Title
Nonoverlapping block smoothers for the Stokes equations

Permalink
https://escholarship.org/uc/item/9rs3k12w

Journal
Numerical Linear Algebra with Applications, 28(6)

ISSN
1070-5325

Authors
Claus, Lisa
Bolten, Matthias

Publication Date
2021-12-01

DOI
10.1002/nla.2389

Peer reviewed

Received: 8 July 2020 | Revised: 4 February 2021 | Accepted: 20 April 2021

DOI: 10.1002/nla.2389

RESEARCH ARTICLE

WILEY

Nonoverlapping block smoothers for the Stokes equations

Lisa Claus[1,2] | **Matthias Bolten**[2]

[1]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California, USA

[2]University of Wuppertal, Wuppertal, Germany

Correspondence
Lisa Claus, Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA.
Email: lclaus@lbl.gov

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: BO 3405/2

Abstract

Overlapping block smoothers efficiently damp the error contributions from highly oscillatory components within multigrid methods for the Stokes equations but they are computationally expensive. This paper is concentrated on the development and analysis of new block smoothers for the Stokes equations that are discretized on staggered grids. These smoothers are nonoverlapping and therefore desirable due to reduced computational costs. Traditional geometric multigrid methods are based on simple pointwise smoothers. However, using multigrid methods to efficiently solve more difficult problems such as the Stokes equations leads to computationally more expensive smoothers, for example, overlapping block smoothers. Nonoverlapping smoothers are less expensive, but have been considered less efficient in the literature. In this paper, we develop new nonoverlapping smoothers, the so-called triad-wise smoothers, and show their efficiency within multigrid methods to solve the Stokes equations. In addition, we compare overlapping and nonoverlapping smoothers by measuring their computational costs and analyzing their behavior by the use of local Fourier analysis.

KEYWORDS

block smoother, finite difference discretization, geometric multigrid methods, local fourier analysis, marker-and-cell scheme, staggered grids, Stokes equations

1 | INTRODUCTION

The Stokes equations are a saddle-point system modeling a slow viscous flow. There is a general interest to find efficient solutions for solving these kind of systems of PDEs, because they describe the physics of many phenomena of scientific and engineering interest.[1] For example, the constant movement of the Earth crust caused by the mantle convection is one application that can be described and simulated by means of the Stokes equations.[2] Large saddle-point linear systems represent a significant challenge for linear solvers owing to their indefiniteness and often poor spectral properties. Their importance motivates the development of effective solvers.

Multigrid methods have been applied successfully to large linear systems.[3-5] The methods find an approximate solution through two complementary processes: smoothing and coarse-grid correction. Smoothing filters parts of the error from the approximate solution in a computationally inexpensive way, while coarse-grid correction constructs

[Correction added on 21 May 2021, after first online publication: affiliation 2 was added to author 'Lisa Claus' in this version.]

a lower dimensional problem to remove the error remaining after smoothing. Coarse grid correction consists of the components restriction, interpolation and solution on the coarsest grid. This paper develops efficient multigrid methods to solve the Stokes equations discretized by the marker-and-cell (MAC) scheme. The multigrid components of restriction, interpolation and solution on the coarsest grid can be immediately generalized to large linear systems of saddle point type.

However, the design of efficient smoothers for solving systems of PDEs requires special attention. We distinguish between decoupled smoothing and collective smoothing. A zero diagonal block that appears in the discrete system hampers a basic numerical treatment of the problem, which would be to relax the discrete equations directly in a decoupled way. A variety of multigrid methods for the Stokes equations based on different smoothing approaches have been developed, used and analyzed.[6-9] We focus on collective relaxation, so-called block smoothers. A comparison of a well-known overlapping block smoother, the Vanka smoother, with a nonoverlapping smoother, the triad-wise smoother, is given. There exist two variants of the Vanka smoother, the so-called multiplicative Vanka smoother and the additive Vanka smoother. In this paper, we focus on the multiplicative version since this smoother leads to better convergence rates considering our aimed discretization method. A study of the additive Vanka smoother with patch-wise procedures can be found in Reference 10. While the multiplicative Vanka smoother is well established to solve the Stokes equations, it is computationally expensive.[11,12] Nonoverlapping block smoothers are less expensive but were considered less efficient in terms of their smoothing and convergence behavior.[11] In this paper, we develop a new nonoverlapping block smoother based on the results of the comparison with both smoothers. For periodic boundary conditions we demonstrate the diminished efficiency of the triad-wise smoother relative to the Vanka smoother. Solving the Stokes equations with Dirichlet boundary conditions leads to an enormous decrease of the efficiency for the triad-wise smoother. We highlight the issues of the nonoverlapping smoother by means of analysis and numerical examples and demonstrate the causes of the high computational costs of the overlapping smoother. Based on these results a new nonoverlapping smoother is constructed for Dirichlet boundary conditions that overcomes the demonstrated issues. To the best of our knowledge, the proposed modified smoother represents the first-ever nonoverlapping block smoother that reaches efficiency close to the overlapping block smoother.

Local Fourier analysis (LFA) is utilized as the main analysis tool for quantitative convergence estimates and to optimize geometric multigrid components such as smoothers or intergrid transfer operators. The idea was introduced by Brandt[13] and later extended and refined in Reference 14. Practical guidelines on the use of LFA can be found in Reference 15. LFA is commonly applied to relaxation schemes using (symmetric) Gauss-Seidel approaches,[16] and has been extended to more sophisticated relaxation schemes.[17-19] In this study, LFA is used to analyze the two-grid method based on the overlapping and nonoverlapping smoothers. In addition to the LFA results, we provide numerical results that guide the development of a new algorithm leading to more efficient multigrid methods.

The outline of the paper is as follows. In Section 2, we introduce the MAC finite-difference discretization of the two-dimensional Stokes equations and the basic idea of the two block relaxation methods used in this paper. In Section 3, we present some basic definitions and notations to introduce LFA for the Stokes equations. In addition, analysis results are given at the end of the section. In Section 4, numerical examples extend the analysis and highlight the differences and issues arising due to different boundary conditions. In Section 5, we develop a new nonoverlapping block smoother, an update of the so-called triad-wise smoothers leading to an enormous increase of efficiency. We give some details about parallelization aspects in Section 6. Conclusions are drawn in Section 7.

2 | SMOOTHER FOR STOKES EQUATIONS

2.1 | Stokes equations

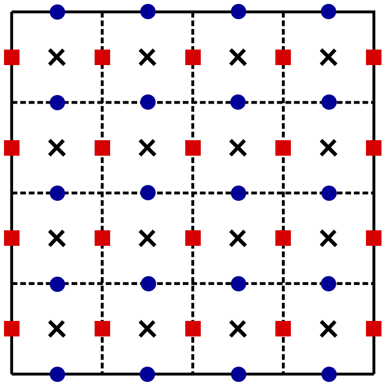
We introduce the Stokes equations in two dimensions here. Let Ω be a domain in $\mathbb{R}^2$ with boundary $\partial\Omega$, and $\mathbf{f} = (\mathrm{f}_1, \mathrm{f}_2)^T$ be a given vector function that describes an external force. Solving the two-dimensional Stokes equations means finding a fluid velocity $\mathbf{u} = (u, v)$ and pressure p such that the following system is fulfilled,

$$\Delta u + \partial_{\mathrm{x}_1} p = \mathrm{f}_1, \tag{1}$$

$$\Delta v + \partial_{\mathrm{x}_2} p = \mathrm{f}_2, \tag{2}$$

$$\partial_{\mathrm{x}_1} u + \partial_{\mathrm{x}_2} v = 0, \tag{3}$$

FIGURE 1 Location of unknowns (u, v, p)



where $\partial_{x_i} := \frac{\partial}{\partial x_i}$. The discretization of the Stokes equations naturally leads to linear systems of the form

$$A_h = \begin{pmatrix} C_1 & & B_1 \\ & C_1 & B_2 \\ -B_1^T & -B_2^T & \end{pmatrix} \begin{pmatrix} u_h \\ v_h \\ p_h \end{pmatrix} = \begin{pmatrix} \mathrm{f}_1 \\ \mathrm{f}_2 \\ 0 \end{pmatrix}, \tag{4}$$

so-called saddle point systems. Here C_1 denotes the negative discrete Laplace operator, $C_1 = -\Delta_h$, and B_1^T and B_2^T are the discrete divergence operators, $B_1^T = (\partial_{x_1})_{h/2}$ and $B_2^T = (\partial_{x_2})_{h/2}$. In this paper, we use a staggered distribution of unknowns, the so-called MAC scheme. The idea of the MAC scheme is to place the unknowns (u, v, p) in different locations. More specifically, the discrete pressure unknowns p are located in the center of each cell and the discrete values of the velocity unknowns are located on the midpoint of vertical edges (x_1-component u) and on the midpoint of horizontal edges (x_2-component v), as seen in Figure 1. We consider uniform meshes with: $h_{x_1} = h_{x_2} = h$ in this paper.

