Linear Algebra and its Applications 656 (2023) 304–323

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Optimized sparse approximate inverse smoothers for solving Laplacian linear systems

Yunhui He [a], Jun Liu [b,*], Xiang-Sheng Wang [c]

[a] *Department of Computer Science, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada*
[b] *Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA*
[c] *Department of Mathematics, University of Louisiana at Lafayette, Lafayette, LA 70503, USA*

ARTICLE INFO

Article history:
Received 10 June 2022
Accepted 3 October 2022
Available online 6 October 2022
Submitted by V. Mehrmann

MSC:
49M25
49K20
65N55
65F10

Keywords:
Multigrid
Local Fourier analysis
Sparse approximate inverse
Smoothing factor
Laplacian

ABSTRACT

In this paper we propose and analyze new efficient sparse approximate inverse (SAI) smoothers for solving the two-dimensional (2D) and three-dimensional (3D) Laplacian linear system with geometric multigrid methods. Local Fourier analysis shows that our proposed SAI smoother for 2D achieves a much smaller smoothing factor than the state-of-the-art SAI smoother studied in Bolten et al. (2016) [12]. The proposed SAI smoother for 3D cases provides smaller optimal smoothing factor than that of weighted Jacobi smoother. Numerical results validate our theoretical conclusions and illustrate the high-efficiency and high-effectiveness of our proposed SAI smoothers. Such SAI smoothers have the advantage of inherent parallelism.

* Corresponding author.
E-mail addresses: yunhui.he@ubc.ca (Y. He), juliu@siue.edu (J. Liu), xswang@louisiana.edu (X.-S. Wang).

https://doi.org/10.1016/j.laa.2022.10.004
0024-3795/

1. Introduction

Laplace operator or Laplacian is used ubiquitously in describing various physical phenomena through partial differential equation (PDE) models, such as Poisson equation, diffusion equation, wave equation, and Stokes equation. For efficient numerical solution of such PDE models, a fundamental task is to (approximately) solve the corresponding sparse discrete Laplacian linear system upon suitable discretization schemes (e.g., finite difference method), where both direct and iterative solvers are extensively studied in the past century. In this paper, we will focus on using multigrid solver for the two-dimensional (2D) and three-dimensional (3D) Laplacian system with the 5-point and 7-point stencil central finite difference scheme, respectively, where new sparse approximate inverse (SAI) smoothers are our major contribution. More specifically, we consider the following Poisson equation on a unit square domain $\Omega = (0,1)^d, d = 2,3$ with Dirichlet boundary condition:

$$-\Delta u = f \quad \text{in} \quad \Omega, \qquad u = g \quad \text{on} \quad \partial\Omega, \tag{1.1}$$

which, upon applying the standard second-order accurate 5-point stencil (2D) and 7-point stencil (3D) central finite difference scheme with a uniform mesh step size $h = 1/N$, leads to a large-scale symmetric positive definite sparse linear system

$$A_h u_h = b_h. \tag{1.2}$$

Here u_h denotes the finite difference approximation to the true solution u over the set Ω_h of all interior grid points, b_h encodes the source term f and boundary data g, and A_h has the following well-known 5-point (2D) or 7-point (3D) stencil representation

$$A_h = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} \text{ or } \frac{1}{h^2}\left\{ \begin{bmatrix} & 0 & \\ 0 & -1 & 0 \\ & 0 & \end{bmatrix} \begin{bmatrix} & -1 & \\ -1 & 6 & -1 \\ & -1 & \end{bmatrix} \begin{bmatrix} & 0 & \\ 0 & -1 & 0 \\ & 0 & \end{bmatrix} \right\} \tag{1.3}$$

For simplicity, we assume f and g to be sufficiently smooth such that the finite difference approximation u_h by (1.2) achieves a second-order accuracy in infinity norm, that is $\|u_h - u|_{\Omega_h}\|_\infty = O(h^2)$. For less regular f and g, finite element discretization may be used to improve approximation accuracy in possible different weaker norms. We numerically verified that our proposed algorithm with appropriate modification also works well for other types of boundary conditions (e.g. periodic and Neumann).

Due to the large condition number of the Laplacian matrix A_h as the mesh step size h is refined, the stationary iterative methods (e.g. Jacobi and Gauss-Seidel iterations) usually converge extremely slowly as the system size n is increased. In contrast, multigrid methods can deliver mesh-independent convergence rate and the optimal computational complexity for solving the above linear system (1.2), where the choice of efficient and effective smoother is the key component. In the past few decades, Poisson equation (1.1) has been numerically solved by various multigrid methods based on

different smoothers and discretization schemes, see [2,13–16,32,36,43,44,53,54] and the references therein. Though more effective than the weighted Jacobi smoother, the popular lexicographical Gauss-Seidel smoother is not very friendly to massively parallel computers due to its sequential nature [1,43]. Nevertheless, a higher parallel efficiency can be achieved with red-black or multi-colored Gauss-Seidel smoothers [1–3,23,38,39,49], for which the communication cost per iteration is proportional to the number of colors. For general symmetric positive definite linear systems, significant efforts in the development of multigrid solvers have been concentrated on the design of effective parallelizable smoothers with smaller smoothing factors (and faster convergence rates), see for example [10,22,30,37,40,41,51] and the references therein. In [18], the authors compared three different Chebyshev polynomial smoothers in the context of aggressive coarsening, where the one-dimensional minimization formulations are defined over a finite interval that bounds all the eigenvalues of diagonally preconditioned system. In this paper, we will only focus on the development of effective and highly parallelizable SAI smoothers, whose convergence rates can be precisely estimated by local Fourier analysis (LFA), a quantitative tool to study multigrid convergence performance and optimize relaxation parameters.

Inspired by the well-studied sparse approximate inverse preconditioners [4–6,8,24,29,31,46,47] for preconditioning sparse linear systems, the class of so-called SAI smoothers were widely studied [12,20,21,48] for general linear systems, where superior smoothing effects were achieved. Besides broad applicability, the inherent parallelism [9,31] in the framework of parallel computing is one major advantage of SAI smoothers. The construction of high quality SAI preconditioners or smoothers of general linear systems is computationally expensive since it often requires to solve (multiple) norm minimization problems. However, for our considered well-structured linear systems, it is possible to analytically find the symbols of highly optimized and effective SAI smoothers through LFA techniques, which completely avoids the expensive numerical construction procedure using various optimization formulations.

