

A local Fourier analysis of additive Vanka relaxation for the Stokes equations

Patrick E. Farrell¹  | Yunhui He²  | Scott P. MacLachlan² 

¹Mathematical Institute, University of Oxford, Oxford, UK

²Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, Newfoundland, Canada

Correspondence

Yunhui He, Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, A1C 5S7, Canada.

Email: yunhui.he@mun.ca

Funding information

Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/K030930/1, EP/R029423/1; Natural Sciences and Engineering Research Council of Canada

Summary

Multigrid methods are popular solution algorithms for many discretized PDEs, either as standalone iterative solvers or as preconditioners, due to their high efficiency. However, the choice and optimization of multigrid components such as relaxation schemes and grid-transfer operators is crucial to the design of optimally efficient algorithms. It is well known that local Fourier analysis (LFA) is a useful tool to predict and analyze the performance of these components. In this article, we develop a local Fourier analysis of monolithic multigrid methods based on additive Vanka relaxation schemes for mixed finite-element discretizations of the Stokes equations. The analysis offers insight into the choice of “patches” for the Vanka relaxation, revealing that smaller patches offer more effective convergence per floating point operation. Parameters that minimize the two-grid convergence factor are proposed and numerical experiments are presented to validate the LFA predictions.

KEYWORDS

additive Vanka, local Fourier analysis, monolithic multigrid, Stokes equations, triangular grids

1 | INTRODUCTION

Saddle-point problems are ubiquitous in applied mathematics.¹ Their importance motivates the development of effective parallel solvers. Block preconditioners and monolithic multigrid methods are established approaches for solving linear (or linearized) saddle-point problems. Block preconditioning is highly effective when the Schur complement of the system is well understood; for the Stokes equations, the Schur complement is spectrally equivalent to a weighted mass matrix, forming the basis for efficient solvers that use multigrid for the viscous term.^{2,3} Monolithic methods that apply multigrid to the entire system can also offer superb efficiency if an appropriate relaxation can be devised. For example, Adler et al⁴ proposed a monolithic multigrid method with Braess–Sarazin relaxation for the Stokes equations that provided the fastest time to solution when compared with several block preconditioners and other monolithic multigrid methods. While many block preconditioners have been successfully employed in (massively) parallel computing environments,^{5,6} the same cannot be said for monolithic multigrid, whose parallelization was largely absent from the literature until recently.^{7–9}

Common approaches to multigrid relaxation for coupled systems are distributive relaxation^{10,11} (which relies on continuum commutativity that may not hold at the discrete level), Braess–Sarazin relaxation,^{12,13} Uzawa-type relaxation,^{14,15} and Vanka relaxation.^{16,17} Based on distributive relaxation, Wang and Chen¹⁸ developed a least squares commutator distributive Gauss–Seidel relaxation for the Stokes equations. Furthermore, this technique has been extended to the Oseen problem by Chen et al.¹⁹ Braess–Sarazin relaxation is known to be highly efficient and has been applied to nematic

liquid crystals,²⁰ magnetohydrodynamics,¹³ and other coupled systems. Considering parallel computation, recently, He and MacLachlan presented a local Fourier analysis (LFA) for both distributive weighted Jacobi and Braess–Sarazin relaxations for the Stokes equations discretized by the marker-and-cell finite-difference scheme²¹ and by mixed finite-element methods,²² showing the power of LFA for designing efficient algorithms.

Vanka-type relaxation has been used in many contexts, such as for the Navier–Stokes equations,^{16,23} and extended to Vanka-like schemes for other problems or to improve performance.^{4,7,13,24–29} However, Vanka relaxation is typically considered in its multiplicative variant. This seems overly constraining, particularly in comparison to Braess–Sarazin relaxation, which can naturally be done in additive form.^{21,22} While multiplicative Vanka relaxation is very efficient, it has a higher cost per iteration than the additive variant and its parallelization is more involved. We therefore consider additive variants of Vanka-type relaxation in this work.

There are two challenging choices to be made for Vanka relaxation, which are observed to be more critical in the additive setting. First, many choices are possible for the underlying patches within the overlapping Schwarz framework. While we would naturally choose small patches for efficiency or large patches for effectiveness, no general results are known. Second, relaxation weights play an important role in ensuring best possible performance of the multigrid algorithm, particularly for additive methods. Thus, there is a need for analysis to inform the algorithmic choices, and LFA seems well suited. LFA has already been applied to Vanka relaxation in the multiplicative^{27,30–33} and multicolored^{28,34} contexts; here, by contrast, we aim to develop LFA for additive schemes and use it to drive parameter choices in practical experiments for the Stokes equations. To our knowledge, this is the first time that LFA has been applied to additive overlapping Vanka relaxation.

We consider the Stokes equations as a model problem, with both $P_2 - P_1$ and $Q_2 - Q_1$ discretizations. We propose two constructions of the patches for Vanka relaxation and two approaches to determine relaxation weights. It is shown that using weighting based on patch geometry outperforms a simpler approach. We also find that using small patches with low-degree Chebyshev iterations leads to more efficient multigrid algorithms than with bigger patches or more relaxation steps per iteration, when cost per sweep is accounted for. Although there are no general rules to facilitate the choice of patches or weights, taking advantage of LFA, we can optimize the weights. For validation, our numerical tests are implemented using Firedrake³⁵ and PETSc.^{36,37} Numerical experiments are shown to match the LFA predictions for both periodic and Dirichlet boundary conditions. We observe that performance is less sensitive to overestimates of the weights for relaxation schemes considered here, which has also been seen in other works.^{4,13} Last but not least, we compare the cost and performance of relaxation schemes considered here.

This article is organized as follows. In Section 2, we introduce the $P_2 - P_1$ and $Q_2 - Q_1$ discretizations considered here and the multigrid framework with additive Vanka relaxation for the Stokes equations. In Section 3, we first give an introduction to LFA, then propose an LFA for the Stokes equations with additive Vanka relaxation. In Section 4, two types of overlapping patches are considered, and we validate the LFA predictions with two-grid experiments. Conclusions and remarks are given in Section 5.

2 | DISCRETIZATION AND SOLUTION OF THE STOKES EQUATIONS

2.1 | Mixed finite-element discretization of the Stokes equations

In this article, we consider the Stokes equations,

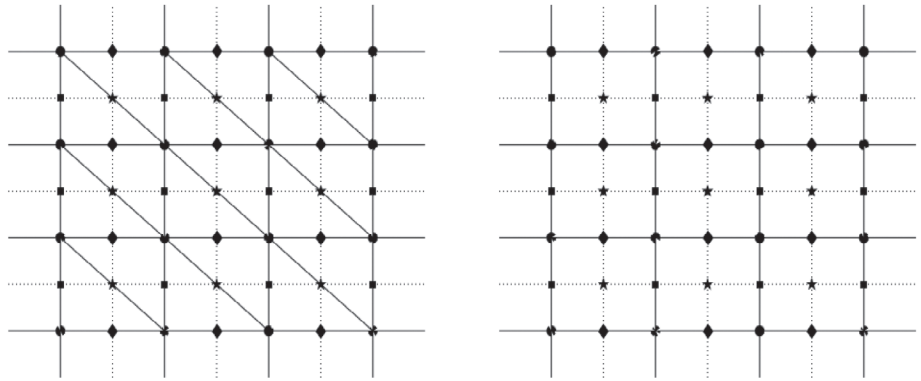
$$\begin{aligned} -\Delta \vec{u} + \nabla p &= \vec{f}, \\ -\nabla \cdot \vec{u} &= 0, \end{aligned} \quad (1)$$

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.

The natural finite-element approximation of Problem (1) when coupled with Dirichlet boundary conditions on \vec{u} on some portion of the domain boundary is: Find $\vec{u}_h \in \mathcal{X}^h$ and $p_h \in \mathcal{H}^h$ such that

$$a(\vec{u}_h, \vec{v}_h) + b(p_h, \vec{v}_h) + b(q_h, \vec{u}_h) = g(\vec{v}_h), \quad \text{for all } \vec{v}_h \in \mathcal{X}_0^h \quad \text{and} \quad q_h \in \mathcal{H}^h, \quad (2)$$

FIGURE 1 Meshes and finite-element degrees of freedom (see definitions in (7)), with \bullet denoting N -type and P_1/Q_1 DoFs, \blacklozenge denoting X -type DoFs, \blacksquare denoting Y -type DoFs, and \star denoting C -type DoFs. At left, $P_2 - P_1$ discretization on triangles. At right, $Q_2 - Q_1$ discretization on quadrilaterals



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 satisfies homogeneous Dirichlet boundary conditions in place of any inhomogeneous essential boundary conditions on \mathcal{X}^h . Problem (2) has a unique solution only when \mathcal{X}^h and \mathcal{H}^h satisfy an inf-sup condition.³⁸⁻⁴¹

Remark 1. If considering an outflow boundary condition, the stress-divergence form of the viscous term should be used instead.⁴² The framework for LFA presented in this article can easily be extended to this situation; however, the two-grid error-propagation operator (and its LFA representation) depends directly on the stencils of the discretized operators. Thus, considering this form instead of (2) may affect the optimal choice of parameters and the resulting performance of the two-grid method.

Here, we consider two types of stable finite-element methods for the Stokes equations. First, we consider the stable mixed approximation for structured meshes of triangular elements using continuous quadratic approximations for the velocity components and continuous linear approximations for the pressure, the $P_2 - P_1$ approximation.³⁸ Second, we consider the stable approximation for rectangular meshes, using continuous biquadratic approximations for the velocity components and continuous bilinear approximations for the pressure, the $Q_2 - Q_1$ (Taylor–Hood) approximation. Both approximations can be represented via nodal basis functions, as shown in Figure 1.

Discretizations of (1) typically lead to linear systems of the form

$$Ky = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} = b, \quad (3)$$

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 $C = 0$.³⁸ For the stable $P_2 - P_1$ and $Q_2 - Q_1$ finite-element discretizations considered here, we take $C = 0$.

2.2 | Monolithic multigrid for the Stokes equations

System (3) is of saddle-point type. Here, we consider the numerical solution of (3) using a monolithic multigrid iteration applied to the full system collectively with a suitable (coupled) relaxation method. As is typical for geometric multigrid, a relaxation technique is employed to quickly damp all oscillatory components of the error. Subsequently, a coarse-grid correction scheme, where a projected problem is solved on a coarser grid and the solution is interpolated as an error correction to the fine-grid approximation, is used to damp the smooth components of the error. In order to describe monolithic multigrid, assume we have two meshes, with fine-grid meshsize h and coarse-grid meshsize H (often, $H = 2h$, by doubling the meshsize in each spatial direction).

For a general nonsingular linear system, $K_h u_h = b_h$, we consider a stationary iteration as the relaxation scheme. Given an approximation, M_h , to K_h that can be inverted easily, the approximate solution is updated via the iteration

$$u_h^{j+1} = (I - M_h^{-1} K_h) u_h^j + M_h^{-1} b_h. \quad (4)$$

The matrix $S_h := I - M_h^{-1} K_h$ is the error propagation operator for relaxation. With restriction and interpolation operators, R_h and P_h , respectively, and coarse-grid matrix K_H , the two-grid error propagation operator corresponding to the relaxation scheme in (4) can be written as

$$E_h = S_h^{v_2} (I - P_h K_H^{-1} R_h K_h) S_h^{v_1}, \quad (5)$$

where $I - P_h K_H^{-1} R_h K_h$ is called the coarse-grid correction operator.

The Jacobi, Gauss–Seidel, and Richardson schemes are often used for relaxation, particularly for discretizations of scalar PDEs. For the restriction operator, R_h , there are many choices, which depend on the problem under consideration. Here, we focus on choices of R_h tied to the mesh and the particular discretization scheme used to generate K_h . The coarse-grid operator, K_H can be the Galerkin operator, $K_H = R_h K_h P_h$, or the natural rediscritization operator (or any other choice). The interpolation operator, P_h , is usually taken to be the conjugate transpose of R_h , with scaling depending on the discretization scheme and the dimension of the considered problem. For more details on the choice of multigrid components, see References 43–45.

If we solve the coarse-grid problem recursively by the two-grid method, then we obtain a multigrid method. Over the past decades, a variety of types of multigrid methods have been developed, including W , V , and F cycles.⁴⁴ In this article, we focus on using additive Vanka-type relaxation in combination with a monolithic multigrid method to solve (3). This means that M_h is constructed by the Vanka approach, and updates both components of the solution to (3) at the same time in the relaxation scheme. Only two-grid schemes are considered.

2.2.1 | Overlapping Schwarz relaxation

Here, we present the multiplicative and additive Schwarz approaches to solve $K_h u_h = b_h$. Let the degrees of freedom (DoFs) of u_h be the set \mathfrak{S} , and $\mathfrak{S}_i, i = 1, 2, \dots, N$, be subsets of unknowns with $\mathfrak{S} = \bigcup_{i=1}^N \mathfrak{S}_i$. Let V_i be the restriction operator mapping from vectors over the set of all unknowns, \mathfrak{S} , to vectors whose unknowns consist of the DoFs in \mathfrak{S}_i . Then $K_i = V_i K_h V_i^T$ is the restriction of K_h to the i th block of DoFs. Moreover, let $D_i = \text{diag}(d_1^i, d_2^i, \dots, d_{m_i}^i)$ for $i = 1, \dots, N$ be a diagonal weight matrix for each block i , where m_i is the dimension of K_i . Then, the multiplicative and additive Schwarz iterations are presented in Algorithm 1.

Algorithm 1. Multiplicative Schwarz iteration

$u^{j,0} = u^{j-1,N}$. For $i = 1, \dots, N$,

$$K_i \delta u_i = V_i (b_h - K_h u^{j,i-1}),$$

and

$$u^{j,i} = u^{j,i-1} + V_i^T D_i \delta u_i.$$

Additive Schwarz iteration:

For $i = 1, \dots, N$,

$$K_i \delta u_i = V_i (b_h - K_h u^j),$$

and

$$u^{j+1} = u^j + \sum_{i=1}^N V_i^T D_i \delta u_i.$$

The error-propagation operator for the multiplicative Schwarz procedure can be written as

$$S_m = \prod_{i=1}^N (I - V_i^T D_i K_i^{-1} V_i K_h)$$

and, for the additive Schwarz procedure, it is

$$S_a = I - M_h^{-1} K_h,$$

where

$$M_h^{-1} = \sum_{i=1}^N V_i^T D_i K_i^{-1} V_i. \quad (6)$$

More details about this algebraic viewpoint on the multiplicative and additive Schwarz iterations can be found, for example, in the work of Saad.⁴⁶ Overlapping multiplicative Schwarz approaches have been used as the relaxation scheme for the Stokes equations,^{4,7,16,24-26,32,33} which we refer to as multiplicative Vanka relaxation. Extending this, we will refer to such overlapping additive Schwarz approaches as additive Vanka relaxation. In this article, we focus on using additive Vanka as a relaxation scheme within monolithic multigrid for the Stokes equations. Some key questions in doing this are

1. How should the subsets \mathfrak{S}_i be chosen?
2. How should D_i be chosen?

In what follows, we consider uniform meshes for the domain, $\Omega = [0, 1]^2$. We will use the pressure DoFs to “seed” the sets, \mathfrak{S}_i , so that (away from the boundary) all sets will have the same structure and size m . In this article, we use local Fourier analysis to guide the choice of \mathfrak{S}_i and other aspects of the relaxation scheme.