The discrete analogue of the x_1-coordinate momentum Equation (1) is defined at vertical edges, the discrete x_2-coordinate momentum Equation (2) at horizontal edges, and the continuity Equation (3) at cell centers using central difference schemes. We introduce the following indexing system with regard to the underlying grid: i is the row index and j is the column index, ranging from $1:n$ or $1:n+1$, where n is the number of cells in one direction. The discrete pressure unknowns are $p_{i,j}$, where $i=1, \dots, n, j=1, \dots, n$, and the velocity unknowns are $u_{i,j}$ with $i=1, \dots, n, j=1, \dots, n+1$ and $v_{i,j}$ with $i=1, \dots, n+1, j=1, \dots, n$. Using this indexing system, the MAC scheme can be written as

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} + \frac{p_{i,j} - p_{i-1,j}}{h} = f_1^{i,j}, \tag{5}$$

$$\frac{4v_{i,j} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1}}{h^2} + \frac{p_{i,j} - p_{i,j-1}}{h} = f_2^{i,j}, \tag{6}$$

$$\frac{u_{i+1,j} - u_{i,j}}{h} + \frac{v_{i,j+1} - v_{i,j}}{h} = 0. \tag{7}$$

Our interest in this paper is to develop efficient multigrid methods to solve the Stokes equations discretized by the MAC scheme. We discuss how the multigrid components of smoothing, restriction, interpolation, and solution on the coarsest grid are generalized to saddle-point systems. A key point of an efficient multigrid method is the right choice of the relaxation procedure. The design of efficient smoothers for solving systems of PDEs often requires special attention. The relaxation method should smooth the error for all unknowns in the equations, which is not an easy task. Therefore, we focus on constructing appropriate relaxation schemes to smooth the error of the unknown functions. The other multigrid components can immediately be extended to systems of PDEs. A zero diagonal block that appears in the discrete system hampers a basic numerical treatment of the problem, which would be to relax the discrete equations directly in a decoupled way. More specifically, decoupled iterative solution methods use blocks on the main diagonal in an equation-wise fashion. Unfortunately, this equation-wise relaxation is not possible due to the zero diagonal block. Moreover, a first obvious choice for these so-called strong off-diagonal operators in the differential system is collective smoothing. However, for staggered grids the unknowns are not defined at the same locations and therefore we define standard collective relaxation as described in the following section.

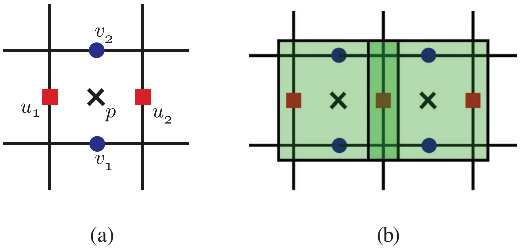


FIGURE 2 The Vanka smoother: five unknowns updated collectively. (a) All five unknowns of one block; (b) Two overlapping cells

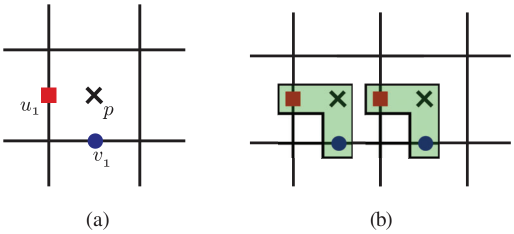


FIGURE 3 Unknowns updated collectively by triad-wise relaxation. (a) All three unknowns of one triad block; (b) Two nonoverlapping blocks

2.2 | Vanka smoother

The basic idea of the Vanka smoother is to solve the discrete Stokes equations locally "cell by cell." This means all five unknowns of one cell are updated collectively, involving the respective four momentum equations at the cell boundaries and one continuity equation in the center of the cell, compare Figure 2 and Equations (5)–(7). This results in an overlapping updating process of 5 × 5 systems of equations. This method was first introduced in Reference 11.

Using this scheme, each velocity component is updated twice and the pressure once per relaxation. The process of relaxation appears in a Gauss–Seidel manner. That means, after smoothing all unknowns of one cell collectively, we use the updated values for the relaxation of the next cell. This relaxation method is commonly referred to as *overlapping Schwarz smoother* or *symmetric coupled Gauss–Seidel*. Especially in the context of the Stokes equations, we use the term *Vanka smoother*, which has robust smoothing properties.[3] Due to the overlap the iteration matrix of the relaxation scheme cannot be easily represented by a banded matrix or equivalently as a stencil. This is possible if a suitable coloring scheme is chosen. More details can be found in Section 6.

2.3 | Triad-wise smoother

For the Stokes system, the Vanka smoother is a very efficient smoother.[11] However, due to the overlap, computational costs are high and parallel implementation aspects are not satisfying due to the required coloring scheme, compare Section 6. We exploit the potential of the Vanka relaxation method and focus on the development of a block smoother that is less expensive and still efficient.

The basic idea of the triad-wise smoother is to update three unknowns collectively, involving two momentum equations at cell boundaries and the continuity equation in the center of the cell. This results in a nonoverlapping updating process of 3 × 3 systems of equations. Using this scheme, one velocity component in each direction and one pressure component are updated simultaneously. An exemplary choice of the block of three unknowns is shown in Figure 3. Alternative choices are possible and will be discussed in Section 3.3. The first version of the triad-wise smoother was introduced in[11] and is referred to as *unsymmetric coupled Gauss–Seidel*.[11] The potential of a different updating process as for example a different order of updating the unknowns or a Jacobi-manner instead of the Gauss–Seidel updating process will be examined in this paper. We stick to the general term triad-wise smoother whenever we do not specify the manner of the updating process. We define the iteration matrix of the relaxation scheme as $S_h := I - M_h^{-1}A_h$, where I represents the identity matrix and the approximate inverse M_h^{-1} can be represented by

$$M_h := \frac{1}{\omega}D_h - L_h = \begin{bmatrix} -\frac{1}{h^2} & & -\frac{1}{h} \\ & -\frac{1}{h^2} & \\ & & \end{bmatrix}\begin{bmatrix} +\frac{4}{\omega h^2} & & +\frac{1}{\omega h} \\ & +\frac{4}{\omega h^2} & +\frac{1}{\omega h} \\ +\frac{1}{\omega h} & +\frac{1}{\omega h} & \end{bmatrix}, \quad \begin{bmatrix} -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & -\frac{1}{h} \\ & & \end{bmatrix}, \tag{8}$$

for a Gauss–Seidel-type updating process and

$$M_h := \frac{1}{\omega} D_h = \frac{1}{\omega} \begin{bmatrix} \frac{4}{h^2} & & +\frac{1}{h} \\ & +\frac{4}{h^2} & +\frac{1}{h} \\ \frac{1}{h} & +\frac{1}{h} & \end{bmatrix}, \tag{9}$$

for Jacobi-type updating steps, with D_h the diagonal and L_h the lower triangular of A_h. In the light of higher computational cost, the triad-wise smoother seems to be a good alternative to the Vanka smoother, since it has no overlap and therefore reduces computational costs. In addition, this relaxation method has better parallelization properties. These characteristics are examined in detail in the next section. However, the unsymmetric coupled (triad-wise) smoother was observed to have poor smoothing and often led to divergence for the Stokes equations.[11]

In the following we analyze and describe the conditions that lead to poor smoothing of the triad-wise smoother. In addition, we investigate different versions of the triad-wise smoother and modify the relaxation method to generate an efficient smoother with good smoothing properties.

3 | LFA FOR STOKES EQUATIONS

We employ LFA to analyze the convergence properties of multigrid methods for the Stokes equations and thereby compare the triad-wise and Vanka smoother.

3.1 | Definitions and notations

In order to describe LFA for staggered grids, we first introduce some terminology. More details can be found, for example, in.[3] We consider two-dimensional infinite uniform grids $G_h = G_h^1 \cup G_h^2 \cup G_h^3$ where

$$G_h^j = \{\boldsymbol{x}^j := \boldsymbol{kh} + \delta^j, \boldsymbol{k} \in \mathbb{Z}^2\}, \quad \text{with } \delta^j = \begin{cases} (0, h/2) & \text{if } j = 1, \\ (h/2, 0) & \text{if } j = 2, \\ (h/2, h/2) & \text{if } j = 3. \end{cases} \tag{10}$$

Here, the grids that correspond to the velocity unknowns u and v, are denoted by G_h^1 and G_h^2. Moreover, the grid that corresponds to pressure unknown p is denoted by G_h^3. The coarse grid, G_H, is constructed by doubling the mesh size, that is, we use standard coarsening. The functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$ are called Fourier modes or Fourier functions with frequency $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and form the basis of the Fourier space $\mathcal{F}_h$. Under general assumptions, each multigrid operator can be replaced by an operator with constant coefficients. Therefore, Fourier modes serve as eigenfunctions of multigrid operators and can be used to analyze the effect of each multigrid component on different modes. For staggered grids, the basis of the Fourier space $\mathcal{F}_h$ is given by the span$\{\varphi^1(\boldsymbol{\theta}, \boldsymbol{x}), \varphi^2(\boldsymbol{\theta}, \boldsymbol{x}), \varphi^3(\boldsymbol{\theta}, \boldsymbol{x})\}$ for $\boldsymbol{\theta} \in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^2$, with

$$\varphi^1(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{pmatrix} e^{\iota \boldsymbol{\theta} \cdot \boldsymbol{x}^1 / \boldsymbol{h}} & 0 & 0 \end{pmatrix}^T, \quad \varphi^2(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{pmatrix} 0 & e^{\iota \boldsymbol{\theta} \cdot \boldsymbol{x}^2 / \boldsymbol{h}} & 0 \end{pmatrix}^T,$$

$$\varphi^3(\boldsymbol{\theta}, \boldsymbol{x}) = \begin{pmatrix} 0 & 0 & e^{\iota \boldsymbol{\theta} \cdot \boldsymbol{x}^3 / \boldsymbol{h}} \end{pmatrix}^T,$$

where $\iota^2 = -1$ and $e^{\iota \boldsymbol{\theta} \cdot \boldsymbol{x} / \boldsymbol{h}} := e^{\iota \theta_1 x_1 / h_1} e^{\iota \theta_2 x_2 / h_2}$. We use the stencil notation and consider the operator A_h of the Stokes system,

$$A_h = \begin{pmatrix} -\Delta_h & & (\partial_{x_1})_{h/2} \\ & -\Delta_h & (\partial_{x_2})_{h/2} \\ -(\partial_{x_1})_{h/2} & -(\partial_{x_2})_{h/2} & \end{pmatrix},$$

with stencils

$$-\Delta_h = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}, \quad (\partial_{\mathrm{x}_1})_{h/2} = \frac{1}{h}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad (\partial_{\mathrm{x}_2})_{h/2} = \frac{1}{h}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

Note that for all Fourier modes $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$,

$$A_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}) = \tilde{A}_h \varphi(\boldsymbol{\theta}, \boldsymbol{x}).$$

Definition 1. We call $\tilde{A}_h$ the symbol of A_h.