In this work, we use LFA to derive new SAI smoothers for 2D and 3D Laplacian problems. Our proposed smoothers are more efficient and effective than these studied in [12,30]. In the literature, LFA has been widely used to study different discretization and relaxation schemes for the Poisson equation (1.1). For example, [34] uses LFA to study Jacobi smoother of multigrid methods for higher-order finite-element approximations to the Laplacian problem. In [25], multiplicative Schwarz smoothers are investigated by LFA for isogeometric discretizations of the Poisson equation. While, [30] studies additive Vanka-type smoothers. Multigrid methods based on triangular grids with standard-coarsening and three-coarsening strategies for the Poisson equation are studied in [26,27]. Our new SAI smoothers are very efficient in inverting Laplacian.

The whole paper is organized as follows. In the next section we recall basic concepts and ideas of LFA that will be used in our analysis. In Section 3, new SAI smoothers are developed and analyzed, where the technical proofs of our main theoretical results Theorems 3.1 and 3.2 are given. In Section 4, several numerical examples are reported to

confirm the LFA predicted multigrid convergence rates of our proposed SAI smoothers. Finally, some conclusions are given in Section 5.

2. A brief review of LFA

Local Fourier analysis (LFA) [49,50] is the standard tool to quantitatively predict the convergence rate of a given multigrid algorithm. In this section we briefly describe the main mechanism of how LFA works. In literature of multigrid methods, LFA is very useful to study the performance of multigrid smoothers. For this, a typical LFA procedure includes the following three steps:

(1) choose a good smoother (e.g. Jacobi) based on the system coefficient matrix;
(2) analyze LFA smoothing factor $\mu(\omega)$ of a ω-parameterized relaxation scheme, and (exactly or approximately) find the best relaxation parameter ω_{opt} that minimizes $\mu(\omega)$, which often requires very technical and tedious analysis;
(3) numerically verify the corresponding LFA two-grid convergence factor and the actual multigrid convergence rate and compare with the obtained optimal smoothing factor $\mu_{\text{opt}} = \mu(\omega_{\text{opt}})$.

To address some predictive limitation of the smoothing factor μ, where coarse-grid correction plays an important role in multigrid performance, the more sophisticated two-grid LFA convergence factor provides a more reliable estimate of actual multigrid convergence rate. For the finite difference discretization considered here, the LFA smoothing factor can offer a sharp prediction of actual multigrid performances. Thus, we will focus on optimizing the smoothing factor μ analytically and then checking the two-grid LFA convergence factor numerically (see below Table 1). Clearly, the choice of an efficient and effective smoother plays a decisive role in determining the practical convergence rate of the overall multigrid algorithm, and the selection of best relaxation parameter ω_{opt} is highly dependent on the chosen smoother too. We aim to design and analyze fast sparse approximate inverse smoothers by LFA techniques.

In the standard (geometric) multigrid method for solving the linear system (1.2), the most commonly used smoothers (e.g., damped Jacobi or Gauss-Seidel) have the following preconditioned Richardson iteration form

$$u_h^{k+1} = u_h^k + \omega M_h(b_h - A_h u_h^k) = \underbrace{(I_h - \omega M_h A_h)}_{S_h(\omega)} u_h^k + \omega M_h b_h, \tag{2.1}$$

where M_h approximates A_h^{-1}, $\omega \in \mathbb{R}$ is a damping parameter to be determined, and $S_h(\omega)$ is called relaxation error operator. For example, the damped Jacobi smoother takes $M_h = \text{diag}(A_h)^{-1}$ and the Gauss-Seidel smoother uses $M_h = \text{tril}(A_h)^{-1}$, where $\text{tril}(A)$ extracts the lower triangular part of A. Let $\varrho(S_h(\omega))$ be the spectral radius of $S_h(\omega)$. Then the above fixed-point iteration (2.1) is asymptotically convergent if and

only if $\varrho(S_h(\omega)) < 1$, which enforces some restrictions on the choice of ω. To estimate the multigrid performance, we can examine the smoothing effects of $S_h(\omega)$, that is how effectively it reduces the high frequency components of approximation errors. However, in practice, it is hard to directly compute or estimate $\varrho(S_h(\omega))$, instead, we can use LFA to study the smoothing properties of $S_h(\omega)$ via its Fourier symbol.

We first give some definitions of LFA following [49]. With the standard coarsening, the low and high frequencies are given by $\boldsymbol{\theta} \in T^{\mathrm{L}} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d$, $\boldsymbol{\theta} \in T^{\mathrm{H}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right)^d \setminus T^{\mathrm{L}}$, respectively. We define the LFA *smoothing factor* for $S_h(\omega)$ as

$$\mu_{\mathrm{loc}}(S_h(\omega)) := \max_{\boldsymbol{\theta} \in T^{\mathrm{H}}} \{\varrho(\widetilde{S}_h(\omega; \boldsymbol{\theta}))\}, \tag{2.2}$$

where the matrix $\widetilde{S}_h(\omega; \boldsymbol{\theta})$ is the symbol of $S_h(\omega)$ and $\varrho(\widetilde{S}_h(\omega; \boldsymbol{\theta}))$ stands for its spectral radius. In particular, for the finite difference Laplacian operator considered here, the symbol $\widetilde{S}_h(\omega; \boldsymbol{\theta})$ is just a scalar number. We also define the optimal smoothing factor μ_{opt} and the corresponding optimal relaxation parameter ω_{opt} as

$$\mu_{\mathrm{opt}} := \min_{\omega \in \mathbb{R}} \mu_{\mathrm{loc}}(S_h(\omega)), \qquad \omega_{\mathrm{opt}} := \arg\min_{\omega \in \mathbb{R}} \mu_{\mathrm{loc}}(S_h(\omega)). \tag{2.3}$$

In general, it is very difficult to analytically find the values of μ_{opt} and ω_{opt}. We point out that LFA assumes periodic boundary conditions and it does not consider the potential influence of different boundary conditions. However, in many applications, either the LFA smoothing factor or two-grid LFA convergence factor offers sharp predictions of problems with other boundary conditions [45].

