymbol{u}_h, \boldsymbol{v}_h) & \text{for all } \boldsymbol{v}_h \in \mathbb{V}_h, \\ B: & \mathbb{V}_h \mapsto \mathbb{Q}_h' & \text{for } \boldsymbol{v}_h \in \mathbb{V}_h,\ \langle B\boldsymbol{v}_h, q_h\rangle = b(\boldsymbol{v}_h, q_h) & \text{for all } q_h \in \mathbb{Q}_h, \end{array}$$

where $\mathbb{X}'$ denotes the dual of space $\mathbb{X}$ and $\langle\cdot,\cdot\rangle$ the duality pairing. We denote B' as the dual operator of B.

Using these notation, the discretization of (3.3) can be written in the form of (2.1):

$$\mathcal{L}\boldsymbol{x} = \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} := \boldsymbol{b}. \tag{3.4}$$

The ordering adopted for $\mathcal{L}$ is an uncoupled ordering of the underlying grid, i.e., the grid values for u were listed first, followed by those for v, and then those for p. Hereafter, subscript MAC and FE, such as $\mathcal{L}_{\text{MAC}}$, $\mathcal{L}_{\text{FE}}$, etc, will be used to distinguish the systems and variables arising from MAC scheme (2.1) and FE discretizations (3.4).

3.2 DGS Smoother

With certain smoothness and boundary conditions, the commutator can be manipulated as

$$(-\Delta + \text{grad div})\text{grad} = \text{curl curl grad} = 0.$$

Here we use the identity $-\Delta = -\text{grad div} + \text{curlcurl}$, which holds in H^{-1} topology, and the fact that $\text{curlgrad} = 0$. In short, the operators Δ and grad are commutative, i.e.,

$$\Delta_u \text{grad}_p = \text{grad}_p \Delta_p, \tag{3.5}$$

where the subscripts are used to indicate different operators associated to the velocity and the pressure. More precisely, Δ_u denotes the vector Laplacian with zero Dirichlet boundary conditions applied to the velocity, Δ_p is the scalar Laplacian to the pressure, and grad_p is the grad operator to the pressure.

The key point to design an effective DGS smoother is the construction of A_p, a discretization of $-\Delta_p$ with zero Neumann boundary conditions, such that (3.5) holds, at least approximately, in the discrete level. We first try to construct $A_p = \alpha M_p^{-1} B M_u^{-1} B'$ by choosing appropriate scaling α and approximations of the mass matrices of pressure M_p and velocity M_u. For example, M_p and M_u can be replaced by their diagonal matrices D_p and D_u, respectively. The scaling α is empirical and depends on the type of elements considered. With the above consideration, the distributive matrix is formed as

$$\mathcal{M} = \begin{pmatrix} I & D_u^{-1}B' \\ 0 & -\alpha D_p^{-1} B D_u^{-1} B' \end{pmatrix}. \tag{3.6}$$

For continuous pressure element discretization, A_p can be chosen as βA_n, where A_n is the Laplacian operator defined on the discrete pressure space, subject to the Neumann boundary condition, and β is a suitable scaling parameter. This kind of selection yields a distributive matrix

$$\mathcal{M} = \begin{pmatrix} I & B' \\ 0 & -\beta A_n \end{pmatrix}. \tag{3.7}$$

3.3 LSC-DGS Smoother

To avoid the difficulty of finding correct scaling parameters case by case, we propose the following distributive matrix

$$\mathcal{M} = \begin{pmatrix} I & B' \\ 0 & -(BB')^{-1}BAB' \end{pmatrix}, \tag{3.8}$$

which gives rise to the transformed system

$$\mathcal{L}\mathcal{M} = \begin{pmatrix} A & PAB' \\ B & BB' \end{pmatrix}, \quad \text{with } P = I - B'(BB')^{-1}B.$$

Note that $P : \mathbb{V}_h \to \ker(B)$ is the orthogonal projection operator in L^2 inner product to $\ker(B)$ and hence $PB' = 0$. Consequently, the commutator reads as

$$W := PAB' = P(AB' - B'A_p).$$

Now the only requirement is the existence of A_p such that (3.5) holds, and no explicit construction is needed.

For a general discrete Stokes system, it is not clear whether such A_p exists or not. However, the projection matrix makes the commutator as small as possible in the least squares sense, i.e.,

$$\|PAB'\|_F \le \min_{X:\mathbb{Q}\to\mathbb{Q}} \|AB' - B'X\|_F, \tag{3.9}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (F-norm) of matrices [22]. Indeed solving the least-squares problem on the right of (3.9) will give the solution $A_p^* = (BB')^{-1}BAB'$ and therefore it will be called Least-Squares Commutator DGS (LSC-DGS) smoother.

The LSC was firstly developed by Elman [20], and used to construct a least-squares approximation to the Schur complement of the linearized Navier-Stokes system, yielding the so-called BFBt preconditioner. Elman, Howle, Shadid, Shuttleworth, and Tuminaro [21] tried to generate an sparse approximate commutator by solving (3.9) over a given sparsity pattern, together with methods of computing sparse approximate inverses. Here we use A_p^* to devise an effective DGS smoother.

Again we take $\widetilde{\mathcal{L}\mathcal{M}}$ and $\mathcal{S}$ in (2.3) to get a DGS smoother with a different distributive matrix. For completeness, we present the LSC-DGS relaxation algorithm below.

```
Algorithm 3.1. $[\boldsymbol{u}^{k+1}, p^{k+1}] \leftarrow \texttt{LSC-DGS}(\boldsymbol{u}^k, p^k)$
  1. Relax momentum equations
       $\boldsymbol{u}^{k+\frac{1}{2}} = \boldsymbol{u}^k + \hat{A}^{-1}(\boldsymbol{f} - A\boldsymbol{u}^k - B'p^k),$
  2. Relax transformed continuity equations
       $\delta q = \hat{A}_p^{-1}(0 - B\boldsymbol{u}^{k+\frac{1}{2}}).$
  3. Transform the correction back to the original variables
       (3.1)   $\boldsymbol{u}^{k+1} = \boldsymbol{u}^{k+\frac{1}{2}} + B'\delta q,$
       (3.2)   $p^{k+1} = p^k - \widetilde{A}_p^{-1} BAB'\delta q.$
```

We now discuss choices of the three approximations $\hat{A}$, $\hat{A}_p$ and $\widetilde{A}_p$ used in LSC-DGS. The smoother $\hat{A}^{-1}$ for the velocity can be the standard Gauss–Seidel relaxation which has been shown to be effective for the Laplacian operator. When applicable, red-black or general multi-coloring ordering is further applied to improve the smoothing effect.

In LSC-DGS, one needs to transfer the correction back to the original pressure variables by applying $(BB')^{-1}BAB'$. In step (3.2), the matrix inversion $(BB')^{-1}$ is replaced by a cheaper relaxation $\widetilde{A}_p^{-1}$ which is in general different with the smoothers $\hat{A}_p$ used in step (2). Recall that the commutator, i.e., (1,2) block of the transformed system $\mathcal{LM}$, will be $W = (I - B'\widetilde{A}_p^{-1}B)AB'$. The closer $\widetilde{A}_p^{-1}$ is to $(BB')^{-1}$, the smaller $\|W\|_F$ is. As a guideline, from our empirical tests, $\widetilde{A}_p$ can be taken to be one symmetric Gauss–Seidel relaxation for structured grids and one $\mathcal{V}$-cycle iteration for unstructured grids.

The smoother $\hat{A}_p$ will affect the projection step (3.1) of $\boldsymbol{u}^{k+\frac{1}{2}}$. Choosing $\hat{A}_p$ closer to $(BB')^{-1}$ will make $\boldsymbol{u}^{k+\frac{1}{2}}$ more divergence free in each level and consequently may help in accelerating the convergence of the whole multigrid procedure. Usually, the smoother $\hat{A}_p$ in step (2) can be just one Gauss–Seidel iteration. For discontinuous pressure finite element approximations, it can be chosen as an element-wise block Gauss–Seidel smoother.