We assume that the system matrix A_h is a block matrix, where each block is an infinite-grid Toeplitz matrix. Then, each block in A_h can be diagonalized with LFA. Each entry in the symbol $\tilde{A}_h$ is computed as the (scalar) symbol of the corresponding block of A_h. Since A_h is a 3×3 block operator, its symbol is a 3×3 matrix,

$$\tilde{A}_h(\boldsymbol{\theta}) = \begin{pmatrix} \frac{1}{h^2}(4 - 2\cos\theta_1 - 2\cos\theta_2) & 0 & 2\imath h \sin\frac{\theta_1}{2} \\ 0 & \frac{1}{h^2}(4 - 2\cos\theta_1 - 2\cos\theta_2) & 2\imath h \sin\frac{\theta_2}{2} \\ -2\imath h \sin\frac{\theta_1}{2} & -2\imath h \sin\frac{\theta_2}{2} & 0 \end{pmatrix}. \tag{11}$$

High and low frequencies for standard coarsening are given by

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

Remark 1. The influence of boundary conditions is not taken into account when applying LFA, since the modes $\varphi_h(\boldsymbol{\theta}, \boldsymbol{x})$ are defined on the infinite grid G_h. Experience with LFA shows that it often serves as an accurate prediction tool for problems with periodic boundary conditions, but degradation in performance may be seen with Dirichlet boundary conditions.[16]

We start with a two-grid analysis and compare results for both smoothers.

3.2 | Two-grid LFA

We start the analysis of the two-grid method with the smoothing procedures. This is followed by an introduction of coarse-grid correction symbols.

Due to the overlapping cells of the Vanka smoother, certain degrees of freedom are updated multiple times over the course of a single sweep of relaxation. As a result, the iteration matrix will not be a Toeplitz matrix, a block Toeplitz representation as for red-black Gauss–Seidel is also not readily available for overlapping smoothers, at least not with lexicographic ordering. Classical LFA is based on the assumption that the iteration matrix M_h can be written as a (block) Toeplitz operator. Therefore, classical LFA techniques fail.

An extension of the classical LFA analysis for the Vanka smoothers was first introduced by Sivaloganathan.[20] MacLachlan and Oosterlee generalized the LFA techniques to analyze overlapping smoothers for other PDEs and discretizations.[21] Their paper provides theoretical results in order to use the Fourier ansatz. A necessary assumption is that the error-propagation operator for coupled (overlapping) relaxation is block-diagonalized by the Fourier matrix, regardless of the distribution of degrees of freedom within the block. The key step in proving this is to show the following inductive step: if the errors before relaxation on degrees of freedom within a block $V_{i,j}$ satisfy a generalized LFA ansatz, then the errors after relaxation on $V_{i,j}$ satisfy the same ansatz advanced by one cell. MacLachlan and Oosterlee show that the error-propagation matrix for any coupled relaxation is an infinite-grid block-multilevel-Toeplitz matrix.[21] This means that we can attempt to analyze these techniques using classical multigrid smoothing and two-grid Fourier analysis tools to measure the effectiveness of the resulting multigrid cycles.

Practical guidelines to obtain the smoothing symbol $\tilde{S}_h(\boldsymbol{\theta})$ of the Vanka smoother to analyze this relaxation method are provided in Reference 12. For more details and a generalization to other overlapping smoothers, we refer to Reference 21.

Due to the nonoverlapping blocks, the LFA analysis for the triad-wise smoother is based on a simple expansion of the assumptions of LFA for scalar PDEs as described in Section 3.1. The error-propagation operator for triad-type smoothers, represented by an operator M_h is

$$\mathcal{S}_h(\omega) = I - M_h^{-1} A_h.$$

The corresponding smoothing symbol $\tilde{S}_h(\omega, \boldsymbol{\theta})$ is given by

$$\tilde{S}_h(\omega, \boldsymbol{\theta}) = I - \tilde{M}_h^{-1}(\boldsymbol{\theta}) \tilde{A}_h(\boldsymbol{\theta}),$$

with I the identity matrix and $\tilde{A}_h(\boldsymbol{\theta})$ as given in (11). The symbol of the block iteration matrix $\tilde{M}_h$ can be given by

$$\tilde{M}_h(\boldsymbol{\theta}) = \begin{pmatrix} \frac{1}{h^2}\left(\frac{4}{\omega} - e^{-\imath\theta_1} - e^{-\imath\theta_2}\right) & 0 & \frac{1}{h}\left(\frac{1}{\omega} e^{\frac{1}{2}\imath\theta_1} - e^{-\frac{1}{2}\imath\theta_1}\right) \\ 0 & \frac{1}{h^2}\left(\frac{4}{\omega} - e^{-\imath\theta_1} - e^{-\imath\theta_2}\right) & \frac{1}{h}\left(\frac{1}{\omega} e^{\frac{1}{2}\imath\theta_2} - e^{-\frac{1}{2}\imath\theta_2}\right) \\ \frac{1}{\omega h} e^{-\frac{1}{2}\imath\theta_1} & \frac{1}{\omega h} e^{-\frac{1}{2}\imath\theta_2} & 0 \end{pmatrix},$$

for a weighted Gauss–Seidel updating process and by

$$\tilde{M}_h(\boldsymbol{\theta}) = \frac{1}{\omega} \begin{pmatrix} \frac{4}{h^2} & 0 & \frac{1}{h} e^{\frac{1}{2}\imath\theta_1} \\ 0 & \frac{4}{h^2} & \frac{1}{h} e^{\frac{1}{2}\imath\theta_2} \\ \frac{1}{h} e^{-\frac{1}{2}\imath\theta_1} & \frac{1}{h} e^{-\frac{1}{2}\imath\theta_2} & 0 \end{pmatrix},$$

for a weighted Jacobi updating process.

For systems of PDEs that are discretized on staggered grids, the analysis of the coarse grid correction is challenging due to the different location of unknowns. In that case, LFA of transfer operators cannot be done as in the scalar case. MacLachlan and Oosterlee[21] discussed Fourier representations of grid-transfer operators for general staggered meshes in the context of systems of PDEs. Here, we have three types of grid points on the fine and coarse grid. The restriction operator can be decomposed based on the partitioning of degrees of freedom (DOFs) associated with the location of the unknowns on the grid. Osterlee and MacLachlan introduced, explained and proved the need to modify LFA due to this staggering. We emphasize this statement in Remark 2. More details can be found in References 18,21,22.

Remark 2. When calculating the symbol of a restriction operator that mixes different types of DOFs, we must split it into the different types of DOFs that it restricts from and to. If the restriction operator is defined on a staggered grid, we have $G_H = G_H^1 \cup G_H^2 \cup G_H^3$ (cf, (10)) and the symbols depend on the location $\boldsymbol{x}$ of the unknowns.

Definition 2. The restriction operators for the Stokes equations on staggered grids are given by

$$R_h^H\Big|_u = \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ & * & \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix}, \quad R_h^H\Big|_v = \begin{bmatrix} \frac{1}{8} & & \frac{1}{8} \\ \frac{1}{4} & * & \frac{1}{4} \\ \frac{1}{8} & & \frac{1}{8} \end{bmatrix}, \quad R_h^H\Big|_p = \begin{bmatrix} \frac{1}{4} & & \frac{1}{4} \\ & * & \\ \frac{1}{4} & & \frac{1}{4} \end{bmatrix},$$

where $*$ denotes the resulting coarse-grid point.

In the next paragraph, we show how to determine the symbol of a restriction operator defined on a staggered mesh. A similar structure for interpolation is established subsequently. Let $\varphi_h(\boldsymbol{\theta}^{\xi}, \boldsymbol{x}) = e^{\imath\boldsymbol{\theta}^{\xi}\boldsymbol{x}/\mathbf{h}}$, with $\boldsymbol{\theta}^{\xi} = \boldsymbol{\theta} + \boldsymbol{\xi}\pi$, where $\boldsymbol{\xi} = (0,0), (1,1), (1,0), (0,1)$. We have the following equality

$$\varphi_h(\boldsymbol{\theta}^{\xi}, \boldsymbol{x}) = e^{\imath\boldsymbol{\xi}\pi\boldsymbol{x}/\mathbf{h}} \varphi_H(2\boldsymbol{\theta}^{(\mathbf{0},\mathbf{0})}, \boldsymbol{x}), \quad \text{for all } \boldsymbol{x} \in G_H.$$

Note that this equality is different to the classical theory and was developed by MacLachlan and Oosterlee.[21] The following relations are based on this update of the classical theory. Due to the definition of $G_H = G_H^1 \cup G_H^2 \cup G_H^3$, we have

$$\boldsymbol{x} = \begin{cases} \left(2jh, \quad 2(l+\frac{1}{2})h\right)^T, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^1, \\ \left(2(j+\frac{1}{2})h, \quad 2lh\right)^T, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^2, \\ \left(2(j+\frac{1}{2})h, \quad 2(l+\frac{1}{2})h\right)^T, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^3. \end{cases}$$

Note that the following relations hold,

$$e^{\iota\xi\pi\boldsymbol{x}/\mathbf{h}} = \begin{cases} e^{\iota\xi_1\pi 2j}e^{\iota\xi_2\pi 2l}e^{\iota\xi_2\pi} = e^{\iota\pi\xi_2} = (-1)^{\xi_2}, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^1, \\ e^{\iota\xi_1\pi 2j}e^{\iota\xi_1\pi}e^{\iota\xi_2\pi 2l} = e^{\iota\pi\xi_1} = (-1)^{\xi_1}, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^2, \\ e^{\iota\pi\xi_1}e^{\iota\pi\xi_2} = (-1)^{\xi_1}(-1)^{\xi_2}, & \text{for } j,l \in \mathbb{Z}, \text{ if } \boldsymbol{x} \in G_H^3. \end{cases}$$

