

Local Fourier analysis for mixed finite-element methods for the Stokes equations

Yunhui He^{a,*}, Scott P. MacLachlan^a

^a*Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada*

Abstract

In this paper, we develop a local Fourier analysis of multigrid methods based on block-structured relaxation schemes for stable and stabilized mixed finite-element discretizations of the Stokes equations, to analyze their convergence behavior. Three relaxation schemes are considered: distributive, Braess-Sarazin, and Uzawa relaxation. From this analysis, parameters that minimize the local Fourier analysis smoothing factor are proposed for the stabilized methods with distributive and Braess-Sarazin relaxation. Considering the failure of the local Fourier analysis smoothing factor in predicting the true two-grid convergence factor for the stable discretization, we numerically optimize the two-grid convergence predicted by local Fourier analysis in this case. We also compare the efficiency of the presented algorithms with variants using inexact solvers. Finally, some numerical experiments are presented to validate the two-grid and multi-grid convergence factors.

Keywords: Monolithic multigrid, Block-structured relaxation, local Fourier analysis, mixed finite-element methods, Stokes Equations

2000 MSC: 65N55, 65F10, 65F08, 76M10

1. Introduction

In recent years, substantial research has been devoted to efficient numerical solution of the Stokes and Navier-Stokes equations, due both to their utility as models of (viscous) fluids and their commonalities with many other physical problems that lead to saddle-point systems (see, for example [1], and many of the other references cited here). In the linear (or linearized) case, solution of the resulting matrix equations is seen to be difficult, due to indefiniteness and the usual ill-conditioning of discretized PDEs. In the literature, block preconditioners (cf. [1] and the references therein) are widely used, due to their easy construction from standard multigrid algorithms for scalar elliptic PDEs, such as algebraic multigrid [2]. However, monolithic multigrid approaches [3, 4, 5, 6, 7] have been shown to outperform these preconditioners when algorithmic parameters are properly chosen [8, 9]. The focus of this work is on the analysis of such monolithic multigrid methods in the case of stable and stabilized finite-element discretizations of the Stokes equations.

*Corresponding author: Yunhui He. E-mail address: yunhui.he@mun.ca.

Email addresses: yunhui.he@mun.ca (Yunhui He), smaclachlan@mun.ca (Scott P. MacLachlan)

Local Fourier analysis (LFA) [10, 11] has been widely used to predict the convergence behavior of multigrid methods, to help design relaxation schemes and choose algorithmic parameters. In general, the LFA smoothing factor provides a sharp prediction of actual multigrid convergence, see [10], under the assumption of an “ideal” coarse-grid correction scheme (CGC) that annihilates low-frequency error components and leaves high-frequency components unchanged. In practice, the LFA smoothing and two-grid convergence factors often exactly match the true convergence factor of multigrid applied to a problem with periodic boundary conditions [12, 13, 10]. Recently, the validity of LFA has been further analysed [14], extending this exact prediction to a wider class of problems. However, the LFA smoothing factor is also known to lose its predictivity of the true convergence in some cases [15, 16, 17]. In particular, the smoothing factor of LFA overestimates the two-grid convergence factor for the Taylor-Hood ($Q_2 - Q_1$) discretization of the Stokes equations with Vanka relaxation [16]. Even for the scalar Laplace operator, the LFA smoothing factor fails to predict the observed multigrid convergence factor for higher-order finite-element methods [15].

Two main questions interest us here. First, we look to extend the study of [16] to consider LFA of block-structured relaxation schemes for finite-element discretizations of the Stokes equations. Secondly, we consider if the LFA smoothing factor can predict the convergence factors for these relaxation schemes. Recently, LFA for multigrid based on block-structured relaxation schemes applied to the marker-and-cell (MAC) finite-difference discretization of the Stokes equations was shown to give a good prediction of convergence [18], in contrast to the results of [16]. Thus, a natural question to investigate is whether the contrasting results between [18] and [16] is due to the differences in discretization or those in the relaxation schemes considered. Here, we apply the relaxation schemes of [18] to the $Q_2 - Q_1$ discretization from [16], as well as an “intermediate” discretization using stabilized $Q_1 - Q_1$ approaches.

In recent decades, many block relaxation schemes have been studied and applied to many problems, including Braess-Sarazin-type relaxation schemes [3, 19, 20, 4, 21], Vanka-type relaxation schemes [3, 22, 4, 16, 23, 24, 7], Uzawa-type relaxation schemes [25, 26, 27, 6, 28], distributive relaxation schemes [29, 5, 30, 31, 32] and other types of methods [33, 34]. Even though LFA has been applied to distributive relaxation [35, 11], Vanka relaxation [16, 24, 36, 37], and Uzawa-type schemes [26] for the Stokes equations, most of the existing LFA has been for relaxation schemes using (symmetric) Gauss-Seidel (GS) approaches, and for simple finite-difference and finite-element discretizations. Considering modern multicore and accelerated parallel architectures, we focus on schemes based on weighted Jacobi relaxation with distributive, Braess-Sarazin, and Uzawa relaxation for common finite-element discretizations of the Stokes equations.

Some key conclusions of this analysis are as follows. First, while the LFA smoothing factor gives a good prediction of the true convergence factor for the stabilized discretizations with distributive weighted Jacobi and Braess-Sarazin relaxation, it does not for the Uzawa relaxation (in contrast to what is seen for the MAC discretization [18, 35]). For no cases does the LFA smoothing factor offer a good prediction of the true convergence behaviour for the (stable) $Q_2 - Q_1$ discretization, suggesting that the discretization is responsible for the lack of predictivity, consistent with the results in [15, 16]. For both stable and stabilized discretizations, we see that standard distributive weighted Jacobi relaxation loses some of its high efficiency, in contrast to what is seen for the MAC scheme [18, 35] but that robustness can be restored with an additional relaxation sweep. Exact Braess-Sarazin relaxation is also highly effective, with LFA-predicted $W(1, 1)$ convergence factors of $\frac{1}{9}$ in the stabilized cases and $\frac{1}{4}$ in the stable case. To realize these rates with inexact cycles, however, requires nested W-cycles to solve the approximate Schur complement equation accurately enough in the stabilized case, although simple weighted Ja-

cobi on the approximate Schur complement is observed to be sufficient in the stable case. For Uzawa-type relaxation, we see a notable gap between predicted convergence with exact inversion of the resulting Schur complement, versus inexact inversion, although some improvement is seen when replacing the approximate Schur complement with a mass matrix approximation, as is commonly used in block-diagonal preconditioners [38, 39, 40]. Overall, however, we see that distributive weighted Jacobi (DWJ) (with 2 sweeps of Jacobi relaxation on the pressure equation) outperforms both Braess-Sarazin relaxation (BSR) and Uzawa relaxation, for the stabilized discretizations, while DWJ and inexact BSR offer comparable performance for the stable discretization.

We organize this paper as follows. In Section 2, we introduce two stabilized $Q_1 - Q_1$ and the stable $Q_2 - Q_1$ mixed finite-element discretizations of the Stokes equations in two dimensions (2D). In Section 3, we first review the LFA approach, then discuss the Fourier representation for these discretizations. In Section 4, LFA is developed for DWJ, BSR, and Uzawa-type relaxation, and optimal LFA smoothing factors are derived for the two stabilized $Q_1 - Q_1$ methods with DWJ and BSR. Multigrid performance is presented to validate the theoretical results. Section 5 exhibits optimized LFA two-grid convergence factors and measured multigrid convergence factors for the $Q_2 - Q_1$ discretization. Furthermore, a comparison of the cost and effectiveness of the relaxation schemes is given. Conclusions are presented in Section 6.

2. Discretizations

In this paper, we consider the Stokes equations,

$$\begin{aligned} -\Delta \vec{u} + \nabla p &= \vec{f}, \\ \nabla \cdot \vec{u} &= 0, \end{aligned} \quad (1)$$

where \vec{u} is the velocity vector, p is the (scalar) pressure of a viscous fluid, and \vec{f} represents a (known) forcing term, together with suitable boundary conditions. Because of the nature of LFA, we validate our predictions against the problem with periodic boundary conditions on both \vec{u} and p . Discretizations of (1) typically lead to a linear system of the following form:

$$Kx = \begin{pmatrix} A & B^T \\ B & -\beta C \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} = b, \quad (2)$$

where A corresponds to the discretized vector Laplacian, and B is the negative of the discrete divergence operator. If the discretization is naturally unstable, then $C \neq 0$ is the stabilization matrix, otherwise $C = 0$. In this paper, we discuss two stabilized $Q_1 - Q_1$ and the stable $Q_2 - Q_1$ finite-element discretizations.

The natural finite-element approximation of Problem (1) is: Find $\vec{u}_h \in \mathcal{X}^h$ and $p_h \in \mathcal{H}^h$ such that

$$a(\vec{u}_h, \vec{v}_h) + b(p_h, \vec{v}_h) + b(q_h, \vec{u}_h) = g(\vec{v}_h), \text{ for all } \vec{v}_h \in \mathcal{X}_0^h \text{ and } q_h \in \mathcal{H}^h, \quad (3)$$

where

$$\begin{aligned} a(\vec{u}_h, \vec{v}_h) &= \int_{\Omega} \nabla \vec{u}_h : \nabla \vec{v}_h, \quad b(p_h, \vec{v}_h) = - \int_{\Omega} p_h \nabla \cdot \vec{v}_h, \\ g(\vec{v}_h) &= \int_{\Omega} \vec{f}_h \cdot \vec{v}_h, \end{aligned}$$

and $\mathcal{X}^h \subset H^1(\Omega)$, $\mathcal{H}^h \subset L_2(\Omega)$ are finite-element spaces. Here, $\mathcal{X}_0^h \subset \mathcal{X}^h$ satisfies homogeneous Dirichlet boundary conditions in place of any non-homogenous essential boundary conditions on \mathcal{X}^h . Problem (3) has a unique solution only when \mathcal{X}^h and \mathcal{H}^h satisfy an inf-sup condition (see [1, 41, 42, 43]).

2.1. Stabilized $Q_1 - Q_1$ discretizations

The standard equal-order approximation of (3) is well-known to be unstable [42, 1]. To circumvent this, a scaled pressure Laplacian term can be added to (3); for a uniform mesh with square elements of size h , we subtract

$$c(p_h, q_h) = \beta h^2 (\nabla p_h, \nabla q_h),$$

for $\beta > 0$. With this, the resulting linear system is given by

$$\begin{pmatrix} A & B^T \\ B & -\beta h^2 A_p \end{pmatrix} \begin{pmatrix} \mathcal{U} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} = b,$$

where A_p is the Q_1 Laplacian operator for the pressure. Denote $S = BA^{-1}B^T$, and $S_\beta = BA^{-1}B^T + \beta C$, where $C = h^2 A_p$. From [1], the red-black unstable mode $\mathbf{p} = \pm \mathbf{1}$, can be moved from a zero eigenvalue to a unit eigenvalue (giving stability without loss of accuracy) by choosing β so that

$$\frac{\mathbf{p}^T S_\beta \mathbf{p}}{\mathbf{p}^T Q \mathbf{p}} = \beta \frac{\mathbf{p}^T C \mathbf{p}}{\mathbf{p}^T Q \mathbf{p}} = 1, \quad (4)$$

where Q is the mass matrix. Substituting the bilinear stiffness and mass matrices into (4), we find $\beta = \frac{1}{24}$. We refer to this method as the Poisson-stabilized discretization (PoSD).

An L_2 projection to stabilize the $Q_1 - Q_1$ discretization, proposed in [43], stabilizes with

$$C(p_h, q_h) = (p_h - \Pi_0 p_h, q_h - \Pi_0 q_h), \quad (5)$$

where Π_0 is the L_2 projection from \mathcal{H}^h into the space of piecewise constant functions on the mesh. We refer to this method as the projection stabilized discretization (PrSD). The 4×4 element matrix C_4 of (5) is given by

$$C_4 = Q_4 - \mathbf{q}\mathbf{q}^T h^2,$$

where Q_4 is the 4×4 element mass matrix for the bilinear discretization and $\mathbf{q} = \left[\frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \right]^T$. In the projection stabilized method, we can write $C = Q - h^2 P$, where P is given by the 9-point stencil

$$P = \frac{1}{4} \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}.$$

Applying (4) to $C = Q - h^2 P$, we find that $\beta = 1$ is the optimal choice.

2.2. Stable $Q_2 - Q_1$ discretizations

In order to guarantee the well-posedness of the discrete system (2) with $C = 0$, the discretization of the velocity and pressure unknowns should satisfy an inf-sup condition,

$$\inf_{q_h \neq 0} \sup_{\vec{v}_h \neq \vec{0}} \frac{|b(q_h, \vec{v}_h)|}{\|\vec{v}_h\|_1 \|q_h\|_0} \geq \Gamma > 0,$$

where Γ is a constant. Taylor-Hood ($Q_2 - Q_1$) elements are well known to be stable [41, 1], where the basis functions associated with these elements are biquadratic for each component of the velocity field and bilinear for the pressure.

3. LFA preliminaries

3.1. Definitions and notations

In many cases, the LFA smoothing factor offers a good prediction of multigrid performance. Thus, we will explore the LFA smoothing factor and true (measured) multigrid convergence for the three types of relaxations considered here. We first introduce some terminology of LFA, following [10, 11]. We consider the following two-dimensional infinite uniform grids,

$$\mathbf{G}_h^j = \{\mathbf{x}^j := (x_1^j, x_2^j) = (k_1, k_2)h + \delta^j, (k_1, k_2) \in \mathbb{Z}^2\},$$

with

$$\delta^j = \begin{cases} (0, 0) & \text{if } j = 1, \\ (0, h/2) & \text{if } j = 2, \\ (h/2, 0) & \text{if } j = 3, \\ (h/2, h/2) & \text{if } j = 4. \end{cases}$$

The coarse grids, \mathbf{G}_{2h}^j , are defined similarly.

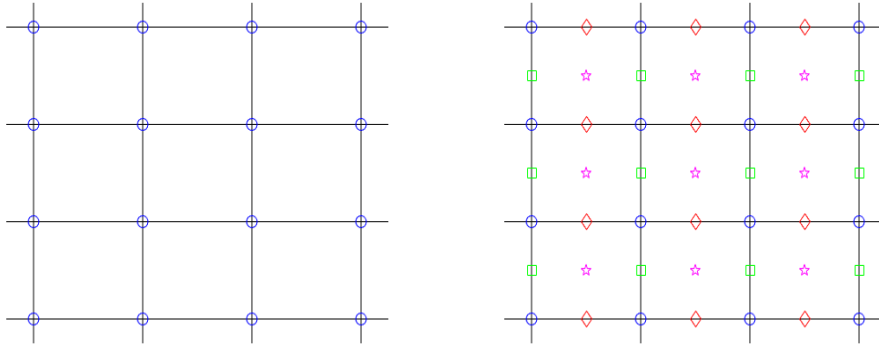


Figure 1: At left, the mesh used for Q_1 discretization. At right, the mesh used for Q_2 discretization. Points marked by \circ correspond to \mathbf{G}_h^1 , those marked by \diamond correspond to \mathbf{G}_h^2 , those marked by \square correspond to \mathbf{G}_h^3 and those marked by \star correspond to \mathbf{G}_h^4 .