Let $\widetilde{A}_h(\boldsymbol{\theta})$ and $\widetilde{M}_h(\boldsymbol{\theta})$ be the *scalar* symbol of A_h and M_h, respectively. Then the symbol of $S_h(\omega) = I_h - \omega M_h A_h$ is obviously given by (note $\widetilde{I}_h = 1$)

$$\widetilde{S_h}(\omega; \boldsymbol{\theta}) = \widetilde{I}_h - \omega \widetilde{M}_h \widetilde{A}_h = 1 - \omega \widetilde{A}_h(\boldsymbol{\theta}) \widetilde{M}_h(\boldsymbol{\theta}), \tag{2.4}$$

which leads to a min-max optimization for finding the optimal smoothing factor

$$\mu_{\mathrm{opt}} = \min_{\omega \in \mathbb{R}} \max_{\boldsymbol{\theta} \in T^{\mathrm{H}}} |1 - \omega \widetilde{A}_h(\boldsymbol{\theta}) \widetilde{M}_h(\boldsymbol{\theta})|. \tag{2.5}$$

Define

$$\lambda_0 := \min_{\boldsymbol{\theta} \in T^{\mathrm{H}}} \widetilde{A}_h(\boldsymbol{\theta}) \widetilde{M}_h(\boldsymbol{\theta}), \qquad \lambda_1 := \max_{\boldsymbol{\theta} \in T^{\mathrm{H}}} \widetilde{A}_h(\boldsymbol{\theta}) \widetilde{M}_h(\boldsymbol{\theta}).$$

If there holds $\lambda_0 > 0$, then it is easy to obtain

$$\mu_{\mathrm{opt}} = |1 - \omega_{\mathrm{opt}} \lambda_0| = |1 - \omega_{\mathrm{opt}} \lambda_1| = \frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0} = 1 - \frac{2}{1 + \lambda_1/\lambda_0} < 1 \tag{2.6}$$

with the optimal relaxation parameter

$$\omega_{\text{opt}} = 2/(\lambda_0 + \lambda_1). \tag{2.7}$$

Hence μ_{opt} is an increasing function of $\lambda_1/\lambda_0 \in [1, \infty)$ and a smaller ratio (or spectral condition number) λ_1/λ_0 gives a smaller smoothing factor. The best choice of $\widetilde{M}_h(\boldsymbol{\theta})$ highly depends on the expression of $\widetilde{A}_h(\boldsymbol{\theta})$ and hence the structure of matrix A_h. Mathematically, one may suggest to select $\widetilde{M}_h(\boldsymbol{\theta}) = 1/\widetilde{A}_h(\boldsymbol{\theta})$ such that $\lambda_1/\lambda_0 = 1$ leading to $\mu_{\text{opt}} = 0$, which is however impractical in computation since it requires the dense matrix $M_h = A_h^{-1}$. Nevertheless, it is still desirable to choose $\widetilde{M}_h(\boldsymbol{\theta}) \approx 1/\widetilde{A}_h(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in T^{\text{H}}$ such that λ_1/λ_0 is minimized in certain sense and meanwhile the matrix-vector product $M_h r_h$ with the residual vector $r_h := b_h - A_h u_h^k$ is efficient to compute. In the next section, we will construct effective M_h by minimizing μ_{opt} over a class of predefined symmetric stencil pattern that leads to parameterized symbol $\widetilde{M}_h(\boldsymbol{\theta})$. From (2.6), we only need to either minimize λ_1/λ_0 or maximize λ_0/λ_1 to obtain the optimal smoothing factor, and the corresponding optimal ω is given by (2.7). We first notice that a scalar multiple of $\widetilde{M}_h(\boldsymbol{\theta})$ does not change μ_{opt}, but it indeed leads to a rescaled ω_{opt}, hence one can normalize the symbol to simplify the analysis.

3. New optimized SAI smoothers

3.1. 2D case

For the 2D 5-point stencil A_h given in (1.3), its symbol reads

$$\widetilde{A}_h(\boldsymbol{\theta}) = \frac{1}{h^2}(4 - e^{i\theta_1} - e^{-i\theta_1} - e^{i\theta_2} - e^{-i\theta_2}) = \frac{2}{h^2}(2 - \cos\theta_1 - \cos\theta_2). \tag{3.1}$$

The weighted point-wise Jacobi smoother $M_J = \text{diag}(A_h)^{-1}$ has a singleton stencil

$$M_J = \frac{h^2}{4}\begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix} \tag{3.2}$$

with symbol $\widetilde{M}_J(\boldsymbol{\theta}) = h^2/4$, which was shown to achieve the optimal smoothing factor $\mu_{\text{opt}} = 3/5 = 0.6$ with $\omega_{\text{opt}} = 4/5 = 0.8$ [49]. Although M_J is very easy to parallelize, its convergence rate of 0.6 becomes rather slow and inefficient for large-scale systems.

In the seminar work [48], the authors derived a 5-point SAI smoother

$$M_{5,\text{TW}} = \frac{h^2}{61}\begin{bmatrix} & 3 & \\ 3 & 17 & 3 \\ & 3 & \end{bmatrix} \tag{3.3}$$

with its symbol $\widetilde{M}_{5,\text{TW}}(\boldsymbol{\theta}) = h^2(17 + 6\cos\theta_1 + 6\cos\theta_2)/61$, which was shown to have a smoothing factor of $21/61 \approx 0.344$ if choosing $\omega = 1$. In fact, LFA shows this particular smoother $M_{5,\text{TW}}$ can achieve the optimal smoothing factor $\mu_{\text{opt}} \approx 0.273$ if taking $\omega_{\text{opt}} \approx$

1.108. Nevertheless, such a 5-point SAI smoother can be further improved to obtain a smaller μ_{opt}. Specifically, in [12,17,21], the authors obtained the following best 5-point SAI smoother (among all symmetric 5-point stencils)

$$M_5 = \frac{8h^2}{41}\begin{bmatrix} & 1 & \\ 1 & 6 & 1 \\ & 1 & \end{bmatrix} \tag{3.4}$$

with its symbol $\widetilde{M}_5(\boldsymbol{\theta}) = (48{+}16\cos\theta_1{+}16\cos\theta_2)/41$, which gives the optimal smoothing factor $\mu_{\text{opt}} = 9/41 \approx 0.220$ with $\omega_{\text{opt}} = 1/4 = 0.25$. One may wondering can we construct a new SAI smoother with a smaller optimal smoothing factor μ_{opt}?