Compared with the standard DGS, step (3.2) of LSC-DGS requires one more relaxation and one more matrix-vector multiplication. On the other hand, LSC-DGS is more robust, efficient, and parameter free. This is a typical trade off between robustness and operation count.

3.4 Comparisons with Inexact Uzawa, Braess-Sarazin and Vanka Smoothers

In order to compare the LSC-DGS smoother with other popular smoothers, we merge step 2 and 3 in Algorithm 3.1 and rewrite LSC-DGS as follows:

$$\boldsymbol{u}^{k+\frac{1}{2}} = \boldsymbol{u}^k + \hat{A}^{-1}(\boldsymbol{f} - A\boldsymbol{u}^k - B'p^k), \tag{3.10}$$

$$\boldsymbol{u}^{k+1} = \hat{P}\boldsymbol{u}^{k+\frac{1}{2}} := (I - B'\hat{A}_p^{-1}B)\boldsymbol{u}^{k+\frac{1}{2}}, \tag{3.11}$$

$$p^{k+1} = p^k - (\hat{A}_p(BAB')^{-1}\widetilde{A}_p)^{-1}(0 - B\boldsymbol{u}^{k+\frac{1}{2}}). \tag{3.12}$$

Inexact Uzawa Smoother. The inexact Uzawa smoother can be summarized as the following two steps [35,60]:

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \hat{A}^{-1}(\boldsymbol{f} - A\boldsymbol{u}^k - B'p^k),$$

$$p^{k+1} = \ p^k - \hat{C}^{-1}(0 - B\boldsymbol{u}^{k+1}).$$

Therefore, the LSC-DGS smoother can be viewed as an inexact Uzawa smoother by taking $\hat{C} = \hat{A}_p(BAB')^{-1}\widetilde{A}_p$, and more importantly adding a projection step (3.11) to ensure that $\boldsymbol{u}^{k+1}$ is more likely to be discretely divergence-free.

Braess-Sarazin Smoother. The Braess-Sarazin [6] or the SIMPLE-type [43,54] smoother is to use $\begin{pmatrix} \hat{A} & B' \\ B & 0 \end{pmatrix}^{-1}$ as a smoother for the saddle point system (3.4), which can be formulated as DGS smoothing by taking the distributive matrix,

$$\mathcal{M}_S = \begin{pmatrix} I & A^{-1}B' \\ 0 & -I \end{pmatrix}.$$

Therefore,

$$\mathcal{L}^{-1} = \mathcal{M}_S(\mathcal{L}\mathcal{M}_S)^{-1} = \begin{pmatrix} I & A^{-1}B' \\ 0 & -I \end{pmatrix}\begin{pmatrix} A & 0 \\ B & BA^{-1}B' \end{pmatrix}^{-1}. \tag{3.13}$$

Replacing A with easily invertible approximation $\hat{A}$ yields the following Braess-Sarazin smoothing procedure:

$$\boldsymbol{u}^{k+\frac{1}{2}} = \boldsymbol{u}^k + \hat{A}^{-1}(f - A\boldsymbol{u}^k - B'p^k), \tag{3.14}$$

$$\boldsymbol{u}^{k+1} = (I - \hat{A}^{-1}B'\hat{S}^{-1}B)\boldsymbol{u}^{k+\frac{1}{2}}, \tag{3.15}$$

$$p^{k+1} = p^k - \hat{S}^{-1}(0 - B\boldsymbol{u}^{k+\frac{1}{2}}), \tag{3.16}$$

where $\hat{S} = B\hat{A}^{-1}B'$.

In Braess-Sarazin or SIMPLE-type smoother, $\hat{A}$ will be chosen as $\hat{A} = \alpha\mathrm{diag}(A)$, with an appropriate damping parameter α, which could depend on the maximum (Braess-Sarazin smoother) or the minimum (SIMPLE-type smoother) eigenvalue of A. Also, for Braess-Sarazin smoother [6], the inverse of the Laplacian-type operator $(B\hat{A}^{-1}B')^{-1}$ should be computed sufficiently accurate, say the relative residual is below 10^{-5}.

Inexact symmetric Uzawa smoother. The inexact symmetric Uzawa smoother introduced in [4] is performed by approximating $(B\hat{A}^{-1}B')^{-1}$ in (3.15) and (3.16) with one or two $\mathcal{V}$-cycle iterations. The analysis of the smoothing property as well as the performance of geometric multigrid methods using such smoothers can be found in [60,61].

The LSC-DGS smoother differs from Braess-Sarazin and inexact symmetric Uzawa smoother in the different projections of projecting $\boldsymbol{u}^{k+1}$ into the divergence-free space (see (3.11) and (3.15)) and in the way of updating the pressure equation. In LSC-DGS (3.12), the pressure is updated using a better approximated Schur complement of the Stokes equations than the one in (3.16).

Vanka smoother. Although Vanka smoothing can be defined for both continuous and discontinuous pressure elements, only discontinuous pressure element discretization is discussed and implemented in this work.

For each cell $\tau \in \mathcal{T}_h$, denote by A_τ, B_τ the element version of A and B, and D_τ the diagonal part of A_τ. The correction proceeds on each cell as follows,

$$\begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} + \begin{pmatrix} D_\tau & B'_\tau \\ B_\tau & 0 \end{pmatrix}^{-1}\begin{pmatrix} r_{\boldsymbol{u}} \\ r_p \end{pmatrix}, \tag{3.17}$$

where $r_{\boldsymbol{u}} = \boldsymbol{f} - A_\tau\boldsymbol{u} - B'_\tau p$ and $r_p = \boldsymbol{0} - B_\tau\boldsymbol{u}$. Since D_τ is a diagonal matrix, the inverse of the small saddle-point system can be computed by inverting the Schur complement $B_\tau D_\tau^{-1}B'_\tau$

which is a matrix of small size. Suitable damping (under-relaxation) can be used to further improve efficiency and ensure the smoothing property; see [58].

Vanka smoother can be viewed as a block-wise Gauss–Seidel procedure of the Braess-Sarazin smoother. Therefore the difference of LSC-DGS smoother with Vanka smoother mainly lies in the different way of updating pressure and enforcing divergence free condition.

4 Auxiliary Space Multigrid Method

In this section we apply the auxiliary space (defect correction) multigrid methods [10,28,29], which consist of an effective smoothing and a simpler discretization serving as a coarse grid correction, to the Stokes equations. Since the MAC scheme is only defined on uniform grids, we restrict ourself to the finite element discretization on the rectangular grids in this section. Similar approaches shall work if we have appropriate generalizations of the MAC scheme and corresponding DGS to triangular grids.

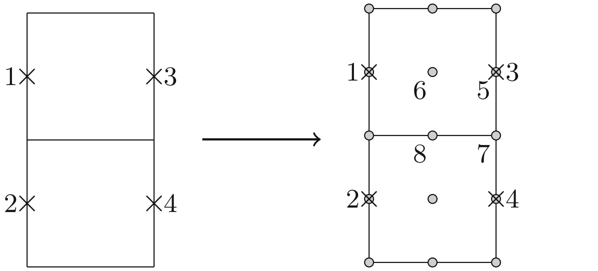
4.1 Transfer Operators

We introduce the transfer operators between a finite element fine-space and the MAC auxiliary-space. One virtue of ASMG is the simplification of the implementation. The user only needs to code the geometric multigrid for MAC, a fine grid smoother, and transfer operators from the discretization of consideration to MAC.