LFA for multiplicative Vanka-type relaxation for the $Q_2 - Q_1$ discretization of the Stokes equations was first performed by MacLachlan and Oosterlee,³² building on early work by Molenaar³¹ and Sivaloganathan.³⁰ This was extended to the $P_2 - P_1$ discretization on triangular meshes by Rodrigo et al..³³ In the multiplicative context, the overlap requires careful treatment within LFA, leading to Fourier representations of so-called “stages” of error within the algorithm. For the additive case considered here, the analysis is much simpler, akin to that of multicolored relaxation.^{28,34} While we do not consider LFA for multiplicative Vanka here, we do offer numerical comparisons of performance between the additive and multiplicative variants below, when using the same subsets, \mathfrak{S}_i , to define the relaxation schemes.

The use of additive Schwarz-type iterations as relaxation schemes in multigrid methods for saddle-point problems was previously analyzed by Schöberl and Zulehner.²⁶ The authors show that, under suitable conditions, the iteration can be interpreted as a symmetric inexact Uzawa method. The advantage of this analysis compared with LFA is that it applies to unstructured grids, although at the cost of giving less quantitative insight. They also propose a general strategy for constructing patches (gathering the velocity degrees of freedom connected to given pressure degrees of freedom) and weights, and apply this to the Crouzeix–Raviart discretization of the Stokes equations. This patch strategy yields the so-called inclusive patches considered subsequently in Figure 5, although with different weights (as these are tuned by LFA in this work).

More generally, both additive and multiplicative Vanka relaxation can be viewed as domain decomposition methods. Thus, the algorithms considered here could also be analyzed from the perspective of one- and two-level domain decomposition approaches, albeit with smaller than typical subdomains. The work of Szyld and Frommer⁴⁷ seems most relevant to the relaxation schemes considered here, analyzing the convergence of overlapping additive Schwarz, but in the case of M -matrices. Similar analysis exists for overlapping multiplicative Schwarz, by Benzi et al..⁴⁸ To our knowledge, such approaches have yet to be applied to Vanka-type relaxation.

3 | LOCAL FOURIER ANALYSIS

3.1 | Definitions and notations

We first introduce some terminology of LFA.^{43,49} We consider the following two-dimensional infinite uniform grids, $\mathbf{G}_h = \bigcup_{j=1}^4 \mathbf{G}_h^j$, where

$$\mathbf{G}_h^j = \left\{ \mathbf{x}^j := (\mathbf{x}_1^j, \mathbf{x}_2^j) = (\mathbf{k}_1, \mathbf{k}_2)h + \delta^j, (\mathbf{k}_1, \mathbf{k}_2) \in \mathbb{Z}^2 \right\}, \quad (7)$$

with

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

We refer to \mathbf{G}_h^1 , \mathbf{G}_h^2 , \mathbf{G}_h^3 , and \mathbf{G}_h^4 as the N -, X -, Y -, and C -type points on the grid \mathbf{G}_h , respectively (see Figure 1). The coarse grids, \mathbf{G}_{2h}^j , are defined similarly. Note that in much of the literature, LFA is applied to discretizations on \mathbf{G}_h^1 . Here, we consider the more general case as needed for P_2 and Q_2 finite elements.

Let L_h be a scalar Toeplitz operator defined by its stencil acting on $l^2(\mathbf{G}_h^j)$ as follows,

$$L_h \hat{=} [s_\kappa]_h \quad (\kappa = (\kappa_1, \kappa_2) \in \mathbf{V}); \quad L_h w_h(\mathbf{x}^j) = \sum_{\kappa \in \mathbf{V}} s_\kappa w_h(\mathbf{x}^j + \kappa h), \quad (8)$$

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, $\mathbf{V} \subset \mathbb{Z}^2$ is a finite index set. Because L_h is formally diagonalized by the Fourier modes $\varphi(\theta, \mathbf{x}^j) = e^{i\theta \cdot \mathbf{x}^j/h} = e^{i\theta_1 \mathbf{x}_1^j/h} e^{i\theta_2 \mathbf{x}_2^j/h}$, where $\theta = (\theta_1, \theta_2)$ and $i^2 = -1$, we use $\varphi(\theta, \mathbf{x}^j)$ as a Fourier basis with $\theta \in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^2$ (or any pair of intervals with length 2π).

For smoothing and two-grid analysis, we have to distinguish high and low frequency components on \mathbf{G}_h^j with respect to \mathbf{G}_{2h}^j .⁴³ Note that for any $\theta' \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2$,

$$\varphi(\theta, \mathbf{x}^j) = \pm \varphi(\theta', \mathbf{x}^j) \quad \text{for } \mathbf{x}^j \in \mathbf{G}_{2h}^j, \quad (9)$$

if and only if $\theta = \theta' \pmod{\pi}$. This means that only those frequency components, $\varphi(\theta, \cdot)$, with $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2$ are distinguishable on \mathbf{G}_{2h}^j . Thus, high and low frequencies for standard coarsening ($H = 2h$) are given by

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

Definition 1. We call $\tilde{L}_h(\theta) = \sum_{\kappa \in \mathbf{V}} s_\kappa e^{i\theta \cdot \kappa}$ the symbol of L_h .

Note that for all functions $\varphi(\theta, \mathbf{x}^j)$,

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

For a relaxation scheme, represented by matrix M_h for operator L_h , the error-propagation operator for relaxation can be written as

$$S_h(\mathbf{p}) = I - M_h^{-1}(\mathbf{p})L_h,$$

where \mathbf{p} represents parameters within M_h . A typical relaxation scheme often reduces high-frequency error components quickly, but is slow to reduce low-frequency errors. Thus, it is natural to define the smoothing factor as follows.

Definition 2. The error-propagation symbol, $\tilde{S}_h(\theta, \mathbf{p})$, for relaxation scheme $S_h(\mathbf{p})$ on the infinite grid \mathbf{G}_h satisfies

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

for all $\varphi(\theta, \mathbf{x})$, and the corresponding smoothing factor for $S_h(\mathbf{p})$ is given by

$$\mu_{\text{loc}} = \mu_{\text{loc}}(S_h(\mathbf{p})) = \max_{\theta \in T^{\text{high}}} \{|\tilde{S}_h(\theta, \mathbf{p})|\}.$$

In many cases, the LFA smoothing factor offers a good prediction of actual two-grid performance, and we can optimize the smoothing factor with respect to the parameters, \mathbf{p} , to obtain an efficient algorithm. However, this is generally not true for higher order finite-element approximations.^{22,32,50} Thus, we next introduce two-grid LFA, which still offers good predictions of performance in this setting.

Remark 2. In many applications of LFA, we consider a system operator rather than the discretization of a scalar PDE, and S_h is a block smoother. However, the definition of symbol and smoothing factor presented here can be extended to a system easily. Details on these extensions will be presented in Section 3.3.

3.2 | Two-grid LFA

In general, LFA smoothing analysis gives a good prediction for the actual multigrid performance, under the assumption that we have an “ideal” coarse-grid-correction operator that annihilates low-frequency error components and leaves high-frequency components unchanged. However, in our setting, this assumption about ideal coarse-grid correction (CGC) does not hold (due to the discretization⁵⁰), but the two-grid LFA convergence factor still offers useful predictions.

To apply LFA to the two-grid operator, (5), and calculate the two-grid convergence factor, we need to analyze how the operators K_h, P_h, R_h , and S_h act on the Fourier components $\varphi(\theta, \mathbf{x}^j)$. From (9), we know that values of $\varphi(\theta, \mathbf{x}^j)$ coincide on \mathbf{G}_{2h}^j for four values of θ , known as harmonic frequencies. Let

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

For a given $\theta \in T^{\text{low}}$, we define the harmonic space

$$\mathcal{F}(\theta) = \text{span} \{ \varphi(\theta^\alpha, \cdot) : \alpha \in \{(0, 0), (1, 0), (0, 1), (1, 1)\} \}.$$

Under standard assumptions, the space $\mathcal{F}(\theta)$ is invariant under the two-grid operator \mathbf{E}_h .^{43,49} We use the ordering of $\alpha = (0, 0), (1, 0), (0, 1), (1, 1)$ for the four harmonics in the following, although, as with any invariant subspace, the ordering of the basis elements is unimportant.

Inserting the representations of S_h, K_h, K_H, P_h, R_h into (5), we obtain the Fourier representation of two-grid error-propagation operator as

$$\tilde{\mathbf{E}}_h(\theta, p) = \tilde{\mathbf{S}}_h^{\nu_2}(\theta, p) (I - \tilde{\mathbf{P}}_h(\theta)(\tilde{\mathbf{K}}_H(2\theta))^{-1}\tilde{\mathbf{R}}_h(\theta)\tilde{\mathbf{K}}_h(\theta)) \tilde{\mathbf{S}}_h^{\nu_1}(\theta, p),$$

where

$$\begin{aligned}\tilde{\mathbf{K}}_h(\theta) &= \text{diag} \{ \tilde{K}_h(\theta^{00}), \tilde{K}_h(\theta^{10}), \tilde{K}_h(\theta^{01}), \tilde{K}_h(\theta^{11}) \}, \\ \tilde{\mathbf{S}}_h(\theta, p) &= \text{diag} \{ \tilde{S}_h(\theta^{00}, p), \tilde{S}_h(\theta^{10}, p), \tilde{S}_h(\theta^{01}, p), \tilde{S}_h(\theta^{11}, p) \}, \\ \tilde{\mathbf{R}}_h(\theta) &= (\tilde{R}_h(\theta^{00}), \tilde{R}_h(\theta^{10}), \tilde{R}_h(\theta^{01}), \tilde{R}_h(\theta^{11})), \\ \tilde{\mathbf{P}}_h(\theta) &= (\tilde{P}_h(\theta^{00}), \tilde{P}_h(\theta^{10}), \tilde{P}_h(\theta^{01}), \tilde{P}_h(\theta^{11})),\end{aligned}$$

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 .

Definition 3. The asymptotic two-grid convergence factor, ρ_{asp} , is defined as

$$\rho_{\text{asp}} = \sup \{ \rho(\tilde{\mathbf{E}}_h(\theta, p)) : \theta \in T^{\text{low}} \}. \quad (10)$$

In practical use, we typically consider a discrete form of ρ_{asp} , denoted by ρ , resulting from sampling ρ_{asp} over only a finite set of frequencies. In many cases, ρ provides a sharp prediction of actual two-grid performance. It is well known, for example, that LFA gives the exact two-grid convergence factor for problems with periodic boundary conditions.⁴⁴ The calculation of ρ is much cheaper, however, than direct calculation of $\rho(\mathbf{E}_h)$ from (5). More importantly, since ρ is a function of the parameters, \mathbf{p} , arising from the relaxation scheme (or the coarse-grid correction), we can optimize ρ to achieve an

optimally efficient algorithm. In our setting, such parameters appear in the diagonal scaling matrices D_i mentioned in Section 2.2.1. One of our goals in this article is to use LFA to optimize the two-grid convergence factor of multigrid when using such relaxation schemes.

Next, we provide details on LFA for additive Vanka relaxation for the Stokes equations. Considering practical use, we focus on the stable $P_2 - P_1$ discretization, as is easily generated using general-purpose FEM tools on simplicial meshes, such as Firedrake³⁵ and FEniCS.^{51,52} While LFA for this discretization has previously been considered by Rodrigo et al,³³ we make use of a different mesh construction and Fourier basis than proposed there. Thus, in the following sections, we provide full details of the LFA for both the discrete operator and finite-element interpolation operators in this setting. We will also show LFA predictions for additive Vanka relaxation for the $Q_2 - Q_1$ discretization. LFA for the $Q_2 - Q_1$ discretization of the Stokes equations has been previously presented in the work of He and MacLachlan,²² where the authors consider different relaxation schemes, but the operator stencil and grid-transfer operators are the same as those used here. Since the key ingredient of the multigrid method of interest here is the use of the additive Vanka relaxation scheme, we present full details of the LFA representation of the relaxation schemes for both $P_2 - P_1$ and $Q_2 - Q_1$ discretizations.

3.3 | LFA for $P_2 - P_1$

In what follows, we consider the discretized Stokes equations, which read

$$K_h = \begin{pmatrix} -\Delta_h & 0 & (\partial_x)_h \\ 0 & -\Delta_h & (\partial_y)_h \\ -(\partial_x)_h & -(\partial_y)_h & 0 \end{pmatrix} := \begin{pmatrix} K_h^{1,1} & 0 & K_h^{1,3} \\ 0 & K_h^{2,2} & K_h^{2,3} \\ K_h^{3,1} & K_h^{3,2} & 0 \end{pmatrix}. \quad (11)$$

For the $P_2 - P_1$ 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 the left of Figure 1. The Laplace operator in (11) is defined by its weak form restricted to the finite-element basis functions. Here, we use the standard nodal basis and, consequently, can write this in stencil form, extending (8), with \mathbf{V} taken to be a finite index set of values, $\mathbf{V} = V_N \cup V_X \cup V_Y \cup V_C$ with $V_N \subset \mathbb{Z}^2$, $V_X \subset \left\{ \left(z_x + \frac{1}{2}, z_y \right) \mid (z_x, z_y) \in \mathbb{Z}^2 \right\}$, $V_Y \subset \left\{ \left(z_x, z_y + \frac{1}{2} \right) \mid (z_x, z_y) \in \mathbb{Z}^2 \right\}$, and $V_C \subset \left\{ \left(z_x + \frac{1}{2}, z_y + \frac{1}{2} \right) \mid (z_x, z_y) \in \mathbb{Z}^2 \right\}$. With this, the (scalar) discrete Laplace operator is naturally treated as a block operator, and the Fourier representation of each block can be calculated based on Definition 1, with the Fourier bases adapted to account for the staggering of the mesh points. Thus, the symbols of $K_h^{1,1}$ and $K_h^{2,2}$ are 4×4 matrices. 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 $K_h^{1,3}$ and $K_h^{2,3}$ are 4×1 matrices, calculated based on Definition 1 adapted for the mesh staggering. The symbols of $K_h^{3,1}$ and $K_h^{3,2}$ are the conjugate transposes of those of $K_h^{1,3}$ and $K_h^{2,3}$, respectively. Accordingly, \tilde{K}_h is a 9×9 matrix for the $P_2 - P_1$ discretization.

We denote the symbols of the finite-element discretizations of the Stokes equations as

$$\tilde{K}_h(\theta_1, \theta_2) = \begin{pmatrix} \tilde{A}(\theta_1, \theta_2) & 0 & \tilde{B}_x^T(\theta_1, \theta_2) \\ 0 & \tilde{A}(\theta_1, \theta_2) & \tilde{B}_y^T(\theta_1, \theta_2) \\ \tilde{B}_x(\theta_1, \theta_2) & \tilde{B}_y(\theta_1, \theta_2) & 0 \end{pmatrix}.$$

Next, we discuss the stencils and symbols for the operators in (11). For the Laplace operator, the stencil can be split into four types, which correspond to the N -, X -, Y -, and C -type points, shown in Figure 2. For the Y -type, the stencil is a 90° rotation of that of X -type, so we do not include it.