The short answer is *yes*, but we have to search from those SAI smoothers with wider/denser stencil. In this paper, for the sake of computational efficiency, we consider a general class of symmetric 9-point stencil smoothers of the following form

$$\Upsilon(\alpha,\beta,\gamma) := h^2\begin{bmatrix} \gamma & \beta & \gamma \\ \beta & \alpha & \beta \\ \gamma & \beta & \gamma \end{bmatrix},$$

where α, β, and γ are to be optimized through LFA. Obviously, the special case $\Upsilon(\alpha,\beta,0)$ with $\gamma = 0$ reduces to the above 5-point stencil and hence we expect a smaller optimal smoothing factor with the free choice of γ. The symbol of Υ reads

$$\widetilde{\Upsilon}(\boldsymbol{\theta}) = h^2(\alpha + 2\beta\cos\theta_1 + 2\beta\cos\theta_2 + 4\gamma\cos\theta_1\cos\theta_2).$$

Based on the idea of domain decomposition method, an element-wise additive Vanka smoother corresponding to the above 9-point stencil was proposed recently in [30]

$$M_{\text{Vanka}} = \frac{h^2}{96}\begin{bmatrix} 1 & 4 & 1 \\ 4 & 28 & 4 \\ 1 & 4 & 1 \end{bmatrix} = \frac{1}{96}\Upsilon(28,4,1), \tag{3.5}$$

with $\widetilde{M}_{\text{Vanka}} = (28 + 8\cos\theta_1 + 8\cos\theta_2 + 4\cos\theta_1\cos\theta_2)/96$, which gives the optimal smoothing factor $\mu_{\text{opt}} = 7/25 = 0.28$ with $\omega_{\text{opt}} = 24/25$. Though with better performance than weighted Jacobi smoother, such a Vanka 9-point stencil smoother M_{Vanka} can be greatly improved to attain a smaller optimal smoothing factor than the above best 5-point SAI smoother M_5, which is one of our major contributions.

To find the best 9-point stencil $\Upsilon(\alpha,\beta,\gamma)$ that achieves the optimal smoother factor μ_{opt}, we essentially need to solve the following min-max optimization problem

$$\begin{aligned}
\mu_{\text{opt}} &= \min_{\alpha,\beta,\gamma,\omega} \mu_{\text{loc}} = \min_{\alpha,\beta,\gamma,\omega} \max_{\boldsymbol{\theta}\in T^{\text{H}}} |1 - \omega\widetilde{A}_h(\boldsymbol{\theta})\widetilde{\Upsilon}(\boldsymbol{\theta})| \\
&= \min_{\alpha,\beta,\gamma,\omega} \max_{\boldsymbol{\theta}\in T^{\text{H}}} |1 - \omega(2\alpha + 4\beta(\cos\theta_1 + \cos\theta_2) + 8\gamma\cos\theta_1\cos\theta_2)(2 - \cos\theta_1 - \cos\theta_2)| \\
&= \min_{a,b,\omega} \max_{\boldsymbol{\theta}\in T^{\text{H}}} |1 - \omega(b + a(\cos\theta_1 + \cos\theta_2) + \cos\theta_1\cos\theta_2)(2 - \cos\theta_1 - \cos\theta_2)
\end{aligned}$$

where we fixed $8\gamma = 1$ and set $b = 2\alpha$ and $a = 4\beta$ to simplify the discussion. Notice that the normalization condition $8\gamma = 1$ only changes the choice of ω_{opt} and it will not affect μ_{opt} since the ratio λ_1/λ_0 remains the same. Based on some technical analysis as stated in the following theorem, we can obtain the following best 9-point SAI smoother M_9 that achieves the optimal smoothing factor

$$M_9 = \frac{1}{24}\Upsilon(44, 10, 3) = \frac{h^2}{24}\begin{bmatrix} 3 & 10 & 3 \\ 10 & 44 & 10 \\ 3 & 10 & 3 \end{bmatrix}. \tag{3.6}$$

Theorem 3.1. *Among all symmetric 9-point stencil of form $\Upsilon(\alpha, \beta, \gamma)$, the SAI smoother $M_9 = \frac{1}{24}\Upsilon(44, 10, 3)$ gives the optimal smoothing factor*

$$\mu_{\text{opt}} = (9 + 8\sqrt{10})/215 \approx 0.1595$$

with the optimal relaxation parameter

$$\omega_{\text{opt}} = \frac{309 - 12\sqrt{10}}{1720} \approx 0.1576.$$

Proof. For better exposition, we will only provide a sketch of the proof in the following. The technical detailed proof will be presented in the corresponding separate *supplementary material* for the interests of readers.

If $(\theta_1, \theta_2) \in T^{\text{H}} = \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right]^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2$, then $(x_1, x_2) := (\cos\theta_1, \cos\theta_2) \in X^{\text{H}} := [-1, 1]^2 \setminus (0, 1]^2$. Given $(a, b) \in \mathbb{R}^2$, we define

$$f_{a,b}(x_1, x_2) := [b + a(x_1 + x_2) + x_1x_2](2 - x_1 - x_2).$$

Then,

$$\mu_{\text{loc}} = \max_{(x_1,x_2)\in X^{\text{H}}} |1 - \omega f_{a,b}(x_1, x_2)|.$$

By symmetry, we may assume without loss of generality that $x_1 \le x_2$. Hence,

$$\mu_{\text{opt}} = \min_{a,b,\omega} \max_{(x_1,x_2)\in X_1\cup X_2} |1 - \omega f_{a,b}(x_1, x_2)|,$$

where

$$\begin{aligned} X_1 :=& \{(x_1, x_2) \in \mathbb{R}^2 : \ -1 \le x_1 \le 0, \ -x_1 \le x_2 \le 1\}, \\ X_2 :=& \{(x_1, x_2) \in \mathbb{R}^2 : \ -1 \le x_1 \le 0, \ x_1 \le x_2 \le -x_1\}. \end{aligned}$$

Since $X_1 \cup X_2$ is compact, the extremes of $f_{a,b}$ on $X_1 \cup X_2$ can be achieved. We denote

$$\chi_{a,b} := \max_{(x_1,x_2)\in X_1\cup X_2} f_{a,b}(x_1, x_2), \quad m_{a,b} := \min_{(x_1,x_2)\in X_1\cup X_2} f_{a,b}(x_1, x_2).$$

Since $\widetilde{A}_h$ is positive, to guarantee the convergence of relaxation scheme, we have to restrict (a, b) in the following region

$$R := \{(a, b) \in \mathbb{R}^2 : m_{a,b} > 0\}.$$

To find a SAI smoother achieving the optimal smoothing factor, from (2.6), we need to solve the following two-variable non-convex constrained minimization problem

$$\mu_{\text{opt}} = \min_{(a,b)\in R} J(a, b), \tag{3.7}$$

where

$$J(a, b) = (\chi_{a,b} - m_{a,b})/(\chi_{a,b} + m_{a,b}),$$

with the corresponding optimal choice of ω in the optimization problem (3.7) is

$$\omega_{\text{opt}} = 2/(\chi_{a,b} + m_{a,b}).$$

If $(a, b) \in R$, then we have

$$b > 1, \quad b > |a|, \quad b - 2a + 1 > 0,$$

and

$$m_{a,b} = \min\{b + a, 2b - 2, 4(b - 2a + 1)\},$$

and

$$\chi_{a,b} = \begin{cases} \max\{\frac{(b+2a)^2}{4a}, 4(b - 2a + 1)\}, & b \le 2a, \\ \max\{2b, 4(b - 2a + 1)\}, & (1 - 2a)^2 < 3(b - 2a), \\ \max\{2b, 4(b - 2a + 1)\}, & (1 - 2a)^2 \ge 3(b - 2a) > 0,\ t_- \notin [0, 2], \\ (b - at_- + t_-^2/4)(2 + t_-), & (1 - 2a)^2 \ge 3(b - 2a) > 0,\ t_- \in [0, 2], \end{cases}$$