Then, the Fourier representation of R_h^H is given by the (3×12) matrix

$$\hat{R}_h^H(\boldsymbol{\theta}) = \left(\tilde{R}_h^H(\boldsymbol{\theta}^{(0,0)}) \quad \tilde{R}_h^H(\boldsymbol{\theta}^{(1,1)}) \quad \tilde{R}_h^H(\boldsymbol{\theta}^{(1,0)}) \quad \tilde{R}_h^H(\boldsymbol{\theta}^{(0,1)})\right),$$

where $\tilde{R}_h^H(\boldsymbol{\theta}^{\xi})$ is given by

$$\tilde{R}_h^H(\boldsymbol{\theta}^{\xi}) = \begin{pmatrix} \tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_u & & \\ & \tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_v & \\ & & \tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_p \end{pmatrix},$$

with $\tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_u := \sum_{\boldsymbol{\kappa}\in\mathbf{V}} \mathrm{r}_{\boldsymbol{\kappa}} e^{\iota\boldsymbol{\theta}^{\xi}\cdot\boldsymbol{\kappa}} e^{\iota\xi\pi\boldsymbol{x}/\mathbf{h}}\Big|_u$, $\tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_v := \sum_{\boldsymbol{\kappa}\in\mathbf{V}} \mathrm{r}_{\boldsymbol{\kappa}} e^{\iota\boldsymbol{\theta}^{\xi}\cdot\boldsymbol{\kappa}} e^{\iota\xi\pi\boldsymbol{x}/\mathbf{h}}\Big|_v$ and $\tilde{R}_h^H(\boldsymbol{\theta}^{\xi})\Big|_p := \sum_{\boldsymbol{\kappa}\in\mathbf{V}} \mathrm{r}_{\boldsymbol{\kappa}} e^{\iota\boldsymbol{\theta}^{\xi}\cdot\boldsymbol{\kappa}} e^{\iota\xi\pi\boldsymbol{x}/\mathbf{h}}\Big|_p$, where $\boldsymbol{\kappa}$ is the multi-index indicating the position within the stencil and $r_{\boldsymbol{\kappa}}$ denotes the entry at this position. Note that the matrices $\tilde{R}_h^H(\boldsymbol{\theta}^{(\mathbf{0,0})}), \tilde{R}_h^H(\boldsymbol{\theta}^{(\mathbf{1,1})}), \tilde{R}_h^H(\boldsymbol{\theta}^{(\mathbf{1,0})}), \tilde{R}_h^H(\boldsymbol{\theta}^{(\mathbf{0,1})})$ are diagonal matrices since the restriction operators of different type of unknowns are not coupled.

Definition 3. We call $\tilde{R}_h^H(\boldsymbol{\theta}^{\xi})$ the symbol of R_h^H.

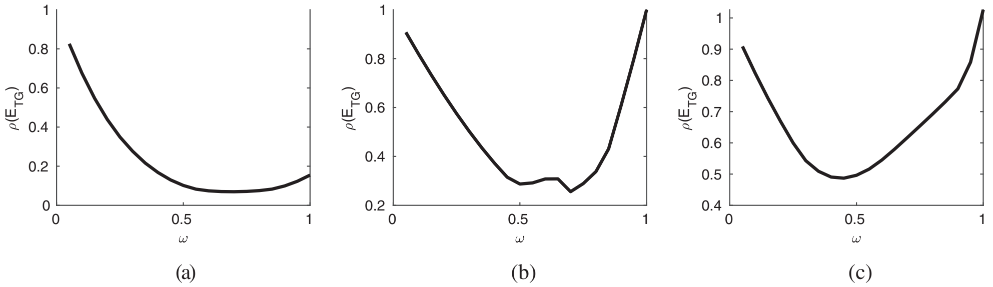
Based on Definition 2, we are able to compute the entries of $\hat{R}_h^H(\boldsymbol{\theta})$. We show the computation of the symbol for the restriction operator $R_h^H\Big|_u$, that is, the first row of the matrix $\hat{R}_h^H(\boldsymbol{\theta})$.

$$\begin{aligned} \tilde{R}_h^H(\boldsymbol{\theta}^{(0,0)})\Big|_u &= \frac{1}{8}e^{-\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{4}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{-\iota\theta_1}e^{\frac{1}{2}\iota\theta_2} + \frac{1}{4}e^{\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{\iota\theta_1}e^{\frac{1}{2}\iota\theta_2}, \\ \tilde{R}_h^H(\boldsymbol{\theta}^{(1,1)})\Big|_u &= -\frac{1}{8}e^{-\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{4}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{-\iota\theta_1}e^{\frac{1}{2}\iota\theta_2} - \frac{1}{4}e^{\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{\iota\theta_1}e^{\frac{1}{2}\iota\theta_2}, \\ \tilde{R}_h^H(\boldsymbol{\theta}^{(1,0)})\Big|_u &= \frac{1}{8}e^{-\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{4}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{-\iota\theta_1}e^{\frac{1}{2}\iota\theta_2} + \frac{1}{4}e^{\frac{1}{2}\iota\theta_2} + \frac{1}{8}e^{\iota\theta_1}e^{\frac{1}{2}\iota\theta_2}, \\ \tilde{R}_h^H(\boldsymbol{\theta}^{(0,1)})\Big|_u &= -\frac{1}{8}e^{-\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{4}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{\iota\theta_1}e^{-\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{-\iota\theta_1}e^{\frac{1}{2}\iota\theta_2} - \frac{1}{4}e^{\frac{1}{2}\iota\theta_2} - \frac{1}{8}e^{\iota\theta_1}e^{\frac{1}{2}\iota\theta_2}. \end{aligned}$$

The calculations for the restriction operators $R_h^H\Big|_v$ and $R_h^H\Big|_p$ are similar. In addition, an equivalent calculation (see the work of MacLachlan and Oosterlee[21]) gives the symbols of interpolation operators $\hat{P}_H^h(\boldsymbol{\theta})$.

Definition 4. The Fourier representation of the coarse-grid operator A_H is given by $\tilde{A}_H(2\boldsymbol{\theta}^{(0,0)})$ such that $A_H\varphi_H(2\boldsymbol{\theta}^{(0,0)},\boldsymbol{x}) = \tilde{A}_H(2\boldsymbol{\theta}^{(0,0)})\varphi_H(2\boldsymbol{\theta}^{(0,0)},\boldsymbol{x})$.

We analyze the two-grid method based on the smoothing operators as introduced in the previous section. Based on the updated restriction and prolongation operators, we perform the analysis of the two-grid method with triad-wise and

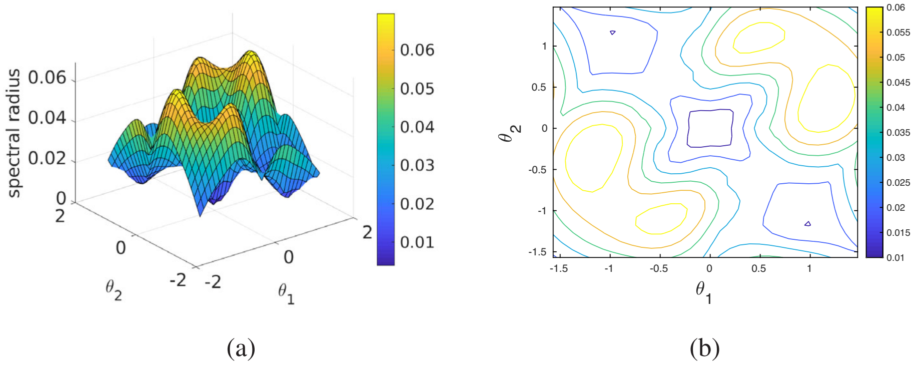


(a)

(b)

(c)

FIGURE 4 Parameter study of the two-grid method for different smoothers. (a) Vanka smoother; (b) triad-Gauss-Seidel smoother; (c) triad-Jacobi smoother



(a)

(b)

FIGURE 5 The spectral radius of the two-grid symbol for the Vanka smoother with $\omega = 0.7$ and different frequencies $\boldsymbol{\theta}$.

Vanka smoothing procedures. The asymptotic two-grid convergence factor is given by the spectral radius

$$\rho(E_{TG}) := \sup_{\boldsymbol{\theta} \in T^{\text{low}}} \rho\left(\tilde{E}_{TG}(\boldsymbol{\theta})\right),$$

where

$$\tilde{E}_{TG}(\boldsymbol{\theta}) = \left(\tilde{S}_h(\boldsymbol{\theta})\right)^{v_2} \left(\tilde{I} - \hat{P}_H^h(\boldsymbol{\theta})\left(\tilde{A}_H(2\boldsymbol{\theta})\right)^{-1} \hat{R}_h^H(\boldsymbol{\theta})\tilde{A}_h(\boldsymbol{\theta})\right) \left(\tilde{S}_h(\boldsymbol{\theta})\right)^{v_1},$$

with v_1 number of presmoothing steps and v_2 number of post-smoothing steps.

3.3 | Analysis results

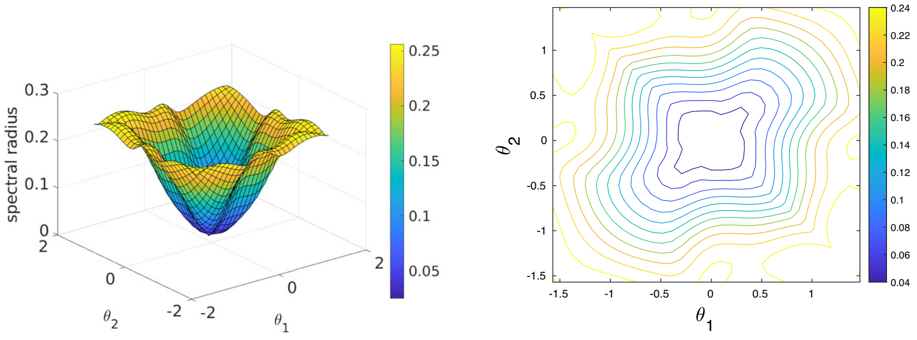
In this section, we present LFA two-grid results with a focus on bilinear interpolation and a Galerkin coarse grid operator. A comparison in Reference 12 shows that the combination of these operators promises best results for the Stokes equations. We compare the results obtained by Vanka and triad-wise smoothers. We use a 33×33 staggered grid and two pre- and postsmoothing steps ($v := v_1 = v_2 = 2$) during the course of this section. Figure 4 presents the LFA convergence factors for the Vanka and triad-wise smoother as a function of the weighting parameter ω. This shows the sensitivity of performance to parameter choice.