Rewriting the stencils shown in Figure 2, we can write the four stencils of $K_h^{1,1}$ as follows,

$$A_N = \frac{1}{3} \begin{bmatrix} & 1 & & & \\ & -4 & & & \\ 1 & -4 & 12 & -4 & 1 \\ & -4 & & & \\ & 1 & & & \end{bmatrix}, \quad A_X = A_Y = A_C = \frac{1}{3} \begin{bmatrix} & -4 & & & \\ -4 & 16 & -4 & & \\ & -4 & & & \end{bmatrix}.$$

Note that each stencil connects multiple types of meshpoints, so we further split each stencil into four substencils based on the type of DoFs. Taking A_N as an example, we see it connects three of the four types of meshpoints. Thus, A_N

FIGURE 2 Stencils for the P_2 finite-element discretization of the Laplace operator on a left triangular grid. Left: connections from DoF at a mesh node. Middle: connections from a horizontal edge. Right: connections from a diagonal mesh edge

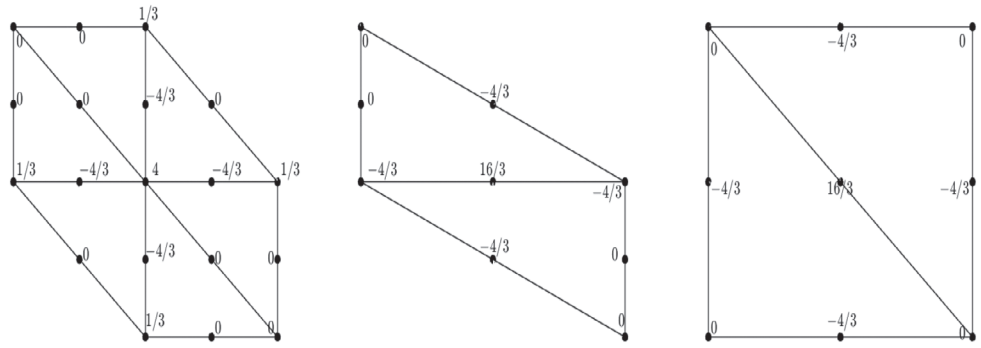
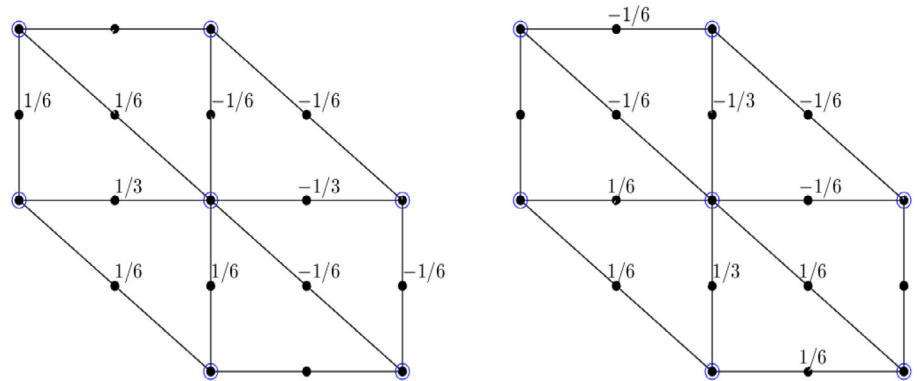


FIGURE 3 Stencils for the $P_2 - P_1$ finite-element discretization of the derivative operators on a left triangular grid. Left: $(\partial_x)_h$ stencil with a scaling h . Right: $(\partial_y)_h$ stencil with a scaling h . The (blue) circles at the center are the locations of pressure unknowns and the marked points without a given weight have value 0



can be written as $A_N = (A_{N,N} \ A_{N,X} \ A_{N,Y} \ A_{N,C})$, where

$$A_{N,N} = \frac{1}{3} \begin{bmatrix} 1 & 12 \\ 1 & 1 \end{bmatrix}, \quad A_{N,X} = \frac{1}{3} \begin{bmatrix} -4 & -4 \end{bmatrix}, \quad A_{N,Y} = \frac{1}{3} \begin{bmatrix} -4 \\ -4 \end{bmatrix}, \quad A_{N,C} = 0.$$

By standard calculation based on Definition 1, we have

$$\tilde{A}_{N,N} = 4 + \frac{2}{3} (\cos \theta_1 + \cos \theta_2), \quad \tilde{A}_{N,X} = -\frac{8}{3} \cos \frac{\theta_1}{2}, \quad \tilde{A}_{N,Y} = -\frac{8}{3} \cos \frac{\theta_2}{2}, \quad \tilde{A}_{N,C} = 0.$$

Similarly, $\tilde{A}_X, \tilde{A}_Y, \tilde{A}_C$ can be treated in this way. Thus, the symbol of $K_h^{1,1}$ and $K_h^{2,2}$ can be written as

$$\tilde{A}(\theta_1, \theta_2) = \begin{pmatrix} \tilde{A}_N \\ \tilde{A}_X \\ \tilde{A}_Y \\ \tilde{A}_C \end{pmatrix} = \begin{pmatrix} \tilde{A}_{N,N} & \tilde{A}_{N,X} & \tilde{A}_{N,Y} & \tilde{A}_{N,C} \\ \tilde{A}_{X,N} & \tilde{A}_{X,X} & \tilde{A}_{X,Y} & \tilde{A}_{X,C} \\ \tilde{A}_{Y,N} & \tilde{A}_{Y,X} & \tilde{A}_{Y,Y} & \tilde{A}_{Y,C} \\ \tilde{A}_{C,N} & \tilde{A}_{C,X} & \tilde{A}_{C,Y} & \tilde{A}_{C,C} \end{pmatrix},$$

with $\tilde{A}(\theta_1, \theta_2)^T = \tilde{A}(\theta_1, \theta_2)$.

From Definition 1 and the stencils in Figure 2, we have the following symbols,

$$\begin{aligned} \tilde{A}_{X,X} &= \frac{16}{3}, & \tilde{A}_{X,Y} &= 0, & \tilde{A}_{X,C} &= -\frac{8}{3} \cos \frac{\theta_2}{2}, \\ \tilde{A}_{Y,Y} &= \frac{16}{3}, & \tilde{A}_{Y,C} &= -\frac{8}{3} \cos \frac{\theta_1}{2}, & \tilde{A}_{C,C} &= \frac{16}{3}. \end{aligned}$$

Similarly to the stencil of the Laplacian operator, the stencils of $(\partial_x)_h$ and $(\partial_y)_h$ can be split into four types of substencil, respectively. Figure 3 shows the stencils of the gradient, that is, the pressure-to-velocity unknown (N -, X -, Y -, and C -type) connections, for the pressure unknown located at the middle of the hexagon.

Thus, the stencil of $(\partial_x)_h$ shown in Figure 3 can be written as $B_x^T = [B_{x,N}; B_{x,X}; B_{x,Y}; B_{x,C}]$. However, here, we calculate the stencils of $-(\partial_x)_h$ and its symbols given by

$$\begin{aligned} B_{x,N} &= 0, \quad \tilde{B}_{x,N}(\theta_1, \theta_2) = 0, \\ B_{x,X} &= \frac{h}{3} \begin{bmatrix} -1 & 1 \end{bmatrix}, \quad \tilde{B}_{x,X}(\theta_1, \theta_2) = \frac{2ih}{3} \sin \frac{\theta_1}{2}, \\ B_{x,Y} &= \frac{h}{6} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad \tilde{B}_{x,Y}(\theta_1, \theta_2) = \frac{ih}{3} \left(\sin \frac{\theta_2}{2} + \sin \theta_1 \cos \frac{\theta_2}{2} - \cos \theta_1 \sin \frac{\theta_2}{2} \right), \\ B_{x,C} &= \frac{h}{6} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \quad \tilde{B}_{x,C}(\theta_1, \theta_2) = \frac{2ih}{3} \sin \frac{\theta_1}{2} \cos \frac{\theta_2}{2}, \end{aligned}$$

respectively.

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

$$\tilde{B}_y(\theta_1, \theta_2)^T = [\tilde{B}_{x,N}(\theta_2, \theta_1); \tilde{B}_{x,Y}(\theta_2, \theta_1); \tilde{B}_{x,X}(\theta_2, \theta_1); \tilde{B}_{x,C}(\theta_2, \theta_1)].$$

Remark 3. Rodrigo et al⁵³ presented a framework for LFA for edge-based discretizations on triangular grids. They consider an expression for the Fourier transform in a nonorthogonal coordinate system in space and frequency variables to adapt to arbitrary structured triangular meshes. This idea can be applied to the discretization considered here. However, for the triangle mesh considered here, it is not necessary to use nonorthogonal coordinates. In the framework of LFA provided here, we use a different Fourier basis to be consistent with the different types of stencil located on different types of grid-points, which simplifies the calculation.

3.4 | LFA representation of grid-transfer operators

Here, we use the standard finite-element interpolation operators and their transposes for restriction. In the following, we discuss the stencils and symbols of these restriction and interpolation operators.

To derive symbols for the grid-transfer operators, we first consider an arbitrary restriction operator characterized by a constant coefficient stencil $R_h \hat{=} [r_\kappa]$. Then, an infinite grid function $w_h : \mathbf{G}_h^1 \rightarrow \mathbb{R}$ (or \mathbb{C}) is transferred to the coarse grid, \mathbf{G}_{2h}^1 , in the following way

$$(R_h w_h)(\mathbf{x}) = \sum_{\kappa \in V} r_\kappa w_h(\mathbf{x} + \kappa h) \quad (\mathbf{x} \in \mathbf{G}_{2h}^1).$$

Taking w_h to be the Fourier mode, $\varphi(\theta^\alpha, \mathbf{x}) = e^{i\theta^\alpha \cdot \mathbf{x}/h}$, we have

$$(R_h \varphi(\theta^\alpha, \cdot))(\mathbf{x}) = \tilde{R}_h(\theta^\alpha) \varphi_{2h}(2\theta^{(0,0)}, \mathbf{x}) \quad (\mathbf{x} \in \mathbf{G}_{2h}^1), \quad (12)$$

with $\tilde{R}_h(\theta^\alpha) = \sum_{\kappa \in V} r_\kappa e^{i\kappa \cdot \theta^\alpha}$, which is called the symbol of R_h . However, since we consider discretizations on staggered meshes, where different “types” of variables interact in the interpolation and restriction operators, the symbol definition for the restriction operator acting on \mathbf{G}_{2h}^j , where $j = 2, 3, 4$, must be modified.

Similarly to the stencils of K_h , the restriction operator for the components of velocity can also be decomposed based on the partitioning of the DoFs associated with the N -, X -, Y -, and C -type meshpoints. Each of these restriction operators connects between all four types of meshpoints, and we partition each restriction operator into four blocks based on the DoFs. For $\mathbf{x} \in \mathbf{G}_{2h}^j$, where $j = 1, \dots, 4$, by standard calculation,⁵⁰ Equation (12) becomes

$$(R_h \varphi(\theta^\alpha, \cdot))(\mathbf{x}) = \sum_{\kappa \in V} r_\kappa e^{i\kappa \cdot \theta^\alpha} e^{i\alpha \cdot \pi \mathbf{x}/h} \varphi_{2h}(2\theta^{(0,0)}, \mathbf{x}).$$

While $e^{i\alpha \cdot \pi \mathbf{x}/h}$ appears in the above formulation, it serves only to indicate which type of DoF R_h is acting on, since

$$e^{i\alpha \cdot \pi \mathbf{x}/h} = \begin{cases} 1, & \text{for } \mathbf{x} \in \mathbf{G}_{2h}^1, \\ (-1)^{\alpha_1}, & \text{for } \mathbf{x} \in \mathbf{G}_{2h}^2, \\ (-1)^{\alpha_2}, & \text{for } \mathbf{x} \in \mathbf{G}_{2h}^3, \\ (-1)^{\alpha_1}(-1)^{\alpha_2}, & \text{for } \mathbf{x} \in \mathbf{G}_{2h}^4. \end{cases}$$

Thus, it is natural to give the following general definition of a restriction symbol on a staggered mesh.

Definition 4. We call $\tilde{R}(\theta^\alpha) = \sum_{\kappa \in V} r_\kappa e^{i\kappa \cdot \theta^\alpha} e^{i\alpha \cdot \pi \mathbf{x}/h}$ the restriction symbol of R_h .

We emphasize that we must first split the restriction operator into the different types of DoFs that it restricts from and to before we can apply Definition 4.

We first consider restriction to the N -type DoFs of a P_2 function, which can be split into four blocks,

$$R_{v,N} = \begin{bmatrix} R_{N,N} & R_{N,X} & R_{N,Y} & R_{N,C} \end{bmatrix}.$$

The N -to- N connection is

$$R_{N,N} = \begin{bmatrix} 1 & \star \end{bmatrix},$$

where the \star denotes the position (on the coarse grid) at which the discrete operator is applied. From Definition 4, $\tilde{R}_{N,N} = 1$. The X -to- N connections yield the stencil

$$R_{N,X} = \frac{1}{8} \begin{bmatrix} -1 & & & -1 \\ -1 & 3 & \star & 3 & -1 \\ & & -1 & & -1 \end{bmatrix}.$$

By standard calculation, we have

$$\tilde{R}_{N,X} = \frac{1}{4} \left(3 \cos \frac{\theta_1}{2} - \cos \frac{3\theta_1}{2} - \cos \frac{\theta_1}{2} \cos \theta_2 + \sin \frac{\theta_1}{2} \sin \theta_2 - \cos \frac{3\theta_1}{2} \cos \theta_2 - \sin \frac{3\theta_1}{2} \sin \theta_2 \right).$$

Similarly, the Y -to- N connection has the stencil

$$R_{N,Y} = \frac{1}{8} \begin{bmatrix} -1 & -1 & & \\ & 3 & -1 & \\ & \star & & \\ -1 & 3 & & \\ & -1 & -1 & \end{bmatrix},$$

with its symbol

$$\tilde{R}_{N,Y} = \frac{1}{4} \left(3 \cos \frac{\theta_2}{2} - \cos \frac{3\theta_2}{2} - \cos \theta_1 \cos \frac{\theta_2}{2} + \sin \theta_1 \sin \frac{\theta_2}{2} - \cos \theta_1 \cos \frac{3\theta_2}{2} - \sin \theta_1 \sin \frac{3\theta_2}{2} \right).$$

The C -to- N connection has the stencil

$$R_{N,C} = \frac{1}{8} \begin{bmatrix} -1 & -1 & & & \\ -1 & 3 & & & \\ & & \star & & \\ & & & 3 & -1 \\ & & & -1 & -1 \end{bmatrix},$$

with its symbol

$$\begin{aligned} \tilde{R}_{N,C} = \frac{1}{8} & \left(3 \cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} - 3 \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2} - \cos \frac{3\theta_1}{2} \cos \frac{3\theta_2}{2} - \sin \frac{3\theta_1}{2} \sin \frac{3\theta_2}{2} \right. \\ & \left. - \cos \frac{3\theta_1}{2} \cos \frac{\theta_2}{2} - \sin \frac{3\theta_1}{2} \sin \frac{\theta_2}{2} - \cos \frac{\theta_1}{2} \cos \frac{3\theta_2}{2} - \sin \frac{\theta_1}{2} \sin \frac{3\theta_2}{2} \right). \end{aligned}$$

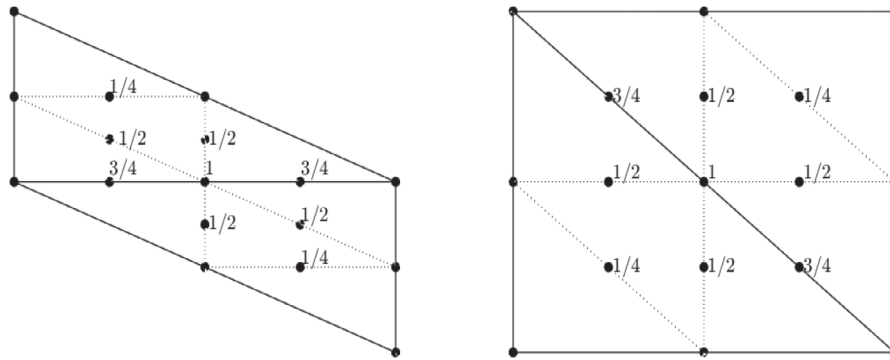


FIGURE 4 Restriction stencils for the P_2 finite-element discretization of the Laplace operator on a left triangular grid. The solid triangles denote elements from the coarse-grid and the dashed triangles are those of the fine-grid. Left: $R_{v,X}$. Right: $R_{v,C}$

*	$R_{*,N}$	$R_{*,X}$	$R_{*,Y}$	$R_{*,C}$
X	1	$\frac{3}{2} \cos \theta_1 + \frac{1}{2} (\cos \frac{\theta_1}{2} \cos \theta_2 + \sin \frac{\theta_1}{2} \sin \theta_2)$	$\cos \frac{\theta_2}{2}$	$\cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} + \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2}$
Y	1	$\cos \frac{\theta_1}{2}$	$\frac{3}{2} \cos \theta_2 + \frac{1}{2} (\cos \theta_1 \cos \frac{\theta_2}{2} + \sin \theta_1 \sin \frac{\theta_2}{2})$	$\cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} + \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2}$
C	1	$\cos \frac{\theta_1}{2}$	$\cos \frac{\theta_2}{2}$	$2 \cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} + \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2}$

TABLE 1 Symbols of $R_{v,X}$, $R_{v,Y}$, and $R_{v,C}$

The weights for the X -type and C -type restrictions for P_2 are shown in Figure 4. For the Y -type DoFs, the restriction stencil is a 90° rotation of that of X -type, so we do not include it. We use the same decomposition for $R_{v,X}$, $R_{v,Y}$, and $R_{v,C}$, and their symbols are listed in Table 1.