where

$$t_- = \frac{2a - 1 - \sqrt{(1 - 2a)^2 - 3(b - 2a)}}{3/2}$$

is the smaller root of the quadratic function $3t^2/4+(1-2a)t+b-2a$ provided $(1-2a)^2 \ge 3(b - 2a) > 0$. For convenience of discussion, we divide R into four disjoint sub-regions:

$$R_0 := \{(a,b) \in R,\ b \le 2a\},$$
$$R_1 := \{(a,b) \in R,\ (1-2a)^2 < 3(b-2a)\},$$
$$R_2 := \{(a,b) \in R,\ (1-2a)^2 \ge 3(b-2a) > 0,\ t_- \notin [0,2]\},$$
$$R_3 := \{(a,b) \in R,\ (1-2a)^2 \ge 3(b-2a) > 0,\ t_- \in [0,2]\}.$$

A tedious but straightforward calculation gives the following results:

1. If $(a,b) \in R_0$, then $J(a,b) \ge 1/5$.
2. If $(a,b) \in R_1 \cup R_2$, then $J(a,b) \ge 3/17$.
3. If $(a,b) \in R_3$, then $J(a,b) \ge (9+8\sqrt{10})/215$.

On the other hand, by choosing $(a,b) = (5/3, 11/3) \in R_3$, we obtain that

$$J(5/3, 11/3) = (9+8\sqrt{10})/215$$

with $m_{5/3,11/3} = 16/3$ and $\chi_{5/3,11/3} = (4352+320\sqrt{10})/729$. Hence, we have

$$\mu_{\text{opt}} = \min_{(a,b)\in R} J(a,b) = \frac{9+8\sqrt{10}}{215},$$

and the corresponding optimal value of ω is calculated as

$$\omega_{\text{opt}} = \frac{2}{m_{5/3,11/3} + \chi_{5/3,11/3}} = \frac{309-12\sqrt{10}}{1720},$$

which completes the proof of Theorem 3.1. □

Compared to M_5 (with $\mu_{\text{opt}} \approx 0.2195$), the proposed M_9 (with $\mu_{\text{opt}} \approx 0.1595$) reduces μ_{opt} by about 30%, which hence leads to a faster multigrid convergence rate. Clearly, M_9 also provides much faster convergence rate than M_{Vanka} with the same cost. Recall the 2D pointwise lexicographic Gauss-Seidel smoother only gives $\mu_{\text{opt}} = 0.5$ [35,49].

It is worthwhile to point out that the LFA technique can be applied to other discretizations to identify optimal smoother. For example, the authors in [12] studied a 9-point stencil arising from linear finite element method for the 2D Poisson equation

$$\widehat{A}_h = \frac{1}{h^2}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix},$$

where the best SAI smoother with a 9-point stencil (of the reduced form $\Upsilon(\alpha,\beta,\beta)$)

$$\widehat{M_9} = \frac{4h^2}{153}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

gives the optimal smoothing factor $\mu_{\text{opt}} = 1/17 \approx 0.0588$ with $\omega_{\text{opt}} = 1$. We emphasize that it is technically more difficult to theoretically find M_9 than both M_5 and $\widehat{M}_9$, since the product symbol $\widetilde{M}_9(\boldsymbol{\theta})\widetilde{A}_h(\boldsymbol{\theta})$ is more complicated than both $\widetilde{M}_5(\boldsymbol{\theta})\widetilde{A}_h(\boldsymbol{\theta})$ and $\widetilde{\widehat{M}}_9(\boldsymbol{\theta})\widetilde{\widehat{A}}_h(\boldsymbol{\theta})$ due to mis-matched stencil patterns (5-point verse 9-point). However, for given structured smoother, one can apply some robust optimization approaches, for example, [22], to numerically identify the (approximate) optimal smoother.

3.2. 3D case

For the 3D 7-point stencil A_h given in (1.3), its symbol reads

$$\widetilde{A}_h(\boldsymbol{\theta}) = \frac{2}{h^2}(3 - (\cos\theta_1 + \cos\theta_2 + \cos\theta_3)). \tag{3.8}$$

The simplest Jacobi smoother is $M_J = (h^2/6)I_h$ with $\widetilde{M}_J(\boldsymbol{\theta}) = h^2/6$ and it achieves a quite large $\mu_{\text{opt}} = 5/7 \approx 0.714$ with the optimal damping parameter $\omega_{\text{opt}} = 6/7$ [49].

Inspired by M_5, we look at all 7-point stencil SAI smoothers of the form

$$\mathcal{T}(\alpha,\beta) := \frac{h^2}{4}\left\{ \begin{bmatrix} & 0 & \\ 0 & \beta & 0 \\ & 0 & \end{bmatrix} \begin{bmatrix} \beta & 2\alpha & \beta \\ & \beta & \end{bmatrix} \begin{bmatrix} & 0 & \\ 0 & \beta & 0 \\ & 0 & \end{bmatrix} \right\} \tag{3.9}$$

with a symbol $\widetilde{\mathcal{T}}(\boldsymbol{\theta}) = (h^2/4)(2\alpha + 2\beta(\cos\theta_1 + \cos\theta_2 + \cos\theta_3))$. Hence there holds

$$\widetilde{\mathcal{T}}(\boldsymbol{\theta})\widetilde{A}_h(\boldsymbol{\theta}) = (\alpha + \beta(\cos\theta_1 + \cos\theta_2 + \cos\theta_3))\,(3 - (\cos\theta_1 + \cos\theta_2 + \cos\theta_3))\,.$$

Following the techniques in [12], we can find the following best 7-point SAI smoother

$$M_7 = \frac{h^2}{10}\left\{ \begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix} \begin{bmatrix} 1 & 8 & 1 \\ & 1 & \end{bmatrix} \begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix} \right\} = \mathcal{T}(8/5, 2/5), \tag{3.10}$$

which gives the optimal smoothing factor as stated in the following theorem.