On the basis of these parameter studies, compare Figure 4, we figured out that the weighting factor $\omega = 0.7$ gives the most promising smoothing behavior for the Vanka and triad-Gauss–Seidel (GS) smoother. While for the triad-Jacobi updating process, we stick to the weighting factor $\omega = 0.45$ in the following. As visualized in Figures 5,6, and 7, the above mentioned combination of MG components lead to the convergence factor

$$\rho(E_{\text{TG}}) = 0.08,$$

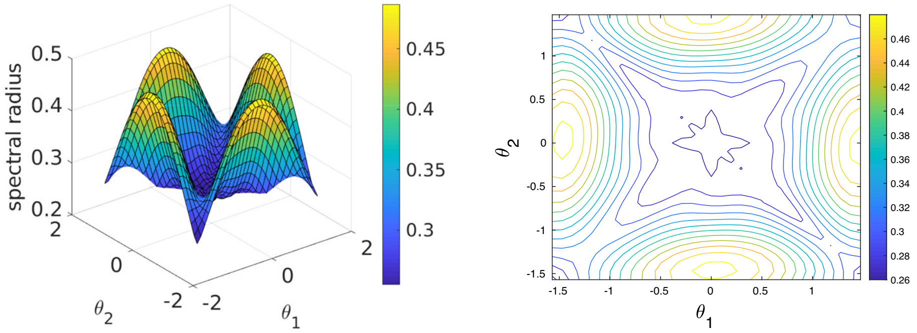
for the Vanka smoother,

$$\rho(E_{\text{TG}}) = 0.26,$$



(a) (b)

FIGURE 6 The spectral radius of the two-grid symbol for the triad-wise relaxation based on a weighted Gauss–Seidel updating process with $\omega = 0.7$ and different frequencies $\boldsymbol{\theta}$



(a) (b)

FIGURE 7 The spectral radius of the two-grid symbol for the triad-wise relaxation based on a weighted Jacobi updating process with $\omega = 0.45$ and different frequencies $\boldsymbol{\theta}$

for the triad-GS smoother, and

$$\rho(E_{\mathrm{TG}}) = 0.49,$$

for the triad-Jacobi smoother.

The results of the triad-wise smoother are based on the choice of the three unknowns that correspond to one triad block, we use a ⌉-shape as seen in Figure 3. Alternative shapes like Γ-shape, L-shape and ⌋-shape lead to similar convergence factors, 0.30, 0.27, and 0.30, respectively. We stick to the ⌉-shape in this paper if not specified otherwise. The performance predicted by LFA shows the potential of the triad-wise smoother. The convergence factor shows that the Vanka smoother serves as a great smoother. However, the triad-wise smoother is still promising because of its lower computational cost and better parallelization potential in combination with a good convergence factor.

4 | NUMERICAL EXAMPLES

In this section, we first present convergence results of a two-grid and a V-cycle multigrid method for the Stokes equations. We start with periodic boundary conditions and observe convergence behavior as predicted by LFA in the previous section. A two-grid method for a homogeneous problem with periodic boundary conditions based on a 33×33 staggered grid is employed. We apply 20 two-grid cycles with two pre- and postsmoothing steps ($\nu = 2$) and start with a random initial guess. We measure the convergence via the 2-norm of the error, that is, we compute $\frac{||\mathbf{e}^k||_2}{||\mathbf{e}^{k-1}||_2}$ after $k = 20$ two-grid cycles. This gives a convergence factor of $\rho = 0.08$ for the Vanka smoother with $\omega = 0.7$, $\rho = 0.24$ for the weighted ($\omega = 0.7$) triad-GS smoother and $\rho = 0.43$ for the weighted ($\omega = 0.45$) triad-Jacobi smoother. The numerical results confirm the LFA convergence results as given in Section 3.3. In addition, Figure 8 shows convergence results for the Vanka and the triad-wise smoothers for the multigrid method. We use the multiplictive Vanka smoother in this paper. In comparision, the additive Vanka smoother leads to a convergence factor of $\rho = 0.26$ for periodic boundary conditions in comparison to 0.08 for the multiplicative Vanka smoother. Patch-wise procedures for the additive Vanka smoother improve this convergence

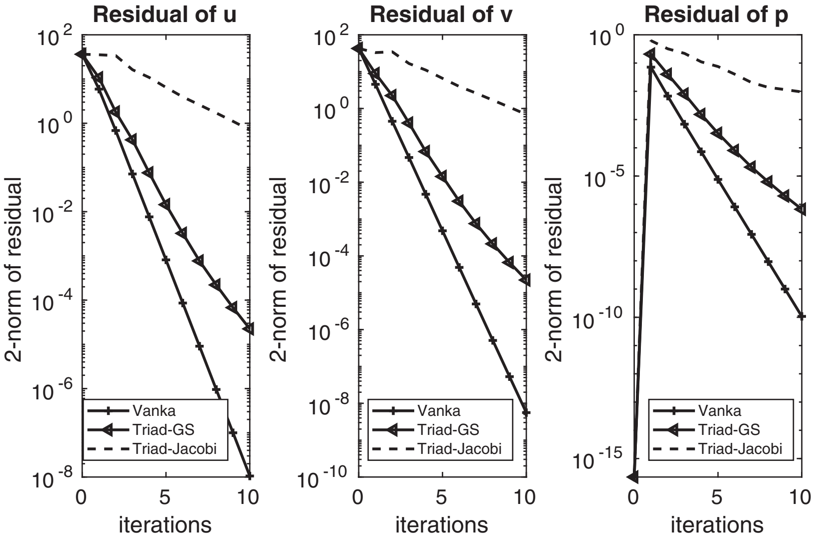


FIGURE 8 Convergence behavior of the multigrid method for the Stokes system with periodic boundary conditions

TABLE 1 Convergence factors for different triad-type relaxation methods with Dirichlet boundary conditions.

	ω-GS (forward)	ω-GS (backward)	ω-GS-RB (backward)	ω-GS, $\nu = 6$ (backward)
ρ	0.36	0.62	0.58	0.29

Abbreviation: GS, Gauss–Seidel; RB, red black.

factors as shown in Reference 10. In this paper, we stick to the multiplicative Vanka smoother and use its convergence factor as benchmark for the triad-wise method.

Figure 8 is based on a 33×33 staggered fine grid and a 3×3 coarse grid on the domain $\Omega = (0, 1)^2$ discretized using the MAC scheme. We use two pre- and postsmoothing steps ($\nu = 2$) and a zero initial guess. The right-hand side is set to

$$\mathbf{f}_1 = 8\pi^2 \cdot \sin(2\pi \mathrm{x}_1^1) \cdot \sin(2\pi \mathrm{x}_2^1) - 2\pi \cdot \sin(2\pi \mathrm{x}_1^1) \cdot \sin(2\pi \mathrm{x}_2^1),$$
$$\mathbf{f}_2 = 8\pi^2 \cdot \cos(2\pi \mathrm{x}_1^2) \cdot \cos(2\pi \mathrm{x}_2^2) + 2\pi \cdot \cos(2\pi \mathrm{x}_1^2) \cdot \cos(2\pi \mathrm{x}_2^2),$$

where the values x^j correspond to the grid points of the discretized domain $\Omega_h^j = \{\boldsymbol{x}^j := \boldsymbol{k}\mathbf{h} + \delta^j, \boldsymbol{k} \in \mathbb{Z}^2\}$, with $\delta^1 = (0, h/2)$ and $\delta^2 = (h/2, 0)$. Figure 8 presents the convergence factors corresponding to the different unknowns u, v, p individually. The convergence factors are obtained by computing the 2-norm of the residual for different numbers of multigrid V-cycles (iterations) for the Stokes equations, that is, we compute $||\mathbf{r}^k||_2$ after each V-cycle. We notice good convergence factors for the Vanka and the triad-GS relaxation methods. However, the Vanka smoother converges even faster than the triad-GS smoother. The triad-Jacobi procedure does not lead to a good convergence rate.

Next, we show convergence results for the Stokes system with Dirichlet boundary conditions. For the triad-wise smoother we focus on a Gauss-Seidel updating process, since we have seen that the triad-Jacobi smoother does not lead to good convergence results. For Dirichlet boundary conditions triad-Jacobi downgrades to a divergent method. Again, practical two-grid convergence factors for a homogeneous problem discretized on a 33×33 grid are given. We apply 20 two-grid cycles, set $\nu = 2$ and start with a random initial guess. That gives a convergence of $\rho = 0.36$ for the triad-wise smoother and $\rho = 0.10$ for the Vanka smoother. Again, the Vanka smoother leads to good convergence results. In contrast, the triad smoother does not show satisfactory convergence for Dirichlet boundary conditions. For this reason, we investigate different possibilities to adapt the smoother. Instead of using a lexicographical updating process, that is, forward direction, we compute the convergence rate for an updating process in backward direction as well as with a red-black(RB)-coloring. Note that an updating process in backward direction is the same as an L-shape updating process. Moreover, we investigate the possibility to use additional smoothing steps. The results can be found in Table 1.