In regard to the prolongation (interpolation) operators, many prolongation methods could be used. Fortunately, for the auxiliary space multigrid methods, the simplest of these is quite effective. For this reason, we apply bilinear interpolation of neighboring auxiliary-space unknowns in the MAC staggered grid. Figure 2 shows an example of the prolongation for u from the MAC space to the bi-quadratic Q_2 space. The prolongation operator takes MAC auxiliary-space vectors ($[\cdots, u_1, u_2, u_3, u_4, \cdots]^t$) and produces Q_2 fine-space vectors ($[\cdots, u_5, u_6, u_7, u_8, \cdots]^t$) according to the following rule:

$$u_5 = u_3,\ u_6 = \frac{1}{2}(u_1 + u_3),\ u_7 = \frac{1}{2}(u_3 + u_4),\ u_8 = \frac{1}{4}(u_1 + u_2 + u_3 + u_4).$$

The prolongation operator for v can be defined in a similar way. Piecewise constant (first-order) interpolation for the p variables is adopted here because of its simplicity and



(a) DOF for u on MAC coarse-space **(b)** DOF for u on Q_2 fine-space

Fig. 2 Sketch of the prolongation from the MAC to Q_2. **a** MAC auxiliary-space points 1, 2, 3, 4. **b** MAC auxiliary-space points 1, 2, 3, 4 and Q_2 fine-space points 5, 6, 7, 8, etc

effectiveness. For the bi-linear Q_1 element approximation of the pressure, we use the average of the pressure on the four cells sharing a vertex. For the discontinuous pressure space P_1^{dc} defined linearly in each element, the degree of freedom approximates the cell center value takes the constant pressure from the MAC, while the degree of freedom approximate the derivatives take values of zero. The interpolation operator will be denoted by $\mathcal{I}_{\mathrm{MAC}}^{\mathrm{FE}}$.

We choose the restriction operator as the transpose of the prolongation operator divided by h^2 which transfers the residual of the finite element discretizations to that of the finite difference discretization. The restriction operator will be denoted by $\mathcal{R}_{\mathrm{FE}}^{\mathrm{MAC}}$.

4.2 Multigrid Procedure

With the transfer operators defined in the previous subsection, one $\mathcal{V}$-cycle multigrid procedure can be summarized in the following algorithm.

Algorithm 4.1. $\boldsymbol{x} \leftarrow$ `asmg-Vcycle`$(\nu_1, \nu_2, \boldsymbol{x}, \boldsymbol{b})$

1. Relax ν_1 steps on $\mathcal{L}_{\mathrm{FE}}\boldsymbol{y} = \boldsymbol{b}$ with initial guess $\boldsymbol{x}$.
2. Form the FE fine-space residual
$$\boldsymbol{r} = \boldsymbol{b} - \mathcal{L}_{\mathrm{FE}}\boldsymbol{x}$$
and restrict it to the MAC auxiliary-space $\boldsymbol{r}_{\mathrm{MAC}} = \mathcal{R}_{\mathrm{FE}}^{\mathrm{MAC}}\boldsymbol{r}$.
3. Solve the MAC auxiliary-space residual equation
$$\mathcal{L}_{\mathrm{MAC}}\boldsymbol{e}_{\mathrm{MAC}} = \boldsymbol{r}_{\mathrm{MAC}}$$
with one multigrid $\mathcal{F}$-cycle procedure.
4. Interpolate the MAC auxiliary-space error to the FE fine-space by $\boldsymbol{e} = \mathcal{I}_{\mathrm{MAC}}^{\mathrm{FE}}\boldsymbol{e}_{\mathrm{MAC}}$ and correct the FE fine-space approximation by:
$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{e}.$$
5. Relax ν_2 steps on $\mathcal{L}_{\mathrm{FE}}\boldsymbol{y} = \boldsymbol{b}$ with initial guess $\boldsymbol{x}$.

We refer to the multigrid method based on `asmg-Vcycle` as `asmg`(ν_1,ν_2). In the framework of defect correction multigrid, the smoothing steps in (1) and (5) can be zero, i.e., `asmg`$(0, 0)$ or it could be replaced by the smoothing for $\mathcal{L}_{\mathrm{MAC}}$. However, such a scheme diverges for all finite elements we have tested. An additional and effective smoothing for the finite element discretization is really necessary and thus our method is in the sprit of auxiliary space method [57]. On solving the MAC auxiliary-space problem, one pre-smoothing and post-smoothing are used for the $\mathcal{F}$-cycle, i.e., $\mathcal{F}(1, 1)$. We can also replace $\mathcal{F}$-cycle with $\mathcal{V}$-cycle or $\mathcal{W}$-cycle. According to our numerical test, $\mathcal{F}(1, 1)$ performs equivalently well as $\mathcal{V}(2, 2)$ or $\mathcal{W}(1, 1)$ cycles in step 3 of `asmg-Vcycle`.

5 Numerical Experiment

The Stokes system to be solved is

$$\mathcal{L}\boldsymbol{x} = \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} := \boldsymbol{b}. \tag{5.1}$$

The selected stopping criterion is

$$\|\boldsymbol{r}_i\|_{l^2}/\|\boldsymbol{r}_0\|_{l^2} < \text{tol},$$

where tol is a tolerance which will be specified later for each example, and $\boldsymbol{r}_i = \boldsymbol{b} - \mathcal{L}\boldsymbol{x}_i$, for multigrid methods and $\boldsymbol{r}_i = \mathcal{P}^{-1}(\boldsymbol{b} - \mathcal{L}\boldsymbol{x}_i)$, for preconditioned GMRES/MINRES method with block or block-triangular preconditioner $\mathcal{P}$ introduced in a moment. Zero initial guess is chosen for all tests. The estimates for asymptotic contraction factors (rate) are the averages of relative residual $(\|\boldsymbol{r}_{3+i}\|_{l^2}/\|\boldsymbol{r}_3\|_{l^2})^{1/i}$ over all steps after the third step. The reason starting from step 3 rather than step 0 is that asymptotic behavior often appears after a few iterations.

All of our algorithms are implemented using MATLAB. The built-in GMRES/MINRES functions are used for the Krylov subspace methods. The finite element matrices on triangular grids and rectangular grids are assembled using the software *i*FEM [16] and IFISS [49], respectively. The AMG/MG implemented in the software package *i*FEM is used for Poisson solvers. All tests are run on a laptop with 2.50GHz Intel(R) Core(TM) i5-2520M processor.

For all tests, the iteration steps and CPU time of each solver are listed in tables. The code has been optimized using vectorization technique so that the CPU time could be a good indicator of the efficiency. To further relieve the affect of programming language and hardware, the operation count is also included. The presentation follows closely the paper by Elman [19].

5.1 Krylov Subspace Methods with Block Preconditioners

For comparison, we will test the Krylov subspace methods with popular block preconditioners. For the Stokes equations, the classical block-diagonal preconditioner for MINRES [19] method is

$$\mathcal{P} = \begin{pmatrix} A & 0 \\ 0 & BA^{-1}B' \end{pmatrix},$$

and the block-triangular preconditioner [19,26] for GMRES method is

$$\mathcal{P} = \begin{pmatrix} A & B' \\ 0 & -BA^{-1}B' \end{pmatrix}.$$

The block preconditioning requires the solution of two systems of equations with matrices A and $BA^{-1}B'$ at each GMRES/MINRES iteration. If $\mathcal{P}^{-1}$ is computed exactly, the preconditioned Krylov methods converges in two or three steps [36]. For practical implementations, the Schur complement $BA^{-1}B'$ is replaced by the mass matrix M_p of the pressure space. For discontinuous pressure space, M_p is block diagonal and easy to invert. For continuous pressure space, say Q_1, the mass matrix M_p can be further replaced by its diagonal matrix [22].

One geometrical multigrid $\mathcal{V}$-cycle is used to invert the Laplacian operator A of the velocity. ASMG is also applied in this step. For example, for Q_2 element, one Gauss–Seidel smoothing for Q_2 element plus an $\mathcal{F}$-cycle for P_1 element serving as the auxiliary space correction.