Theorem 3.2. *Among all symmetric 7-point stencil of form $\mathcal{T}(\alpha,\beta)$, the SAI smoother $M_7 = \mathcal{T}(8/5, 2/5)$ gives the optimal smoothing factor*

$$\mu_{\text{opt}} = 25/73 \approx 0.343$$

with the optimal relaxation parameter

$$\omega_{\text{opt}} = 20/73 \approx 0.274.$$

Proof. By normalization, we assume without loss of generality that $\beta = 2/5$ and $\alpha = a\beta$. Let $x_i = \cos\theta_i$ with $i = 1, 2, 3$, and transform the high frequency variable $\boldsymbol{\theta} \in T^{\mathrm{H}}$ to $(x_1, x_2, x_3) \in X^{\mathrm{H}} := [-1, 1]^3 \setminus (0, 1]^3$. By standard calculation, we have

$$\mu_{\mathrm{loc}} = \max_{(x_1,x_2,x_3)\in X^H} |1 - 2/5 \cdot \omega f_a(x_1, x_2, x_3)|,$$

where

$$f_a(x_1, x_2, x_3) := (a + x_1 + x_2 + x_3)(3 - x_1 - x_2 - x_3).$$

Denote the extreme values of f_a as

$$\chi_a := \max_{(x_1,x_2,x_3)\in X^{\mathrm{H}}} f_a(x_1, x_2, x_3), \quad m_a := \min_{(x_1,x_2,x_3)\in X^{\mathrm{H}}} f_a(x_1, x_2, x_3).$$

From (2.6), to obtain the optimal smoothing factor, we only need to maximize the ratio $r_a := m_a/\chi_a$.

We restrict a in the region $R := \{a \in \mathbb{R} : \ m_a > 0\}$ to guarantee the convergence of the relaxation scheme. Note that $f_a(x_1, x_2, x_3) = V_a(t) := (a + t)(3 - t)$ with $t := x_1 + x_2 + x_3 \in [-3, 2]$. Using $f_a > 0$, we have $a > 3$. Since $V_a(t)$ is a concave function in t (i.e., $V_a''(t) < 0$) with a unique critical point at $t = (3 - a)/2$, we obtain

$$m_a = \min\{V_a(-3), V_a(2)\} = \min\{6a - 18, a + 2\},$$

and

$$\chi_a = \begin{cases} (a + 3)^2/4, & 3 < a \le 9, \\ 6a - 18, & a \ge 9. \end{cases}$$

If $3 < a \le 4$, then $r_a = \frac{6a-18}{(a+3)^2/4} \le \frac{24}{49}$. If $4 \le a \le 9$, then $r_a = \frac{a+2}{(a+3)^2/4} \le \frac{24}{49}$. If $a \ge 9$, then $r_a = \frac{a+2}{6a-18} \le \frac{11}{36}$. A combination of the above three cases gives $r_a \le r_4 = 24/49$ with $m_4 = 6$ and $\chi_4 = 49/4$. Hence, from (2.6) we obtain

$$\mu_{\mathrm{opt}} = \frac{\chi_4 - m_4}{\chi_4 + m_4} = \frac{25}{73} \approx 0.343$$

with optimal ω_{opt} satisfied $\beta\omega_{\mathrm{opt}} = 2/(\chi_4 + m_4)$, where $\beta = \frac{2}{5}$, that is

$$\omega_{\mathrm{opt}} = \frac{2}{\beta(\chi_4 + m_4)} = \frac{5}{2}\frac{8}{73} = \frac{20}{73} \approx 0.274. \quad \square$$

Similar to the above discussed 2D cases (from M_5 to M_9), one may obtain a smaller optimal smoothing factor if a wider/denser (e.g. 19-point or 27-point) symmetric stencil is used to construct the SAI smoother for 3D problem. Nevertheless, the resulting product

symbol becomes much more complicated, which is overwhelmingly tedious to optimize analytically. In such situations, one may resort to some robust optimization approaches (based on the same min-max formulation), see e.g. [22]. Although further discussion is beyond the scope of this paper, we numerically verified such heuristically optimized 19-point or 27-point SAI smoothers indeed deliver slightly faster convergence rates than that of M_7. We also mention the 3D pointwise lexicographic Gauss-Seidel smoother only attains $\mu_{\text{opt}} = (4+\sqrt{5})/11 \approx 0.567$ [35], which is much larger than that of the above optimized SAI smoother M_7.

4. Numerical results

In this section, we present some numerical tests to illustrate the effectiveness of our proposed multigrid algorithms for 2D and 3D Poisson problems with Dirichlet, periodic and Neumann boundary conditions. All simulations are implemented with MATLAB on a Dell Precision 5820 Workstation with Intel(R) Core(TM) i9-10900X CPU@3.70 GHz and 64 GB RAM, where the CPU times (in seconds) are estimated by the timing functions `tic/toc`. In our multigrid algorithms, we use the coarse operator from re-discretization with a coarse mesh step size $H = 2h$, full weighting restriction and linear interpolation operators, W or V cycle with ν_1-pre and ν_2-post smoothing iteration, the coarsest mesh step size $h_0 = 1/4$, and the stopping tolerance $tol = 10^{-10}$ based on reduction in relative residual norms. We will test both $W(\nu_1+\nu_2) = W(1+0)$ and $V(\nu_1+\nu_2) = V(1+1)$ cycles in our numerical examples. The initial guess $u_h^{(0)}$ is chosen as uniformly distributed random numbers in $(0,1)$. The multigrid convergence rate (factor) of k-th iteration is computed as [49]

$$\widehat{\rho}^{(k)} = \left(\|r_k\|_2/\|r_0\|_2\right)^{1/k}, \tag{4.1}$$

where $r_k = b_h - A_h u_h^{(k)}$ denotes the residual vector after the k-th multigrid iteration. We will report $\widehat{\rho}^{(k)}$ of the last multigrid iteration as the actual convergence rate. The MATLAB codes for reproducing the following figures are available online at the link: https://github.com/junliu2050/SAI-MG-Laplacian.