Table 1 shows that a different ordering like an updating process in backward direction or a RB-coloring worsens the convergence. Results for a Γ-shape or ⌋-shape are similar to the results of the updating process in backward direction, that

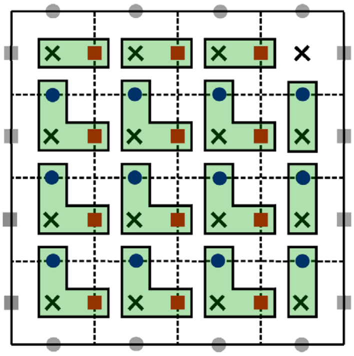
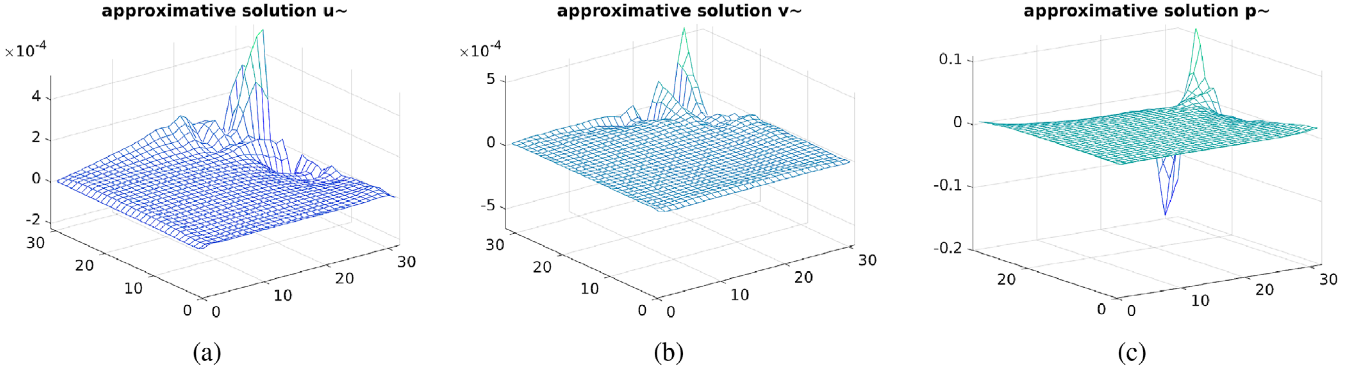


FIGURE 9 Examplary choice of triad blocks with Dirichlet boundary conditions



(a) (b) (c)

FIGURE 10 Approximate solution of the Stokes equations with Dirichlet boundary conditions after 20 multigrid cycles based on the triad smoother

is, the convergence rate in forward as well as in backward direction for a Γ-shape or a ⌋-shape is $\rho = 0.58$. An increase in the number of smoothing steps leads to better convergence results. However, we obtain a convergence factor of $\rho = 0.29$ if we apply six pre- and postsmoothing steps. This is as expensive as using the Vanka smoother but less efficient in terms of the convergence rate.

The poor convergence behavior of the triad-wise smoother results from the boundary treatment, as it does not treat the unknowns at the boundary (or near the boundary) in an appropriate way. Since the unknowns at the boundary are predefined, they are not treated by the relaxation. The idea of the triad-wise smoother is to relax three unknowns at the same time, which is not possible with Dirichlet boundary conditions regardless of the choice of the three unknowns. An illustration with an exemplary choice of triad blocks is given in Figure 9.

We notice the blocks include three unknowns whenever possible. Close to the boundary we have blocks with two unknowns only. At one corner we have one unknown only. Figure 10 illustrates the incorrect treatment by a visualization of the approximate solution after 20 multigrid cycles.

This observation leads to the following three adaptations of the boundary treatment that are able to improve the smoother but still do not lead to satisfying results. First, after each triad-wise-smoothing cycle, we add an additional smoothing at the critical corner by using an overlapping block as visualized in Figure 11(a). This gives a convergence factor of $\rho = 0.41$. By adding overlapping blocks not only at the corner but also at the boundaries, see Figure 11(a), we get a convergence factor of $\rho = 0.37$. By combining the triad updating process at the interior of the grid with the Vanka smoother at the boundary cells, see Figure 11(c), we improve the convergence factor to $\rho = 0.22$.

In addition, we investigate results for a multigrid method as depicted in Figure 12. We focus on the Vanka smoother and two exemplary triad-wise smoothers, compare Figure 11(b) and (c). We discretize on a 33×33 grid with the domain $\Omega = (0, 1)^2$, use two pre- and postsmoothing steps ($\nu = 2$) and employ the zero solution as an initial guess.

The right-hand side is set to

$$\mathbf{f}_1 = 2\pi^2 \cdot \sin(\pi x_1^1) \cdot \sin(\pi x_2^1) + \pi \cdot \cos(\pi x_1^1),$$
$$\mathbf{f}_2 = 2\pi^2 \cdot \cos(\pi x_1^2) \cdot \cos(\pi x_2^2) - \pi \cdot \sin(\pi x_2^2).$$

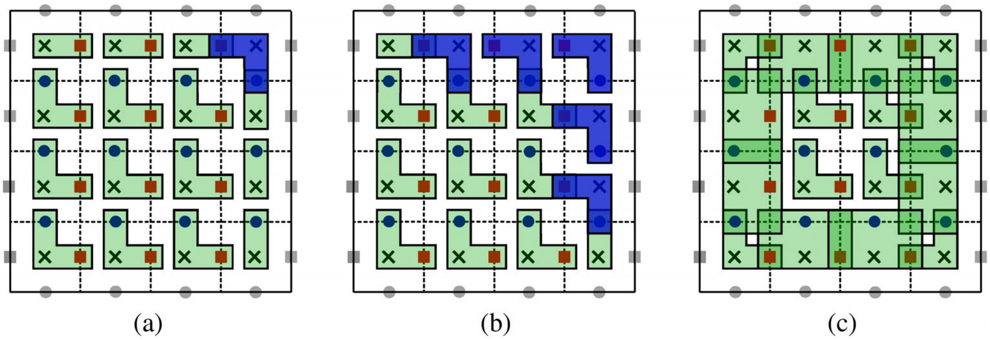


FIGURE 11 Triad updating blocks with additional overlapping blocks

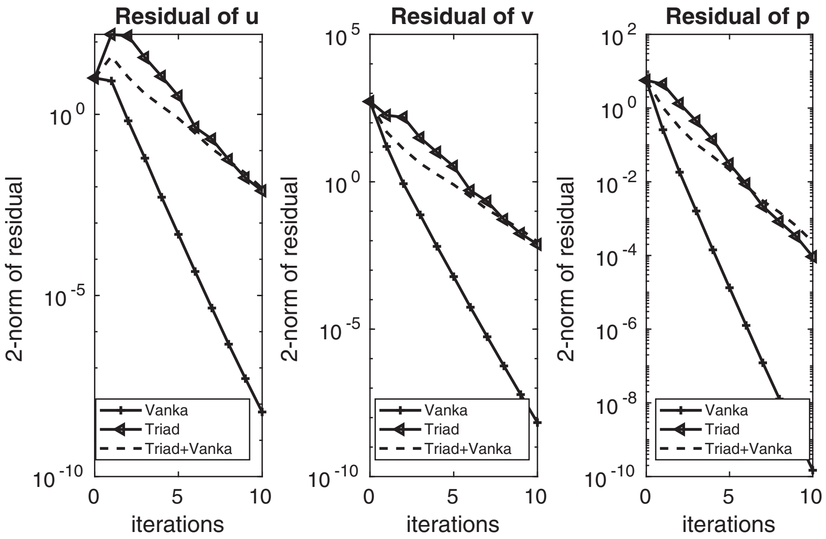


FIGURE 12 Convergence behavior of the multigrid method for the Stokes system with Dirichlet boundary conditions

We choose zero boundary conditions for the x-component u and $\pm\cos(\pi y)$ and $\pm\cos(\pi x)$ for the y-component v of the velocity unknown. Figure 12 shows good convergence for the Vanka smoother (similar to that in the periodic case). In contrast, none of the triad-wise smoothing methods show satisfactory convergence for Dirichlet boundary conditions.

5 | ALGORITHM DEVELOPMENT

The results of the LFA, compare Section 3.3, and the numerical results for periodic boundary conditions, c.f. Section 4, in combination with the better parallelization and the reduced computational work demonstrate the potential of the triad-wise relaxation method. However, the numerical results for the Stokes system with Dirichlet boundary conditions indicate the issue of the triad-wise method. In what follows, we show how to fix this issue. For this purpose more computational work is needed. The idea is to repeat the relaxation process four times while changing the unknowns contained in one block after each iteration. This idea is illustrated in Figure 13. An exemplary order of the four iterations is visualized from left to right, that is, Figure 13(a) illustrates the first iteration and Figure 13(d) the last one. As we will see in the following section, varying the order of the four iterations leads to different convergence factors.

5.1 | LFA results

In this section, we present LFA two-grid results with a focus on bilinear interpolation and a Galerkin coarse grid operator. We compare the results obtained by the Vanka smoother and different versions of the fourfold triad smoother. We use a 33×33 staggered grid and two pre- and postsmoothing steps ($\nu := \nu_1 = \nu_2 = 2$) during the course of this section. We stick

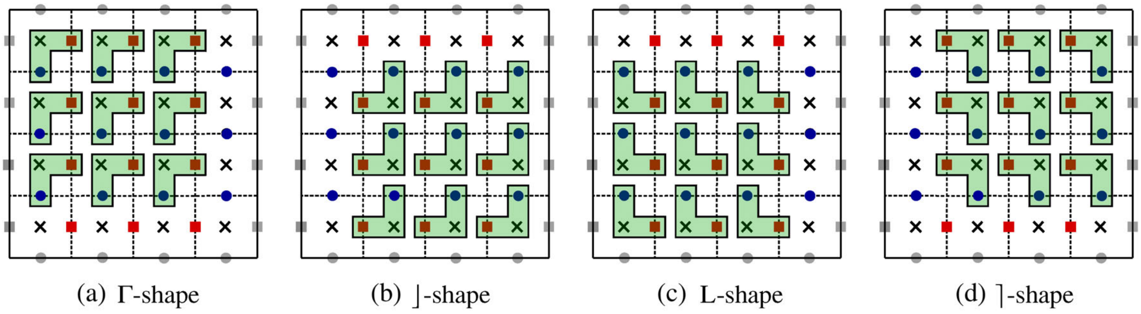

(a) Γ-shape (b) ⌋-shape (c) L-shape (d) ⌉-shape

FIGURE 13 New algorithm for the triad relaxation method. The order of the procedure is performed as visualized in the figures from left to right. (a) Γ-shape; (b) ⌋-shape; (c) L-shape; (d) ⌉-shape

	Vanka	Γ, ⌋, L, ⌉,	Γ, ⌋, ⌉, L	Γ, L, ⌋, ⌉,	⌋, L, ⌉, Γ	⌉, ⌋, L, Γ
ρ	0.08	0.21	0.17	0.08	0.21	0.08

TABLE 2 Convergence factors of the two-grid method with Vanka and different fourfold triad-type relaxation methods based on local Fourier analysis

to the weighting factor $\omega = 0.7$ as indicated in Section 3.3. The above mentioned combination of MG components leads to the following convergence factor for Vanka and different versions of the fourfold triad smoother, as visualized in Table 2.