Let L_h be a scalar Toeplitz operator defined by its stencil acting on \mathbf{G}_h^j as follows:

$$L_h \stackrel{\wedge}{=} [s_\kappa]_h \ (\kappa = (\kappa_1, \kappa_2) \in V); \quad L_h w_h(\mathbf{x}^j) = \sum_{\kappa \in V} s_\kappa w_h(\mathbf{x}^j + \kappa h), \quad (6)$$

with constant coefficients $s_\kappa \in \mathbb{R}$ (or \mathbb{C}), where $w_h(\mathbf{x}^j)$ is a function in $l^2(\mathbf{G}_h^j)$. Here, $V \subset \mathbb{Z}^2$ is a finite index set. Because L_h is formally diagonalized by the Fourier modes $\varphi(\boldsymbol{\theta}, \mathbf{x}^j) = e^{i\boldsymbol{\theta} \cdot \mathbf{x}^j/h} = e^{i\theta_1 x_1^j/h} e^{i\theta_2 x_2^j/h}$, where $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and $i^2 = -1$, we use $\varphi(\boldsymbol{\theta}, \mathbf{x}^j)$ as a Fourier basis with $\boldsymbol{\theta} \in [-\frac{\pi}{2}, \frac{3\pi}{2}]^2$. High and low frequencies for standard coarsening (as considered here) are given by

$$\boldsymbol{\theta} \in T^{\text{low}} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2, \quad \boldsymbol{\theta} \in T^{\text{high}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right]^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2.$$

Definition 3.1. *If, for all functions $\varphi(\boldsymbol{\theta}, \mathbf{x}^j)$,*

$$L_h \varphi(\boldsymbol{\theta}, \mathbf{x}^j) = \widetilde{L}_h(\boldsymbol{\theta}) \varphi(\boldsymbol{\theta}, \mathbf{x}^j),$$

we call $\widetilde{L}_h(\boldsymbol{\theta}) = \sum_{\kappa \in V} s_\kappa e^{i\boldsymbol{\theta} \cdot \kappa}$ the symbol of L_h .

In what follows, we consider (3×3) linear systems of operators, which read

$$\mathcal{L}_h = \begin{pmatrix} L_h^{1,1} & L_h^{1,2} & L_h^{1,3} \\ L_h^{2,1} & L_h^{2,2} & L_h^{2,3} \\ L_h^{3,1} & L_h^{3,2} & L_h^{3,3} \end{pmatrix} = \begin{pmatrix} -\Delta_h & 0 & (\partial_x)_h \\ 0 & -\Delta_h & (\partial_y)_h \\ -(\partial_x)_h & -(\partial_y)_h & L_h^{3,3} \end{pmatrix}, \quad (7)$$

where $L_h^{3,3}$ depends on which discretization we use.

For the stabilized $Q_1 - Q_1$ approximations, the degrees of freedom for both velocity and pressure are only located on \mathbf{G}_h^1 as pictured at left of Figure 1. In this setting, the $L_h^{k,\ell}$ ($k, \ell = 1, 2, 3$) in (7) are scalar Toeplitz operators. Denote $\widetilde{\mathcal{L}}_h$ as the symbol of \mathcal{L}_h . Each entry in $\widetilde{\mathcal{L}}_h$ is computed as the (scalar) symbol of the corresponding block of $L_h^{k,\ell}$, following Definition 3.1. Thus, $\widetilde{\mathcal{L}}_h$ is a 3×3 matrix. All blocks in \mathcal{L}_h are diagonalized by the same transformation on a collocated mesh.

However, for the $Q_2 - Q_1$ discretization, the degrees of freedom for velocity are located on $\mathbf{G}_h = \bigcup_{j=1}^4 \mathbf{G}_h^j$, containing four types of meshpoints as shown at right of Figure 1. The Laplace operator in (7) is defined by extending (6), with V taken to be a finite index set of values, $V = V_N \cup V_X \cup V_Y \cup V_C$ with $V_N \subset \mathbb{Z}^2$, $V_X \subset \{(z_x + \frac{1}{2}, z_y) | (z_x, z_y) \in \mathbb{Z}^2\}$, $V_Y \subset \{(z_x, z_y + \frac{1}{2}) | (z_x, z_y) \in \mathbb{Z}^2\}$, and $V_C \subset \{(z_x + \frac{1}{2}, z_y + \frac{1}{2}) | (z_x, z_y) \in \mathbb{Z}^2\}$. With this, the (scalar) Q_2 Laplace operator is naturally treated as a block operator, and the Fourier representation of each block can be calculated based on Definition 3.1, with the Fourier bases adapted to account for the staggering of the mesh points. Thus, the symbols of $L_h^{1,1}$ and $L_h^{2,2}$ are 4×4 matrices. For more details of LFA for the Laplace operator using higher-order finite-element methods, refer to [15]. Similarly to the Laplace operator, both terms in the gradient, $(\partial_x)_h$ and $(\partial_y)_h$, can be treated as (4×1) -block operators. Then, the symbols of $L_h^{1,3}$ and $L_h^{2,3}$ are 4×1 matrices, calculated based on Definition 3.1 adapted for the mesh staggering. The symbols of $L_h^{3,1}$ and $L_h^{3,2}$ are the conjugate transposes of those of $L_h^{1,3}$ and $L_h^{2,3}$, respectively. Finally, $L_h^{3,3} = 0$. Accordingly, $\widetilde{\mathcal{L}}_h$ is a 9×9 matrix for the $Q_2 - Q_1$ discretization.

Definition 3.2. The error-propagation symbol, $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$, for a block smoother \mathcal{S}_h on the infinite grid \mathbf{G}_h satisfies

$$\mathcal{S}_h\varphi(\boldsymbol{\theta}, \mathbf{x}) = \widetilde{\mathcal{S}}_h\varphi(\boldsymbol{\theta}, \mathbf{x}), \quad \boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^2,$$

for all $\varphi(\boldsymbol{\theta}, \mathbf{x})$, and the corresponding smoothing factor for \mathcal{S}_h is given by

$$\mu_{\text{loc}} = \mu_{\text{loc}}(\mathcal{S}_h) = \max_{\boldsymbol{\theta} \in \mathcal{T}^{\text{high}}} \{|\lambda(\widetilde{\mathcal{S}}_h(\boldsymbol{\theta}))|\},$$

where λ is an eigenvalue of $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$.

In Definition 3.2, $\mathbf{G}_h = \mathbf{G}_h^1$ for the stabilized case (and $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$ is a 3×3 matrix) and $\mathbf{G}_h = \bigcup_{j=1}^4 \mathbf{G}_h^j$ for the stable case (where $\widetilde{\mathcal{S}}_h(\boldsymbol{\theta})$ is a 9×9 matrix).

The error-propagation symbol for a relaxation scheme, represented by matrix M_h , applied to either the stabilized or stable scheme is written as

$$\widetilde{\mathcal{S}}_h(\mathbf{p}, \omega, \boldsymbol{\theta}) = I - \omega \widetilde{M}_h^{-1}(\boldsymbol{\theta}) \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}),$$

where \mathbf{p} represents parameters within M_h , the block approximation to \mathcal{L}_h , ω is an overall weighting factor, and \widetilde{M}_h and $\widetilde{\mathcal{L}}_h$ are the symbols for M_h and \mathcal{L}_h , respectively. Note that μ_{loc} is a function of some parameters in Definition 3.2. In this paper, we focus on minimizing μ_{loc} with respect to these parameters, to obtain the optimal LFA smoothing factor.

Definition 3.3. Let \mathcal{D} be the set of allowable parameters and define the optimal smoothing factor over \mathcal{D} as

$$\mu_{\text{opt}} = \min_{\mathcal{D}} \mu_{\text{loc}}.$$

If the standard LFA assumption of an ‘‘ideal’’ CGC holds, then the two-grid convergence factor can be estimated by the smoothing factor, which is easy to compute. However, as expected, we will see that this idealized CGC does not lead to a good prediction for some cases we consider below. When the LFA smoothing factor fails to predict the true two-grid convergence factor, the LFA two-grid convergence factor can still be used. Thus, we give a brief introduction to the LFA two-grid convergence factor in the following.

Let

$$\begin{aligned} \boldsymbol{\alpha} &= (\alpha_1, \alpha_2) \in \{(0, 0), (1, 0), (0, 1), (1, 1)\}, \\ \boldsymbol{\theta}^\alpha &= (\theta_1^{\alpha_1}, \theta_2^{\alpha_2}) = \boldsymbol{\theta} + \pi \cdot \boldsymbol{\alpha}, \quad \boldsymbol{\theta} := \boldsymbol{\theta}^{00} \in \mathcal{T}^{\text{low}}. \end{aligned}$$

We use the ordering of $\boldsymbol{\alpha} = (0, 0), (1, 0), (0, 1), (1, 1)$ for the four harmonics. To apply LFA to the two-grid operator,

$$\mathbf{M}_h^{\text{TGM}} = \mathcal{S}_h^{v_2} \mathcal{M}_h^{\text{CGC}} \mathcal{S}_h^{v_1}, \quad (8)$$

we require the representation of the CGC operator,

$$\mathcal{M}_h^{\text{CGC}} = I - P_h(\mathcal{L}_{2h}^*)^{-1} R_h \mathcal{L}_h,$$

where P_h is the multigrid interpolation operator and R_h is the restriction operator. The coarse-grid operator, \mathcal{L}_{2h}^* , can be either the Galerkin or rediscrization operator.

Inserting the representations of $\mathcal{S}_h, \mathcal{L}_h, \mathcal{L}_{2h}^*, P_h, R_h$ into (8), we obtain the Fourier representation of two-grid error-propagation operator as

$$\widetilde{\mathbf{M}}_h^{\text{TGM}}(\boldsymbol{\theta}) = \widetilde{\mathbf{S}}_h^{\nu_2}(\boldsymbol{\theta})(I - \widetilde{\mathbf{P}}_h(\boldsymbol{\theta})(\widetilde{\mathcal{L}}_{2h}^*(2\boldsymbol{\theta}))^{-1}\widetilde{\mathbf{R}}_h(\boldsymbol{\theta})\widetilde{\mathcal{L}}_h(\boldsymbol{\theta}))\widetilde{\mathbf{S}}_h^{\nu_1}(\boldsymbol{\theta}),$$

where

$$\begin{aligned}\widetilde{\mathcal{L}}_h(\boldsymbol{\theta}) &= \text{diag}\{\widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{00}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{10}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{01}), \widetilde{\mathcal{L}}_h(\boldsymbol{\theta}^{11})\}, \\ \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}) &= \text{diag}\{\widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{00}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{10}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{01}), \widetilde{\mathcal{S}}_h(\boldsymbol{\theta}^{11})\}, \\ \widetilde{\mathbf{P}}_h(\boldsymbol{\theta}) &= (\widetilde{P}_h(\boldsymbol{\theta}^{00}); \widetilde{P}_h(\boldsymbol{\theta}^{10}); \widetilde{P}_h(\boldsymbol{\theta}^{01}); \widetilde{P}_h(\boldsymbol{\theta}^{11})), \\ \widetilde{\mathbf{R}}_h(\boldsymbol{\theta}) &= (\widetilde{R}_h(\boldsymbol{\theta}^{00}), \widetilde{R}_h(\boldsymbol{\theta}^{10}), \widetilde{R}_h(\boldsymbol{\theta}^{01}), \widetilde{R}_h(\boldsymbol{\theta}^{11})),\end{aligned}$$

in which $\text{diag}\{T_1, T_2, T_3, T_4\}$ stands for the block diagonal matrix with diagonal blocks, T_1, T_2, T_3 , and T_4 .

Here, we use the standard finite-element interpolation operators and their transposes for restriction. For Q_1 , the symbol is well-known [10] while, for the nodal basis for Q_2 , the symbol is given in [15].

Definition 3.4. *The asymptotic two-grid convergence factor, ρ_{asp} , is defined as*

$$\rho_{\text{asp}} = \sup\{\rho(\widetilde{\mathbf{M}}_h(\boldsymbol{\theta})^{\text{TGM}}) : \boldsymbol{\theta} \in T^{\text{low}}\}.$$

In what follows, we consider a discrete form of ρ_{asp} , denoted by ρ_h , resulting from sampling ρ_{asp} over only a finite set of frequencies. For simplicity, we drop the subscript h throughout the rest of this paper, unless necessary for clarity.

3.2. Fourier representation of discretization operators

3.2.1. Fourier representation of the stabilized $Q_1 - Q_1$ discretization

By standard calculation, the symbols of the Q_1 stiffness and mass stencils are

$$\begin{aligned}\widetilde{A}_{Q_1}(\theta_1, \theta_2) &= \frac{2}{3}(4 - \cos \theta_1 - \cos \theta_2 - 2 \cos \theta_1 \cos \theta_2), \\ \widetilde{M}_{Q_1}(\theta_1, \theta_2) &= \frac{h^2}{9}(4 + 2 \cos \theta_1 + 2 \cos \theta_2 + \cos \theta_1 \cos \theta_2),\end{aligned}$$

respectively. The stencils of the partial derivative operators $(\partial_x)_h$ and $(\partial_y)_h$ are

$$B_x^T = \frac{h}{12} \begin{bmatrix} -1 & 0 & 1 \\ -4 & 0 & 4 \\ -1 & 0 & 1 \end{bmatrix}, \quad B_y^T = \frac{h}{12} \begin{bmatrix} 1 & 4 & 1 \\ 0 & 0 & 0 \\ -1 & -4 & -1 \end{bmatrix},$$

respectively, and the corresponding symbols are

$$\widetilde{B}_x^T(\theta_1, \theta_2) = \frac{ih}{3} \sin \theta_1 (2 + \cos \theta_2), \quad \widetilde{B}_y^T(\theta_1, \theta_2) = \frac{ih}{3} (2 + \cos \theta_1) \sin \theta_2,$$

where T denotes the conjugate transpose. Thus, the symbols of the stabilized finite-element discretizations of the Stokes equations are given by

$$\widetilde{\mathcal{L}}(\theta_1, \theta_2) = \begin{pmatrix} \widetilde{A}_{Q_1} & 0 & \widetilde{B}_x^T \\ 0 & \widetilde{A}_{Q_1} & \widetilde{B}_y^T \\ \widetilde{B}_x & \widetilde{B}_y & \widetilde{L}_h^{3,3} \end{pmatrix} := \begin{pmatrix} a & 0 & b_1 \\ 0 & a & b_2 \\ -b_1 & -b_2 & -c \end{pmatrix}.$$

For the Poisson-stabilized discretization, the symbol of $-L_h^{3,3}$ is $c = c_1 = a\beta h^2$. For the projection stabilized method, following (5), the symbol of $-L_h^{3,3}$ is

$$c_2 = \left(\frac{4 + 2 \cos \theta_1 + 2 \cos \theta_2 + \cos \theta_1 \cos \theta_2}{9} - \frac{(1 + \cos \theta_1)(1 + \cos \theta_2)}{4} \right) h^2. \quad (9)$$

For convenience, we write $-C$ for the last block of Equation (2), and its symbol as $-c$ in the rest of this paper.