5.2 Operation Count

Simple comparison of iteration steps and CPU time may not be fair since the performance of a specific method depends on the implementation and testing environment: the programming language, optimization of codes, and the hardware (memory and cache), etc. To fairly compare

Table 1 The operations used for each smoother/preconditioner

Smoother/Precond.	Matrix-vector product required					
LSCDGS	$2A$	$2B$	$2B'$	$\hat{A}_p^{-1}$	$\widetilde{A}_p^{-1}$	
DGS ($\mathcal{M}$ in (3.6))	A	$2B$	$2B'$	$\hat{A}_p^{-1}$	$2D_u^{-1}$	D_p^{-1}
DGS ($\mathcal{M}$ in (3.7))	A	B	$2B'$	$\hat{A}_p^{-1}$	A_n	
Vanka	NA_τ	$3NB_\tau$	NB'_τ	$2ND_\tau^{-1}$		
Inexact Sym. Uzawa	A	B	$2B'$	$\hat{A}^{-1}$	$\mathcal{V}_{B\hat{A}^{-1}B'}(1,1)$	
Block-diag precond.		$\mathcal{V}_A(1,1)$	D_p^{-1}			
Block-tri precond.		$\mathcal{V}_A(1,1)$	D_p^{-1}	B'		

Table 2 The number of operations for smoothers and preconditioners

Smoother/preconditioner	$Q_1^\perp - P_0$	$P_1^B - P_1$	$Q_2 - P_1^{\text{dc}}$	$Q_2 - Q_1$	$P_2^{\text{iso}} - P_1$
MAC one F-cycle	$2.8\mathcal{L}$	$1.9\mathcal{L}$	$0.4\mathcal{L}$	$0.4\mathcal{L}$	$0.4\mathcal{L}$
LSCDGS	$2.3\mathcal{L}$	$2.4\mathcal{L}$	$2.2\mathcal{L}$	$2.2\mathcal{L}$	$2.2\mathcal{L}$
DGS	$1.3\mathcal{L}$	$1.6\mathcal{L}$	$1.5\mathcal{L}$	$1.5\mathcal{L}$	$1.5\mathcal{L}$
Vanka	$2.4\mathcal{L}$	-	$2.5\mathcal{L}$	-	-
Inexact Sym. Uzawa	$1.7\mathcal{L}$	$2.6\mathcal{L}$	$1.7\mathcal{L}$	$1.7\mathcal{L}$	$1.7\mathcal{L}$
Block-diag precond.	$2.9\mathcal{L}$	$1\mathcal{L}$	$2.0\mathcal{L}$	$2.1\mathcal{L}$	$2.1\mathcal{L}$
Block-tri precond.	$3.0\mathcal{L}$	$1.3\mathcal{L}$	$2.2\mathcal{L}$	$2.3\mathcal{L}$	$2.3\mathcal{L}$

Table 3 Cost for one multigrid cycle or one step of the preconditioned Krylov subspace methods

Methods	$Q_1^\perp - P_0$	$P_1^b - P_1$	$Q_2 - P_1^{\text{dc}}$	$Q_2 - Q_1$	$P_2^{\text{iso}} - P_1$
MG-LSCDGS	$8.5\mathcal{L}$	$7.7\mathcal{L}$	$5.8\mathcal{L}$	$5.8\mathcal{L}$	$5.8\mathcal{L}$
MG-DGS	$6.4\mathcal{L}$	$6.1\mathcal{L}$	$4.4\mathcal{L}$	$4.4\mathcal{L}$	$4.4\mathcal{L}$
MG-Vanka	$8.6\mathcal{L}$	-	$6.4\mathcal{L}$	-	-
MG-iUzawa	$7.2\mathcal{L}$	$8.1\mathcal{L}$	$4.8\mathcal{L}$	$4.8\mathcal{L}$	$4.8\mathcal{L}$
MINRES	$4.7\mathcal{L}$	$2.8\mathcal{L}$	$3.3\mathcal{L}$	$3.3\mathcal{L}$	$3.3\mathcal{L}$
GMRES	$4.0\mathcal{L}$	$2.3\mathcal{L}$	$3.2\mathcal{L}$	$3.3\mathcal{L}$	$3.3\mathcal{L}$

the efficiency of different solvers, following Elman [19], we list the operation count of each method but only consider the dominating portion of the matrix-vector multiplication. We use one matrix-vector product $\mathcal{L}\boldsymbol{x}$ as a unit, which consists of matrix-vector multiplications of A, B, and B'. The cost of a matrix-vector product is estimated to be the number of non-zeros in the matrix used.

We gather the matrix-vector products used in different smoothers and preconditioners in Table 1 and present the costs of smoothers and preconditioners in Table 2. The costs for one multigrid cycle `asmg`(1,1) and Krylov subspace iteration are given in Table 3. For multigrid methods, the costs of one $\mathcal{V}(1,1)$-cycle and $\mathcal{F}(1,1)$-cycle step are estimated as $11/3$ and $9/2$ work units, respectively, including one residual evaluation on the finest grid; see e.g., [15,50]. Since the iteration cost for each step of GMRES is not linear, we only count the cost for the

Table 4 Example 1. The number of iterations and CPU time (in parentheses) of the asmg multigrid method with different smoothers and preconditioned Krylov subspace methods for solving the system (5.1)

Smoother	h	$Q_1^{\perp} - P_0$	$P_1^b - P_1$	$Q_2 - P_1^{\mathrm{dc}}$	$Q_2 - Q_1$	$P_2^{\mathrm{iso}} - P_1$
	1/64	6 (0.07 s)	10 (0.22 s)	7 (0.25 s)	10 (0.30 s)	13 (0.35 s)
LSC-DGS	1/128	5 (0.23 s)	10 (0.78 s)	7 (0.97 s)	9 (1.00 s)	13 (1.30 s)
	1/256	5 (0.90 s)	10 (3.03 s)	7 (4.40 s)	9 (4.01 s)	12 (4.81 s)
	1/64	9 (0.13 s)	23 (0.39 s)	11 (0.31 s)	21 (0.84 s)	29 (1.20 s)
DGS	1/128	9 (0.40 s)	23 (1.41 s)	11 (1.15 s)	20 (2.89 s)	28 (4.03 s)
	1/256	9 (1.63 s)	22 (5.21 s)	11 (4.51 s)	20 (10.7 s)	27 (12.5 s)
	1/64	5 (0.06 s)	–	6 (0.24 s)	–	–
Vanka	1/128	5 (0.24 s)	–	6 (0.94 s)	–	–
	1/256	5 (0.89 s)	–	6 (4.30 s)	–	–
	1/64	11 (0.24 s)	11 (0.33 s)	9 (0.32 s)	14 (0.56 s)	20 (0.68 s)
iUzawa	1/128	12 (1.67 s)	12 (1.28 s)	9 (1.11 s)	14 (2.76 s)	20 (2.96 s)
	1/256	12 (5.66 s)	12 (7.90 s)	9 (4.23 s)	15 (9.31 s)	21 (10.3 s)
Diag-block	1/64	46 (0.55 s)	37 (0.30 s)	30 (0.74 s)	50 (0.85 s)	47 (0.85 s)
precond.	1/128	46 (1.25 s)	37 (1.25 s)	30 (3.01 s)	48 (2.30 s)	45 (2.30 s)
(MINRES)	1/256	44 (5.07 s)	37 (4.90 s)	32 (11.7 s)	48 (10.2 s)	43 (10.2 s)
Tri-block	1/64	25 (0.60 s)	24 (1.14 s)	22 (0.96 s)	32 (1.45 s)	24 (1.05 s)
precond.	1/128	26 (1.64 s)	25 (4.99 s)	22 (4.32 s)	32 (5.35 s)	19 (4.35 s)
(GMRES)	1/256	27 (6.64 s)	25 (20.3 s)	23 (14.7 s)	33 (22.8 s)	18 (17.8 s)

preconditioner and the multiplication of $\mathcal{L}$ in Table 3 for GMRES. Additional cost will be counted for the whole iteration process in the numerical experiment.

The counting is based on an $N \times N$ uniform rectangular grid. For general unstructured grids, the number of operation for each method depends also on the topology of the underline grids and could be slightly different.