Table 2 displays convergence factors obtained by LFA. We include a range of choices for the order of the fourfold triad smoother as well as results for the Vanka smoother. It is obvious that varying the order of the four iterations for the fourfold triad smoother changes the convergence factors drastically. The best variant leads to a convergence of $\rho = 0.08$ which equals the convergence of the two-grid method that utilizes the Vanka smoother. These LFA results demonstrate that the new fourfold algorithm has potential to improve the triad-wise smoother.

5.2 | Numerical results

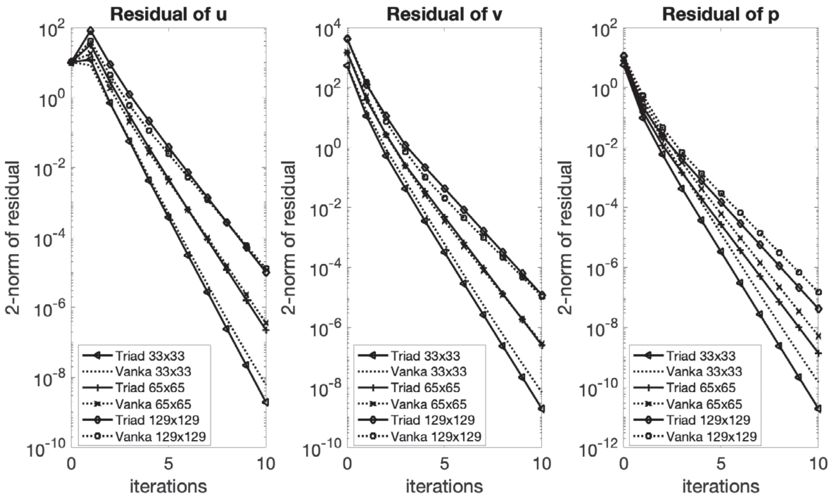
In this section, we present convergence results of a two-grid method as well as a V-cycle multigrid method for the Stokes equations using the fourfold triad smoother in comparison to the Vanka smoother. Section 5.1 illustrates that varying the order of the four iterations for the fourfold triad smoother changes the results significantly. However, numerical examples with Dirichlet boundary conditions allow to change the order in which the blocks of each iteration are updated. This leads to new insights. Examples to modify the order within each of the four iterations include a backward or forward as well as a column or row-wise updating process.

The two-grid convergence factor of the updated version of the triad-wise smoother for a homogeneous problem with Dirichlet boundary conditions are as follows. Starting with a naïve process of updating in lexicographical order, that is, row-wise starting with the bottom left block, during all four iterations leads already to promising convergence results of $\rho = 0.07$ for example for the Γ, ⌋, L, ⌉ updating process. We obtain an even better convergence rate of $\rho = 0.04$ for Γ, ⌋, ⌉, L. A more sophisticated updating process promises more stable results: a column-wise forward Γ-shape, followed by a column-wise backward ⌋-shape, followed by a row-wise forward L-shape and row-wise backward ⌉-shape updating process, where different kinds of ordering lead to the same convergence rate of $\rho = 0.04$.

Again, we applied 20 two-grid cycles with two pre- and postsmoothing steps starting with a random initial guess. Results of a multigrid method are visualized in Figure 14. We discretize on a variety of grid sizes, ranging from a 33×33 grid to a 129×129 grid each with the domain $\Omega = (0, 1)^2$. We use two pre- and postsmoothing steps and employ the zero solution as an initial guess. We stick to the Γ, ⌋, L, ⌉ ordering for the fourfold triad smoother with the sophisticated updating process.

The modified procedure causes four times more computational work in comparison with the results of the original version of the triad-wise smoother, compare Section 4. This leads to a higher number of arithmetic operations for the triad-wise smoother compared to the Vanka smoother. We have a total number of $139 \cdot (N - 1)^2$ for the Vanka smoother versus $176 \cdot (N - 1)^2$ operations for the updated triad smoother. However, the new triad procedure convergences even better than the Vanka smoother. We have $\rho = 0.04$ for the triad-wise smoother compared to $\rho = 0.10$ for the Vanka smoother.

FIGURE 14 Convergence behavior of the multigrid method using two presmoothing and two postsmoothing steps for the Stokes system with Dirichlet boundary conditions including the new triad algorithm.



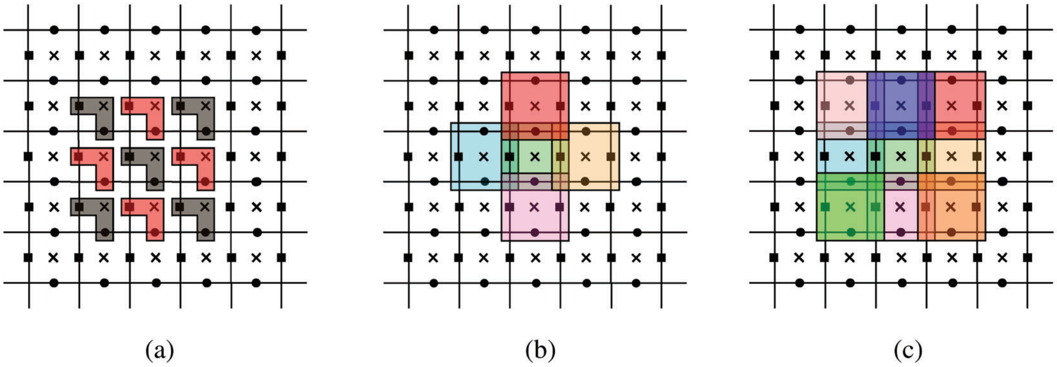


FIGURE 15 Coloring schemes. (a) The triad-wise smoother; (b) The Vanka smoother (5 color); (c) The Vanka smoother (9 color)

Each of the individual substeps of the modified method can be parallelized in the same way as a whole step of the nonmodified method. Nevertheless, the four steps are processed consecutively resulting in more synchronization steps.

6 | PARALLEL IMPLEMENTATION ASPECTS

So far, we have seen a comparison between an overlapping and a nonoverlapping smoother, analyzed their convergence behavior and potential for optimization. Applications for the Stokes equations result in systems of equations that are much larger than the test problems within this paper, that is, with several million unknowns. Consequently, a serial implementation is usually not sufficient. As a result, we compare the parallelization potential of the overlapping smoother with the nonoverlapping smoother in the following.

The obvious difference between the two smoothers is the number of unknowns and equations that are included in one block, while the number of blocks that have to be updated per relaxation is the same for both smoothers. The arithmetic operations that we need for one relaxation step are $139 \cdot N^2$ for the Vanka smoother compared to $44 \cdot N^2$ for the original version of the triad-wise smoother, where N describes the number of blocks. More details regarding the computation of the number of arithmetic operations can be found in Reference 12. Parallel implementation aspects that we take into account are the required coloring and the communication effort that is needed for computations in parallel.

The required colorings to perform the smoothing steps in parallel are visualized in Figure 15. The Vanka smoother needs five different colors, while the triad-wise smoother requires a red-black coloring (i.e., two colors), only.

An important parallelization aspect is the "dependencies," that is, all neighboring unknowns that are needed to update an unknown. To perform a method in parallel without read and write conflicts it is necessary to avoid simultaneous updates of dependent unknowns. The coloring scheme is based on depenencies. The unknowns of one triad block depend

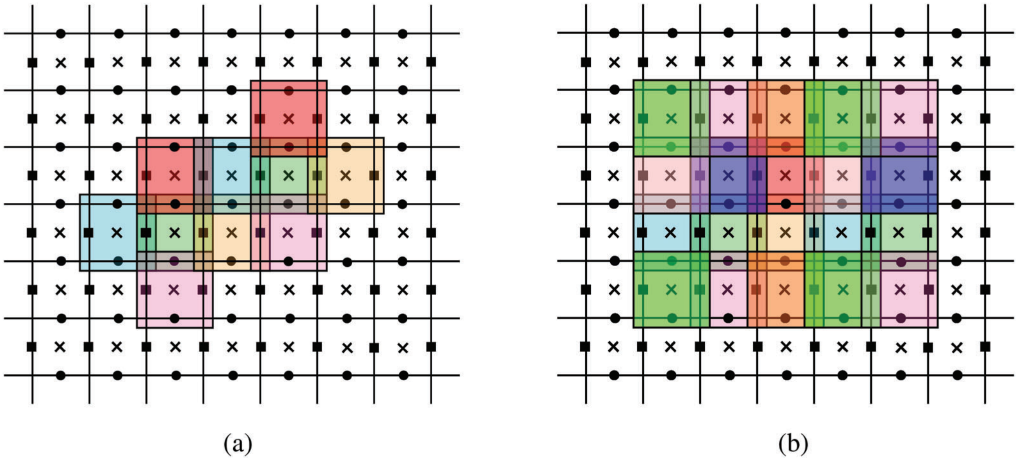


FIGURE 16 Repetition of the coloring pattern of the Vanka smoother. (a) The five color Vanka smoother; (b) The nine color Vanka smoother

on the unknowns of adjacent cells. Consequently, a red-black coloring is sufficient, compare Figure 15(a). That means all unknowns of the red triad blocks can be updated simultaneously while all unknowns of the black triad blocks can be updated simultaneously. While, the coloring scheme of the Vanka smoother is illustrated in Figure 15(b). Due to the additional unknowns that are included in one Vanka block compared to the triad block, the unknowns do not solely depend on the adjacent blocks. Due to the MAC scheme as given in Equations (5)–(7) a five coloring is sufficient to perform the Vanka smoother in parallel. The repetition of the coloring schemes are visualized in Figure 15(a) for the red-black pattern of the triad-wise smoother and in Figure 16(a) for the five-color Vanka pattern.