3.2.2. Fourier representation of stable $Q_2 - Q_1$ discretizations

The symbols of the stiffness and mass stencils for the Q_2 discretization using nodal basis functions in 1D are

$$\widetilde{A}_{Q_2}(\theta) = \frac{1}{3h} \begin{pmatrix} 14 + 2 \cos \theta & -16 \cos \frac{\theta}{2} \\ -16 \cos \frac{\theta}{2} & 16 \end{pmatrix}, \quad \widetilde{M}_{Q_2}(\theta) = \frac{h}{30} \begin{pmatrix} 8 - 2 \cos \theta & 4 \cos \frac{\theta}{2} \\ 4 \cos \frac{\theta}{2} & 16 \end{pmatrix},$$

respectively [15]. Here, we note that the (1, 1) entries correspond to the symbols associated with basis functions at the nodes of the mesh, while the (2, 2) entries correspond to the symbols associated with cell-centre (bubble) basis functions. The off-diagonal entries express the interaction between the two types of basis functions. Then, the Fourier representation of $-\Delta_h$ in 2D can be written as a tensor product,

$$\widetilde{A}_2(\theta_1, \theta_2) = \widetilde{A}_{Q_2}(\theta_2) \otimes \widetilde{M}_{Q_2}(\theta_1) + \widetilde{M}_{Q_2}(\theta_2) \otimes \widetilde{A}_{Q_2}(\theta_1).$$

The tensor product preserves block structuring; that is, $\widetilde{A}_2(\theta_1, \theta_2)$ is a 4×4 matrix, ordered as mesh nodes, x -edge midpoints, y -edge midpoints, and cell centres. Each row of $\widetilde{A}_2(\theta_1, \theta_2)$ reflects the connections between one of the four types of degrees of freedom with each of these four types. Similarly, there are four types of stencils for $(\partial_x)_h$ and $(\partial_y)_h$.

The stencils and the symbols of $(\partial_x)_h$ for the nodal, x -edge, y -edge, and cell-centre degrees of freedom are

$$\begin{aligned} B_N &= \frac{h}{18} \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, & \widetilde{B}_N(\theta_1, \theta_2) &= \frac{ih}{9} \sin \theta_1, \\ B_X &= \frac{h}{18} \begin{bmatrix} 0 & 0 \\ -4 & 4 \\ 0 & 0 \end{bmatrix}, & \widetilde{B}_X(\theta_1, \theta_2) &= \frac{2ih}{9} \sin \frac{\theta_1}{2}, \\ B_Y &= \frac{h}{18} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, & \widetilde{B}_Y(\theta_1, \theta_2) &= \frac{2ih}{9} \sin \theta_1 \cos \frac{\theta_2}{2}, \\ B_C &= \frac{h}{18} \begin{bmatrix} -4 & 4 \\ -4 & 4 \end{bmatrix}, & \widetilde{B}_C(\theta_1, \theta_2) &= \frac{8ih}{9} \sin \frac{\theta_1}{2} \cos \frac{\theta_2}{2}, \end{aligned}$$

respectively. Denote $\widetilde{B}_{Q_2,x}(\theta_1, \theta_2)^T = [\widetilde{B}_N; \widetilde{B}_X; \widetilde{B}_Y; \widetilde{B}_C]$.

Similarly to $\widetilde{B}_{Q_2,x}(\theta_1, \theta_2)^T$, the symbol of the stencil of $(\partial_y)_h$ can be written as

$$\widetilde{B}_{Q_2,y}(\theta_1, \theta_2)^T = [\widetilde{B}_N(\theta_2, \theta_1); \widetilde{B}_Y(\theta_2, \theta_1); \widetilde{B}_X(\theta_2, \theta_1); \widetilde{B}_C(\theta_2, \theta_1)].$$

Thus, the Fourier representation of the $Q_2 - Q_1$ finite-element discretization of the Stokes equations can be written as

$$\tilde{\mathcal{L}}_h(\theta_1, \theta_2) = \begin{pmatrix} \tilde{A}_2(\theta_1, \theta_2) & 0 & \tilde{B}_{Q_2,x}(\theta_1, \theta_2)^T \\ 0 & \tilde{A}_2(\theta_1, \theta_2) & \tilde{B}_{Q_2,y}(\theta_1, \theta_2)^T \\ \tilde{B}_{Q_2,x}(\theta_1, \theta_2) & \tilde{B}_{Q_2,y}(\theta_1, \theta_2) & 0 \end{pmatrix}. \quad (10)$$

Note that the Fourier symbol for the $Q_2 - Q_1$ discretization is a 9×9 matrix, and that the LFA smoothing factor for the Q_2 approximation generally fails to predict the true two-grid convergence factor [15, 16]. The same behavior is seen for the relaxation schemes considered here. Therefore, we do not present smoothing factor analysis for this case and only optimize two-grid LFA predictions numerically.

4. Relaxation for $Q_1 - Q_1$ discretizations

4.1. DWJ relaxation

Distributive GS (DGS) relaxation [5, 32] is well known to be highly efficient for the MAC finite-difference discretization [10], and other discretizations [33, 44]. Its sequential nature is often seen as a significant drawback. However, Distributive weighted Jacobi (DWJ) relaxation was recently shown to achieve good performance for the MAC discretization [18]. Thus, we consider DWJ relaxation for the finite-element discretizations considered here. The discretized distribution operator can be represented by the preconditioner

$$\mathcal{P} = \begin{pmatrix} I_h & 0 & (\partial_x)_h \\ 0 & I_h & (\partial_y)_h \\ 0 & 0 & \Delta_h \end{pmatrix}.$$

Then, we apply blockwise weighted-Jacobi relaxation to the distributed operator

$$\mathcal{L}\mathcal{P} \approx \mathcal{L}^* = \begin{pmatrix} -\Delta_h & 0 & 0 \\ 0 & -\Delta_h & 0 \\ -(\partial_x)_h & -(\partial_y)_h & -(\partial_x)_h^2 - (\partial_y)_h^2 + L^{3,3}\Delta_h \end{pmatrix}, \quad (11)$$

where we note that the operators $(\partial_x)_h^2$ and $(\partial_y)_h^2$ are formed by taking products of the discrete derivative operators and, thus, do not satisfy the identity $(\partial_x)_h^2 + (\partial_y)_h^2 = \Delta_h$.

The discrete matrix form of \mathcal{P} is

$$\mathcal{P} = \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix},$$

where A_p is the Laplacian operator discretized at the pressure points. For standard distributive weighted-Jacobi relaxation (with weights α_1, α_2), we need to solve a system of the form

$$M_D \delta \hat{x} = \begin{pmatrix} \alpha_1 \text{diag}(A) & 0 \\ B & \alpha_2 h^2 I \end{pmatrix} \begin{pmatrix} \delta \hat{\mathcal{U}} \\ \delta \hat{p} \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \end{pmatrix}, \quad (12)$$

then distribute the updates as $\delta x = \mathcal{P} \delta \hat{x}$. We use h^2 in the (2, 2) block of (12), because the diagonal entries of the (2, 2) block will be of the form of a constant times h^2 (up to boundary

conditions), for both stabilization terms. The error propagation operator for the scheme is, then, $I - \omega \mathcal{P} M_D^{-1} \mathcal{L}$.

The symbol of the blockwise weighted-Jacobi operator, M_D , is

$$\widetilde{M}_D(\theta_1, \theta_2) = \begin{pmatrix} \frac{8}{3}\alpha_1 & 0 & 0 \\ 0 & \frac{8}{3}\alpha_1 & 0 \\ -b_1 & -b_2 & h^2\alpha_2 \end{pmatrix}.$$

By standard calculation, the eigenvalues of the error-propagation symbol, $\widetilde{\mathcal{S}}_D(\alpha_1, \alpha_2, \omega, \theta) = I - \omega \widetilde{\mathcal{P}} \widetilde{M}_D^{-1} \widetilde{\mathcal{L}}$, are

$$1 - \frac{\omega}{\alpha_1} y_1, \quad 1 - \frac{\omega}{\alpha_1} y_1, \quad 1 - \frac{\omega}{\alpha_2} y_2, \quad (13)$$

where $y_1 = \frac{3a}{8}$ and $y_2 = \frac{-b_1^2 - b_2^2 + ac}{h^2}$.

Noting that $y_1 = \frac{3a}{8}$ is very simple, we first consider a lower bound on the optimal LFA smoothing factor corresponding to y_1 .

Lemma 4.1.

$$\mu^* := \min_{(\alpha_1, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ \left| 1 - \frac{\omega}{\alpha_1} y_1 \right| \right\} = \frac{1}{3},$$

and this value is achieved if and only if $\frac{\omega}{\alpha_1} = \frac{8}{9}$.

Proof. It is easy to check that $a = \frac{2(4 - \cos \theta_1 - \cos \theta_2 - 2 \cos \theta_1 \cos \theta_2)}{3} \in [2, 4]$ for $\theta \in T^{\text{high}}$. The minimum of y_1 is $y_{1,\min} = \frac{3}{4}$ with $(\cos \theta_1, \cos \theta_2) = (0, 1)$ or $(1, 0)$ and the maximum is $y_{1,\max} = \frac{3}{2}$ with $(\cos \theta_1, \cos \theta_2) = (1, -1)$ or $(-1, 1)$. Thus, $\mu^* = \frac{y_{1,\max} + y_{1,\min}}{y_{1,\max} - y_{1,\min}} = \frac{1}{3}$ under the condition $\frac{\omega}{\alpha_1} = \frac{2}{y_{1,\min} + y_{1,\max}} = \frac{8}{9}$. \square

Remark 4.1. The optimal smoothing factor for damped Jacobi relaxation for the Q_1 finite-element discretization of the Laplacian is $\frac{1}{3}$ with $\frac{\omega}{\alpha} = \frac{8}{9}$. Thus, this offers an intuitive lower bound on the possible performance of block relaxation schemes that include this as a piece of the overall relaxation.

From (13), we see that the only difference between the eigenvalues of DWJ relaxation for the Poisson-stabilized and projection stabilized methods is in the third eigenvalue, which depends on y_2 and, consequently, on the stabilization term.

4.1.1. Poisson-stabilized discretization with DWJ relaxation

For the Poisson-stabilized case, $y_2 = \frac{-b_1^2 - b_2^2 + ac}{h^2}$ with $c = \beta \alpha h^2$ and $\beta = \frac{1}{24}$. By standard calculation, $y_{2,\min} = \frac{8}{27}$, with $(\cos \theta_1, \cos \theta_2) = (-1, -1)$, and $y_{2,\max} = \frac{64}{51}$ with $(\cos \theta_1, \cos \theta_2) = (\frac{8}{17}, 0)$ or $(0, \frac{8}{17})$.

Theorem 4.1. The optimal smoothing factor for the Poisson-stabilized discretization with DWJ relaxation is $\frac{55}{89}$, that is,

$$\mu_{\text{opt}} = \min_{(\alpha_1, \omega, \alpha_2)} \max_{\theta \in T^{\text{high}}} \left\{ \left| \lambda(\widetilde{\mathcal{S}}_D(\alpha_1, \alpha_2, \omega, \theta)) \right| \right\} = \frac{55}{89} \approx 0.618,$$

and is achieved if and only if

$$\frac{\omega}{\alpha_2} = \frac{459}{356}, \quad \frac{136}{267} \leq \frac{\omega}{\alpha_1} \leq \frac{96}{89}. \quad (14)$$

Proof. $\min_{(\alpha_2, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ \left| 1 - \frac{\omega}{\alpha_2} y_2 \right| \right\} = \frac{y_{2,\text{max}} - y_{2,\text{min}}}{y_{2,\text{max}} + y_{2,\text{min}}} = \frac{55}{89}$ with the condition that $\frac{\omega}{\alpha_2} = \frac{2}{y_{2,\text{max}} + y_{2,\text{min}}} = \frac{459}{356}$. Because $\frac{55}{89} > \frac{1}{3}$, we need to require $|1 - \frac{\omega}{\alpha_1} y_1| \leq \frac{55}{89}$ for all y_1 to achieve this factor. It follows that $\frac{136}{267} \leq \frac{\omega}{\alpha_1} \leq \frac{96}{89}$. \square

4.1.2. Projection stabilized discretization with DWJ relaxation

For the projection stabilized discretization, y_2 depends on c_2 given in (9), and standard calculation gives $y_{2,\text{min}} = \frac{8}{27}$ with $(\cos \theta_1, \cos \theta_2) = (-1, -1)$ and $y_{2,\text{max}} = \frac{3}{2}$ with $(\cos \theta_1, \cos \theta_2) = (-\frac{1}{2}, 1)$ or $(1, -\frac{1}{2})$.

Theorem 4.2. *The optimal smoothing factor for the projection stabilized discretization with DWJ relaxation is $\frac{65}{97}$, that is,*

$$\mu_{\text{opt}} = \min_{(\alpha_1, \omega, \alpha_2)} \max_{\theta \in T^{\text{high}}} \left\{ \left| \lambda(\tilde{\mathcal{S}}_D(\alpha_1, \alpha_2, \omega, \theta)) \right| \right\} = \frac{65}{97} \approx 0.670,$$

and is achieved if and only if

$$\frac{\omega}{\alpha_2} = \frac{108}{97}, \quad \frac{128}{291} \leq \frac{\omega}{\alpha_1} \leq \frac{108}{97}. \quad (15)$$

Proof. $\min_{(\alpha_2, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ \left| 1 - \frac{\omega}{\alpha_2} y_2 \right| \right\} = \frac{y_{2,\text{max}} - y_{2,\text{min}}}{y_{2,\text{max}} + y_{2,\text{min}}} = \frac{65}{97}$ with the condition that $\frac{\omega}{\alpha_2} = \frac{2}{y_{2,\text{max}} + y_{2,\text{min}}} = \frac{108}{97}$. Since $\frac{65}{97} > \frac{1}{3}$, we need to require $|1 - \frac{\omega}{\alpha_1} y_1| \leq \frac{65}{97}$ for all y_1 to achieve this factor, which leads to $\frac{128}{291} \leq \frac{\omega}{\alpha_1} \leq \frac{108}{97}$. \square

Comparing the Poisson-stabilized and projection stabilized discretizations using DWJ, we see that the optimal LFA smoothing factor for the Poisson-stabilized discretization slightly outperforms that of the projection stabilized discretization. In both cases, a stronger relaxation on the (3, 3) block of (11) would be needed in order to improve performance to match the lower bound on the convergence factor of $\frac{1}{3}$. A natural approach is to using more iterations to solve the pressure equation in DWJ. We explore the LFA predictions for this case in the following.