Following Definition 4 to account for staggering, we can write

$$\tilde{R}_v(\theta^{00}) = \begin{pmatrix} \tilde{R}_{v,N}(\theta^{00}) \\ \tilde{R}_{v,X}(\theta^{00}) \\ \tilde{R}_{v,Y}(\theta^{00}) \\ \tilde{R}_{v,C}(\theta^{00}) \end{pmatrix}, \quad \tilde{R}_v(\theta^{10}) = \begin{pmatrix} \tilde{R}_{v,N}(\theta^{10}) \\ -\tilde{R}_{v,X}(\theta^{10}) \\ \tilde{R}_{v,Y}(\theta^{10}) \\ -\tilde{R}_{v,C}(\theta^{10}) \end{pmatrix}, \quad \tilde{R}_v(\theta^{01}) = \begin{pmatrix} \tilde{R}_{v,N}(\theta^{01}) \\ \tilde{R}_{v,X}(\theta^{01}) \\ -\tilde{R}_{v,Y}(\theta^{01}) \\ -\tilde{R}_{v,C}(\theta^{01}) \end{pmatrix}, \quad \tilde{R}_v(\theta^{11}) = \begin{pmatrix} \tilde{R}_{v,N}(\theta^{11}) \\ -\tilde{R}_{v,X}(\theta^{11}) \\ -\tilde{R}_{v,Y}(\theta^{11}) \\ \tilde{R}_{v,C}(\theta^{11}) \end{pmatrix}.$$

Interpolation, P_v , is taken to be the transpose of R_v , with symbol $\tilde{P}_v(\theta^\alpha) = \frac{1}{4} \tilde{R}_v^T(\theta^\alpha)$.

All of the pressure DoFs are located on \mathbf{G}_h^1 . Thus, the restriction operator for pressure acts only on one type of grid point, the nodes of the mesh. The stencil of the restriction operator for pressure is given by

$$R_p = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 2\star & 1 \\ & 1 & 1 \end{bmatrix},$$

with its symbol

$$\tilde{R}_p(\theta_1, \theta_2) = 1 + \cos \theta_1 + \cos \theta_2 + \cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2.$$

The interpolation, P_p , for pressure is taken to be the transpose of R_p , with symbol $\tilde{P}_p(\theta_1, \theta_2) = \frac{1}{4} \tilde{R}(\theta_1, \theta_2)_p^T$.

Finally, the restriction operator for the Stokes system can be written as

$$R_h = \begin{bmatrix} R_v & 0 & 0 \\ 0 & R_v & 0 \\ 0 & 0 & R_p \end{bmatrix},$$

with its symbol (over all four harmonics) being a 9×36 matrix. For the interpolation operator, P_h , we consider the transpose of R_h , and the symbols satisfy

$$\tilde{P}_h(\theta) = \frac{1}{4} \tilde{R}_h^T(\theta).$$

For the coarse-grid operator, K_H , we consider rediscritization, which is equivalent to Galerkin coarsening.

3.5 | LFA for additive Vanka relaxation

When applying LFA to overlapping multiplicative Vanka relaxation, multiple Fourier representations are required for each variable in a given patch to account for the intermediate “stages” in the relaxation.³² This is not needed in the additive case considered here. Unlike classical relaxation, additive Vanka relaxation is a block relaxation scheme. Since the variables on each block are updated at the same time, based on the same block system, we can apply LFA ideas, but must modify the standard LFA to include a block Fourier basis to represent all the information in each block. By Fourier transformation the operator M_h^{-1} , defined on an infinite mesh, can be block diagonalized (by appropriately ordering the unknowns).

Recall $M_h^{-1} = \sum_{i=1}^N V_i^T D_i K_i^{-1} V_i$. Under the Fourier ansatz, D_i , V_i and K_i have the same representation for all i . Note that since D_i is a diagonal scaling matrix, its Fourier representation is itself. Similarly, V_i is a projector, and its symbol is itself. Note that the representation of K_i^{-1} is equal to the inverse of the representation of K_i . Thus, we only need to consider the representation of K_i . Assume that the set of grid points corresponding to the DoFs of K_i is $\Xi_i = \{\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_m^{(i)}\} \subseteq \mathbf{G}_h$ and assume the ordering of the DoFs in K_i is consistent with the ordering of points in Ξ_i . Note that, due to the overlap between subdomains in the Schwarz relaxation, Ξ_i might contain multiple points with the same Fourier representation in the symbol of (11), but we treat them separately in the representation of K_i , since these points correspond to different unknowns in \mathfrak{S}_i . Let

$$\varpi = \text{span} \left\{ \psi_j(\theta) = e^{i\theta \cdot \mathbf{x}_j^{(i)}/h} \cdot \chi_j, j = 1, \dots, m \right\},$$

where χ_j is an $m \times 1$ vector with only one nonzero element with value 1 located in the j th position.

Let us consider the action of K_i on this Fourier basis on Ξ_i , defining the symbol, \tilde{K}_i , so that

$$K_i \psi(\theta) = \tilde{K}_i \psi(\theta), \quad \forall \psi \in \varpi. \quad (13)$$

Let Φ be an $m \times m$ diagonal matrix with diagonal elements, $e^{i\theta \cdot \mathbf{x}_j^{(i)}/h}$ for $j = 1, 2, \dots, m$, and $\mathbf{x}_j^{(i)} \in \Xi_i$, and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ be an arbitrary vector. Then, (13) is equivalent to

$$K_i \sum_{j=1}^m \alpha_j \psi_j = K_i \Phi \alpha = \Phi \tilde{K}_i \alpha. \quad (14)$$

Thus, from (14), the Fourier representation of K_i is

$$\tilde{K}_i = \Phi^{-1} K_i \Phi = \Phi^T K_i \Phi. \quad (15)$$

Note that, under the Fourier ansatz, all of these matrices are independent of i except for our construction of Φ . However, due to the special structure of Φ , we can use relative values of nodal position to replace the matrix Φ in (15) by a simple matrix (called the relative Fourier matrix), scaled by $e^{i\theta \cdot \mathbf{x}_s^{(i)}/h}$, for some $\mathbf{x}_s^{(i)} \in \Xi_i$. This is equivalent to considering the Fourier basis acting on a local offset, $\mathbf{x}_j^{(i)} := \mathbf{x}_j^{(i)} - \mathbf{x}_s^{(i)}$ for some fixed point $\mathbf{x}_s^{(i)}$ in the patch, usually the pressure node. As this scaling simplifies the calculation, we will use the relative Fourier matrix in the rest of the article.

3.6 | LFA for additive Vanka relaxation for the Stokes equations

Now, we consider LFA for two types of Vanka relaxation schemes that differ only in the choice of the relaxation blocks. First, for each pressure DoF, we consider patches containing all velocity DoFs included in a hexagon centered at the node associated with the pressure, see left of Figure 5 for the $P_2 - P_1$ discretization. This patch is the smallest one that contains

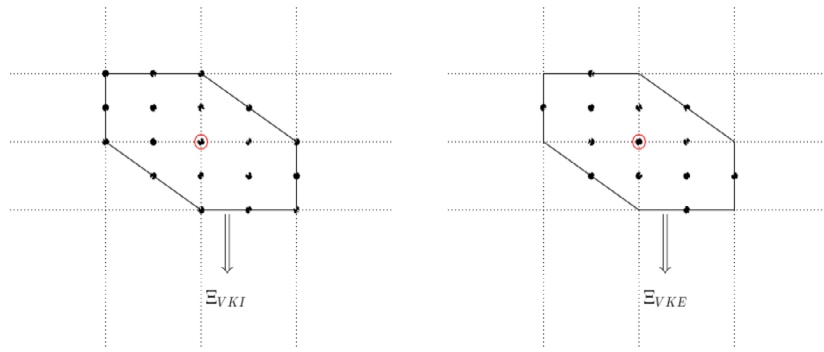


FIGURE 5 Left: Vanka-inclusive patch for the $P_2 - P_1$ discretization. Right: Vanka-exclusive patch for the $P_2 - P_1$ discretization

Patch:	$P_2 - P_1$		$Q_2 - Q_1$	
#	Ξ_{VKI}	Ξ_{VKE}	Ξ_{VKI}	Ξ_{VKE}
Each component of velocity				
N	7	1	9	1
X	4	4	6	6
Y	4	4	6	6
C	4	4	4	4
Pressure				
N	1	1	1	1
Total	39	27	51	35

TABLE 2 The number (#) of different type of DoFs for Vanka-inclusive and Vanka-exclusive patches for the $P_2 - P_1$ and $Q_2 - Q_1$ discretizations

all velocity DoFs with connections to this pressure DoF in the symbolic nonzero pattern of the matrix. For this reason, we refer to this patch construction as Vanka-inclusive (VKI). The number of DoFs in this patch for $P_2 - P_1$ is 39. To be specific, there are seven N -type points, four points for X -, Y -, and C -type, respectively, for each of the two components of the velocity, and one N -type point for the central pressure (see Table 2). Since the patches and submatrices K_i are the same, it is natural to consider D_i to be the same for all i . For D_i in (6), a simple idea is to take D_i to be the identity, which we refer to as using *no weights*. Another choice for D_i is to take d_{m_i} in D_i to be the reciprocal of the number of patches that each type of DoF appears in. We refer to this as using *geometric weights* (or *natural weights*), and will denote this by VKIW in the results to follow. To be specific, the weights for the velocity DoFs are $1/7$, $1/4$, $1/4$, and $1/4$ for the N -, X -, Y -, and C -type points, respectively. For pressure, the natural weight is 1. Thus, D_i is a 39×39 matrix given by

$$D_i = \begin{pmatrix} D_v & 0 & 0 \\ 0 & D_v & 0 \\ 0 & 0 & D_p \end{pmatrix}, \quad \text{with } D_v = \begin{pmatrix} D_N & 0 & 0 & 0 \\ 0 & D_X & 0 & 0 \\ 0 & 0 & D_Y & 0 \\ 0 & 0 & 0 & D_C \end{pmatrix}, \quad D_p = 1,$$

where $D_N = \frac{1}{7}I_{7 \times 7}$ and $D_X = D_Y = D_C = \frac{1}{4}I_{4 \times 4}$. Since \tilde{K}_h is a 9×9 matrix, the symbol of V_i is a 39×9 projection matrix. Finally, \tilde{K}_i is a 39×39 matrix, as is D_i .

Now, we give more details about how to calculate the relative Fourier matrix, Φ . For $P_2 - P_1$, with patches Ξ_{VKI} as shown at the left of Figure 5, we take $x_s^{(i)}$ be the node located directly below the central pressure DoF. We use lexicographical order to order the remaining points in Ξ_{VKI} for each type of point (N -, X -, Y -, and C -ordering), that is, from left to right and bottom to top. Then,

$$\Phi(\theta_1, \theta_2) = \begin{pmatrix} \Phi_v & 0 & 0 \\ 0 & \Phi_v & 0 \\ 0 & 0 & \Phi_p \end{pmatrix}, \quad (16)$$

where

$$\begin{aligned}\Phi_n &= [1, e^{i\theta_1}, e^{-i\theta_1+i\theta_2}, e^{i\theta_2}, e^{i\theta_1+i\theta_2}, e^{-i\theta_1+i2\theta_2}, e^{i2\theta_2}], \\ \Phi_x &= [e^{i/2\theta_1}, e^{-i/2\theta_1+i\theta_2}, e^{i/2\theta_1+i\theta_2}, e^{-i/2\theta_1+i2\theta_2}], \\ \Phi_y &= [e^{i/2\theta_2}, e^{i\theta_1+i/2\theta_2}, e^{-i\theta_1+i3/2\theta_2}, e^{i3/2\theta_2}], \\ \Phi_c &= [e^{-i/2\theta_1+i/2\theta_2}, e^{i/2\theta_1+i/2\theta_2}, e^{-i/2\theta_1+i3/2\theta_2}, e^{i/2\theta_1+i3/2\theta_2}], \\ \Phi_0 &= [\Phi_n, \Phi_x, \Phi_y, \Phi_c], \\ \Phi_v &= \text{diag}(\Phi_0), \\ \Phi_p &= e^{i\theta_2}.\end{aligned}$$

Remark 4. Note that matrix Φ in (16) is only a function of frequency, $\theta = (\theta_1, \theta_2)$, and is independent of the meshsize, h , and the index i .

For V_i , mapping the global vector to the vector on the patch, we only need to account for the duplication that arises from the representation of the global Fourier basis of dimension 9 to the local block representation. Here, the order follows N -type, X -type, Y -type, then C -type, and velocity is first, then followed by the pressure. The structure is

$$\tilde{V}_i = \begin{pmatrix} V_v & 0 & 0 \\ 0 & V_v & 0 \\ 0 & 0 & V_p \end{pmatrix},$$

where $V_p = 1$ and V_v is a 19×4 matrix defined as follows

$$V_v = \begin{pmatrix} I_N & 0 & 0 & 0 \\ 0 & I_X & 0 & 0 \\ 0 & 0 & I_Y & 0 \\ 0 & 0 & 0 & I_C \end{pmatrix} \quad (17)$$

where I_N is the 7×1 vector of all ones, and $I_X = I_Y = I_C$ are the 4×1 vector of all ones.

Considering the stencils in Figure 3, we note that of the adjacent nodal velocity DoFs, only the central one appears in the gradient operator. This motivates construction of a second patch that contains all of the DoFs in VKI except for the N -type velocity DoFs at the edges of the VKI patch, see the right of Figure 5. We refer to this as Vanka-exclusive (VKE) when used with no weights ($D_i = I$). Table 2 presents the number of the four types of DoFs for VKE, which contains $39 - 2 \times 6 = 27$ DoFs in total. Another choice for D_i is to use the natural weights, which we refer to as Vanka-exclusive with natural weights (VKEW). Here, the weights are 1, $1/4$, $1/4$, and $1/4$ for N -, X -, Y -, and C -type velocity DoFs, respectively, and 1 for pressure. Similarly to VKI, we can calculate the matrix Φ whose size is 27×27 following (16), replacing Φ_n by $\Phi_n = e^{i\theta_2}$. For \tilde{V}_i , there is only 1 N -type point for velocity, so we only need to modify I_N to be the scalar 1 in (17), then we obtain \tilde{V}_i , which is a 27×9 matrix. In each of these cases, the overall symbol for relaxation is a 9×9 matrix. Since we consider four harmonics, following Subsection 3.2, \tilde{E}_h is a 36×36 matrix. In the rest of this article, we omit the subscript h unless it is necessary to avoid confusion.