The communication effort differs due to the different numbers of unknowns that are updated within each block smoothing step. Moreover, the dependence on updated unknowns of neighboring cells leads to further differences. For the Vanka smoother $60 \cdot N^2$ communication steps are necessary, while for the triad-wise smoother $12 \cdot N^2$ are needed for one smoothing iteration. One smoothing step of the Vanka smoother requires as many communication steps as five smoothing steps of the triad-wise smoother. These results show the low computational cost in combination with good parallelization properties of the triad-wise smoother. However, a parallel implementation may lead to different convergence factors due to the coloring scheme, that is, a modified order of updating the unknowns.

We implement a two-grid method in serial with the colored versions of both smoothers to get an idea how the coloring influences the convergence factors. We start with a comparison of numerial results based on the overlapping and nonoverlapping smoother for the Stokes equations with periodic boundary conditions. The results are based on the same example and settings as described in Section 4. That leads to convergence factors of $\rho = 0.29$ for the triad-wise smoother with red-black coloring. Compared to a convergence factor of $\rho = 0.06$ for the five-color Vanka smoother. We have similar results compared to the serial implementation without a coloring. In consideration of the low computational effort for the triad-wise smoother the convergence factor is reasonable. Multigrid results show the same effect, see Figure 17. We discretize on a 33×33 grid with the domain $\Omega = (0, 1)^2$, use two pre- and postsmoothing steps and employ the zero solution as an initial guess.

This is not true for Dirichlet boundary conditions. As we already notice in Section 4, a different ordering of updating the unknowns may change the convergence results especially with Dirichlet boundary conditions. The two-grid method for the colored versions of both smoothers with Dirichlet boundary conditions leads to the following results. The triad-wise smoother with a GS update of unknowns and a RB-coloring leads to $\rho = 0.58$ as seen in Table 1. However, if we implement the modified version of the triad-wise smoother as described in Section 5 with a RB coloring, we get a convergence factor of $\rho = 0.21$. The novel triad-wise smoother consists of four consecutive steps, in which one single step is similar to the original triad-wise smoother. The consecutive execution of the four steps results in more synchronization steps. In comparison, the five-color Vanka smoother leads to convergence results of $\rho = 0.99$. This extremely bad convergence rate for the Vanka smoother can be immensely improved by a nine-coloring (cf, Figures 15 and 16) because of symmetry that is lost in the five-color version. We obtain a convergence rate of $\rho = 0.10$ for the two-grid method with the 9-color Vanka smoother. In comparison with the triad-wise smoother, the Vanka smoother with nine colors clearly leads to better convergence results. As discussed before the computational costs of the Vanka smoother with five colors

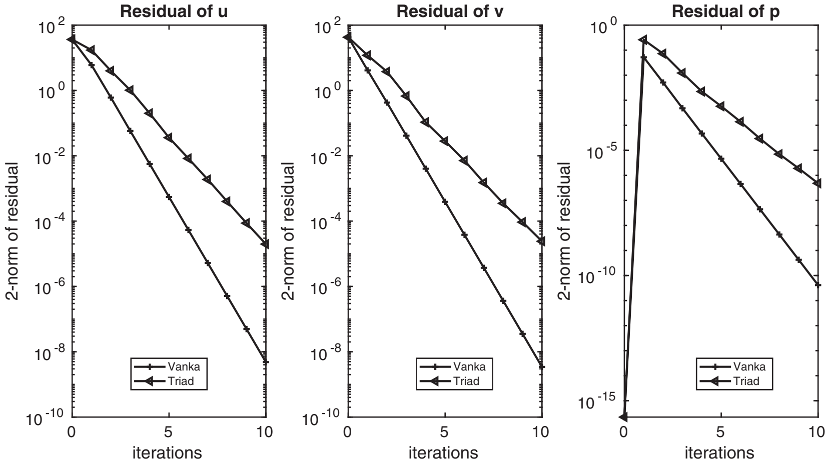


FIGURE 17 Convergence behavior of the multigrid method using two presmoothing and two postsmoothing steps on a 33×33 grid for the Stokes system with periodic boundary conditions.

are approximately the same as the costs of the modified triad-wise smoother. The communication costs of the triad-wise smoother increase from $12 \cdot N^2$ communication steps to $48 \cdot N^2$ due to the repetition of the original triad-wise smoother for four times, which is still below the costs for the Vanka smoother. In addition, the nine-coloring leads to more consecutive processing and therefore decreases the parallelization of the Vanka smoother even more. Using the triad-wise smoother instead of the Vanka smoother compromises the lower convergence rate with slightly lower communication costs and additionally good parallelization properties.

7 | CONCLUSIONS

Focusing on the results of the serial implementation we draw the following conclusions. For the Stokes equations with periodic boundary conditions the two-grid and multigrid convergence of the well-known Vanka smoother is better than the convergence of the triad-wise smoother. However, the triad-wise smoother is less expensive than the Vanka smoother.

For the Stokes equations with Dirichlet boundary conditions the good convergence of the Vanka smoother remains unchanged, while the convergence of the triad-wise smoother is not satisfying. However, the modified fourfold triad-wise smoother leads to convergence results that are even better than the convergence results of the Vanka smoother. In comparison to the original triad-wise smoother we need to accept less parallelization potential for the fourfold triad-wise smoother. These results give a good understanding of the challenges arising from different boundary conditions.

For the parallel implementation, we observe similar characteristics. A clear advantage of the triad-wise smoother over the Vanka smoother is given in respect to its low computational costs and its good parallelization properties with a reasonable higher convergence rate. This is especially true for the Stokes equations with periodic boundary conditions. Dirichlet boundary conditions lead to similar difficulties for the parallel implementation compared to the serial algorithm for the triad smoother. In addition, the Vanka smoother needs additional coloring for the two-grid method to obtain a good convergence rate. This leads to a further decrease of parallelization properties of the Vanka smoother and highlights the advantages of our novel triad-wise smoother.

Similar strategies can also be imagined for other discretizations, including stabilized finite elements. For that purpose, a general strategy for choosing the nonoverlapping blocks would be favorable. Applying these ideas to different discretizations as well as developing such strategies will be subject of further research.

ACKNOWLEDGEMENTS

This work was partly supported by the German Research Foundation (DFG) through Priority Programme 1648, “Software for Exascale Computing” (SPPEXA).

CONFLICT OF INTEREST

This study does not have any conflicts to disclose.

DATA AVAILABILITY STATEMENT

Data available on request from the authors

ORCID

Lisa Claus https://orcid.org/0000-0001-5854-157X

REFERENCES

1. Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Numerical mathematics and scientific computation. New York, NY: Oxford University Press; 2005.
2. Bauer S, Bunge HP, Drzisga D, et al. Hybrid parallel multigrid methods for geodynamical simulations. Software for exascale computing—SPPEXA 2013–2015. Lecture Notes Computing in Science and Engineering. Volume 113. Cham, Swizerland: Springer; 2016. p. 211–35.
3. Trottenberg U, Oosterlee C, Schuller A. Multigrid. London, UK: Elsevier Academic Press; 2001.
4. John V. On the parallel performance of coupled multigrid methods for the solution of the incompressible Navier-Stokes equations. Large-scale scientific computations of engineering and environmental problems (Varna, 1997). Notes Numerical Fluid Mechanics. Volume 62. Braunschweig, Germany: Friedr. Vieweg; 1998. p. 269–80.
5. Pe de la Riva A, Rodrigo C, Gaspar FJ. A robust multigrid solver for isogeometric analysis based on multiplicative Schwarz smoothers. SIAM J Sci Comput. 2019;41(5):S321–45.
6. 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):509–21.
7. 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):C143–68.
8. Drzisga D, John L, Rüde U, Wohlmuth B, Zulehner W. On the analysis of block smoothers for saddle point problems. SIAM J Matrix Anal Appl. 2018;39(2):932–60.
9. Schöberl J, Zulehner W. On Schwarz-type smoothers for saddle point problems. Numer Math. 2003;95(2):377–99.
10. Farrell PE, He Y, and MacLachlan SP. A local Fourier analysis of additive Vanka relaxation for the Stokes equations; 2020.
11. Vanka SP. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. J Comput Phys. 1986;65(1):138–58.
12. Claus L. Multigrid smoothers for saddle point systems. Wuppertal: Universitätsbibliothek Wuppertal; 2019.
13. Brandt A. Multi-level adaptive solutions to boundary-value problems. Math Comput. 1977;31(138):333–90.
14. Brandt A. Rigorous quantitative analysis of multigrid, I. constant coefficients two-level cycle with L_2-norm. SIAM J Numer Anal. 1994;31(6):1695–730.
15. Wienands R, Joppich W. Practical Fourier analysis for multigrid methods. vol. 4 of Numerical Insights. Boca Raton, FL: Chapman & Hall/CRC; 2005 With 1 CD-ROM (Windows and UNIX).
16. Niestegge A, Witsch K. Analysis of a multigrid Stokes solver. Appl Math Comput. 1990;35(3):291–303.
17. He Y, MacLachlan S. Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations. Numer Linear Algebra Appl. 2018;25(3):e214, 28.
18. He Y, MacLachlan S. Local Fourier analysis for mixed finite-element methods for the Stokes equations. J Comput Appl Math. 2019;357:161–83.
19. 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.
20. Sivaloganathan S. The use of local mode analysis in the design and comparison of multigrid methods. Comput Phys Commun. 1991;65(1):246–52.
21. MacLachlan S, Oosterlee C. Local Fourier analysis for multigrid with overlapping smoothers applied to systems of PDEs. Numer Linear Algebra Appl. 2011;18(08):751–74.
22. 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:e2285.

How to cite this article: Claus L, Bolten M. Nonoverlapping block smoothers for the Stokes equations. *Numer Linear Algebra Appl*. 2021;e2389. https://doi.org/10.1002/nla.2389