4.1.3. Stabilized discretization with 2 sweeps of Jacobi for DWJ relaxation

Denote the (3, 3) block of (11) as G . We consider applying two sweeps of weighted-Jacobi relaxation with equal weights, ω_J , on the pressure equation. As before, we note that G has a constant diagonal entry proportional to h^2 , so we write weighted Jacobi relaxation on G as $I - G_J^{-1}G$ for $G_J = \frac{h^2}{\omega_J}I$. Thus, we can represent this relaxation scheme as solving

$$M_{D,J} \delta \hat{x} = \begin{pmatrix} \alpha_1 \text{diag}(A) & 0 \\ B & \hat{G} \end{pmatrix} \begin{pmatrix} \delta \hat{u} \\ \delta \hat{p} \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \end{pmatrix}, \quad (16)$$

where $\hat{G} = (2G_J^{-1} - G_J^{-1}GG_J^{-1})^{-1}$. The symbol of $M_{D,J}$, is

$$\tilde{M}_{D,J}(\theta_1, \theta_2) = \begin{pmatrix} \frac{8}{3}\alpha_1 & 0 & 0 \\ 0 & \frac{8}{3}\alpha_1 & 0 \\ -b_1 & -b_2 & \frac{h^2}{2\omega_J - \omega_J^2 y_2} \end{pmatrix}.$$

By standard calculation, the eigenvalues of the error-propagation symbol, $\widetilde{\mathcal{S}}_{D,J}(\alpha_1, \omega_J, \omega, \theta) = I - \omega \widetilde{\mathcal{P}} \widetilde{M}_{D,J}^{-1} \widetilde{\mathcal{L}}$, are

$$1 - \frac{\omega}{\alpha_1} y_1, \quad 1 - \frac{\omega}{\alpha_1} y_1, \quad 1 - \omega y_3, \quad (17)$$

where $y_3 = \omega_J y_2 (2 - \omega_J y_2)$, where the symbol of G is $h^2 y_2$, with y_2 defined as in (13). Note that $\widetilde{\mathcal{S}}_{D,J}$ has the same eigenvalue, $1 - \frac{\omega}{\alpha_1} y_1$ as that of $\widetilde{\mathcal{S}}_D$. A natural question is whether

$\min_{(\alpha_1, \omega_J, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ |1 - \omega y_3| \right\} = \frac{1}{3}$, which is shown in the following theorems.

Theorem 4.3. *The optimal smoothing factor for the Poisson-stabilized discretization with 2 sweeps of Jacobi for DWJ relaxation is $\frac{1}{3}$, that is,*

$$\mu_{\text{opt}} = \min_{(\alpha_1, \omega_J, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ |\lambda(\widetilde{\mathcal{S}}_{D,J}(\alpha_1, \omega_J, \omega, \theta))| \right\} = \frac{1}{3},$$

and is achieved if and only if $\frac{\omega}{\alpha_1} = \frac{8}{9}$ and either

$$\begin{aligned} \frac{459}{356} &\leq \omega_J \leq \frac{51}{64} \left(1 + \frac{\sqrt{2}}{2}\right), \\ \frac{2}{3 \left(\frac{64}{51} \omega_J (2 - \frac{64}{51} \omega_J)\right)} &\leq \omega \leq \frac{4}{3}, \end{aligned}$$

or

$$\begin{aligned} \frac{27}{8} \left(1 - \frac{\sqrt{2}}{2}\right) &\leq \omega_J \leq \frac{459}{356}, \\ \frac{2}{3 \left(\frac{8}{27} \omega_J (2 - \frac{8}{27} \omega_J)\right)} &\leq \omega \leq \frac{4}{3}. \end{aligned}$$

Proof. Recall that $y_3 = \omega_J y_2 (2 - \omega_J y_2) := \xi(2 - \xi)$, where $\xi = \omega_J y_2$. Let

$$\mu^{**} = \min_{(\alpha_1, \omega_J, \omega)} \max_{\theta \in T^{\text{high}}} \left\{ |1 - \omega y_3| \right\}. \quad (18)$$

We first show that $\mu^{**} \leq \frac{1}{3}$ under some conditions on the parameters, ω_J and ω . Let $y_{3,\min}$ and $y_{3,\max}$ be the maximum and minimum of $y_3(\xi) = \xi(2 - \xi)$, respectively. If $\mu^{**} \leq \frac{1}{3}$, then it must be that

$$\frac{2}{3y_{3,\min}} \leq \omega \leq \frac{4}{3y_{3,\max}}. \quad (19)$$

Next, we need to find what $y_{3,\min}$ and $y_{3,\max}$ are. As discussed earlier, $y_2 \in [\frac{8}{27}, \frac{64}{51}]$. Thus, $\xi \in [\frac{8}{27} \omega_J, \frac{64}{51} \omega_J]$, where $\omega_J > 0$. Note that $y_3(\xi) = \xi(2 - \xi) = -(\xi - 1)^2 + 1$ is a quadratic function with the axis of symmetric, $\xi = 1$. Thus, the extreme values of $y_3(\xi)$ are achieved at the points $\frac{8}{27} \omega_J$, $\frac{64}{51} \omega_J$ or 1. Based on $\frac{64}{51} \omega_J \leq 1$ and $\frac{64}{51} \omega_J \geq 1$, we consider two cases.

1. If $\frac{64}{51} \omega_J \leq 1$, we have

$$y_{3,\min} = \frac{8}{27} \omega_J \left(2 - \frac{8}{27} \omega_J\right), \quad y_{3,\max} = \frac{64}{51} \omega_J \left(2 - \frac{64}{51} \omega_J\right). \quad (20)$$

Note that (19) indicates that $y_{3,\max} \leq 2y_{3,\min}$. Combining with (20) leads to $\omega_J \geq \frac{373}{394}$. However, $\omega_J \leq \frac{51}{64} < \frac{373}{394}$. Thus, there is no ω_J such that $\mu^{**} \leq \frac{1}{3}$ in this case.

2. To guarantee that $|1 - \omega y_3| = |1 - \omega \xi(2 - \xi)| < 1$, we require that $0 < \xi < 2$. Assume that $1 \leq \frac{64}{51}\omega_J < 2$. It follows that $\frac{8}{27}\omega_J < 1 \leq \frac{64}{51}\omega_J$. Recall that $y_3(\xi) = \xi(2 - \xi) = -(\xi - 1)^2 + 1$.

- If $(\frac{64}{51}\omega_J - 1) \geq (1 - \frac{8}{27}\omega_J)$, we have

$$\frac{459}{356} \leq \omega_J < \frac{51}{32}. \quad (21)$$

Then, the extreme values of $y_3(\xi)$ are

$$y_{3,\min} = \frac{64}{51}\omega_J \left(2 - \frac{64}{51}\omega_J\right), \quad y_{3,\max} = y_3(1) = 1. \quad (22)$$

Substituting (22) in to (19), we have

$$\frac{2}{3\left(\frac{64}{51}\omega_J(2 - \frac{64}{51}\omega_J)\right)} \leq \omega \leq \frac{4}{3}. \quad (23)$$

To guarantee (23) makes sense, in combination with (21) gives

$$\frac{459}{356} \leq \omega_J \leq \frac{51}{64}\left(1 + \frac{\sqrt{2}}{2}\right). \quad (24)$$

Recall that there is another eigenvalue, $1 - \frac{\omega}{\alpha_1}y_1$, of $\widetilde{\mathcal{S}}_{D,J}$. In order to obtain $\mu_{\text{opt}} = \frac{1}{3}$, we thus require

$$\begin{aligned} \frac{459}{356} &\leq \omega_J \leq \frac{51}{64}\left(1 + \frac{\sqrt{2}}{2}\right), \\ \frac{2}{3\left(\frac{64}{51}\omega_J(2 - \frac{64}{51}\omega_J)\right)} &\leq \omega \leq \frac{4}{3}, \\ \frac{\omega}{\alpha_1} &= \frac{8}{9}. \end{aligned}$$

- A similar argument holds if $(\frac{64}{51}\omega_J - 1) \leq (1 - \frac{8}{27}\omega_J)$, leading to the second set of conditions.

□

Note that the set of parameters values defined in Theorem 4.3 is not empty, with parameters $\alpha_1 = \frac{3}{2}$, $\omega = \frac{4}{3}$ and $\omega_J = 1$ in the set.

Theorem 4.4. *The optimal smoothing factor for the projection stabilized discretization with two sweeps of Jacobi for DWJ relaxation is $\frac{1}{3}$, that is,*

$$\mu_{\text{opt}} = \min_{(\alpha_1, \omega, \alpha_2)} \max_{\theta \in T^{\text{high}}} \left\{ |\lambda(\widetilde{\mathcal{S}}(\alpha_1, \alpha_2, \omega, \theta))| \right\} = \frac{1}{3},$$

and is achieved if and only if $\frac{\omega}{\alpha_1} = \frac{8}{9}$ and either

$$\begin{aligned} \frac{108}{97} &\leq \omega_J \leq \frac{2}{3}\left(1 + \frac{\sqrt{2}}{2}\right), \\ \frac{2}{3\left(\frac{3}{2}\omega_J(2 - \frac{3}{2}\omega_J)\right)} &\leq \omega \leq \frac{4}{3}, \end{aligned}$$

or

$$\frac{27}{8} \left(1 - \frac{\sqrt{2}}{2}\right) \leq \omega_J \leq \frac{108}{97},$$

$$\frac{2}{3 \left(\frac{8}{27} \omega_J \left(2 - \frac{8}{27} \omega_J\right)\right)} \leq \omega \leq \frac{4}{3}.$$

Proof. The proof is similar to that of Theorem 4.3. \square

Remark 4.2. Theorems 4.3 and 4.4 tell us that two sweeps of weighted-Jacobi relaxation on the pressure equation in DWJ are required to achieve optimal performance. This is different than the case of DWJ for the MAC discretization [18], where the optimal convergence factor of $\frac{3}{5}$ is attained with one sweep of relaxation on the pressure equation.

Remark 4.3. Red-black Gauss-Seidel relaxation [10] is an attractive tool for parallel computation as it typically offers better relaxation properties while retaining parallelism. However, due to the added coupling of the finite-element operators considered here, four-colour or nine-colour relaxation would be needed to decouple the updates. Thus, we restrict ourselves to weighted Jacobi relaxation.

4.2. Braess-Sarazin relaxation

Although DWJ relaxation is efficient, we see clearly in the above that it “underperforms” in relation to weighted Jacobi relaxation for the scalar Poisson problem unless additional work is done on the pressure equation. Furthermore, proper construction of the preconditioner, \mathcal{P} , is not always possible or straightforward, especially for other types of saddle-point problems. Considering these obstacles, we also analyse other block-structured relaxation schemes. Braess-Sarazin-type algorithms were originally developed as a relaxation scheme for the Stokes equations [19], requiring the solution of a greatly simplified but global saddle-point system. The (exact) BSR approach was first introduced in [19], where it was shown that a multigrid convergence rate of $O(k^{-1})$ can be achieved, where k denotes the number of smoothing steps on each level. As a relaxation scheme for the system in (2), one solves a system of the form

$$M_E \delta x = \begin{pmatrix} \alpha D & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \delta \mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_{\mathcal{U}} \\ r_p \end{pmatrix}, \quad (25)$$

where D is an approximation to A , the inverse of which is easy to apply, for example I , or $\text{diag}(A)$. Solutions of (25) are computed in two stages as

$$S \delta p = \frac{1}{\alpha} B D^{-1} r_{\mathcal{U}} - r_p, \quad (26)$$

$$\delta \mathcal{U} = \frac{1}{\alpha} D^{-1} (r_{\mathcal{U}} - B^T \delta p),$$

where $S = \frac{1}{\alpha} B D^{-1} B^T + C$, and $\alpha > 0$ is a chosen weight for D to obtain a better approximation to A . We consider an additional weight, ω , for the global update, δx , to improve the effectiveness of the correction to both the velocity and pressure unknowns.

There is a significant difficulty in practical use of exact BSR because it requires an exact inversion of the approximate Schur complement, S , which is typically very expensive. A broader

class of iterative methods for the Stokes problem is discussed in [21], which demonstrated that the same $O(k^{-1})$ performance can be achieved as with exact BSR when the pressure correction equation is not solved exactly. In practice, an approximate solve is sufficient for the Schur complement system, such as with a few sweeps of weighted Jacobi relaxation or a few multigrid cycles. In what follows, we take $D = \text{diag}(A)$ and analyze exact BSR; to see what convergence factor can be achieved. In numerical experiments, we then consider whether it is possible to achieve the same convergence factor using an inexact solver. Note that some studies [3, 8, 45] have shown the efficiency of inexact Braess-Sarazin relaxation. The symbol of M_E is given by

$$\tilde{M}_E(\theta_1, \theta_2) = \begin{pmatrix} \frac{8}{3}\alpha & 0 & b_1 \\ 0 & \frac{8}{3}\alpha & b_2 \\ -b_1 & -b_2 & -c \end{pmatrix}.$$

The symbol of the error-propagation matrix for weighted exact BSR is $\tilde{S}_E(\alpha, \omega, \theta) = I - \omega \tilde{M}_E^{-1} \tilde{\mathcal{L}}$. A standard calculation shows that the determinant of $\tilde{\mathcal{L}} - \lambda \tilde{M}_E$ is

$$\pi_E(\lambda; \alpha) = (1 - \lambda)(a - \frac{8}{3}\alpha\lambda) \left[(1 - \lambda)(b_1^2 + b_2^2) + (\frac{8}{3}\alpha\lambda - a)c \right]. \quad (27)$$

We first establish a lower bound on the LFA smoothing factor for the stabilized method with BSR.

Theorem 4.5. *The optimal LFA smoothing factor for the Poisson-stabilized and projection stabilized discretizations with exact BSR is not less than $\frac{1}{3}$.*

Proof. From (27), two eigenvalues of $\tilde{M}_E^{-1} \tilde{\mathcal{L}}$ are given by

$$\lambda_1 = 1, \quad \lambda_2 = \frac{3a}{8\alpha},$$

which are independent of the stabilization term, c . From Lemma 4.1, we know that for λ_2 , the optimal smoothing factor is $\frac{1}{3}$, under the condition that $\frac{\omega}{\alpha} = \frac{8}{9}$. Note that if $|1 - \omega\lambda_1| \leq \frac{1}{3}$, then $\frac{2}{3} \leq \omega \leq \frac{4}{3}$. Because there is another eigenvalue, λ_3 , the optimal LFA smoothing factor is not less than $\frac{1}{3}$. \square

Similarly to DWJ, we see that the Jacobi relaxation for the Laplacian discretization places a limit on the overall performance of BSR. From (27), the third eigenvalue of $\tilde{M}_E^{-1} \tilde{\mathcal{L}}$ is $\lambda_3 = \frac{ac+b}{\frac{8}{3}ac+b}$, where $b = -(b_1^2 + b_2^2) \geq 0$ (because both b_1 and b_2 are imaginary). Thus, we only need to check whether we can choose α and ω so that $|1 - \omega\lambda_3| \leq \frac{1}{3}$ over all high frequencies, while also ensuring $|1 - \omega\lambda_1| \leq \frac{1}{3}$ and $|1 - \omega\lambda_2| \leq \frac{1}{3}$.