5.3 Numerical Examples: Rectangular Grids

The first numerical experiment is carried out to show the efficiency of the LSC-DGS smoother and the ASMG method. The grids considered are the uniform rectangular grids. A uniform triangulation can be obtained by drawing the diagonal with positive slope in each rectangular cell. Five stable finite element discretizations on uniform rectangular or triangular grids are considered: $Q_1^{\perp} - P_0$, $P_1^b - P_1$, $Q_2 - Q_1$, $P_2^{\mathrm{iso}} - P_1$ and $Q_2 - P_1^{\mathrm{dc}}$. Here, $Q_1^{\perp}$ denotes rotated nonconforming bilinear element [46], P_1^b stands for the piecewise linear and continuous space with cubic element bubbles [1] on a uniform triangulation, P_2^{iso} is the linear element on a uniform refined triangulation, and P_1^{dc} represents disconutinous element in each small rectangle [22]. Other low order elements including $Q_2 - Q_0$, $Q_2^{\mathrm{iso}} - Q_0$, etc., have also been tested and similar conclusions were drawn. Since the rate of convergence of these schemes are lower than the MAC scheme, results for these schemes are not reported here.

For the ASMG algorithm, the solver for the MAC scheme is one $\mathcal{F}(1, 1)$ cycle, i.e., an $\mathcal{F}$-cycle with one $\mathcal{V}(1, 1)$ in each level. Only asmg$(1, 1)$ is tested. The basic set up for the smoothers is listed below.

Table 5 Example 1. The averaged contraction rate and number of operation counts (in parentheses) of the `asmg` multigrid method and preconditioned Krylov subspace methods with $h = 1/256$, tol $= 10^{-6}$

Smoother	$Q_1^{\perp} - P_0$	$P_1^b - P_1$	$Q_2 - P_1^{\rm dc}$	$Q_2 - Q_1$	$P_2^{\rm iso} - P_1$
LSC-DGS	0.12 (43 $\mathcal{L}$)	0.33 (85 $\mathcal{L}$)	0.13 (41 $\mathcal{L}$)	0.28 (52 $\mathcal{L}$)	0.30 (56 $\mathcal{L}$)
DGS	0.27 (58 $\mathcal{L}$)	0.49 (132 $\mathcal{L}$)	0.37 (48 $\mathcal{L}$)	0.58 (88 $\mathcal{L}$)	0.61 (93 $\mathcal{L}$)
Vanka	0.05 (43 $\mathcal{L}$)	–	0.11 (38 $\mathcal{L}$)	–	–
iUzawa	0.29 (108 $\mathcal{L}$)	0.31 (122 $\mathcal{L}$)	0.21 (54 $\mathcal{L}$)	0.30 (84 $\mathcal{L}$)	0.35 (98 $\mathcal{L}$)
MINRES	0.70 (207 $\mathcal{L}$)	0.74 (104 $\mathcal{L}$)	0.67 (106 $\mathcal{L}$)	0.77 (158 $\mathcal{L}$)	0.74 (152 $\mathcal{L}$)
GMRES	0.42 (513 $\mathcal{L}$)	0.63 (382 $\mathcal{L}$)	0.39 (172 $\mathcal{L}$)	0.70 (281 $\mathcal{L}$)	0.51 (201$\mathcal{L}$)

(1) *LSC-DGS*: One Gauss–Seidel iteration for computing $\hat{A}_p^{-1}$ and one symmetric Gauss–Seidel iteration for $\widetilde{A}_p^{-1}$.

(2) *DGS*: Distributive matrix (3.6) is used for $Q_1^{\perp} - P_0$, $Q_2 - P_1^{\rm dc}$, $Q_2 - Q_1$ and $P_2^{\rm iso} - P_1$ with α taking as $3/2$, 1, 2, 2, respectively, and distributive matrix (3.7) is used for $P_1^b - P_1$ with $\beta = 3/2$. The parameters α and β have been tuned in each test.

(3) *Inexact symmetric Uzawa smoother*: The approximation $\hat{A} = 2\text{diag}(A)$ for elements of $P_2^{\rm iso} - P_1$ and $P_1^b - P_1$ [60], and $\hat{A} = \text{diag}(A)$ for other finite elements are used on rectangular grids. One AMG $\mathcal{V}(1, 1)$-cycle is used to solve the approximate Schur complement matrix $B\hat{A}^{-1}B'$. The set up time of AMG is estimated as $1/4$ of $\mathcal{V}$-cycle iterations.

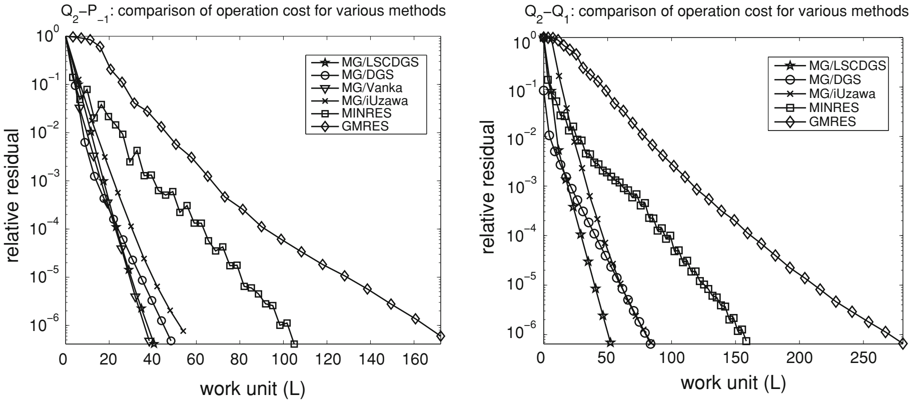
(4) *Vanka*: Multicolor ordering of the cells is used, and one type of the matrices A_τ, B_τ and D_τ are assembled for each colored cell. The set up time is negligible.

For all elements and methods tested in this example, the tolerance is tol $= 10^{-6}$.

Example 1 Driven cavity flow problem. We consider the leaky cavity problem on a unit square, i.e, the force $\boldsymbol{f} = 0$ in system (1.1), and homogenous Dirichlet boundary condition except on the top

$$\{y = 1; 0 \le x \le 1 | \boldsymbol{u} = (1, 0)\}.$$

The number of iterations and CPU time (in parentheses) of the `asmg` multigrid method with different smoothers and preconditioned Krylov subspace methods is summarized in Table 4. The estimated contraction rate and the operation count is presented in Table 5. The cost for m steps of MINRES is counted as $m(\mathcal{L} + \mathcal{P}^{-1}) + 6m \cdot$#DOF; see [42]. For GMRES, the total costs for m steps iteration is roughly $m(\mathcal{L} + \mathcal{P}^{-1}) + 4m^2\cdot$ #DOF; see ([47, p. 158]). The presented results of preconditioned Krylov subspace methods agree with those in the book by Elman et al. [22]. We have also tried other alternatives in the preconditioned Krylov subspace method by solving the mass matrix for pressure space more accurately, or more $\mathcal{V}$-cycles for Poisson equation for velocity, etc. We found that inverting Laplacian for velocity with one $\mathcal{V}$-cycle and approximating Schur complement with the diagonal matrix of the mass matrix is almost optimal in view of operation counts for almost all the tested elements. We also notice that one may gain marginal efficiency by using Preconditioned Conjugate Residual (PCR) method [19]. One may accelerate the convergence by using Chebyshev semi-iteration in the inversion of the mass matrix [52]. As pointed out in [52], the Chebychev-semi-iteration for the pressure mass matrix is very attractive, especially when more accurate solution is required (smaller tolerance). For the tolerance 10^{-6}, however, the speed up is almost negligible, see Figure 4.1 in [52].



Q_2-P_1^{dc} Q_2-Q_1

Fig. 3 Comparison of operation counts for multigrid method and Krylov subspace method. **a** $Q_2 - P_1^{\mathrm{dc}}$. **b** $Q_2 - Q_1$ (Example 1)

The residual norms against operation counts of different solvers for $Q_2 - P_1^{\mathrm{dc}}$ and $Q_2 - Q_1$ is plotted in Fig. 3. Similar behavior is also observed for other elements and thus only two typical finite element pairs, one with continuous and another with discontinuous pressure space, are shown here.