For the $Q_2 - Q_1$ discretization of the Stokes equations, the patches of Vanka-inclusive and Vanka-exclusive are a little different than the structure with the $P_2 - P_1$ discretizations. Figure 6 shows the Vanka-inclusive and Vanka-exclusive patches for the $Q_2 - Q_1$ discretization, and Table 2 lists the details. For VKI, the numbers of unknowns for the velocity are 9, 6, 6, and 4 for N -, X -, Y -, and C -types, respectively. In total, there are $2(9 + 6 + 6 + 4) + 1 = 51$ DoFs in one patch. To construct V_v , we only need to change I_N in (17) to be the 9×1 vector of all ones, and $I_X = I_Y$ to be the 6×1 vectors of all ones. As in the $P_2 - P_1$ case, the $Q_2 - Q_1$ gradient operator on a uniform mesh uses only the central nodal velocity DoF. Thus, the second patch contains all of the DoFs in VKI except for the N -type points at the edges of the block, giving $51 - 2 \times 8 = 35$ DoFs. To obtain the representation of V_i using VKE for $Q_2 - Q_1$ from the construction of V_v for VKI with the $Q_2 - Q_1$ discretization, we only need to change I_N to a scalar 1.

We again use the relative Fourier matrix, Φ , to transform, the block matrix, K_i . We set $\mathbf{x}_s^{(i)}$ to be the lower-left corner at the block, giving,

$$\Phi(\theta_1, \theta_2) = \begin{pmatrix} \Phi_{Qv} & 0 & 0 \\ 0 & \Phi_{Qv} & 0 \\ 0 & 0 & \Phi_{Qp} \end{pmatrix}, \quad (18)$$

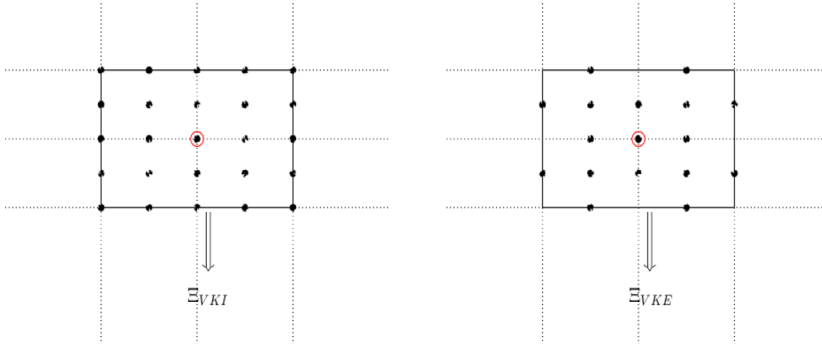


FIGURE 6 Left: Vanka-inclusive patch for the $Q_2 - Q_1$ discretization. Right: Vanka-exclusive patch for the $Q_2 - Q_1$ discretization

where

$$\begin{aligned}
 \Phi_{Qn} &= [1, e^{i\theta_1}, e^{i2\theta_1}, e^{i\theta_2}, e^{i\theta_1+i\theta_2}, e^{i2\theta_1+i\theta_2}, e^{i2\theta_2}, e^{i\theta_1+i2\theta_2}, e^{i2\theta_1+i2\theta_2}], \\
 \Phi_{Qx} &= [e^{i/2\theta_1}, e^{i3/2\theta_1}, e^{i/2\theta_1+i\theta_2}, e^{i3/2\theta_1+i\theta_2}, e^{i/2\theta_1+i2\theta_2}, e^{i3/2\theta_1+i2\theta_2}], \\
 \Phi_{Qy} &= [e^{i/2\theta_2}, e^{i\theta_1+i/2\theta_2}, e^{i2\theta_1+i/2\theta_2}, e^{i3/2\theta_2}, e^{i\theta_1+i3/2\theta_2}, e^{i2\theta_1+i3/2\theta_2}], \\
 \Phi_{Qc} &= [e^{i/2\theta_1+i/2\theta_2}, e^{i3/2\theta_1+i/2\theta_2}, e^{i/2\theta_1+i3/2\theta_2}, e^{i3/2\theta_1+i3/2\theta_2}], \\
 \Phi_{Q0} &= [\Phi_{Qn}, \Phi_{Qx}, \Phi_{Qy}, \Phi_{Qc}], \\
 \Phi_{Qv} &= \text{diag}(\Phi_{Q0}), \\
 \Phi_{Qp} &= e^{i\theta_1+i\theta_2}.
 \end{aligned}$$

For VKE with $Q_2 - Q_1$, we truncate Φ_{Qn} to the scalar $e^{i\theta_1+i\theta_2}$.

4 | PARAMETER CHOICE AND VALIDATION

As a relaxation scheme, we consider the Chebyshev iteration⁴⁶ on K_h preconditioned with overlapping additive Vanka, with the two patches discussed before. We note that this is the natural extension of weighted Jacobi to block relaxation when considering more than a single relaxation sweep per level in a two-grid cycle. The key point to tuning the Chebyshev iteration is the choice of the lower and upper bounds for the interval that determines the Chebyshev polynomials. LFA is useful here, and we employ it to find optimal bounds for different degrees of Chebyshev polynomials. Relaxation using both no weights and natural weights will be considered. As a special case, we also consider a simple preconditioned Richardson relaxation, for the cases $\nu_1 + \nu_2 = 1$ and $\nu_1 + \nu_2 = 2$, where we again use LFA to help find the optimal weights. The goal of these experiments is to use LFA to determine “best practices” in terms of how to choose patches and weights, with relaxation parameters optimized for these choices. Finally, we compare the cost and performance among these approaches.

In practice, the LFA two-grid convergence factors often exactly match the true convergence factor of multigrid applied to a problem with periodic boundary conditions.^{43,54} For the case of Dirichlet boundary conditions, a gap between the LFA predictions and the actual performance is sometimes observed.^{10,21,22} In order to see the influence of boundary conditions on multigrid performance, we present data for both Dirichlet and periodic boundary conditions. Our tests are implemented using Firedrake and PETSc³⁷ for both the $P_2 - P_1$ and $Q_2 - Q_1$ discretizations. The subproblems associated with K_i are solved by LU decomposition. We focus on optimizing the interval used for Chebyshev relaxation with symmetric pre- and post-relaxation.

4.1 | Numerical results for the $P_2 - P_1$ discretization

4.1.1 | No weights

First, we consider $D_i = I$ and the two different patches discussed above. The two-grid error iteration matrix is

$$E_k = p_k(T) \mathcal{M}^{\text{CGC}} p_k(T), \quad (19)$$

TABLE 3 Two-grid LFA predictions for Chebyshev Vanka relaxation and two-grid performance with periodic ($\hat{\rho}$) and Dirichlet ($\hat{\rho}^D$) boundary conditions for the $P_2 - P_1$ discretization for VKI

LFA predictions		$h = \frac{1}{20}$			$h = \frac{1}{40}$			$h = \frac{1}{80}$		
k	$[\alpha, \beta]_{\text{VKI}}$	ρ_{VKI}	$\hat{\rho}_{\text{VKI}}$	$\hat{\rho}_{\text{VKI}}^D$	$\hat{\rho}_{\text{VKI}}$	$\hat{\rho}_{\text{VKI}}^D$	$\hat{\rho}_{\text{VKI}}$	$\hat{\rho}_{\text{VKI}}^D$	$\hat{\rho}_{\text{VKI}}$	$\hat{\rho}_{\text{VKI}}^D$
1	[0.1, 8.3]	0.672	0.672	0.699	0.671	0.699	0.671	0.699	0.671	0.699
2	[0.9, 7.8]	0.295	0.296	0.585	0.296	0.585	0.296	0.585	0.296	0.585
3	[0.9, 7.9]	0.120	0.107*	0.148	0.133*	0.148	0.143*	0.148	0.143*	0.148
4	[1.4, 7.2]	0.102	0.102	0.133	0.102	0.133	0.102	0.133	0.102	0.133
5	[1.3, 7.4]	0.051	0.050	0.074	0.049	0.074	0.047	0.074	0.047	0.074

Note: Optimal intervals for different k with no weights. An alternating convergence pattern was observed, so the reported convergence factor is averaged over the final seven iterations before convergence (denoted *).

TABLE 4 Two-grid LFA predictions for Chebyshev Vanka relaxation and two-grid performance with periodic ($\hat{\rho}$) and Dirichlet ($\hat{\rho}^D$) boundary conditions for the $P_2 - P_1$ discretization for VKE

LFA predictions		$h = \frac{1}{20}$			$h = \frac{1}{40}$			$h = \frac{1}{80}$		
k	$[\alpha, \beta]_{\text{VKE}}$	ρ_{VKE}	$\hat{\rho}_{\text{VKE}}$	$\hat{\rho}_{\text{VKE}}^D$	$\hat{\rho}_{\text{VKE}}$	$\hat{\rho}_{\text{VKE}}^D$	$\hat{\rho}_{\text{VKE}}$	$\hat{\rho}_{\text{VKE}}^D$	$\hat{\rho}_{\text{VKE}}$	$\hat{\rho}_{\text{VKE}}^D$
1	[0.3, 6.0]	0.475	0.476	0.571	0.491	0.571	0.475	0.571	0.475	0.571
2	[0.5, 4.7]	0.440	0.412	0.392	0.485	0.419	0.388	0.463	0.388	0.463
3	[1.2, 4.6]	0.168	0.169	0.175	0.169	0.176	0.168	0.176	0.168	0.176
4	[1.6, 4.6]	0.127	0.125	0.124	0.128	0.127	0.127	0.127	0.127	0.127
5	[2.6, 3.7]	0.112	0.111	0.108	0.111	0.111	0.112	0.111	0.112	0.111

Note: Optimal intervals for different k with no weights.

where $T = M_h^{-1}K_h$ and p_k is the Chebyshev polynomial with degree k on a given interval. Below, we take ρ to be the LFA prediction sampled at 32 equispaced points in each dimension of the Fourier domain. (When sampling the frequency space at more points, the predictions go up only slightly.) We take $\hat{\rho}$ to be the measured convergence factor per iteration defined by $\hat{\rho}_j = \frac{\|e_j\|}{\|e_{j-1}\|}$, where $e_j = b_h - K_h u_j$, measured for problem (3) with zero right-hand side and random initial guess at the iteration when the residual norm is first below 10^{-150} . Since the discretized problem is singular, we project the approximate solution after each iteration to ensure that it remains orthogonal to the null space.

In order to find the optimal intervals for the Chebyshev iteration for different k , we use a brute-force search with stepsize 0.1 to optimize the intervals with LFA. The optimal interval, denoted as $[\alpha, \beta]$, and the corresponding convergence factors are given in the following tables. Tables 3 and 4 give the measured convergence for periodic and Dirichlet boundary conditions versus the LFA predictions for different k and the corresponding optimal intervals with $h = \frac{1}{20}, \frac{1}{40}$, and $\frac{1}{80}$. We see that the measured convergence factors are largely independent of the meshsize, h . A good match is generally seen between the LFA predictions and the measured convergence factors, with only a tiny gap between the LFA prediction and the measured convergence factor. It is reasonable that such a gap exists for the Dirichlet case, since the LFA prediction is of the expected asymptotic convergence factor for the problem with periodic boundary conditions. This might also suggest that extra work is needed to reduce this gap.¹⁰ Note that only for $k = 2$ and VKI is this gap significant. We have also tested some other intervals for this case, seeing that the gap between the LFA prediction and the measured results with Dirichlet boundary conditions is quite variable, with some parameter values coming close to the LFA prediction, but never achieving a good agreement. For VKI with $k = 3$ and periodic boundary conditions, the per iteration measured convergence factor oscillates, but the long-term average still approaches that given by LFA. For these cases, we present averaged convergence over the final seven iterations. These tables indicate that when $k = 1$, VKE is more effective than VKI; however, for $k > 1$, the opposite occurs.

Remark 5. In the numerical tests, we see that for some cases, the optimal pair of $[\alpha, \beta]$ are not unique. We break such ties arbitrarily.

In Figure 7, we show the LFA amplification factors using Chebyshev Vanka relaxation with $k = 1$ and the parameters from Tables 3 and 4 in (19) for the $P_2 - P_1$ discretization. We see that the VKE reduces the high-frequency error faster

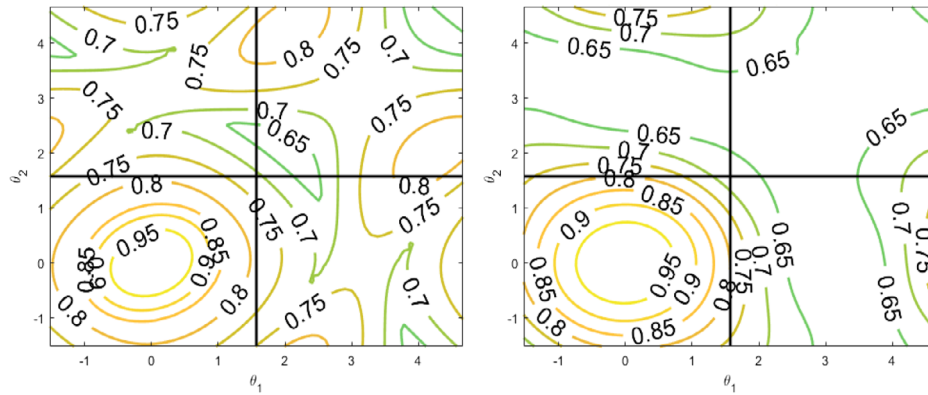


FIGURE 7 The LFA amplification factors for Vanka-relaxation with $k = 1$ for the $P_2 - P_1$ discretization. Left: VKI with $k = 1$. Right: VKE with $k = 1$

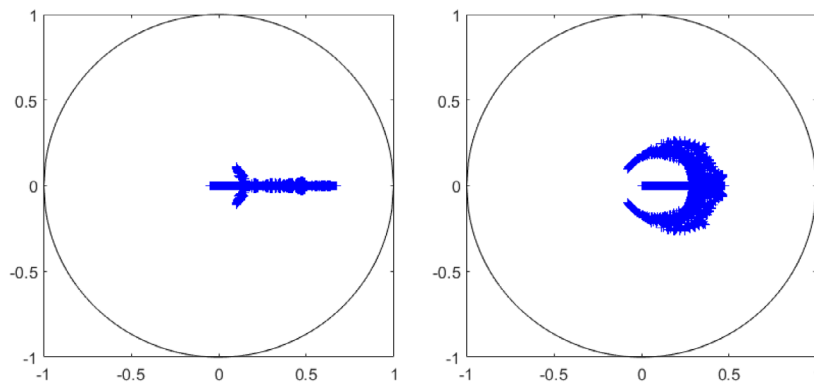


FIGURE 8 The LFA-predicted spectra of the two-grid error-propagation operators for the $P_2 - P_1$ discretization. Left: VKI with $k = 1$. Right: VKE with $k = 1$

TABLE 5 Two-grid LFA predictions for Chebyshev Vanka relaxation and multigrid performance with periodic ($\hat{\rho}$) and Dirichlet ($\hat{\rho}^D$) boundary conditions for the $P_2 - P_1$ discretization

k	$[\alpha, \beta]_{\text{VKIW}}$	ρ_{VKIW}	$\hat{\rho}_{\text{VKIW}}$	$\hat{\rho}_{\text{VKIW}}^D$	$[\alpha, \beta]_{\text{VKEW}}$	ρ_{VKEW}	$\hat{\rho}_{\text{VKEW}}$	$\hat{\rho}_{\text{VKEW}}^D$
1	[0.9, 2.9]	0.518	0.518	0.517	[1.3, 4.0]	0.584	0.584	0.589
2	[1.1, 1.7]	0.196	0.197	0.196	[0.5, 3.5]	0.376	0.426	0.279
3	[1.4, 2.0]	0.106	0.126	0.103	[1.3, 3.6]	0.233	0.234	0.232
4	[1.8, 2.2]	0.085	0.085	0.085	[2.0, 3.5]	0.149	0.149	0.148
5	[1.3, 1.8]	0.070	0.071	0.069	[2.5, 3.5]	0.108	0.107	0.108

Note: Optimal intervals for different k with natural weights. $h = \frac{1}{40}$.

than VKI, and both reduce the low-frequency error slowly. In Figure 8, we present the spectrum of the associated two-grid error-propagation operators. The distribution of the eigenvalues is notably different: For VKI, most of the eigenvalues are real, while, for VKE, the eigenvalues are clustered around a circle in the complex plane.