Theorem 4.6. *The optimal smoothing factor for both the Poisson-stabilized and projection stabilized discretizations with exact BSR is*

$$\mu_{\text{opt}} = \min_{(\alpha, \omega)} \max_{\theta \in T^{\text{high}}} |\lambda(\tilde{S}(\alpha, \omega, \theta))| = \frac{1}{3},$$

if and only if

$$\frac{\omega}{\alpha} = \frac{8}{9}, \quad \frac{3}{4} \leq \alpha \leq \frac{3}{2}.$$

Proof. Note that $a \in [2, 4]$, and choose α such that $2 = a_{\min} \leq \frac{8}{3}\alpha \leq a_{\max} = 4$. If c is positive, the following always holds

$$\frac{3}{4\alpha} = \frac{a_{\min}}{\frac{8}{3}\alpha} \leq \frac{a_{\min}c + b}{\frac{8}{3}\alpha c + b} \leq \frac{ac + b}{\frac{8}{3}\alpha c + b} \leq \frac{a_{\max}c + b}{\frac{8}{3}\alpha c + b} \leq \frac{a_{\max}}{\frac{8}{3}\alpha} = \frac{3}{2\alpha}.$$

Furthermore, if $\frac{\omega}{\alpha} = \frac{8}{9}$, we have

$$\frac{2}{3} = \frac{3}{4\alpha} \cdot \frac{8}{9}\alpha \leq \omega\lambda_3 \leq \frac{3}{2\alpha} \cdot \frac{8}{9}\alpha = \frac{4}{3}. \quad (28)$$

For both discretizations, we can check that $c > 0$ over the high frequencies. From (28), it is easy to see that $|1 - \omega\lambda_3| \leq \frac{1}{3}$, with $\alpha = \frac{9}{8}\omega \in [\frac{3}{4}, \frac{3}{2}]$. \square

4.3. Inexact Braess-Sarazin relaxation

Here, we also consider solving the Schur complement equation, (26), by weighted Jacobi relaxation with weight, ω_J . Following [21], we refer to this as inexact Braess-Sarazin relaxation (IBSR). Let the corresponding block preconditioner be M_I , given by

$$M_I = \begin{pmatrix} \alpha D & B^T \\ B & \hat{S} + B(\alpha D)^{-1}B^T \end{pmatrix}$$

where \hat{S} is the approximation of $-S = -B(\alpha D)^{-1}B^T - C$ used in (26). For one sweep of weighted Jacobi relaxation, \hat{S}_1 is given by

$$\hat{S}_1 = -\frac{1}{\omega_J} \text{diag}(S),$$

and for 2 sweeps of weighted Jacobi relaxation with equal weights, \hat{S} is given by

$$\hat{S}_2 = \hat{S}_1(2I + \hat{S}_1^{-1}S)^{-1}.$$

By direct computation, $B(\alpha D)^{-1}B^T := S_0$ can be written in terms of a 5×5 stencil:

$$S_0 = \frac{h^2}{\alpha} \begin{pmatrix} -1/192 & -1/48 & -1/24 & -1/48 & -1/192 \\ -1/48 & 0 & 1/24 & 0 & -1/48 \\ -1/24 & 1/24 & 3/16 & 1/24 & -1/24 \\ -1/48 & 0 & 1/24 & 0 & -1/48 \\ -1/192 & -1/48 & -1/24 & -1/48 & -1/192 \end{pmatrix}. \quad (29)$$

The symbol of S_0 is $\widetilde{S}_0 = \frac{3b}{8\alpha} := \varsigma$ for $b = -b_1^2 - b_2^2$. In fact, $\varsigma = \widetilde{B}(\widetilde{\alpha D})^{-1}\widetilde{B}^T$. Let γ be the symbol of \hat{S}_1 ,

$$\gamma = \begin{cases} -\frac{h^2}{24\omega_J} \left(\frac{9}{2\alpha} + \frac{8}{3} \right), & \text{for PoSD} \\ -\frac{h^2}{24\omega_J} \left(\frac{9}{2\alpha} + \frac{14}{3} \right), & \text{for PrSD} \end{cases}$$

Similarly, let η be the symbol of $\hat{S} + B(\alpha D)^{-1}B^T$,

$$\eta = \begin{cases} \gamma + \varsigma, & \text{for one sweep } (\hat{S}_1) \\ (2 + \tau\gamma^{-1})^{-1}\gamma + \varsigma, & \text{for two sweeps } (\hat{S}_2) \end{cases}$$

where

$$\tau = \begin{cases} \zeta + c_1, & \text{for PoSD} \\ \zeta + c_2, & \text{for PrSD} \end{cases}$$

Finally, the symbol of M_I is given by

$$\tilde{M}_I(\theta_1, \theta_2) = \begin{pmatrix} \frac{8}{3}\alpha & 0 & b_1 \\ 0 & \frac{8}{3}\alpha & b_2 \\ -b_1 & -b_2 & \eta \end{pmatrix}. \quad (30)$$

The symbol of the error-propagation matrix for IBSR is $\tilde{S}_I(\alpha, \omega, \theta) = I - \omega \tilde{M}_I^{-1} \tilde{\mathcal{L}}$. A standard calculation shows that the determinant of $\tilde{\mathcal{L}} - \lambda \tilde{M}_I$ is

$$\pi_I(\lambda; \alpha, \omega, \omega_J) = -(a - \frac{8}{3}\alpha\lambda) \left[(b - \frac{8\alpha\eta}{3})\lambda^2 + (a\eta - \frac{8\alpha c}{3} - 2b)\lambda + ac + b \right]. \quad (31)$$

From (31), we see there is an eigenvalue $\frac{3a}{8\alpha}$, which is the same as that of exact BSR. As before, the question now becomes whether there is a choice of ω, α and ω_J such that convergence equal to that of exact BSR can be achieved. We leave this as an open question for future work and, instead, numerically optimize the two-grid convergence factor over these parameters.

Remark 4.4. *A similar form to (30) occurs for inexact BSR applied to the stable $Q_2 - Q_1$ approximation, modifying the stencil of C to be zero, and accounting for the block structure shown in (10).*

4.4. Numerical experiments for stabilized discretizations

We now present LFA predictions, validating DWJ, (I)BSR, and the related Uzawa iteration against measured multigrid performance for these schemes. We consider the homogeneous problem in (1), with periodic boundary conditions, and a random initial guess, $x_h^{(0)}$.

Convergence is measured using the averaged convergence factor, $\hat{\rho}_h = \sqrt[k]{\frac{\|d_h^{(k)}\|_2}{\|d_h^{(0)}\|_2}}$, with $k = 100$, and $d_h^{(k)} = b - Kx_h^{(k)}$. The LFA predictions are made with $h = 1/128$, for both the smoothing factor, μ , and two-grid convergence factor, ρ_h . For testing, we use standard $W(\nu_1, \nu_2)$ cycles with bilinear interpolation for Q_1 variables and biquadratic interpolation for Q_2 variables, and their adjoints for restriction. We consider both rediscrretization and Galerkin coarsening, noting that they coincide for all terms except the stabilization terms that include a scaling of h^2 . The coarsest grid is a mesh with 4 elements. Where significant differences arise, we also report two-grid convergence rates for $TG(\nu_1, \nu_2)$ cycles.

4.4.1. PoSD with DWJ

From the range of parameters allowed in (14), we select $\alpha_1 = 1.451$, $\alpha_2 = 1.000$, and $\omega = 1.290$ (for convenience, satisfying the equality in (14)) to compute the LFA predictions. Figure 2 shows the spectrum of the two-grid error-propagation operators for DWJ relaxation with rediscrretization and Galerkin coarsening. Note that the two-grid convergence factor is the same as the optimal smoothing factor for rediscrretization, but not for Galerkin coarsening.

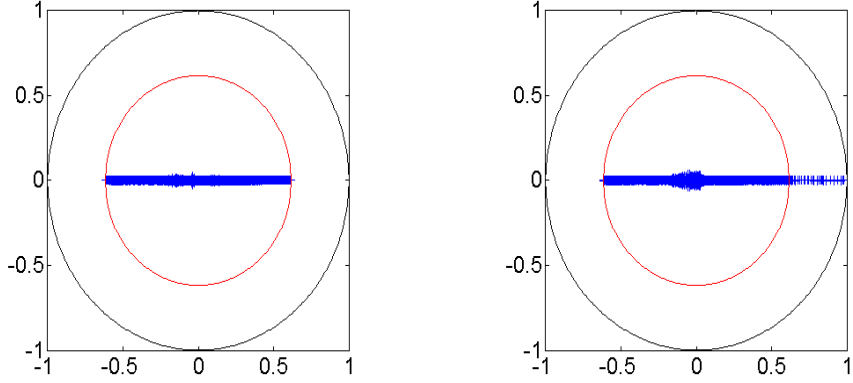


Figure 2: The spectrum of the two-grid error-propagation operator using DWJ for PoSD. Results with rediscretization are shown at left, while those with Galerkin coarsening are at right. In both figures, the inner circle has radius equal to the LFA smoothing factor.

In order to see the sensitivity of performance to parameter choice, we consider the two-grid LFA convergence factor with rediscretization coarsening. From (14), we know that there are many optimal parameters. To fix a single parameter for DWJ, we consider the case of $\omega = \frac{459}{356}$ and, at the left of Figure 3, we present the LFA-predicted two-grid convergence factors for DWJ with variation in α_1 and α_2 . Here, we see strong sensitivity to “too small” values of both parameters, for $\alpha_1 < 1$ and $\alpha_2 < 0.9$, including a notable portion of the optimal range of values predicted by the LFA smoothing factor. At the right of Figure 3, we fix $\alpha_2 = \frac{356}{459}\omega$ and vary ω and α_1 . The two lines are the lower and upper bounds from (14), between which LFA predicts the optimal convergence factor should be achieved. Note that not all of the allowed parameters obtain the optimal convergence factor. Here, we see great sensitivity for large values of ω , but a large range with generally similar performance as in the optimal parameter case.

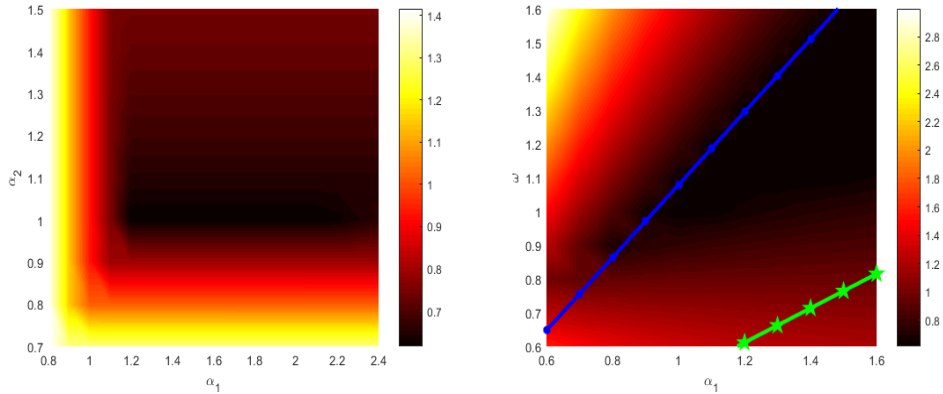


Figure 3: The two-grid LFA convergence factor for the PoSD using DWJ and rediscretization. At left, we fix $\omega = \frac{459}{356}$ and vary α_1 and α_2 . At right, we fix $\alpha_2 = \frac{356}{459}\omega$ and vary ω and α_1 .

In Table 1, we present the multigrid performance of DWJ with W -cycles for rediscrretization coarsening. These results show measured multigrid convergence factors that coincide with the LFA-predicted two-grid convergence factors. Similar results are seen for V -cycles with rediscrretization. For Galerkin coarsening, nearly identical W -cycle results are seen when $\nu_1 + \nu_2 > 2$, but divergence is seen for W -cycles with $\nu_1 + \nu_2 = 1$ or 2, and for all V -cycles tested. In Table 2, we report the multigrid performance of DWJ using 2 sweeps of Jacobi relaxation on the pressure equation with rediscrretization for PoSD. Here, we take $\alpha_1 = 3/2, \omega_J = 1, \omega = 4/3$ as in Theorem 4.3. We see that the LFA convergence factor accurately predicts the measured performance.

Table 1: W -cycle convergence factors, $\hat{\rho}_h$, for DWJ with rediscrretization for PoSD, compared with LFA two-grid predictions, ρ_h . Here, the algorithmic parameters are $\alpha_1 = 1.451, \alpha_2 = 1.000, \omega = 1.290$ and the LFA smoothing factor is $\mu = 0.618$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.618	0.618	0.382	0.236	0.236	0.146
$\hat{\rho}_{h=1/64}$	0.564	0.568	0.349	0.215	0.214	0.133
$\hat{\rho}_{h=1/128}$	0.561	0.568	0.348	0.215	0.214	0.132

Table 2: W -cycle convergence factors, $\hat{\rho}_h$, for DWJ with 2 sweeps of Jacobi on the pressure equation for PoSD with rediscrretization, compared with LFA two-grid predictions, ρ_h . Here, the algorithmic parameters are $\alpha_1 = 3/2, \omega_J = 1, \omega = 4/3$ and the LFA smoothing factor is $\mu = 0.333$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.338	0.338	0.115	0.078	0.078	0.061
$\hat{\rho}_{h=1/64}$	0.324	0.324	0.112	0.074	0.075	0.074
$\hat{\rho}_{h=1/128}$	0.324	0.324	0.112	0.075	0.075	0.073

4.4.2. PrSD with DWJ

From the range of parameters allowed in (15), we choose $\alpha_1 = 1, \alpha_2 = 1, \omega = \frac{108}{97}$. Figure 4 shows that the smoothing factor provides a good prediction for the two-grid convergence factor with rediscrretization, but not with Galerkin coarsening.

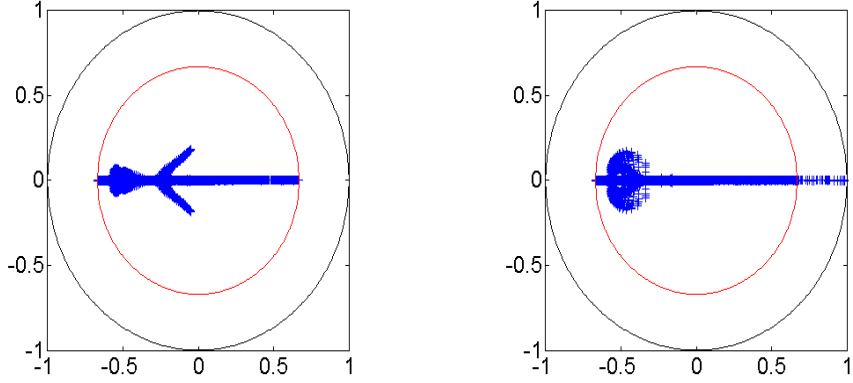


Figure 4: The spectrum of the two-grid error-propagation operator using DWJ for PrSD. Results with rediscritization are shown at left, while those with Galerkin coarsening are at right. In both figures, the inner circle has radius equal to the LFA smoothing factor.