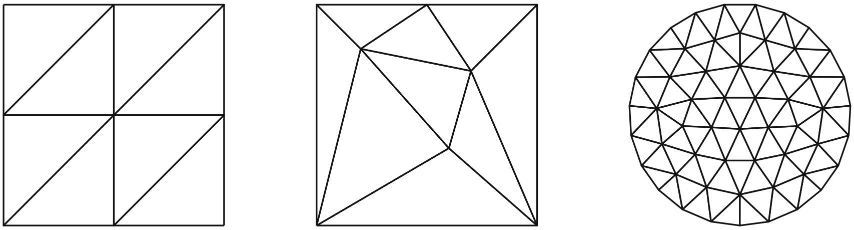
We have tested other settings having analytical solutions, similar behavior was observed. We shall not report them.

Based on these results, we may draw the following conclusion for the performance on uniform grids.

(1) All methods are uniformly convergent with respect to h. Namely all of them are of optimal linear complexity.
(2) Multigrid `asmg(1,1)`, i.e., combination of smoothing in the FE space and correction using multigrid for MAC scheme, outperforms the popular preconditioned Krylov spaces methods. For example, ASMG using LSC-DGS is in average two or three times faster than preconditioned MINRES.
(3) Among the various smoothers, LSC-DGS smoother is the most efficient one in most cases. For discontinuous pressure, Vanka smoothing is the best in terms of operation counts, which in part is due to the multicolor ordering for the grids. For continuous pressure spaces, due to the continuity of the pressure, it is not easy to code Vanka smoothing for overlapping patches [33,34]. In general, geometric and basis information is needed to code an effective Vanka smoother while LSC-DGS only uses the given matrices and thus is more user-friendly.

5.4 Numerical Examples: Triangular Grids

For general triangular grids, we do not have efficient MG-DGS methods for MAC-type schemes. Therefore we test the standard $\mathcal{V}$-cycle or $\mathcal{W}$-cycle multigrid based on the LSC-DGS smoother. For saddle point problems, we expect the uniform convergence of $\mathcal{W}$-cycle with enough smoothing steps while $\mathcal{V}$-cycle is less stable. In [60] Zulehner pointed out that the contraction rates of multigrid methods (using an inexact Uzawa smoother) are certainly still unsatisfactory for unstructured grids. Therefore we apply tests on three different kinds of grids including both structured and unstructured grids.



(a) A structured grid of a square domain.

(b) An unstructured grid of a square domain.

(c) An unstructured grid of a circular domain.

Fig. 4 Three types of meshes used in Example 2

As in [6,60], we consider the $P_2^{\text{iso}} - P_1$ element (modified Taylor-Hood element) discretization: linear shape-functions on a triangular grid for the pressure and linear shape-functions for the velocity on an uniformly refined mesh (where each triangle is divided into four similar small triangles) [14,45]. The performance of other elements are similar. The symbol "×" in the following tables means that the corresponding algorithm diverges.

Example 2 Classical multigrid methods with LSC-DGS smoother on triangular grids. This example is taken from [6,60]. The analytical solution $\boldsymbol{u}$ and p are chosen as follows:

$$\boldsymbol{u}(x, y) = \begin{pmatrix} \sin x \sin y \\ \cos x \cos y \end{pmatrix}, \quad p(x, y) = 2 \cos x \sin y + C,$$

where C is chosen for each specific example such that $p \in L_0^2(\Omega)$.

Three different types of meshes are considered and the coarsest meshes are plotted in Fig 4. The level k in Tables 6, 7, 8 stands for k times uniform refinement of the coarsest mesh. For the circular domain, after each refinement, the boundary nodes are projected onto the circle and thus results in non-nested finite element spaces. The grid in Fig. 4(b) is taken from [6,60].

The choice of tolerance is tol $= 10^{-8}$ in this example.

Example 2.1 Structured grids of a square domain. The performance of both $\mathcal{V}$-cycles and $\mathcal{W}$-cycles is present in Table 6 for the structured grid shown in Fig. 4 (a). In the LSC-DGS smoother, $\hat{A}^{-1}$ is chosen as one Gauss–Seidel iteration, and $\widetilde{A}_p^{-1}$ is one symmetric Gauss–Seidel iteration. In the inexact symmetric Uzawa smoother, following Zulehner [60], the smoother $\hat{A}$ is chosen as $\hat{A} = 2\text{diag}(A)$, and one $\mathcal{V}(2, 2)$-cycle is used to solve the approximate Schur complement $\hat{S} = B\hat{A}^{-1}B'$. For comparison, the numerical result for ASMG-LSC-DGS is also included in Table 6.

Example 2.2 Unstructured grids of a square domain. The numerical results are reported in Table 7 for the unstructured grids shown in Fig. 4(b). In the LSC-DGS smoother, $\hat{A}^{-1}$ is one Gauss–Seidel iteration, while $\widetilde{A}_p^{-1}$ is chosen as one $\mathcal{V}(2, 2)$-cycle, since more accurate solver is required to guarantee the commutator W is small for the unstructured grids. In the inexact symmetric Uzawa smoother, $\hat{A}$ is chosen as $\hat{A} = 2\text{diag}(A)$, and two $\mathcal{V}(2, 2)$-cycles are used to solve the approximate Schur complement $\hat{S} = B\hat{A}^{-1}B'$ to ensure the Schur complement $BA^{-1}B'$ being solved accurately enough.

For this example, only $\mathcal{W}$-cycle is reported since $\mathcal{V}$-cycle converge but not uniformly within 5 steps of pre-smoothing and post-smoothing for both of MG-LSCDGS and

Table 6 Example 2.1: structured grids of a square domain

level	#DOF	$\mathcal{V}$-cycle		$\mathcal{W}$-cycle	
		(1,1)	(2,2)	(1,1)	(2,2)
(a) MG-LSC-DGS					
5	37,507	21 (0.68s)	10 (0.44s)	10 (0.51s)	7 (0.56s)
6	148,739	31 (3.64s)	10 (1.66s)	10 (2.07s)	7 (2.32s)
7	592,387	58 (27.1s)	10 (7.07s)	10 (7.22s)	7 (9.66s)
(b) ASMG-LSC-DGS					
5	37,507	16 (0.43s)	13 (0.51s)		
6	148,739	16 (1.42s)	13 (1.72s)		
7	592,387	16 (5.80s)	13 (6.18s)		
(c) MG-iUzawa					
5	37,507	×	16 (1.21s)	29 (3.12s)	14 (2.30s)
6	148,739	×	16 (3.58s)	29 (8.21s)	14 (6.73s)
7	592,387	×	16 (12.9s)	29 (27.1s)	14 (22.6s)

Comparison of the iterations for three different kinds of multigrid solvers: (a) MG-LSC-DGS, (b) ASMG-LSC-DGS, and (c) MG-iUzawa (tol = 10^{-8})

Table 7 Example 2.2: Unstructured grids of a square domain

level	#DOF	$\mathcal{W}$-cycle			
		(1,1)	(2,2)	(3,3)	(4,4)
(a) MG-LSC-DGS					
5	41,875	35 (4.13s)	18 (3.28s)	13 (3.61s)	11 (4.04s)
6	166,691	34 (16.2s)	18 (15.3s)	13 (15.9s)	11 (16.1s)
7	665,155	34 (65.5s)	18 (56.7s)	13 (58.8s)	11 (63.6s)
(b) MG-iUzawa					
5	41,875	×	×	33 (11.2s)	26 (10.9s)
6	166,691	×	×	33 (38.3s)	25 (37.1s)
7	665,155	×	×	33 (126s)	25 (120s)

Comparison of the iterations for two different kinds of multigrid solvers: (a) MG-LSC-DGS and (b) MG-iUzawa (tol = 10^{-8})

MG-iUzawa. Due to the poor mesh quality, both the cost of one iteration and the number of iterations increases comparing to the structured grids case.