4.1.2 | Geometric weights

As seen in Tables 3 and 4, the measured convergence factors are largely independent of meshsize. Thus, we only consider $h = \frac{1}{40}$ in the following tests. Table 5 gives the measured convergence versus LFA predictions for relaxation with natural weights and the corresponding optimal intervals. We see a good agreement between the LFA predictions and the measured convergence factors. For the case of Dirichlet boundary conditions, the LFA predictions match the measured convergence factors very well except for VKEW with $k = 2$. Comparing Tables 3 and 4 with Table 5, the big difference is that, for all k , VKIW outperforms VKEW. Moreover, for the inclusive patch, VKIW performs better than VKI. However, this is not true for the exclusive patch, where we see VKE has better convergence factors than VKEW.

TABLE 6 Two-grid LFA predictions, $\rho^{(v_1, v_2)}$, for Richardson relaxation optimizing outer weights for the $P_2 - P_1$ discretization

Method	ω_{opt}	$\rho^{(1,0)}$	$\rho^{(1,1)}$	$\rho^{(2,2)}$	$\rho^{(3,3)}$	$\rho^{(4,4)}$	$\rho^{(5,5)}$	$(\omega_{1,\text{opt}}, \omega_{2,\text{opt}})$	$\rho^{(1,1)}$
VKI	0.24	0.819	0.670	0.509	0.411	0.334	0.273	(0.14, 0.50)	0.556
VKIW	0.78	0.587	1.061	0.337	0.386	0.107	0.106	(0.16, 0.84)	0.507
VKE	0.36	0.669	0.497	0.638	0.298	0.126	0.149	(0.22, 0.56)	0.356
VKEW	0.68	0.574	1.507	0.591	0.919	0.292	0.473	(0.00, 0.68)	0.574

4.1.3 | Optimized weights

From the above results, we see that LFA provides a good prediction for the actual two-grid performance, especially for the periodic problem. We also see that using geometric weights can improve performance. Motivated by this, we now consider whether using different weights for each different type of DoF within the relaxation scheme can improve performance. In this subsection, we apply LFA to optimize such weights. Here, we consider a preconditioned Richardson iteration, with corresponding two-grid error propagation operator

$$E_R = (I - \omega_2 M_h^{-1} K_h)^{v_2} \mathcal{M}^{\text{CGC}} (I - \omega_1 M_h^{-1} K_h)^{v_1}. \quad (20)$$

Our target is to optimize the corresponding convergence factor, ρ , by brute-force search or using other optimization algorithms. We also consider the effect of using different pre- and post-relaxation weights in (20).

Table 6 shows results for the $P_2 - P_1$ discretization when optimizing only the outer parameters in the Richardson relaxation. We use brute-force search with sampling points taken in steps of 0.02 on the interval $[0, 1]$ for the case of $v_1 + v_2 = 1$. Fixing this weight, we then consider the performance of symmetric cycles with $v_1 = v_2 = 1, \dots, 5$, for comparison to the results using a Chebyshev iteration presented in Tables 3 to 5. We note both that the optimization of the Chebyshev intervals is more stable than simple use of multiple steps of a preconditioned Richardson iteration with fixed weight, and that the corresponding convergence factors presented in Table 6 are generally noticeably worse than those in Tables 3 to 5, although perhaps further optimization of the Richardson weight would lead to some improvements. Furthermore, for the case of $v_1 + v_2 = 2$, we consider independent choices of ω_1 and ω_2 , using sampling points taken in steps of 0.02 on the intervals $[0, 0.5]$ and $[0.5, 0.9]$, respectively, to find the optimal results. Note that these intervals were selected based on results from a coarse sampling of wider intervals. We note that using $v_1 + v_2 = 1$ seems to be more efficient than using $v_1 + v_2 = 2$, except for the case of VKI, even when using two different weights for the pre- and post-relaxation parameters. We note that using different pre- and post-relaxation parameters gives notable improvements for VKE and VKI, but at best marginal gains for VKIW and VKEW.

Next, we consider fixing $\omega_1 = \omega_2 = 1$ in (20) and using three different weights for D_i : d_1 for N -type velocity DoFs, d_2 for X -, Y -, and C -type velocity DoFs, and d_3 for the pressure. We make use of a robust optimization algorithm designed for LFA optimization⁵⁵ to find the optimal parameters, rather than brute-force searches. For the optimal parameters, we truncate the results obtained by the robust optimization to two digits, noting that the performance is not overly sensitive to this truncation. We first optimize for a single-relaxation sweep, that is, $v_1 + v_2 = 1$. Table 7 shows that, in this setting, Vanka-inclusive achieves a convergence factor of 0.581, while Vanka-exclusive has a better convergence factor, 0.456. Both of these are significantly better than the corresponding results for VKI and VKE from Table 6, but only slightly better than VKIW and VKEW. Then, we optimize for the case of $v_1 + v_2 = 2$, showing that there is no significant improvement compared with a single relaxation. Comparing Tables 6 with 7 suggests that optimizing weights in D_i obtains better performance than doing so for the outer weights, although we note that nonsymmetric weighting of VKE in Table 6 outperforms the symmetric results in Table 7.

In order to see whether using more weights for different types of DoFs can improve the performance of Vanka relaxation, we consider fixing $\omega_1 = \omega_2 = 1$ and using four weights, d_1, d_2, d_3 , and d_4 , for the N -, X -, Y -, and C -type velocity DoFs, respectively, and one weight, d_5 , for the pressure in D_i . We again use the robust optimization algorithm.⁵⁵ Similarly, we first consider a single relaxation. The optimal weights and corresponding LFA predictions are presented in Table 8, showing that VKE achieves better performance than VKI. Then, we optimize with $v_1 + v_2 = 2$. We achieve a

Method	$d_{1,opt}$	$d_{2,opt}$	$d_{3,opt}$	ρ
VKI ($v_1 + v_2 = 1$)	0.19	0.22	0.71	0.581
VKE ($v_1 + v_2 = 1$)	0.54	0.26	0.68	0.456
VKI ($v_1 + v_2 = 2$)	0.22	0.29	0.47	0.436
VKE ($v_1 + v_2 = 2$)	0.39	0.29	0.37	0.406

TABLE 7 Two-grid LFA predictions for Richardson relaxation for the $P_2 - P_1$ discretization with three weights in D_i

Method	$d_{1,opt}$	$d_{2,opt}$	$d_{3,opt}$	$d_{4,opt}$	$d_{5,opt}$	ρ
VKI ($v_1 + v_2 = 1$)	0.19	0.21	0.17	0.35	0.74	0.571
VKE ($v_1 + v_2 = 1$)	0.49	0.25	0.24	0.30	0.68	0.452
VKI ($v_1 + v_2 = 2$)	0.22	0.28	0.28	0.30	0.51	0.415
VKE ($v_1 + v_2 = 2$)	0.43	0.33	0.30	0.32	0.36	0.408

TABLE 8 Two-grid LFA predictions for Richardson relaxation for the $P_2 - P_1$ discretization with five weights in D_i

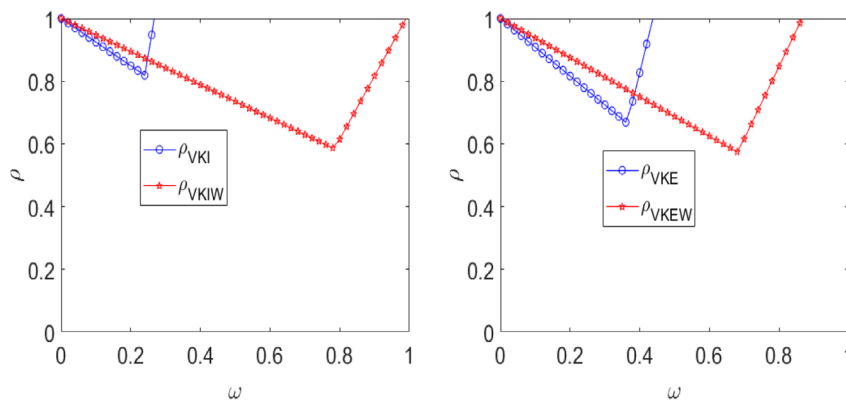


FIGURE 9 Two-grid LFA convergence factor as a function of ω for Richardson relaxation with $v_1 + v_2 = 1$. Left: Vanka-inclusive and Vanka-inclusive with natural weights. Right: Vanka-exclusive and Vanka-exclusive with natural weights

convergence factor of 0.415 for Vanka-inclusive and 0.408 for Vanka-exclusive, which are only slightly better than results using three weights. Table 8 suggests that a single relaxation is again more efficient for VKI and VKE, especially for the case of VKE. All in all, comparing Table 7 with Table 8 shows that using three weights for D_i is enough to obtain near-optimal performance. It is not necessary to use five weights.

4.1.4 | Sensitivity of convergence factors to relaxation weights

Here, we present LFA results to show the sensitivity of performance to parameter choice for Richardson relaxation with VKI, VKIW, VKE, and VKEW. In Figure 9, we show results for a single Richardson relaxation for Vanka-inclusive and Vanka-exclusive, sampling ω in steps of 0.02 on the interval $[0, 1]$. Figure 9 shows that it is better to underestimate the optimal parameter than to overestimate it. Similar behavior is seen in other works. Adler et al¹³ explored different types of Vanka as a preconditioner for GMRES for magnetohydrodynamics, showing the same preference for underestimation of the optimal parameter. In the application of Vanka relaxation for the Stokes equations with an $H(\text{div})$ conforming discretization,⁴ even when using different weights for the velocity and pressure DoFs, it also appears better to underestimate the optimal parameters. We note that for both Vanka-inclusive and Vanka-exclusive with natural weights, convergence is observed for a wider range of Richardson weights than for Vanka-inclusive and Vanka-exclusive.

In Figures 10 and 11, we present the LFA convergence factor as function of ω_1 and ω_2 , sampling in steps of 0.05 on the interval $[0, 1]$, for Richardson relaxation with $v_1 + v_2 = 2$. We see similar behavior, that performance degrades rapidly for weights that are “too large.” For Vanka-inclusive, the case with no weights is more sensitive to the outer parameters, while the opposite is seen for Vanka-exclusive.

FIGURE 10 Two-grid LFA convergence factor as a function of ω_1 and ω_2 for Richardson relaxation with $\nu_1 + \nu_2 = 2$. The white circle marks the optimal point. Left: Vanka-inclusive. Right: Vanka-inclusive with natural weights

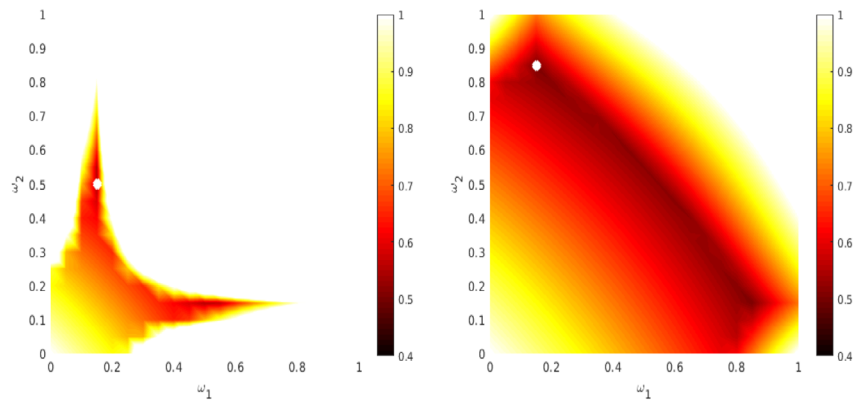


FIGURE 11 Two-grid LFA convergence factor as a function of ω_1 and ω_2 for Richardson relaxation with $\nu_1 + \nu_2 = 2$. The white circle marks the optimal points. Left: Vanka-exclusive. Right: Vanka-exclusive with natural weights

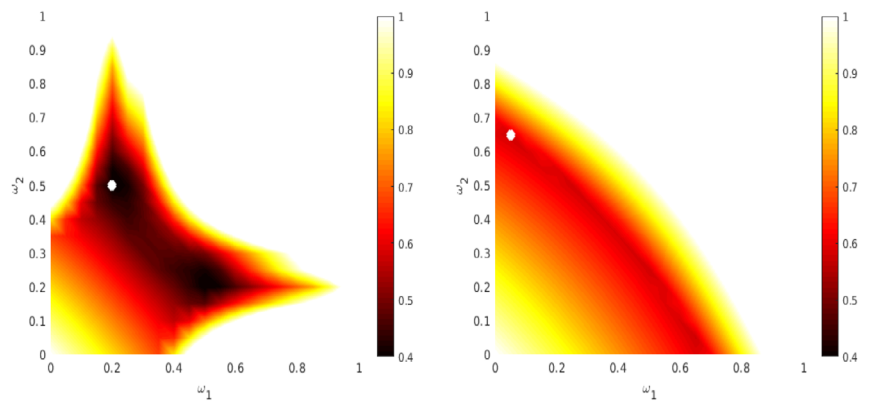
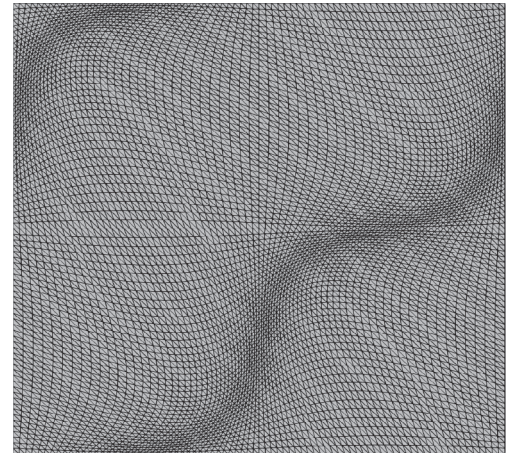


FIGURE 12 Distorted mesh with $\varepsilon = 0.1$



4.1.5 | Sensitivity of convergence factors to mesh distortion

Since local Fourier analysis applies only in the case of infinite or periodic meshes, a natural question is how accurate the predictions are in the case of distorted or otherwise nonuniform meshes. Here, we consider smoothly distorted meshes, where the coordinates of the nodes of the (uniform) mesh are mapped under the transformation $(x, y) \rightarrow (x + \varepsilon \sin(2\pi x) \sin(2\pi y), y - \varepsilon \sin(2\pi x) \sin(2\pi y))$, and the connectivity of the mesh is preserved. Figure 12 shows the mesh distortion for the case of $\varepsilon = 0.1$. Tables 9 and 10 show measured convergence factors for the two-grid algorithm as we distort a uniform mesh with $h = 1/80$, taking ε to be integer multiples of the meshsize and fixing the Chebyshev relaxation parameters to be the same as the optimal choices for the uniform mesh.