Similarly to the discussion above, we consider the sensitivity to parameter choice for DWJ applied to PrSD. To fix a single parameter for DWJ, we consider the case of $\omega = \frac{108}{97}$. At the left of Figure 5, we present the LFA-predicted convergence factors for DWJ with variation in α_1 and α_2 , again seeing a strong sensitivity to “too small” values of the parameters. At the right of Figure 5, we fix $\alpha_2 = \frac{97}{108}\omega$. The two lines are the lower and upper bounds from (15), between which LFA predicts the optimal convergence factor should be achieved. Note that not all of the parameters in this range obtain the optimal convergence factor. We see that, for small α_1 , the convergence factor is very sensitive to large values of ω .

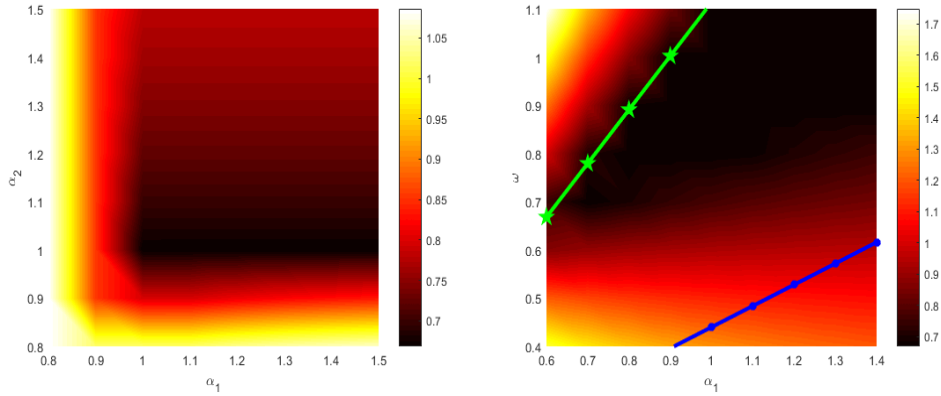


Figure 5: The two-grid LFA convergence factor for the PrSD using DWJ and rediscritization. At left, we fix $\omega = \frac{108}{97}$ and vary α_1 and α_2 . At right, we fix $\alpha_2 = \frac{97}{108}\omega$ and vary ω and α_1 .

In Table 3, we present the multigrid performance of DWJ relaxation with W -cycles for rediscritization coarsening. We see that the measured multigrid convergence factors match well with the LFA-predicted two-grid convergence factors. For Galerkin coarsening, as in the case of

PoSD, we see divergence when $\nu_1 + \nu_2 \leq 2$, but performance matching that of rediscrretization for $\nu_1 + \nu_2 > 2$. Here, V -cycle results are similar to the W -cycle results for both rediscrretization and Galerkin coarsening approaches. In Table 4, we compare the LFA predictions with multigrid performance for DWJ using 2 sweeps of Jacobi relaxation on the pressure equation. Here, we take $\alpha_1 = 3/2$, $\omega_J = 1$, $\omega = 4/3$ as in Theorem 4.4, and observe a good match between the LFA predictions and measured performance.

Table 3: W -cycle convergence factors, $\hat{\rho}_h$, for DWJ with rediscrretization for PrSD, compared with LFA two-grid predictions, ρ_h . Here, the algorithmic parameters are $\alpha_1 = 1$, $\alpha_2 = 1$, $\omega = 108/97$ and the LFA smoothing factor is $\mu = 0.670$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.670	0.670	0.449	0.300	0.300	0.201
$\hat{\rho}_{h=1/64}$	0.652	0.652	0.436	0.291	0.292	0.196
$\hat{\rho}_{h=1/128}$	0.651	0.652	0.435	0.291	0.291	0.195

Table 4: W -cycle convergence factors, $\hat{\rho}_h$, for DWJ with 2 sweeps of Jacobi on the pressure equation for PrSD with rediscrretization, compared with LFA two-grid predictions, ρ_h . Here, the algorithmic parameters are $\alpha_1 = 3/2$, $\omega_J = 1$, $\omega = 4/3$ and the LFA smoothing factor is $\mu = 0.333$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.333	0.333	0.112	0.079	0.079	0.062
$\hat{\rho}_{h=1/64}$	0.324	0.324	0.112	0.074	0.075	0.074
$\hat{\rho}_{h=1/128}$	0.324	0.324	0.112	0.075	0.075	0.073

4.4.3. PoSD with BSR

Next, we consider BSR for PoSD, first displaying the two-grid LFA convergence factor as a function of α for rediscrretization coarsening with $\omega = \frac{8}{9}\alpha$ in Figure 6. Comparing the convergence factor with μ^2 , for $\nu_1 = \nu_2 = 1$, we see a good match over the interior of the interval $\frac{3}{4} \leq \alpha \leq \frac{3}{2}$ predicted by Theorem 4.6. For larger values of $\nu_1 + \nu_2$, this agreement deteriorates as is typical when the behavior of coarse-grid correction becomes dominant. At the right of Figure 6, we see good agreement between ρ and μ when $\nu_1 + \nu_2 = 1$ with fixed $\alpha = 1$. In both cases, similar behaviour is seen with Galerkin coarsening.

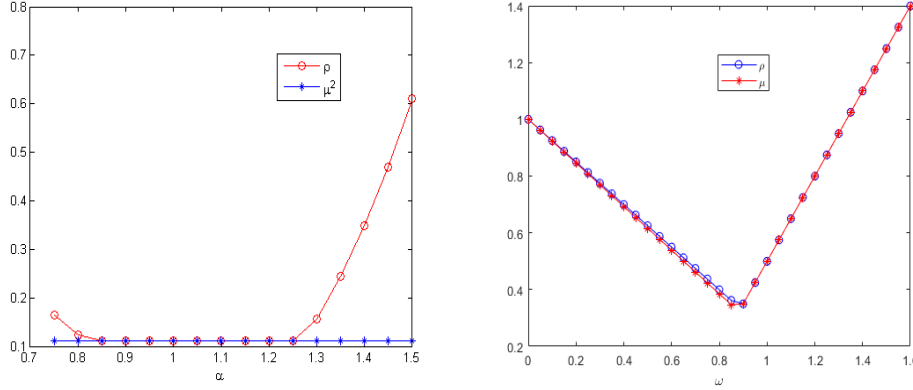


Figure 6: Two-grid and smoothing factors for BSR with rediscritization for PoSD. At left, comparing ρ with μ^2 for $\nu_1 = \nu_2 = 1$ with $\omega = \frac{8}{9}\alpha$. At right, comparing ρ with μ for $\nu_1 + \nu_2 = 1$ with $\alpha = 1$.

Motivated by the above, we use $\alpha = 1$ and $\omega = \frac{8}{9}$ for multigrid experiments with rediscritization, solving the Schur complement equation exactly. Table 5 shows that the measured multigrid convergence factors match well with the LFA-predicted two-grid convergence factors for W -cycles with rediscritization coarsening, and similar results are seen for Galerkin coarsening.

Table 5: W -cycle convergence factors, $\hat{\rho}_h$, for BSR with rediscritization for PoSD, compared with LFA two-grid predictions, ρ_h . Here, the algorithmic parameters are $\alpha = 1$, $\omega = \frac{8}{9}$ and the LFA smoothing factor is $\mu = 0.333$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.333	0.333	0.111	0.079	0.079	0.062
$\hat{\rho}_{h=1/64}$	0.324	0.323	0.112	0.075	0.075	0.058
$\hat{\rho}_{h=1/128}$	0.323	0.323	0.112	0.075	0.075	0.058

In Table 6, we report the LFA prediction for IBSR with different parameters and one or two sweeps of Jacobi relaxation on the approximate Schur complement for PoSD. Here, we clearly see the benefit of two sweeps of relaxation on the approximate Schur complement over a single sweep, as well as that better performance is possible when (numerically) optimizing all of the parameters for IBSR independent of the optimization for exact BSR.

Table 6: LFA predictions: two-grid convergence factor, $\rho_{h=1/128}$, and smoothing factor, μ , of IBSR with rediscritization for PoSD with $\nu_1 + \nu_2 = 1$.

Sweep	α	ω	ω_J	μ	$\rho_{h=1/128}$
1 (Optimized)	1.2	1.1	0.7	0.679	0.679
1	1.0	8/9	1.0	0.669	0.735
2 (Optimized)	1.1	1.0	1.0	0.366	0.366
2	1.0	8/9	1.0	0.461	0.461

Table 7 shows that the LFA-predicted 2-grid convergence factors closely match those seen

in practice. However, as shown in Table 8, significant degradation is seen when considering W -cycles, particularly as $\nu_1 + \nu_2$ increases.

Table 7: Two-grid convergence factor, $\hat{\rho}_h$ for IBSR with 2 sweeps of Jacobi with rediscrctization for PoSD, compared with LFA two-grid predictions, ρ_h , with optimized parameters, $\alpha = 1.1, \omega = 1.0$ and $\omega_J = 1$.

Cycle	$TG(0, 1)$	$TG(1, 0)$	$TG(1, 1)$	$TG(1, 2)$	$TG(2, 1)$	$TG(2, 2)$
$\rho_{h=1/128}$	0.366	0.366	0.167	0.128	0.128	0.106
$\hat{\rho}_{h=1/64}$	0.352	0.353	0.160	0.120	0.120	0.100
$\hat{\rho}_{h=1/128}$	0.352	0.353	0.160	0.122	0.122	0.100

Table 8: W -cycles convergence factors, $\hat{\rho}_h$, for IBSR with 2 sweeps of Jacobi with rediscrctization for PoSD, compared with LFA two-grid predictions, ρ_h with optimized parameters, $\alpha = 1.1, \omega = 1.0$ and $\omega_J = 1$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.366	0.366	0.167	0.128	0.128	0.106
$\hat{\rho}_{h=1/64}$	0.456	0.453	0.245	0.197	0.200	0.167
$\hat{\rho}_{h=1/128}$	0.459	0.462	0.257	0.206	0.211	0.175

The gap between the results seen for exact BSR in Table 5 and those for IBSR in Table 8 is quite significant. To maintain the performance observed for exact BSR, we could simply use more Jacobi iterations on the Schur complement system in IBSR; however, experiments showed that this did not lead to a scalable algorithm. Instead, we consider solving the Schur complement system by applying a multigrid $W(1, 1)$ -cycle using weighted Jacobi relaxation with weight ω_J , shown in Table 9. From Table 9, we observe that using only 1 or 2 $W(1, 1)$ -cycles on the approximate Schur complement achieves convergence factors essentially matching those in Table 5, showing that the $W(1, 1)$ cycle is the most cost effective.

Table 9: W -cycle convergence factors, $\hat{\rho}_h$, for IBSR with inner $W(1, 1)$ -cycle for PoSD and $(\alpha, \omega, \omega_J) = (1, 8/9, 1)$. In brackets, minimum value of the number of inner $W(1, 1)$ -cycles that achieves the same convergence factors as those of LFA predictions, ρ_h , for exact BSR.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.333	0.333	0.111	0.079	0.079	0.062
$\hat{\rho}_{h=1/64}$	0.368(2)	0.346(2)	0.131(2)	0.075(2)	0.075(2)	0.059(1)
$\hat{\rho}_{h=1/128}$	0.343(2)	0.351(2)	0.111(2)	0.075(2)	0.075(2)	0.063(1)

4.4.4. PrSD with BSR

We now consider BSR for PrSD. At the left of Figure 7, we see a good agreement between the two-grid convergence factor and μ^2 for $\nu_1 = \nu_2 = 1$ for some parameters in the range defined in Theorem 4.6 when using rediscrctization. A larger interval of agreement is seen for the corresponding results for Galerkin coarsening. In both cases, agreement between the two-grid convergence factor and $\mu^{\nu_1 + \nu_2}$ degrades as $\nu_1 + \nu_2$ increases, as expected.

Note that Theorem 4.6 demonstrates that the smoothing factor for BSR is a function of $\frac{\omega}{\alpha}$ (but the same is not necessarily true for the convergence factor). In Figure 7, we plot the LFA smoothing and convergence factors for BSR with rediscrctization as a function of ω , with $\alpha = 0.8$ and see that these factors generally agree, although the smoothing factor slightly underestimates

the convergence factor. As two-grid convergence is, however, sensitive to the choice of α , the smoothing factor generally underestimates the convergence factor for other values of α .

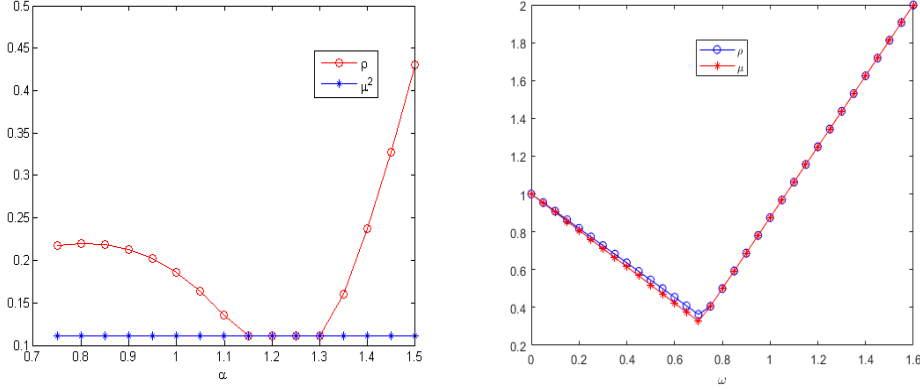


Figure 7: Two-grid and smoothing factors for BSR with rediscrctization for PrSD. At left, comparing ρ with μ^2 for $\nu_1 = \nu_2 = 1$ with $\omega = \frac{8}{9}\alpha$. At right, comparing ρ with μ for $\nu_1 + \nu_2 = 1$ with $\alpha = \frac{4}{5}$.

Fixing $\omega = \frac{8}{9}\alpha$ with $\alpha = 1.2$ (as suggested by Figure 7 for $\nu_1 = \nu_2 = 1$), Table 10 shows that the measured multigrid convergence factors again match well with the LFA-predicted two-grid convergence factors for W -cycles with rediscrctization coarsening. Note, however, the degradation for $\nu_1 + \nu_2 = 1$, where the smoothing factor analysis predicts a convergence factor of $\frac{1}{3}$ that is not realized. The convergence factor of $\frac{1}{3}$ can be achieved by choosing $\alpha = \frac{4}{5}$ and $\omega = \frac{8}{9}\alpha$ in the BSR scheme with either $W(1, 0)$ or $W(0, 1)$ cycles, but these choices lead to a slight degradation with $\nu_1 + \nu_2 > 1$. Similar results are seen for Galerkin coarsening with $\alpha = 1$ and $\omega = \frac{8}{9}\alpha$ with the notable exception that the smoothing factor prediction was matched by both the two-grid LFA convergence factor and true W -cycle convergence in this case for all experiments.

Table 10: W -cycle convergence factors, $\hat{\rho}_h$, for BSR with rediscrctization for PrSD, compared with LFA two-grid predictions, ρ_h , with algorithmic parameters, $\alpha = 1.2$ and $\omega = \frac{8}{9}\alpha$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.673	0.673	0.111	0.079	0.079	0.062
$\hat{\rho}_{h=1/64}$	0.585	0.585	0.112	0.075	0.075	0.058
$\hat{\rho}_{h=1/128}$	0.584	0.584	0.112	0.075	0.075	0.058

In Table 11, we report the LFA prediction for IBSR with one or two sweeps of Jacobi relaxation on the approximate Schur complement with different parameters for PrSD. As in the case of PoSD, one sweep is not enough to obtain performance comparable to exact BSR, and there is a significant advantage to independently optimizing the parameters for IBSR.