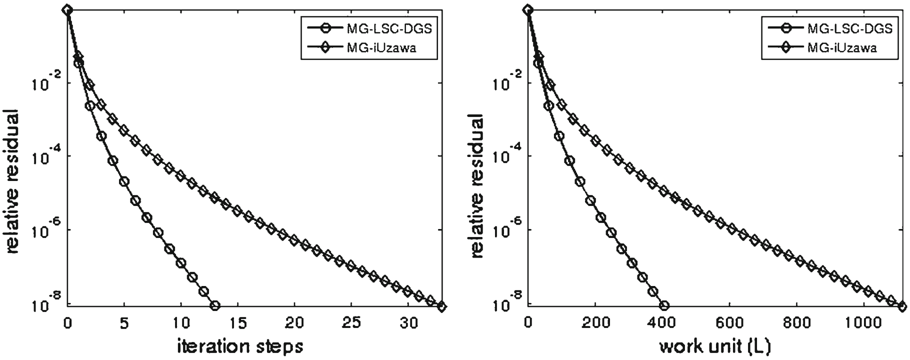
We estimate the operation cost for LSC-DGS smoother and inexact symmetric Uzawa smoother in the unstructured grid case for $P_2^{\text{iso}} - P_1$ element. The cost of one $\mathcal{V}(2,2)$-cycle and $\mathcal{W}(3,3)$-cycle for inverting a sparse matrix with N non-zeros is about $19/3N$ and $13N$, respectively [50]. The cost for one LSC-DGS smoother is about $2.4\mathcal{L}$, and the cost for one inexact symmetric Uzawa smoother is about $2.6\mathcal{L}$. Therefore, the cost of one multigrid $\mathcal{W}(3,3)$-cycle is $31.2\mathcal{L}$ and $33.8\mathcal{L}$ for MG-LSC-DGS and MG-iUzawa, respectively. The comparison is plotted in Fig. 5 and the result for MG-iUzawa is consistent with that in [60].

Example 2.3 Unstructured grids of a circular domain. This example is devoted to show the performance of multigrid $\mathcal{V}$-cycle and $\mathcal{W}$-cycle for a unstructured grid of a circular domain.

Table 8 Example 2.3: Unstructured grids of a circular domain

level	#DOF	$\mathcal{V}$-cycle		$\mathcal{W}$-cycle	
		(1,1)	(2,2)	(1,1)	(2,2)
(a) MG-LSC-DGS					
4	98,843	15 (1.97s)	9 (1.12s)	11 (1.72s)	7 (1.51s)
5	393,523	20 (6.56s)	9 (4.77s)	11 (6.10s)	7 (5.22s)
6	1,570,403	36 (50.8s)	9 (18.6s)	12 (27.5s)	7 (21.7s)
(b) MG-iUzawa					
4	98,843	×	16 (1.91s)	29 (5.22s)	15 (4.9s)
5	393,523	×	16 (8.73s)	30 (17.7s)	15 (14.2s)
6	1,570,403	×	16 (34.7s)	30 (69.1s)	15 (56.7s)

Comparison of the iterations for two different kinds of multigrid solvers: (a) MG-LSC-DGS and (b) MG-iUzawa (tol = 10^{-8})



(a) Converge history of LSCDGS v.s. iUzawa in items of iteration steps

(b) Converge history of LSCDGS v.s. iUzawa in items of work units

Fig. 5 Example 2.2: comparison of MG-LSCDGS and MG-iUzawa on unstructured grids shown in Fig. 4b. **a** Comparisons of the convergence history of $\mathcal{W}(3,3)$ in items of iteration steps, **b** comparisons of the convergence history of $\mathcal{W}(3,3)$ in items of work units

In the LSC-DGS smoother, $\hat{A}$ is chosen as one Gauss–Seidel iteration, and $\widetilde{A}_p$ is chosen as one symmetric Gauss–Seidel iteration. In the inexact symmetric Uzawa smoother, $\hat{A}$ is chosen as $\hat{A} = 2\text{diag}(A)$, and one $\mathcal{V}(2,2)$-cycle is used to solve the approximate Schur complement $\hat{S} = B\hat{A}^{-1}B'$. Due to the nice mesh quality, the performance of MG-LSC-DGS is almost identical to the structured grid case, see Table 8.

Based on these results, we may conclude

(1) The performance of multigrid methods depends crucially on the quality of the mesh. For unstructured grids with poor mesh quality, one or two $\mathcal{V}$-cycles are needed for smoothers and the iteration steps are almost doubled, which in turn increase the CPU time a lot.
(2) For both structured and unstructured grids, MG-LSC-DGS is faster than MG-iUzawa by a factor of two or three.
(3) For structured grids, ASMG using MAC as a coarse grid correction works best. The reason might be due to the point-wise divergence free property of the velocity approximation in the MAC scheme.

6 Conclusions and Future Research

The Stokes system is the first step for the numerical computation of incompressible flow equations. Our method can be extended to the Navier-Stokes equations, since no explicit construction of A_p is required. On rectangular grids we will design fast solvers for the MAC discretization of Navier-Stokes equation with the LSCDGS smoother, combining the idea in [12,13] for the high-Reynolds incompressible flow, and use it as a coarse grid solver to design an efficient multigrid solver for finite element discretization of Navier-Stokes equations.

Another direction is to develop an efficient ASMG for triangular grids. We will first investigate the generalization of MAC to triangular grids [23,38] and design corresponding DGS-type smoother and multigrid methods.

Acknowledgments The work of the first author was supported by 2010-2012 China Scholarship Council (CSC). The second author was supported by NSF Grant DMS-1115961, and in part by Department of Energy prime award # DE-SC0006903. The authors are grateful to the discussions with Professors A. Brandt, J. Hu, J. Xu, I. Yavneh, and L. Zikatanov. The second author was introduced to DGS by Prof. Yavenh during the IMA workshop 'Numerical Solutions of Partial Differential Equations: Fast Solution Techniques', Dec 2010, and then had further discussion with these experts in the workshop 'Algebraic Multigrid Methods with Applications to Fluids and Structure Interactions and Other Multi-Physical Systems' in Kunming Aug 2011. We would also like to thank IMA and Kunming University of Science and Technology for their support and hospitality, as well as for their exciting research atmosphere. The authors also would like to thank two referees for their thorough review. The quality of the paper is improved by addressing their constructive comments and suggestions.

References

1. Arnold, D., Brezzi, F., Fortin, M.: A stable finite element for the Stokes equations. Calcolo **21**(4), 337–344 (1984)
2. Auzinger, W., Stetter, H.: Defect correction and multigrid iterations. In: Hackbusch, W., Trottenberg, U. (eds.) Multigrid Methods, vol. 960, pp. 327–351 (1982)
3. Bacuta, C., Vassilevski, P., Zhang, S.: A new approach for solving Stokes systems arising from a distributive relaxation method. Numer. Methods Partial Differ. Equ. **27**(4), 898–914 (2011)
4. Bank, R., Welfert, B., Yserentant, H.: A class of iterative methods for solving saddle point problems. Numerische Mathematik **56**(7), 645–666 (1989)
5. Benzi, M., Golub, G., Liesen, J.: Numerical solution of saddle point problems. Acta Numerica. **14**(1), 1–137 (2005)
6. Braess, D., Sarazin, R.: An efficient smoother for the Stokes problem. Appl. Numer. Math. **23**(1), 3–19 (1997)
7. Bramble, J., Pasciak, J.: A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comput. **50**(181), 1–17 (1988)
8. Bramble, J., Pasciak, J.: Iterative techniques for time dependent Stokes problems. Comput. Math. Appl. **33**(1–2), 13–30 (1997)
9. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comp. **31**, 333–390 (1977)
10. Brandt, A.: Multigrid techniques: 1984 guide with applications to fluid dynamics. Ges. für Mathematik u, Datenverarbeitung (1984)
11. Brandt, A., Dinar, N.: Multi-grid Solutions to Elliptic Llow Problems. Institute for Computer Applications in Science and Engineering, NASA Langley Research Center (1979)
12. Brandt, A., Yavneh, I.: On multigrid solution of high-Reynolds incompressible entering flows. J. Comput. Phys. **101**, 151–164 (1992)
13. Brandt, A., Yavneh, I.: Accelerating multigrid convergence and high-Reynolds recirculating flows. SIAM J. Sci. Comput. **14**, 607–626 (1993)
14. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Element Methods. Springer, Berlin (1991)
15. Briggs, W., McCormick, S., et al.: A Multigrid Tutorial, vol. 72. Society for Industrial Mathematics (2000)