Considering the results in Table 9, we see generally small changes in the measured convergence factor from those predicted by LFA for $\varepsilon = h = 0.0125$, although some changes, particularly for $k = 2, 3$ are more significant. Slightly larger

TABLE 9 Measured two-grid performance for multigrid with Chebyshev Vanka relaxation with no weights applied to the $P_2 - P_1$ discretization on distorted meshes

ε :	VKI					VKE				
k	0	0.0125	0.025	0.05	0.1	0	0.0125	0.025	0.05	0.1
1	0.672	0.673	0.675	0.679	0.718	0.475	0.471	0.479	0.506*	0.557*
2	0.295	0.347	0.405	0.525	0.755	0.440	0.540	0.663	div	div
3	0.120	0.176	0.150*	0.166	0.266	0.168	0.210	0.265	0.393	0.668
4	0.102	0.103	0.104	0.109	0.127	0.127	0.141	0.161	0.184	0.405
5	0.051	0.062	0.079	0.159	0.316	0.112	0.117	0.124	0.139	0.255

Note: An alternating convergence pattern was observed, so the reported convergence factor is averaged over the final few iterations before convergence (denoted *). Diverging iterations are denoted by “div.”

TABLE 10 Measured two-grid performance for multigrid with Chebyshev Vanka relaxation with geometric weights applied to the $P_2 - P_1$ discretization on distorted meshes

ϵ :	VKIW					VKEW					
	k	0	0.0125	0.025	0.05	0.1	0	0.0125	0.025	0.05	0.1
1		0.518	0.540	0.571	0.636	0.751	0.584	0.586	0.588	0.607	0.647
2		0.196	0.268	0.382	0.701	div	0.376	0.644	div	div	div
3		0.106	0.113*	0.126	0.228	0.614	0.233	0.302	0.408	0.740	div
4		0.085	0.118	0.140	0.169	0.357	0.149	0.175	0.251	0.440	div
5		0.070	0.088	0.113	0.168	0.548	0.108	0.153	0.185	0.256	0.668

Note: An alternating convergence pattern was observed, so the reported convergence factor is averaged over the final few iterations before convergence (denoted *). Diverging iterations are denoted by “div.”

changes are seen in the case of $\epsilon = 2h = 0.025$; however, for most cases, the LFA predictions from the uniform mesh are surprisingly accurate. Only for $k = 2$ do we see significant deviation, measuring a convergence factor of 0.405 for VKI, compared with a prediction of 0.295, and 0.663 for VKE, compared with 0.440. For larger ϵ , the discrepancy grows more significant (as expected), and we start to see true cases of “failure,” with the optimal parameters from the uniform mesh leading to divergent stationary iterations. Similar results are observed in Table 10 for the case of using geometric weights, with the most sensitivity to ϵ seen for VKEW.

4.1.6 | Comparison with multiplicative Vanka

A recognized downside of additive relaxation schemes, as considered here, is that their convergence suffers in comparison with equivalent multiplicative schemes. To some extent, this degradation in performance is seen as acceptable, since the additive schemes offer easy parallelization and are more efficient per iteration (since they require only one global residual evaluation per iteration, and not local residual updates at every stage). Nonetheless, it is worthwhile to understand how much performance is sacrificed when moving from a multiplicative to an additive scheme. Table 11 provides measured two-grid convergence factors for the finest grid considered above with $h = 1/80$, for a parallelized variant of multigrid with multiplicative Vanka relaxation run on eight cores. Here, we use the standard “hybrid” version of the relaxation scheme, which is multiplicative on a core, but additive across cores.

To optimize the relaxation parameters, we use a two-stage brute-force approach. In the first stage, a coarse sampling of the endpoints of the Chebyshev interval was taken, sweeping over the lower endpoint in steps of 0.5, for values $0.5 \leq \alpha \leq 9.5$, and, for each value of α , the upper endpoint in steps of 0.5, for values $\alpha + 0.5 \leq \beta \leq 10.0$. From the optimal parameters identified by this sweep, $[\alpha_c, \beta_c]$ (with lowest measured convergence factor at the iteration where the relative norm of

TABLE 11 Measured two-grid performance for multigrid with multiplicative Chebyshev Vanka relaxation with no weights, using brute-force optimization to determine the Chebyshev interval, applied to the $P_2 - P_1$ discretization

	$k = 1$		$k = 2$		$k = 3$		$k = 4$	
	$[\alpha, \beta]$	ρ	$[\alpha, \beta]$	ρ	$[\alpha, \beta]$	ρ	$[\alpha, \beta]$	ρ
VKI	[0.5, 3.6]	0.308	[1.2, 3.0]	0.068	[1.8, 2.5]	0.044	[2.2, 2.3]	0.031
VKE	[0.5, 3.3]	0.338	[1.1, 2.7]	0.075	[1.3, 2.4]	0.047	[1.5, 2.1]	0.038

TABLE 12 Two-grid LFA predictions for Chebyshev Vanka relaxation and two-grid performance with periodic boundary conditions ($\hat{\rho}$) for the $Q_2 - Q_1$ discretization

k	$[\alpha, \beta]_{\text{VKI}}$	ρ_{VKI}	$\hat{\rho}_{\text{VKI}}$	$[\alpha, \beta]_{\text{VKE}}$	ρ_{VKE}	$\hat{\rho}_{\text{VKE}}$
1	[4.61, 4.81]	0.866	0.864	[3.40, 3.49]	0.770	0.769
2	[4.72, 4.73]	0.752	0.752	[0.96, 5.99]	0.506	0.509
3	[3.66, 5.82]	0.642	0.635	[0.79, 6.18]	0.326	0.327
4	[0.39, 9.14]	0.203	0.201	[0.91, 6.03]	0.262	0.263
5	[0.37, 11.42]	0.162	0.166*	[3.29, 3.67]	0.278	0.277

Note: Optimal intervals for different k with no weights. $h = \frac{1}{40}$. An alternating convergence pattern was observed, so the reported convergence factor is averaged over the final five iterations before convergence (denoted *).

the residual reaches 10^{-20}), a finer sampling is used, taking steps of 0.1 in both α and β , from $\alpha_c - 0.5$ to $\beta_c + 0.5$, again ensuring $\alpha < \beta$. The parameters reported in Table 11 are those identified as optimal from this search and the convergence factors, ρ , reported are those measured at the iteration where the relative norm of the residual reaches 10^{-20} ; in cases where the optimal parameters occurred as boundary cases at the lower/upper limits of the search range, the search range was extended by 0.5 to resolve the possible ambiguity.

Comparing performance between the multiplicative results in Table 11 and those in Tables 3 and 4, we see that the multiplicative form of the relaxation yields substantially better convergence, particularly for $k = 1$ or 2. On the one hand, this emphasizes the common outcome for multiplicative schemes over their additive counterparts. On the other hand, with increasing trends toward processors that excel at fine-grained parallelism, the fact that the additive schemes offer at least comparable performance is encouraging.

4.2 | Numerical results for the $Q_2 - Q_1$ discretization

Similarly to the case of the $P_2 - P_1$ discretization, we consider Chebyshev Vanka relaxation and preconditioned Richardson relaxation within monolithic multigrid methods for the Stokes equations for the $Q_2 - Q_1$ discretization. Here, all of the optimal parameters are obtained by using the robust optimization algorithm,⁵⁵ and we compare the LFA predictions with performance observed for the discretization with periodic boundary conditions with $h = \frac{1}{40}$. We remark that the measured convergence factors with Dirichlet boundary conditions also match well with the LFA predictions. For the optimal parameters, we truncate the results obtained by the robust optimization to two digits, noting that the performance is not sensitive to this truncation.

Tables 12 and 13 present the measured convergence for periodic boundary conditions versus the LFA predictions for different k and the corresponding optimal intervals for Chebyshev Vanka relaxation with no weights and natural weights, respectively. Similarly to the case of the $P_2 - P_1$ discretizations, we see that the optimal pair of $[\alpha, \beta]$ is not unique in some cases. We break such ties arbitrarily. We see good agreement between the LFA predictions and the multigrid performance, except for VKI with $k = 5$ and VKEW with $k = 2$. For the former case, the per iteration convergence factor oscillates in the range from 0.12 to 0.22, averaging close to that predicted by LFA. For the latter, more variation is seen in the per-cycle convergence factors, between 0.07 and 1.19, but the long-term average still approaches that given by LFA. For these cases, we present averaged convergence over the final five iterations. Comparing Tables 12 with 13 shows that the patches with natural weights outperform those with no weights, in some cases by a large margin. An interesting observation from these tables, particularly Table 13, is that the optimization has a natural preference for giving very small intervals. This would normally be a cause for concern, but clearly leads to excellent behavior in this setting.

k	$[\alpha, \beta]_{\text{VKIW}}$	ρ_{VKIW}	$\hat{\rho}_{\text{VKIW}}$	$[\alpha, \beta]_{\text{VKEW}}$	ρ_{VKEW}	$\hat{\rho}_{\text{VKEW}}$
1	[0.15, 2.95]	0.637	0.636	[1.32, 3.51]	0.681	0.683
2	[0.71, 1.97]	0.288	0.290	[1.50, 1.54]	0.271	0.265*
3	[0.90, 1.42]	0.153	0.161	[1.93, 1.99]	0.234	0.241
4	[1.23, 1.24]	0.097	0.098	[2.38, 2.39]	0.217	0.221
5	[1.17, 1.40]	0.071	0.072	[1.57, 2.41]	0.129	0.128

Note: Optimal intervals for different k with natural weights. $h = \frac{1}{40}$. An alternating convergence pattern was observed, so the reported convergence factor is averaged over the final five iterations before convergence (denoted *).

TABLE 13 Two-grid LFA predictions for Chebyshev Vanka relaxation and two-grid performance with periodic boundary conditions ($\hat{\rho}$) for the $Q_2 - Q_1$ discretization

TABLE 14 Two-grid LFA predictions, $\rho^{(v_1, v_2)}$, for Richardson relaxation optimizing outer weights for the $Q_2 - Q_1$ discretization

Method	ω_{opt}	$\rho^{(1,0)}$	$\rho^{(1,1)}$	$\rho^{(2,2)}$	$\rho^{(3,3)}$	$\rho^{(4,4)}$	$\rho^{(5,5)}$	$(\omega_{1,\text{opt}}, \omega_{2,\text{opt}})$	$\rho^{(1,1)}$
VKI	0.21	0.931	0.867	0.752	0.651	0.565	0.490	(0.13, 0.60)	0.809
VKIW	0.92	0.712	0.913	0.574	0.290	0.155	0.134	(0.65, 0.65)	0.637
VKE	0.29	0.878	0.770	0.602	0.488	0.403	0.337	(0.76, 0.17)	0.639
VKEW	0.73	0.697	1.519	0.507	0.296	0.577	0.305	(0.41, 0.41)	0.684

Method	$d_{1,\text{opt}}$	$d_{2,\text{opt}}$	$d_{3,\text{opt}}$	ρ
VKI ($v_1 + v_2 = 1$)	0.10	0.13	1.00	0.695
VKE ($v_1 + v_2 = 1$)	0.88	0.20	0.84	0.648
VKI ($v_1 + v_2 = 2$)	0.15	0.09	0.79	0.583
VKE ($v_1 + v_2 = 2$)	0.18	0.22	0.48	0.646

TABLE 15 Two-grid LFA predictions for Richardson relaxation for the $Q_2 - Q_1$ discretization with three weights in D_i

Table 14 gives results for the $Q_2 - Q_1$ discretization when optimizing only the outer parameter as in Table 6. As above, we then fix this weight and consider symmetric cycles with $v_1 = v_2 = 1, \dots, 5$. For small numbers of sweeps, we see here that Richardson acceleration is as effective as Chebyshev for VKI and VKE, although Chebyshev is more effective for larger numbers of relaxation sweeps. For VKIW and VKEW, by contrast, the Chebyshev-based relaxation schemes are always significantly more effective. While using two different weights within a (1, 1) cycle is effective for VKI and VKE, better efficiency is seen for a single sweep with natural weights. We also consider fixing $\omega_1 = \omega_2 = 1$ in (20) and optimizing the inner weights. Table 15 shows the optimal results for preconditioned Richardson relaxation with three parameters. Note that, as above, optimizing five weights does not improve the performance; thus, we omit the results here. Optimizing three weights obtains significantly better results than above for VKI, but shows little improvement when increasing the number of relaxation sweeps. As above, optimizing three weights offers some improvement over natural weights, but not enough to suggest further optimization is worthwhile.

4.3 | Comparing cost and performance

4.3.1 | Cost of relaxation for the $P_2 - P_1$ discretization

All schemes have the same cost for computing the initial residual. The P_2 Laplacian, A , contains four types of stencils, that is, a 9-point stencil for N -type points, and 5-point stencils for X -, Y -, and C -type points, and B_x and B_y have 10-point stencils. Away from boundaries, we can naturally associate each DoF with a node, to see that a mesh with n nodes also has about n of each type of edge DoF. So, the cost of a single-residual evaluation on a mesh with n nodes is (roughly) that of $(9 + 3 \times 5)n \times 2 + 10 \times n \times 4 = 88n$ multiply-add operations, coming from the six nonzero blocks in the matrix.

TABLE 16 The number (#) of nonzero elements of L and U ($LU = K_i$) for Vanka-inclusive and Vanka-exclusive patches for the $P_2 - P_1$ and $Q_2 - Q_1$ discretizations

Patch: #	$P_2 - P_1$		$Q_2 - Q_1$	
	Ξ_{VKI}	Ξ_{VKE}	Ξ_{VKI}	Ξ_{VKE}
L	284	115	617	335
U	282	115	617	337
Total	566	230	1234	672

TABLE 17 Comparing cost and performance for the $P_2 - P_1$ discretization

	VKI	VKE	VKIW	VKEW
W_s ($k = 1, v_1 + v_2 = 1$)	$654n$	$318n$	$693n$	$345n$
Most effective ($k, v_1 + v_2$)	(3, 2)	(1, 2)	(1, 1)	(1, 1)
ρ	0.120	0.356	0.571	0.452
$W_t = k \cdot (v_1 + v_2) \cdot W_s$				
W_t	$3924n$	$636n$	$693n$	$345n$
Relative efficiency	0.830	0.571	0.757	0.452

For the $P_2 - P_1$ discretization, the remaining cost is that of solving a small problem, $K_i x_i = b_i$, in each patch. Here, we use LU decomposition to solve these subproblems. Assume that the LU decomposition is precomputed, as this can be done once per patch and used for all of the solves over that patch. As in Table 2, for VKI and VKIW, the size of the patch problem is $2(7 + 4 + 4 + 4) + 1 = 39$. For VKE and VKEW, it is $2(1 + 2 + 2 + 4) + 1 = 27$. Note that, similarly to K , K_i is a block (sparse) matrix, and its LU factors will retain some sparsity as well. Thus, the cost of solving the sparse systems with L or U will require a number of multiply-add operations equal to the number of nonzero entries in L or U . Table 16 presents these numbers, based on direct calculation of the factorizations. For VKI and VKIW, the cost of applying the inverses of L and U is 566 multiply-add operations. For the whole system, we need to solve roughly n subproblems, giving a total cost of $566n$. Similarly, it costs $230n$ multiply-add operations for VKE and VKEW. For the approaches with natural weights, we need an additional scaling for each subproblem. Thus, there will be an additional cost of $39n$ and $27n$ for VKIW and VKEW, respectively.

Accumulating the costs of a residual evaluation with these, we have total costs of $88n + 566n = 654n$ multiply-add operations per sweep of VKI, $88n + 230n = 318n$ multiply-add operations per sweep of VKE, $654n + 39n = 693n$ multiply-add operations per sweep of VKIW, and $318n + 27n = 345n$ multiply-add operations per sweep of VKEW. To compare these costs, we omit the cost of the coarse-grid correction and only consider the cost of the relaxation scheme. We denote the above cost as W_s , which corresponds to $k = 1$ and $v_1 + v_2 = 1$. Comparing efficiencies can now be easily done by appropriately weighting either measured or predicted convergence factors relative to their work (here, we use the predicted convergence factors): 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 with 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. Here, we compare the efficiency relative to VKEW.