Table 11: LFA predictions: two-grid convergence factor, $\rho_{h=1/128}$, and smoothing factor, μ , of IBSR with rediscrretization for PrSD with $\nu_1 + \nu_2 = 1$.

Sweep	α	ω	ω_J	μ	$\rho_{h=1/128}$
1 (Optimized)	1.6	0.8	1	0.714	0.714
1	1.2	16/15	1.0	0.718	1.027
2 (Optimized)	1.2	0.9	1.2	0.494	0.445
2	1.2	16/15	1.0	0.431	0.549

Considering, then, the two-grid method with these optimized parameters and two relaxation sweeps on the approximate Schur complement, Table 12 shows that two-grid LFA offers a good prediction of performance. In Table 13, however, we see degraded performance when using W -cycles.

Table 12: Two-grid convergence factors, $\hat{\rho}_h$ for IBSR with 2 sweeps of Jacobi for PrSD with rediscrretization, compared with LFA two-grid predictions, ρ_h , with optimized parameters, $\alpha = 1.2$, $\omega = 0.9$ and $\omega_J = 1.2$.

Cycle	$TG(0, 1)$	$TG(1, 0)$	$TG(1, 1)$	$TG(1, 2)$	$TG(2, 1)$	$TG(2, 2)$
$\rho_{h=1/128}$	0.445	0.445	0.319	0.262	0.262	0.225
$\hat{\rho}_{h=1/64}$	0.418	0.420	0.301	0.251	0.250	0.212
$\hat{\rho}_{h=1/128}$	0.420	0.420	0.304	0.250	0.249	0.212

Table 13: W -cycles convergence factors, $\hat{\rho}_h$ for IBSR with 2 sweeps of Jacobi for PrSD with rediscrretization, compared with LFA two-grid predictions, ρ_h , with optimized parameters, $\alpha = 1.2$, $\omega = 0.9$ and $\omega_J = 1.2$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.445	0.445	0.319	0.262	0.262	0.225
$\hat{\rho}_{h=1/64}$	0.739	0.740	0.340	0.304	0.299	0.268
$\hat{\rho}_{h=1/128}$	0.736	0.735	0.342	0.309	0.311	0.276

Thus, we again consider solving the Schur complement system by applying a multigrid $W(1, 1)$ -cycle. Table 14 shows that this IBSR is seen to be effective, requiring 1 to 4 $W(1, 1)$ cycles on the Schur complement system to match the convergence seen in Table 10. Again, $W(1, 1)$ cycles seem to be the most cost effective option for the approximate Schur complement.

Table 14: W -cycle convergence factors, $\hat{\rho}_h$, for IBSR with inner $W(1, 1)$ -cycle for PrSD and $(\alpha, \omega, \omega_J) = (6/5, 16/15, 1.1)$. In brackets, minimum value of the number of inner $W(1, 1)$ -cycles that achieves the same convergence factors as those of LFA predictions, ρ_h , for exact BSR.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	0.673	0.673	0.111	0.079	0.079	0.062
$\hat{\rho}_{h=1/64}$	0.680(4)	0.677(1)	0.112(3)	0.075(2)	0.075(2)	0.059(1)
$\hat{\rho}_{h=1/128}$	0.659(1)	0.662(1)	0.112(3)	0.075(2)	0.075(2)	0.067(1)

4.5. Stabilized discretizations with Uzawa relaxation

Multigrid methods using Uzawa relaxation schemes [6, 26, 27] are popular approaches due to their low cost per iteration. We consider Uzawa relaxation as a simplification of BSR, deter-

mining the update as the (weighted) solution of

$$M\delta x = \begin{pmatrix} \alpha D & 0 \\ B & -\hat{S} \end{pmatrix} \begin{pmatrix} \delta \mathcal{U} \\ \delta p \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \end{pmatrix},$$

where αD is an approximation to A and $-\hat{S}$ is an approximation of the Schur complement, $-BA^{-1}B^T - C$.

Here, we consider an analogue to exact BSR with $D = \text{diag}(A)$. The choice of \hat{S} is discussed later. In this setting, we observe that minimizing the LFA smoothing factor does not minimize the LFA convergence factor. Thus, we consider minimizing the two-grid convergence factor numerically for $\nu_1 + \nu_2 = 1$ and $\nu_1 = \nu_2 = 1$ with rediscrretization coarsening, and compare with measured multigrid performance.

We consider three approximations to the Schur complement, starting from the true approximate Schur complement, $C + B(\alpha \text{diag}(A))^{-1}B^T$. Motivated by the stable finite-element case, we also consider replacing $B(\alpha \text{diag}(A))^{-1}B^T$ in this matrix by a weighted mass matrix, yielding $\hat{S} = C + \delta Q$. Finally, motivated by the finite-difference case and efficiency of implementation, we consider taking $\hat{S} = \sigma h^2 I$, for a scalar weight, σ , to be optimized by the LFA. Note that, due to the constant-coefficient stencils assumed by LFA, this corresponds to using a single sweep of Jacobi to approximate solution of either of the two above approximations.

For the case of $C + B(\alpha \text{diag}(A))^{-1}B^T$, the optimized LFA two-grid convergence factors for $\nu_1 + \nu_2 = 1$ with rediscrretization coarsening are 0.428 for PoSD and 0.436 for PrSD. These are notably worse than the BSR smoothing factor of $\frac{1}{3}$, which is achieved for $W(1, 0)$ or $W(0, 1)$ cycles. Here, $W(1, 0)$ cycles reflect this convergence, achieving measured convergence factors of 0.417 for PoSD and 0.526 for PrSD. Increasing the number of relaxation sweeps per iteration yields some improvement in the predicted LFA convergence factors when optimizing parameters again, but not enough to outperform repeated $W(1, 0)$ cycles.

For the mass-matrix-based approximation, $\hat{S} = C + \delta Q$, the optimized two-grid convergence factors for $\nu_1 + \nu_2 = 1$ with rediscrretization coarsening are 0.5 for PoSD and 0.417 for PrSD. While poorer convergence might be expected in both cases, the addition of an extra parameter, δ , allows a (slight) improvement for PrSD. In both cases, we observe consistent performance with numerical experiments, achieving convergence factors of 0.493 for PoSD and 0.392 for PrSD using $W(0, 1)$ or $W(1, 0)$ cycles.

Finally, for the diagonal approximation $\hat{S} = \sigma h^2 I$, we achieve notably better performance optimizing with $\nu_1 = \nu_2 = 1$ than for $\nu_1 + \nu_2 = 1$. For PoSD, the optimized two-grid LFA convergence factor is 0.382, while it is 0.497 for PrSD. In practice, we achieve slightly worse convergence factors using $W(1, 1)$ cycles with rediscrretization coarsening, of 0.531 for PoSD and 0.543 for PrSD. These are both significantly worse than the convergence factors of $\frac{1}{9}$ observed using inexact BSR; however, it must be noted that W -cycles on the Schur complement system were needed in that case. A better approximation to inverting the true approximate Schur complement would be to apply multigrid to it, just as was done for IBSR above. Here, we observe that significant work may be needed to achieve convergence similar to that of Uzawa where the Schur complement is exactly inverted, requiring 10 $W(1, 1)$ -cycles on the approximate Schur complement to achieve a convergence factor of 0.416 for PoSD and 0.522 for PrSD, suggesting that the Jacobi version of Uzawa is ultimately more efficient.

4.6. Comparing cost and performance

For convenience, we denote standard DWJ as DWJ(1) and DWJ with 2 sweeps of Jacobi relaxation on the pressure equation as DWJ(2) in the following.

The above results give a clear comparison of the effectiveness of the multigrid cycles with the considered relaxation schemes, but not of their relative efficiencies. To translate from effectiveness to efficiency, we must properly account for the cost per iteration of each relaxation scheme. All schemes assume the residual is already calculated; for the 9-point stencils in A , B , and stabilization terms, C , the cost of a single residual evaluation on a mesh with n points is (roughly) that of $63n$ multiply-add operations, coming from the 7 nonzero blocks in the matrix. For DWJ(1), the rest of the cost of relaxation is fairly easy to account, requiring one diagonal scaling operation on each of the three components of the solution vector, plus matrix-vector products with the pressure Laplacian, A_p , and with both B and B^T . Counting multiply-add operations for these on a grid with n points, we have $3n$ for the diagonal scalings, and $9n$ each for the multiplication with A_p and with B_x and B_y and their transposes, totalling $48n$ multiply-add operations. For DWJ(2), we need $48n$ multiply-add operators plus the cost of the second sweep. For the second sweep, we need to compute a residual related to G , the (3, 3) block of (11), and a diagonal scaling. Note that the cost of the residual is $54n$ ($9n$ each for the multiplication with A_p , C , and with B_x and B_y and their transposes). In total, the cost of DWJ(2) is $103n$ multiply-add operators. For IBSR, following (26), we require two diagonal scaling operations on each of the velocity components, one matrix-vector product with each of B and B^T , and 2 or 3 W(1,1) cycles on the pressure variable. To account for the costs of the W(1,1) cycles, we use the standard cost estimate for W-cycles, as requiring 4 “Work Units” per iteration, where a Work Unit is the cost of forming a residual for the pressure equation. Here, given the 25-point stencil structure seen in Equation (29), each Work Unit requires 25 multiply-add operations, so the total cost of IBSR with 2 W(1,1) cycles on the Schur complement is $4n + 36n + 200n = 240n$ multiply-add operations (and $340n$ multiply-add operations if 3 W(1,1) cycles are needed). Finally, Uzawa relaxation with diagonal scaling on the pressure has a cost less than that of DWJ(1), as it requires diagonal scaling again for all three components of the solution, but only one matrix-vector multiplication, with B . These total $21n$ multiply-add operations.

Accumulating the costs of a residual evaluation with these, we have total costs of $111n$ multiply-add operations per sweep of DWJ(1), $166n$ multiply-add operations per sweep of DWJ(2), $303n$ multiply-add operations per sweep of IBSR with 2 W(1,1) cycles per Schur-complement solve, and $84n$ multiply-add operations per sweep of Uzawa with diagonal scaling. Considering these relative to one-another, we see that DWJ(1) has a cost of about $4/3$ per cycle as Uzawa, that DWJ(2) has a per-cycle cost of about 2 times that of Uzawa, and 1.5 times that of DWJ(1), that IBSR has a per-cycle cost of about 3.6 times that of Uzawa, 2.7 and 1.8 times that of DWJ(1) and DWJ(2), respectively. The per-cycle convergence factors observed above are 0.35 per cycle for W(1,1) cycles of DWJ(1) for PoSD and 0.44 per cycle for W(1,1) cycles of DWJ(1) for PrSD, 0.11 per cycle for W(1,1) cycles of DWJ(2) and IBSR for both stabilizations, and 0.53 or 0.54 per cycle for W(1,1) cycles with Uzawa. Comparing efficiencies can now be easily done by appropriately weighting these convergence factors relative to their work: if one iteration costs W times that of another, and yields a convergence factor of ρ_1 , then we can easily compare $\rho_1^{1/W}$ directly to the second convergence factor, ρ_2 , to see if the *effective* error reduction achieved by the first algorithm in an equal amount of work to the second is better or worse than that achieved by the second. Comparing DWJ(1) to Uzawa, then, for PoSD, we compare $0.35^{3/4} \approx 0.46$ to 0.53 and see that DWJ(1) is more efficient. For PrSD, we compare $0.44^{3/4} \approx 0.54$ and see that DWJ(1) and Uzawa are similarly efficient for PrSD. Comparing DWJ(2) to Uzawa, we compare $0.11^{1/2} \approx 0.33$ to $0.53(0.54)$, showing that DWJ(2) is much efficient than Uzawa. Comparing DWJ(2) to DWJ(1), we compare $0.11^{1/1.5} \approx 0.23$ to $0.35(0.45)$ and see DWJ(2) is more efficient than DWJ(1). Comparing IBSR to Uzawa, we compare $0.11^{1/3.6} \approx 0.54$, and see that it as also

comparable in efficiency to the others for the case of PrSD, but slightly less efficient than DWJ(1) for PoSD. DWJ(2) and IBSR have the same per-cycle convergence factor, but the cost of DWJ(2) ($166n$) is less than IBSR ($303n$). Thus DWJ(2) is more efficient than IBSR. Overall, DWJ(2) outperforms Uzawa, IBSR and DWJ(1). We note that these results are a little different than those seen for the MAC discretization in [18], where IBSR outperforms other schemes. Differences seen in practice (and the influence of factors ignored in the LFA, such as boundary conditions) are important to consider.

An important practical consideration commonly observed in the LFA literature (see, for example, [18, 35]) is the influence of boundary conditions. In numerical experiments not shown here, we often see significant degradation in convergence between the results for periodic boundary conditions and those for Dirichlet boundary conditions, particularly for cases with larger numbers of relaxation sweeps per cycle. For DWJ(2) with $\nu_1 + \nu_2 = 1$, changing from periodic to Dirichlet boundary conditions results in convergence factors increasing from 0.324 reported in Tables 2 and 4 to about 0.46 (PrSD) or 0.56 (PoSD) for two-grid cycles, and to about 0.64 (PrSD) and 0.7 (PoSD) for W-cycles. For IBSR with $\nu_1 + \nu_2 = 1$, however, the degradation is much less, with W-cycle convergence rates of 0.38 for PoSD (still with 2 inner W(1,1)-cycles for the Schur complement system) and 0.35 for PrSD (with $\alpha = 4/5$, $\omega = 32/45$, and 4 W(1,1) cycles with $\omega_J = 1.1$ for the Schur complement system). Clearly this difference in performance is enough to change the balance above, with the added cost of IBSR with inner W-cycles paying off over DWJ(2).

5. Relaxation for $Q_2 - Q_1$ discretization

As explored in [15], classical LFA smoothing factor analysis is unreliable for Q_2 discretizations, making it unsuitable for analysis of the standard stable $Q_2 - Q_1$ discretization of the Stokes equations. Thus, we consider only numerical (“brute force”) optimization of two-grid LFA convergence factors in this setting.

For DWJ, we find optimal convergence factors of 0.619 for $\nu_1 + \nu_2 = 1$ and 0.558 for $\nu_1 = \nu_2 = 1$. While the former is quite comparable to convergence predicted and achieved for both stabilized discretizations with $\nu_1 + \nu_2 = 1$, we see a significant lack of improvement with increased relaxation, in contrast to the equal-order case. The same is observed for multigrid W-cycle performance, with $W(1, 0)$ convergence measured at 0.620 and $W(1, 1)$ convergence measured at 0.510.

For exact BSR, we find optimal convergence factors of 0.551 for $\nu_1 + \nu_2 = 1$ and 0.250 for $\nu_1 = \nu_2 = 1$. While these are slightly larger than the comparable factors of $\frac{1}{3}$ and $\frac{1}{9}$, respectively, for the stabilized discretizations, they still reflect good performance of the underlying method.