16. Chen, L.: iFEM: An Integrated Finite Element Methods Package in MATLAB. University of California at Irvine, Technical Report (2009)
17. Chen, L.: Finite difference method (MAC) for Stokes equations. Lecture notes (2012)
18. Chen, L., Wang, M., Zhong, L.: Second order accuracy of a MAC scheme for the Stokes equations (in preparation) (2013)
19. Elman, H.: Multigrid and Krylov subspace methods for the discrete Stokes equations. Int. J. Numer. Methods Fluids **22**(8), 755–770 (1996)
20. Elman, H.: Preconditioning for the steady-state Navier-Stokes equations with low viscosity. SIAM J. Sci. Comput. **20**(4), 1299–1316 (1999)
21. Elman, H., Howle, V., Shadid, J., Shuttleworth, R., Tuminaro, R.: Block preconditioners based on approximate commutators. SIAM J. Sci. Comput. **27**(5), 1651–1668 (2006)
22. Elman, H., Silvester, D., Wathen, A.: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. Oxford University Press, USA (2005)
23. Eymard, R., Fuhrmann, J., Linke, A.: MAC schemes on triangular meshes. Finite Vol Complex Appl VI Problems Perspect. **4**, 399–407 (2011)
24. Gaspar, F., Lisbona, F., Oosterlee, C., Vabishchevich, P.: An efficient multigrid solver for a reformulated version of the poroelasticity system. Comput. Methods Appl. Mech. Eng. **196**(8), 1447–1457 (2007)
25. Gaspar, F., Lisbona, F., Oosterlee, C., Wienands, R.: A systematic comparison of coupled and distributive smoothing in multigrid for the poroelasticity system. Numer. Linear Algebra Appl. **11**(2–3), 93–113 (2004)
26. Geenen, T., Vuik, C., Segal, G., MacLachlan, S.: On iterative methods for the incompressible Stokes problem. Int. J. Numer. Methods fluids **65**(10), 1180–1200 (2011)
27. Gresho, P., Sani, R.: On pressure boundary conditions for the incompressible Navier-Stokes equations. Int. J. Numer. Methods Fluids **7**(10), 1111–1145 (1987)
28. Hackbusch, W.: On multigrid iterations with defect correction. In: Hackbusch, W., Trottenberg, U., (eds.) Multigrid Methods, pp. 461–473 (1982)
29. Hackbusch, W.: Multi-grid Methods and Applications, vol. 4 of Springer Series in Computational Mathematics (1985)
30. Han, H., Wu, X.: A new mixed finite element formulation and the MAC method for the Stokes equations. SIAM J. Numer. Anal. **35**(2), 560–571 (1998)
31. Harlow, F., Welch, J., et al.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys. fluids **8**(12), 2182 (1965)
32. Hu, Q., Zou, J.: Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems. Numer. Math. **93**(2), 333–359 (2002)
33. John, V., Matthies, G.: Higher-order finite element discretizations in a benchmark problem for incompressible flows. Int. J. Numer. Methods Fluids **37**(8), 885–903 (2001)
34. Larin, M., Reusken, A.: A comparative study of efficient iterative solvers for generalized stokes equations. Numer. Linear Algebra Appl. **15**(1), 13–34 (2008)
35. Maitre, J.F., Musy, F., Nigòn, P.: Fast solver for the Stokes equations using multigrid with a Uzawa smoother. Notes Numer. Fluid Mech. **11**, 77–83 (1985)
36. Murphy, M., Golub, G., Wathen, A.: A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comput. **21**(6), 1969–1972 (1999)
37. Nicolaides, R.: Analysis and convergence of the MAC scheme I. The linear problem. SIAM J. Numer. Anal. **29**(6), 1579–1591 (1992)
38. Nicolaides, R., Porsching, T., Hall, C.: Covolume Methods in Computational Fluid Dynamics, vol. 279. Wiley, New York (1995)
39. Nicolaides, R., Wu, X.: Analysis and convergence of the MAC scheme II. Navier-Stokes equations. Math. Comput. **65**(213), 29–44 (1996)
40. Niestegge, A., Witsch, K.: Analysis of a multigrid Stokes solver. Appl. Math. Comput. **35**(3), 291–303 (1990)
41. Oosterlee, C., Lorenz, F.: Multigrid methods for the Stokes system. Comput. Sci. Eng. **8**(6), 34–43 (2006)
42. Paige, C., Saunders, M.: Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. **12**(4), 617–629 (1975)
43. Patankar, S., Spalding, D.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. Int. J. Heat Mass Transf. **15**(10), 1787–1806 (1972)
44. Peric, M., Kessler, R., Scheuerer, G.: Comparison of finite-volume numerical methods with staggered and colocated grids. Comput. Fluids **16**(4), 389–403 (1988)
45. Pironneau, O.: Finite Element Methods for Fluids. NASA STI/Recon technical report A, vol. 90, p. 24264 (1989)

46. Rannacher, R., Turek, S.: Simple nonconforming quadrilateral Stokes element. Numer. Methods Partial Differ. Equ. **8**(2), 97–111 (1992)
47. Saad, Y.: Iterative Methods for Sparse Linear Systems. PWS Pub Co, USA (1996)
48. Shaw, G., Sivaloganathan, S.: On the smoothing properties of the simple pressure-correction algorithm. Int. J. Numer. Methods Fluids **8**(4), 441–461 (1988)
49. Silvester, D., Elman, H., Ramage, A.: Incompressible flow and iterative solver software (IFISS) version 3.1. Available online at http://www.manchester.ac.uk/ifiss (2011)
50. Trottenberg, U., Oosterlee, C., Schüller, A.: Multigrid. Academic Press, London (2001)
51. Vanka, S.: Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. J. Comput. Phys. **65**(1), 138–158 (1986)
52. Wathen, A., Rees, T.: Chebyshev semi-iteration in preconditioning for problems including the mass matrix. Electron. Trans. Numer. Anal. **34**, 125–135 (2009)
53. Wienands, R., Gaspar, F., Lisbona, F., Oosterlee, C.: An efficient multigrid solver based on distributive smoothing for poroelasticity equations. Computing **73**(2), 99–119 (2004)
54. Wittum, G.: Multi-grid methods for Stokes and Navier-Stokes equations. Numerische Mathematik **54**(5), 543–563 (1989)
55. Wittum, G.: On the convergence of multi-grid methods with transforming smoothers. Numerische Mathematik **57**(1), 15–38 (1990)
56. Xu, J.: Iterative methods by space decomposition and subspace correction. SIAM Rev. **34**(4), 581–613 (1992)
57. Xu, J.: The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. Computing **56**(3), 215–235 (1996)
58. Zhang, L.: A second-order upwinding finite difference scheme for the steady Navier-Stokes equations in primitive variables in a driven cavity with a multigrid solver. M2AN **24**, 133–150 (1990)
59. Zhu, Y., Sifakis, E., Teran, J., Brandt, A.: An efficient multigrid method for the simulation of high-resolution elastic solids. ACM Trans. Graph. (TOG) **29**(2), 16 (2010)
60. Zulehner, W.: A class of smoothers for saddle point problems. Computing **65**(3), 227–246 (2000)
61. Zulehner, W.: Analysis of iterative methods for saddle point problems: a unified approach. Math. Comput. **71**(238), 479–506 (2002)