Let $\nu = \nu_1 + \nu_2$ be the total number of smoothing steps in each multigrid cycle. In practice, the LFA smoothing factor often offers a sharp prediction of LFA two-grid convergence factor ρ_h and actual two-grid performance, which also predicts the W-cycle multigrid convergence rate [49,50]. In Table 1, we numerically optimize the LFA two-grid convergence factor $\rho_h(\nu = 1)$ with respect to the relaxation parameter $\omega \in (0,1]$, and then use the numerically obtained optimal parameter ω_{opt}^{TG} to compute the corresponding smoothing factor $\mu(\omega_{\text{opt}}^{TG})$, and $\rho_h(\nu)$ as a function of $\nu = 2,3,4$. We observe that two-grid LFA convergence factor $\rho_h(\nu = 1)$ is the same as the LFA smoothing factor $\mu(\omega_{\text{opt}}^{TG})$, and the approximately optimal ω_{opt}^{TG} and $\rho_h(\nu = 1)$ match with our theoretical smoothing analysis, ω_{opt} and μ_{opt}, respectively. As compared in Table 1, we also include the damped Jacobi smoother M_J and the SAI smoother M_5. Both our proposed SAI smoothers M_9

Table 1
LFA predicted two-grid convergence factor $\rho_h(\nu)$ using ω_{opt}^{TG} obtained from numerically minimizing two-grid LFA convergence factor $\rho_h(\nu = 1)$ and the corresponding LFA smoothing factor $\mu(\omega_{\text{opt}}^{TG})$ with $h = \frac{1}{256}$ (for 2D) and $h = \frac{1}{64}$ (for 3D).

		ω_{opt}^{TG}	$\mu(\omega_{\text{opt}}^{TG})$	$\rho_h(\nu=1)$	$\rho_h(\nu=2)$	$\rho_h(\nu=3)$	$\rho_h(\nu=4)$
$2D$	M_J	0.800	0.600	0.600	0.360	0.216	0.137
	M_5	0.250	0.220	0.220	0.087	0.056	0.044
	M_9	0.158	0.160	0.160	0.070	0.046	0.035
$3D$	M_J	0.857	0.714	0.714	0.510	0.364	0.260
	M_7	0.274	0.343	0.343	0.152	0.107	0.085

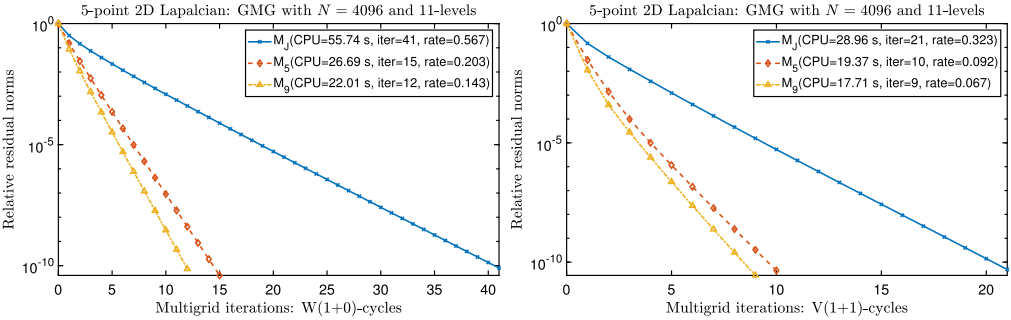

Fig. 1. Example 1: comparison of multigrid convergence with different smoothers. Left: W(1+0)-cycle. Right: V(1+1)-cycle.

and M_7 significantly outperform the Jacobi smoother M_J, which are also confirmed by the following several 2D and 3D numerical examples with various boundary conditions.

4.1. Example 1 [19]

In the first example we consider the following data

$$u = (x^2 - x^4)(y^4 - y^2),\ \ f = 2(1 - 6x^2)(y^2 - y^4) + 2(1 - 6y^2)(x^2 - x^4),\ \ g = 0.$$

In Fig. 1, we compare the multigrid convergence performance of our considered three SAI-type smoothers: M_J, M_5, and M_9, where the estimated convergence rates match with the LFA preconditions shown in Table 1. Moreover, Fig. 1 reveals that using V-cycle is much cheaper than W-cycle. Clearly, both SAI smoothers M_5 and M_9 attain significantly faster convergence rates and also cost less CPU times than the Jacobi smoother M_J. In serial computation, we only observe marginal speed up in CPU times for M_9 over M_5, since M_9 has a wider stencil and higher operation cost in each iteration. But we expect to achieve even more significant speedup in parallel computation since SAI smoothers are embarrassingly parallelizable and the parallel CPU times will be mainly determined by the required sequential iteration numbers.

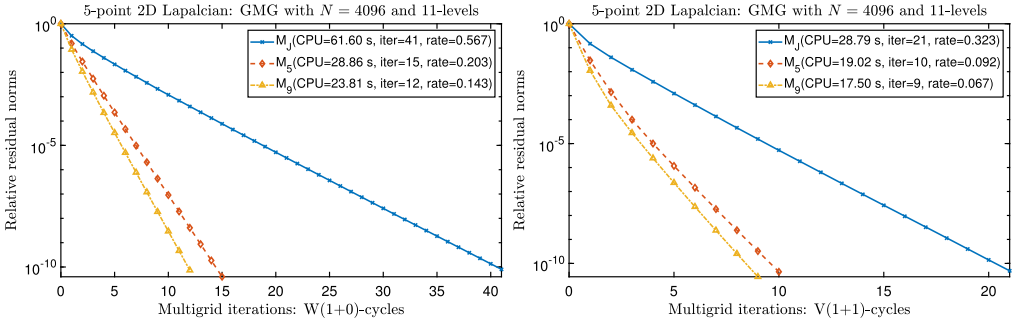

Fig. 2. Example 2: comparison of multigrid convergence with different smoothers. Left: W(1+0)-cycle. Right: V(1+1)-cycle.

4.2. Example 2

In the second example we consider the following data

$$u = x\ln(x)y\ln(y),\ \ f = -x\ln(x)/y - y\ln(y)/x,\ \ g = 0,$$

where f has singularity near the boundary with $x = 0$ or $y = 0$. In Fig. 2, we compare the multigrid convergence performance of our considered three SAI-type smoothers: M_J, M_5, and M_9, where the observed convergence rates are the same as those reported in Example 1. Again, we see V-cycle multigrid is more efficient than W-cycle multigrid. This example shows that the convergence rates of SAI smoothers are not obviously influenced by the lower regularity of the given source term f.