At left of Figure 8, we show the spectral radius of the error-propagation symbol for exact BSR as a function of Fourier frequency, θ , noting that predicted reduction over the high frequencies is not as good as would be needed to equal two-grid convergence in the equal-order case. In order to see how the convergence factor changes with the parameters α and ω , we display the convergence factor as a function of α and ω at the right of Figure 8. The optimal choice, of $\alpha = 1.1$ and $\omega = 1.05$, occurs in a narrow band of ω values, but larger range of α values lead to reasonable results.

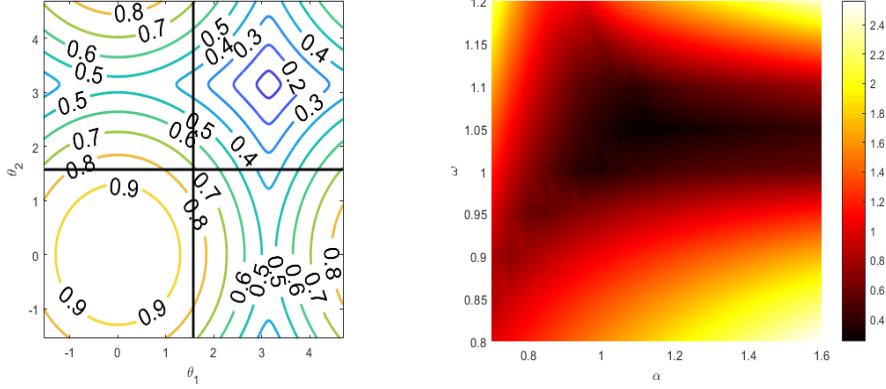


Figure 8: At left, the spectral radius of the error-propagation symbol for exact BSR applied to the $Q_2 - Q_1$ discretization, as a function of the Fourier mode, θ . At right, the LFA-predicted two-grid convergence factor for BSR applied to the $Q_2 - Q_1$ discretization as a function of α and ω , with $(\nu_1, \nu_2) = (1, 1)$.

As always, an inexact solve of the Schur complement system is needed to yield a practical variant of BSR. While 2 sweeps of Jacobi appears sufficient to achieve scalable W -cycle convergence when $\nu_1 + \nu_2 > 2$ (see Table 15), we find 3 sweeps are needed to achieve $W(1, 1)$ convergence factors of 0.240 (see Table 16), in contrast to results in [21] and for the equal-order discretizations considered here, where a much stronger iteration was needed. Similar results were seen for $V(1, 1)$ cycles when 3 sweeps of Jacobi were used for the Schur complement system.

Table 15: W -cycles convergence factors, $\hat{\rho}_h$, for IBSR with 2 sweeps Jacobi for $Q_2 - Q_1$ approximation with red discretization, compared with LFA two-grid predictions, ρ_h , for exact BSR with algorithmic parameters, $\alpha = 1.1, \omega = 1.05$ and $\omega_J = 1.0$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	4.893	4.893	0.249	0.109	0.109	0.090
$\hat{\rho}_{h=1/64}$	NAN	NAN	0.434	0.131	0.130	0.085
$\hat{\rho}_{h=1/128}$	NAN	NAN	0.437	0.130	0.130	0.085

Table 16: W -cycles convergence factors, $\hat{\rho}_h$, for IBSR with 3 sweeps Jacobi for $Q_2 - Q_1$ approximation with red discretization, compared with LFA two-grid predictions, ρ_h , for exact BSR with algorithmic parameters, $\alpha = 1.1, \omega = 1.05$ and $\omega_J = 1.0$.

Cycle	$W(0, 1)$	$W(1, 0)$	$W(1, 1)$	$W(1, 2)$	$W(2, 1)$	$W(2, 2)$
$\rho_{h=1/128}$	4.893	4.893	0.249	0.109	0.109	0.090
$\hat{\rho}_{h=1/64}$	491.373	492.094	0.240	0.104	0.104	0.085
$\hat{\rho}_{h=1/128}$	NAN	NAN	0.240	0.104	0.104	0.085

Finally, we consider the same three variants of Uzawa relaxation as examined above for the equal-order case. For $\hat{S} = B(\alpha \text{diag}(A))^{-1} B^T$, the best convergence factor found for $\nu_1 + \nu_2 = 1$ was 0.729, while better convergence was predicted for $\hat{S} = \delta Q$, with factor 0.554. This is to be expected, perhaps, since the Q_1 mass matrix is well-known to be a better approximation of

the true Schur complement than the classical BSR approximate Schur complement. However, approximating either by a single sweep of Jacobi, yielding $\hat{S} = \sigma h^2 I$, gives a convergence factor 0.717. While 2-grid cycles with $\nu_1 + \nu_2 = 1$ match the predicted convergence factor, W -cycles did not converge for these parameters.

Comparing, then, the efficiency of inexact BSR and DWJ for the $Q_2 - Q_1$ discretization, we see that inexact BSR, where $W(1, 1)$ cycles achieve a convergence factor of 0.24 provides roughly the same reduction as 3 cycles with 1 DWJ sweep per cycle, where LFA predicts $\rho = 0.619$. Noting that inexact BSR is relatively more expensive in this case, with cost dominated by the two diagonal scalings per sweep on the Q_2 velocity degrees of freedom, we suggest a proper implementation study is needed to determine which, if either, provides best performance in practice.

6. Conclusion

In this paper, LFA is presented for block-structured relaxation schemes for stabilized and stable finite-element discretizations of the Stokes equations. The convergence and smoothing factors exhibited here provide optimized parameters for DWJ with one or two sweeps of Jacobi relaxation on the pressure equation and BSR for the stabilized discretizations. The convergence of (inexact) BSR clearly outperforms multigrid with both standard DWJ and Uzawa relaxation. However, standard DWJ can be improved by additional relaxation on the pressure equation, and the improved version is more efficient than IBSR. While the LFA smoothing factor loses its predictivity of the two-grid convergence factor for the stable $Q_2 - Q_1$ discretization and for Uzawa relaxation for both stabilized and stable discretizations, the two-grid LFA convergence factor can still provide useful predictions. We consider as well the inexact case for BSR, with Jacobi iterations or multigrid cycles used to approximate solution of the Schur complement system, as is suitable for use on modern parallel and graphics processing unit (GPU) architectures. From numerical experiments, we see that inexact BSR can be as good as the exact iteration for solving the Stokes equations. The analysis and LFA predictions demonstrated here offer good insight into the use of block-structured relaxation for other types of saddle-point problems, which will be considered in future work.

Acknowledgements

The work of S.M. was partially supported by an NSERC discovery grant.

References

- [1] H. C. Elman, D. J. Silvester, A. J. Wathen, Finite elements and fast iterative solvers with applications in incompressible fluid dynamics, 2nd Edition, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2014.
- [2] J. W. Ruge, K. Stüben, Algebraic multigrid, Multigrid methods 3 (13) (1987) 73–130.
- [3] J. Adler, T. R. Benson, E. Cyr, S. P. MacLachlan, R. S. Tuminaro, Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics, SIAM J. Sci. Comput. 38 (1) (2016) B1–B24.
- [4] J. H. Adler, D. B. Emerson, S. P. MacLachlan, T. A. Manteuffel, Constrained optimization for liquid crystal equilibria, SIAM J. Sci. Comput. 38 (1) (2016) B50–B76.
- [5] A. Brandt, N. Dinar, Multigrid solutions to elliptic flow problems, in: Numerical methods for partial differential equations, Vol. 42 of Publ. Math. Res. Center Univ. Wisconsin, Academic Press, New York-London, 1979, pp. 53–147.
- [6] M. A. Olshanskii, Multigrid analysis for the time dependent Stokes problem, Math. Comp. 81 (277) (2012) 57–79.

- [7] J. Schöberl, W. Zulehner, On Schwarz-type smoothers for saddle point problems, *Numer. Math.* 95 (2) (2003) 377–399.
- [8] J. H. Adler, T. R. Benson, S. P. MacLachlan, Preconditioning a mass-conserving discontinuous Galerkin discretization of the Stokes equations, *Numer. Linear Algebra Appl.* 24 (3) (2017) e2047, 23.
- [9] P. E. Farrell, L. Mitchell, F. Wechsung, An augmented lagrangian preconditioner for the 3D stationary incompressible Navier-Stokes equations at high Reynolds number, arXiv preprint arXiv:1810.03315.
- [10] U. Trottenberg, C. W. Oosterlee, A. Schüller, *Multigrid*, Academic Press, Inc., San Diego, CA, 2001, with contributions by A. Brandt, P. Oswald and K. Stüben.
- [11] R. Wienands, W. Joppich, *Practical Fourier analysis for multigrid methods*, CRC press, 2004.
- [12] A. Brandt, Rigorous quantitative analysis of multigrid, I. constant coefficients two-level cycle with L_2 -norm, *SIAM Journal on Numerical Analysis* 31 (6) (1994) 1695–1730.
- [13] R. P. Stevenson, On the validity of local mode analysis of multi-grid methods, Ph.D. thesis (1990).
- [14] C. Rodrigo, F. J. Gaspar, L. T. Zikatanov, On the validity of the local Fourier analysis, *Journal of Computational Mathematics* 37 (3) (2019) 340–348.
- [15] Y. He, S. P. MacLachlan, Two-level Fourier analysis of multigrid for higher-order finite-element methods, submitted.
- [16] S. P. MacLachlan, C. W. Oosterlee, Local Fourier analysis for multigrid with overlapping smoothers applied to systems of PDEs, *Numer. Linear Algebra Appl.* 18 (4) (2011) 751–774.
- [17] S. Friedhoff, S. MacLachlan, C. Borgers, Local Fourier analysis of space-time relaxation and multigrid schemes, *SIAM Journal on Scientific Computing* 35 (5) (2013) S250–S276.
- [18] Y. He, S. P. MacLachlan, Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations, *Numerical Linear Algebra with Applications* 25 (3), e2147.
- [19] D. Braess, R. Sarazin, An efficient smoother for the Stokes problem, *Appl. Numer. Math.* 23 (1) (1997) 3–19.
- [20] D. Braess, W. Dahmen, A cascadic multigrid algorithm for the Stokes equations, *Numer. Math.* 82 (2) (1999) 179–191.
- [21] W. Zulehner, A class of smoothers for saddle point problems, *Computing* 65 (3) (2000) 227–246.
- [22] S. P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, *J. Comput. Phys.* 65 (1) (1986) 138–158.
- [23] S. Manservigi, Numerical analysis of Vanka-type solvers for steady Stokes and Navier-Stokes flows, *SIAM J. Numer. Anal.* 44 (5) (2006) 2025–2056.
- [24] C. Rodrigo, F. J. Gaspar, F. J. Lisbona, On a local Fourier analysis for overlapping block smoothers on triangular grids, *Appl. Numer. Math.* 105 (2016) 96–111.
- [25] L. John, U. Rüde, B. Wohlmuth, W. Zulehner, On the analysis of block smoothers for saddle point problems, arXiv preprint arXiv:1612.01333.
- [26] F. J. Gaspar, Y. Notay, C. W. Oosterlee, C. Rodrigo, A simple and efficient segregated smoother for the discrete Stokes equations, *SIAM J. Sci. Comput.* 36 (3) (2014) A1187–A1206.
- [27] B. Gmeiner, M. Huber, L. John, U. Rüde, B. Wohlmuth, A quantitative performance study for Stokes solvers at the extreme scale, *J. Comput. Sci.* 17 (part 3) (2016) 509–521.
- [28] J.-F. Maitre, F. Musy, P. Nigon, A fast solver for the Stokes equations using multigrid with a Uzawa smoother, in: *Advances in multigrid methods* (Oberwolfach, 1984), Vol. 11 of *Notes Numer. Fluid Mech.*, Vieweg+Teubner Verlag, Wiesbaden, 1985, pp. 77–83.
- [29] C. Bacuta, P. S. Vassilevski, S. Zhang, A new approach for solving Stokes systems arising from a distributive relaxation method, *Numerical Methods for Partial Differential Equations* 27 (4) (2011) 898–914.
- [30] G. Wittum, Multi-grid methods for Stokes and Navier-Stokes equations, *Numer. Math.* 54 (5) (1989) 543–563.
- [31] M. Wang, L. Chen, Multigrid methods for the Stokes equations using distributive Gauss-Seidel relaxations based on the least squares commutator, *J. Sci. Comput.* 56 (2) (2013) 409–431.
- [32] C. W. Oosterlee, F. J. Gaspar, Multigrid methods for the Stokes system, *Computing in Science & Engineering* 8 (6) (2006) 34–43.
- [33] L. Chen, Multigrid methods for saddle point systems using constrained smoothers, *Comput. Math. Appl.* 70 (12) (2015) 2854–2866.
- [34] S. Takacs, A robust multigrid method for the time-dependent Stokes problem, *SIAM J. Numer. Anal.* 53 (6) (2015) 2634–2654.
- [35] A. Niestegge, K. Witsch, Analysis of a multigrid Stokes solver, *Appl. Math. Comput.* 35 (3) (1990) 291–303.
- [36] S. Sivaloganathan, The use of local mode analysis in the design and comparison of multigrid methods, *Computer Physics Communications* 65 (1-3) (1991) 246–252.
- [37] J. Molenaar, A two-grid analysis of the combination of mixed finite elements and Vanka-type relaxation, in: *Multigrid methods, III* (Bonn, 1990), Vol. 98 of *Internat. Ser. Numer. Math.*, Birkhäuser, Basel, 1991, pp. 313–323.
- [38] A. J. Wathen, T. Rees, Chebyshev semi-iteration in preconditioning for problems including the mass matrix, *Electronic Transactions on Numerical Analysis* 34 (125-135) (2009) 125–135.

- [39] A. Wathen, D. Silvester, Fast iterative solution of stabilised Stokes systems. part I: Using simple diagonal preconditioners, *SIAM J. Numer. Anal.* 30 (3) (1993) 630–649.
- [40] D. Silvester, A. Wathen, Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners, *SIAM J. Numer. Anal.* 31 (5) (1994) 1352–1367.
- [41] S. C. Brenner, L. R. Scott, *The mathematical theory of finite element methods*, 3rd Edition, Vol. 15 of *Texts in Applied Mathematics*, Springer, New York, 2008.
- [42] F. Brezzi, J. Douglas, Stabilized mixed methods for the Stokes problem, *Numer. Math.* 53 (1) (1988) 225–235.
- [43] C. R. Dohrmann, P. B. Bochev, A stabilized finite element method for the Stokes problem based on polynomial pressure projections, *International Journal for Numerical Methods in Fluids* 46 (2) (2004) 183–201.
- [44] L. Chen, X. Hu, M. Wang, J. Xu, A multigrid solver based on distributive smoother and residual overweighting for Oseen problems, *Numerical Mathematics: Theory, Methods and Applications* 8 (02) (2015) 237–252.
- [45] J. H. Adler, D. Emerson, S. P. MacLachlan, T. A. Manteuffel, Constrained optimization for liquid crystal equilibria, *SIAM Journal on Scientific Computing* 38 (1) (2016) B50–B76.