Next, using the data presented above, we find the most effective parameters for each of VKI, VKE, VKIW, and VKEW, and compare their costs. From Tables 3 and 4, we see the most effective choice for VKI is $k = 3$ giving $\rho = 0.120$, while, for VKE, it is $k = 1$ giving $\rho = 0.475$. Note, however, that a more efficient set of parameters for VKE is found in Table 6, giving a convergence factor of 0.356 per iteration. From Table 5, the most effective choice for VKIW is $k = 2$ giving $\rho = 0.196$, and $k = 1$ giving $\rho = 0.584$ for VKEW. Here, optimizing with more weights, as in Tables 7 and 8, was more effective, and we consider the best results with a nontrivial scaling matrix, D_i , in place of the geometric weights, giving 0.571 for VKIW and 0.452 for VKEW. Now, we can calculate the efficiency for each relaxation scheme relative to its total cost, $W_t = k \cdot (v_1 + v_2) \cdot W_s$, where W_s denotes the cost of a single relaxation. Table 17 details the cost-effectiveness for each scheme for the $P_2 - P_1$ discretization, showing that VKEW offers the most efficient relaxation scheme. In particular, this shows that the substantially smaller cost per iteration of the exclusive patches offers greater efficiency, despite the improved convergence when using inclusive patches.

	VKI	VKE	VKIW	VKEW
$W_s (k = 1, v_1 + v_2 = 1)$	$1338n$	$776n$	$1389n$	$811n$
Most effective $(k, v_1 + v_2)$	$(4, 2)$	$(1, 2)$	$(1, 1)$	$(1, 1)$
ρ	0.203	0.639	0.695	0.648
$W_t = k \cdot (v_1 + v_2) \cdot W_s$				
W_t	$10704n$	$1552n$	$1389n$	$811n$
Relative efficiency	0.886	0.791	0.809	0.648

TABLE 18 Comparing cost and performance for the $Q_2 - Q_1$ discretization

4.3.2 | Cost of relaxation for the $Q_2 - Q_1$ discretization

The Q_2 Laplacian, A , contains four types of stencils, a 25-point stencil for N -type points, 15-point stencils for X -, and Y - DoFs, and a 9-point stencil for C -type points, while B_x and B_y have 12-point stencils. So, the cost of a single-residual evaluation on a mesh with n nodes is (roughly) that of $(25 + 2 \cdot 15 + 9)n \cdot 2 + 12 \cdot n \cdot 4 = 104n$ multiply-add operations, coming from the six nonzero blocks in the matrix.

For the $Q_2 - Q_1$ approximation with additive Vanka-type relaxation, there is little difference in the cost calculation compared with that of the $P_2 - P_1$ discretization. Here, we again use the LU decomposition of the patch matrices, K_i , and assume the LU decomposition is done. For VKI, the cost of applying the inverses of L and U is 1,234 multiply-add operations per block (see Table 16). Thus, the cost for a full sweep is $1,234n$ multiply-add operations. Similarly, it costs $672n$ multiply-add operations per sweep of VKE. For the approaches with natural weights, there will be an additional cost of $51n$ and $35n$ for VKIW and VKEW, respectively, for these scaling operations. Accumulating the costs of a residual evaluation with these, we have total costs of $104n + 1,234n = 1,338n$ multiply-add operations per sweep of VKI, $104n + 672n = 776n$ multiply-add operations per sweep of VKE, $1,338 + 51n = 1,389n$ multiply-add operations per sweep of VKIW, and $776n + 35n = 811n$ multiply-add operations per sweep of VKEW.

From Tables 12 and 13, we see the most effective choice for VKI is $k = 4$ giving $\rho = 0.203$, while, for VKE, it is $k = 3$ giving $\rho = 0.326$. Note, however, that a more efficient set of parameters for VKE is found in Table 14, giving a convergence factor of 0.639 per iteration with $v_1 + v_2 = 2$. From Table 13, the most effective choice for VKIW is $k = 3$ giving $\rho = 0.153$, and $k = 2$ giving $\rho = 0.271$ for VKEW. Here, optimizing with more weights, as in Table 15, was more effective, and we consider the best results with a nontrivial scaling matrix, D_i , in place of the geometric weights, giving 0.695 for VKIW and 0.648 for VKEW with $v_1 + v_2 = 1$. Table 18 compares the efficiency relative to VKEW, showing that the most efficient choice is VKEW, the same as for the $P_2 - P_1$ discretization.

5 | CONCLUSIONS AND FUTURE WORK

We present a local Fourier analysis for a monolithic multigrid method based on overlapping additive Vanka-type relaxation for the Stokes equations. Two choices of patches for the overlapping schemes are discussed for the $P_2 - P_1$ and $Q_2 - Q_1$ discretizations. A general framework of LFA for additive Vanka relaxation is developed to help choose algorithmic parameters, which can be applied to other problems and to different discretizations. The LFA shows that using smaller patches can outperform relaxation using bigger patches due to the lower cost per sweep of relaxation. Moreover, to improve the performance, we use LFA to optimize the weights, yielding notable improvement. Numerical performance with periodic and Dirichlet boundary conditions validate the LFA predictions, showing that these Vanka relaxation schemes are robust to the different boundary conditions.

Extending Vanka relaxation for other types of problems is an interesting topic. We note that this LFA framework of additive Vanka relaxation has a limitation: since we need to know the patch first, selection of the patches is not readily optimized in this framework. Another interesting question is the use of other boundary conditions on the patches, such as are used in optimized Schwarz, which could also be tuned using LFA. Developing general-purpose LFA software for additive Vanka with automatic evaluation of patch choices and optimizing the weights are also topics for future work.

ACKNOWLEDGEMENTS

The work of P.E.F. was supported by the Engineering and Physical Sciences Research Council (grant numbers EP/K030930/1 and EP/R029423/1). The work of S.P.M. was partially supported by an NSERC Discovery Grant. This work does not have any conflicts of interest.

ORCID

Patrick E. Farrell  <https://orcid.org/0000-0002-1241-7060>

Yunhui He  <https://orcid.org/0000-0003-3476-2487>

Scott P. MacLachlan  <https://orcid.org/0000-0002-6364-0684>

REFERENCES

1. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numer.* 2005;14:1–137.
2. Wathen A, Silvester D. Fast iterative solution of stabilised Stokes systems. Part I: using simple diagonal preconditioners. *SIAM J Numer Anal.* 1993;30(3):630–649.
3. Silvester D, Wathen A. Fast iterative solution of stabilised Stokes systems. Part II: using general block preconditioners. *SIAM J Numer Anal.* 1994;31(5):1352–1367.
4. Adler JH, Benson TR, MacLachlan SP. Preconditioning a mass-conserving discontinuous Galerkin discretization of the Stokes equations. *Numer Linear Algebra Appl.* 2017;24(3):e2047.
5. Geenen T, ur Rehman M, MacLachlan SP, Segal G, Vuik C, van den Berg AP, Spakman W. Scalable robust solvers for unstructured FE geodynamic modeling applications: Solving the Stokes equation for models with large localized viscosity contrasts, *Geochem. Geophys. Geosyst.* 2009;10:Q09002, doi:10.1029/2009GC002526.
6. Rudi J, Malossi ACI, Isaac T, et al. An extreme-scale implicit solver for complex PDEs: highly heterogeneous flow in Earth's mantle. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '15.* New York, NY, USA: ACM; 2015. p. 5:1–5:12.
7. John V, Tobiska L. Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier-Stokes equations. *Int J Numer Methods Fluids.* 2000 Jan;33(4):453–473.
8. Gmeiner B, Rüde U, Stengel H, Waluga C, Wohlmuth B. Performance and scalability of hierarchical hybrid multigrid solvers for Stokes systems. *SIAM J Sci Comput.* 2015;37(2):C143–C168.
9. Gmeiner B, Huber M, John L, Rüde U, Wohlmuth B. A quantitative performance study for Stokes solvers at the extreme scale. *J Comput Sci.* 2016;17(part 3):509–521.
10. Niestegge A, Witsch K. Analysis of a multigrid Stokes solver. *Appl Math Comput.* 1990;35(3):291–303.
11. Oosterlee CW, Lorenz FJG. Multigrid methods for the Stokes system. *Comput Sci Eng.* 2006;8(6):34–43.
12. Braess D, Sarazin R. An efficient smoother for the Stokes problem. *Appl Numer Math.* 1997;23:3–20.
13. Adler JH, Benson TR, Cyr EC, MacLachlan SP, Tuminaro RS. Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics. *SIAM J Sci Comput.* 2016;38(1):B1–B24.
14. Gaspar FJ, Notay Y, Oosterlee CW, Rodrigo C. A simple and efficient segregated smoother for the discrete Stokes equations. *SIAM J Sci Comput.* 2014;36(3):A1187–A1206.
15. Luo P, Rodrigo C, Gaspar FJ, Oosterlee CW. On an Uzawa smoother in multigrid for poroelasticity equations. *Numer Linear Algebra Appl.* 2017;24(1):e2074.
16. Vanka SP. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J Comput Phys.* 1986;65:138–158.
17. Franco SR, Rodrigo C, Gaspar FJ, Pinto MAV. A multigrid waveform relaxation method for solving the poroelasticity equations. *Comput Appl Math.* 2018;37(4):4805–4820.
18. Wang M, Chen L. Multigrid methods for the Stokes equations using distributive Gauss–Seidel relaxations based on the least squares commutator. *J Sci Comput.* 2013;56(2):409–431.
19. Chen L, Hu X, Wang M, Xu J. A multigrid solver based on distributive smoother and residual overweighting for Oseen problems. *Numer Math Theory Methods Appl.* 2015;8(02):237–252.
20. Adler JH, Emerson DB, MacLachlan SP, Manteuffel TA. Constrained optimization for liquid crystal equilibria. *SIAM J Sci Comput.* 2016;38(1):B50–B76.
21. He Y, MacLachlan SP. Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations. *Numerical Linear Algebra with Applications.* 2018;25(3):e2147.
22. He Y, MacLachlan SP. Local Fourier analysis for mixed finite-element methods for the Stokes equations. *J Comput Appl Math.* 2019;357:161–183.
23. John V. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations. *Int J Numer Methods Fluids.* 2002;40(6):775–798.
24. John V, Matthies G. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *Int J Numer Methods Fluids.* 2001;37(8):885–903.
25. Manservigi S. Numerical analysis of Vanka-type solvers for steady Stokes and Navier-Stokes flows. *SIAM J Numer Anal.* 2006;44(5):2025–2056.

26. Schöberl J, Zulehner W. On Schwarz-type smoothers for saddle point problems. *Numerische Mathematik*. 2003;95(2):377–399.
27. de la Riva AP, Rodrigo C, Gaspar FJ. A robust multigrid solver for isogeometric analysis based on multiplicative Schwarz smoothers. *SIAM J Sci Comput*. 2019;41(5):S321–S345.
28. Kahl K, Kintscher N. Automated local Fourier analysis (aLFA). *BIT Numer Math*. 2020. <https://doi.org/10.1007/s10543-019-00797-w>.
29. Claus L. Multigrid smoothers for saddle point systems. Wuppertal, Germany: Bergische Universität Wuppertal, 2019.
30. Sivaloganathan S. The use of local mode analysis in the design and comparison of multigrid methods. *Comput Phys Commun*. 1991;65(1-3):246–252.
31. Molenaar J. A two-grid analysis of the combination of mixed finite elements and Vanka-type relaxation, Birkhäuser, Basel. *Multigrid Methods III.*, Springer, 1991;p. 313–323.
32. MacLachlan SP, Oosterlee CW. Local Fourier analysis for multigrid with overlapping smoothers applied to systems of PDEs. *Numer Linear Algebra Appl*. 2011;18:751–774.
33. Rodrigo C, Gaspar FJ, Lisbona FJ. On a local Fourier analysis for overlapping block smoothers on triangular grids. *Appl Numer Math*. 2016;105:96–111.
34. Boonen T, Van Lent J, Vandewalle S. Local Fourier analysis of multigrid for the curl-curl equation. *SIAM J Sci Comput*. 2008;30(4):1730–1755.
35. Rathgeber F, Ham DA, Mitchell L, et al. Firedrake: automating the finite element method by composing abstractions. *ACM Trans Math Softw*. 2017;43(3):24.
36. Balay S, Abhyankar S, Adams M, et al. PETSc Users Manual: Revision 3.10. Argonne, IL: Argonne National Lab (ANL), 2018.
37. Kirby RC, Mitchell L. Solver composition across the PDE/linear algebra barrier. *SIAM J Sci Comput*. 2018;40(1):C76–C98.
38. Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers with applications in incompressible fluid dynamics. *Numerical Mathematics and Scientific Computation*. 2nd ed. Oxford, UK: Oxford University Press, 2014.
39. Brenner SC, Scott LR. The mathematical theory of finite element methods. vol. 15 of Texts in Applied Mathematics. 3rd ed. New York, NY: Springer, 2008.
40. Brezzi F, Douglas J. Stabilized mixed methods for the Stokes problem. *Numerische Mathematik*. 1988;53(1):225–235.
41. Dohrmann CR, Bochev PB. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *Int J Numer Methods Fluids*. 2004;46(2):183–201.
42. Limache A, Idelsohn S, Rossi R, Oñate E. The violation of objectivity in Laplace formulations of the Navier–Stokes equations. *Int J Numer Methods Fluids*. 2007;54(6-8):639–664.
43. Trottenberg U, Oosterlee CW, Schüller A. Multigrid. San Diego, CA: Academic Press, Inc., 2001 With contributions by A. Brandt, P. Oswald and K. Stüben.
44. Stüben K, Trottenberg U. Multigrid methods: fundamental algorithms, model problem analysis and applications. *Multigrid methods*, Berlin, Heidelberg: Springer, 1982; p. 1–176.
45. Wesseling P. An introduction to multigrid methods. *Pure and Applied Mathematics (New York)*. Chichester: John Wiley & Sons, Ltd., 1992.
46. Saad Y. Iterative methods for sparse linear systems. vol. 82. SIAM; 2003.
47. Frommer A, Szyld DB. Weighted max norms, splittings, and overlapping additive Schwarz iterations. *Numer Math*. 1999;83(2):259–278.
48. Benzi M, Frommer A, Nabben R, Szyld DB. Algebraic theory of multiplicative Schwarz methods. *Numer Math*. 2001;89(4):605–639.
49. Wienands R, Joppich W. Practical Fourier analysis for multigrid methods, Vol. 4, Boca Raton, FL: Chapman & Hall/CRC, 2005.
50. He Y, MacLachlan SP. Two-level Fourier analysis of multigrid for higher-order finite-element discretizations of the Laplacian. *Numer Linear Algebra Appl*. 2020;27(3):e2285.
51. Logg A, Mardal KA, Wells GN, et al. Automated Solution of Differential Equations by the Finite Element Method. The FEniCS book. Vol 84. Springer Science & Business Media, 2012.
52. Alnæs MS, Blechta J, Hake J, et al. The FEniCS Project Version 1.5. *Arch Numer Softw*. 2015;3(100):9–23.
53. Rodrigo C, Sanz F, Gaspar FJ, Lisbona FJ. Local Fourier analysis for edge-based discretizations on triangular grids. *Numer Math Theory Methods Appl*. 2015;8(1):78–96.
54. Stevenson RP. On the Validity of Local Mode Analysis of Multi-Grid Methods. Utrecht, The Netherlands: Utrecht University, 1990.
55. Brown J, He Y, MacLachlan SP, Menickelly M, and Wild S. Robust optimization in local Fourier analysis. arXiv preprint arXiv:200100887. 2020.

How to cite this article: Farrell PE, He Y, MacLachlan SP. A local Fourier analysis of additive Vanka relaxation for the Stokes equations. *Numer Linear Algebra Appl*. 2021;28:e2306. <https://doi.org/10.1002/nla.2306>