4.3. Example 3 (Periodic boundary conditions)

In this example we consider the same 2D Poisson equation

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1)^2 \tag{4.2}$$

with periodic boundary conditions

$$u(0,y) = u(1,y),\ \ u(x,0) = u(x,1),\ \ u_x(0,y) = u_x(1,y),\ \ u_y(x,0) = u_y(x,1),$$

where f is assumed to satisfy the compatibility condition such that there exists a unique zero-mean solution. In particular, we choose the following data

$$u = \sin(2\pi x)\cos(2\pi y),\ \ f = 8\pi^2\sin(2\pi x)\cos(2\pi y).$$

To accommodate the periodic structure of the system matrix A_h, we also enforce periodic structure in constructing the SAI smoothers. In Fig. 3, we compare the multigrid convergence performance of our considered three SAI-type smoothers: M_J, M_5, and M_9,

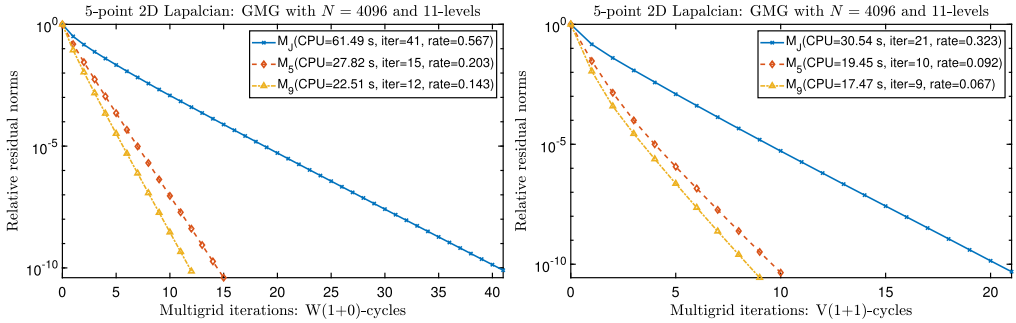

Fig. 3. Example 3: comparison of multigrid convergence with different smoothers. Left: W(1+0)-cycle. Right: V(1+1)-cycle.

where the observed W-cycle and V-cycle convergence rates are the same as the cases of Dirichlet boundary conditions reported in Example 1 and 2. This is anticipated since LFA predictions essentially assume periodic boundary conditions.

4.4. Example 4 (Neumann boundary condition)

In this example we consider 2D Poisson equation with pure Neumann boundary condition:

$$-\Delta u = f \quad \text{in} \quad \Omega, \qquad \frac{\partial u}{\partial n} = \varphi, \quad \text{on} \quad \partial\Omega, \tag{4.3}$$

where f and φ are assumed to satisfy the compatibility condition such that there exists a unique zero-mean solution. In particular, we choose the following data

$$u = (x^3/3 - x^5/5 - 1/20)(y^5/4 - y^3/3 - 1/20), \ f = -\Delta u, \ \varphi = 0.$$

As suggested in [19, Chap. 7], we have introduced ghost points near boundary to obtain a symmetric discrete system with a second-order accuracy, and modified the restriction and interpolation operators in multigrid to effectively handle Neumann boundary points. In Fig. 4, we compare the multigrid convergence performance of our considered three SAI-type smoothers: M_J, M_5, and M_9, where the observed W-cycle and V-cycle convergence rates are the same as those reported in previous 2D examples with Dirichlet and periodic boundary conditions. This example indicates our proposed SAI smoother M_9 also works very well for the case with pure Neumann boundary conditions that lead to a singular linear system [52]. We mention that the Neumann boundary conditions are also widely used in time-dependent diffusion models from fluid dynamics and many other applications.

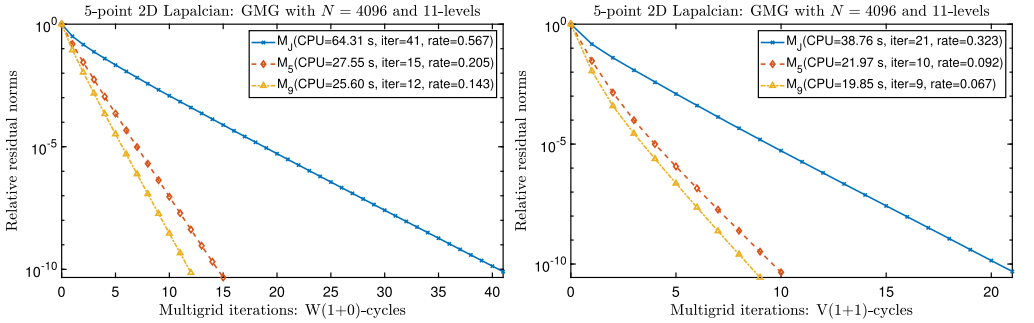

Fig. 4. Example 4: comparison of multigrid convergence with different smoothers. Left: W(1+0)-cycle. Right: V(1+1)-cycle.

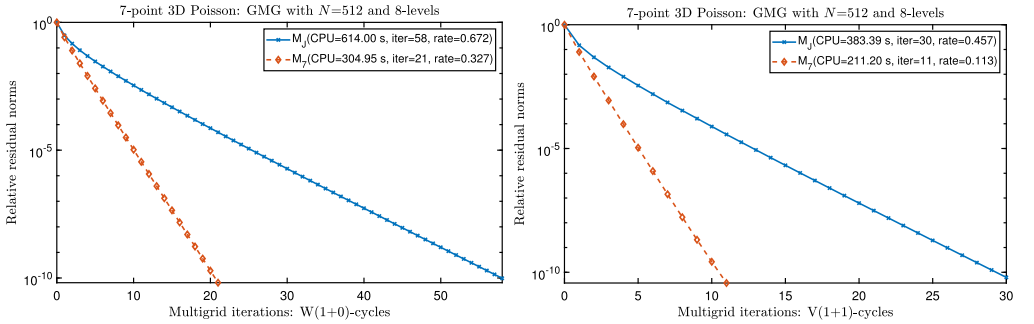

Fig. 5. Example 5: comparison of multigrid convergence with different smoothers. Left: W(1+0)-cycle. Right: V(1+1)-cycle.

4.5. Example 5 [54]

In the last example, we consider the following 3D data

$$u = \sin(\pi x)\sin(\pi y)\sin(\pi z),\ \ f = 3\pi^2 \sin(\pi x)\sin(\pi y)\sin(\pi z),\ \ g = 0.$$

In Fig. 5, we compare the multigrid convergence performance of the two SAI-type smoothers: M_J and M_7, where the observed convergence rates are compatible with the LFA predictions presented in Table 1. For both W and V cycles, M_7 takes about half of the CPU times by M_J, as predicted by Table 1. For all smoothers, we see that V-cycle is more efficient that W-cycle. We highlight that with $N = 512$ the system has about 134 million unknowns, which takes about 200 seconds for M_7 with V-cycle.

5. Conclusion

In this paper, we proposed and analyzed new 9-point and 7-point stencil based SAI multigrid smoothers for solving 2D and 3D Laplacian linear systems respectively. The obtained optimal LFA smoothing factors are significantly smaller than that of the state-

of-the-art SAI smoothers in literature. The crucial optimal relaxation parameters are exactly derived through rigorous analysis. Numerical results with 2D and 3D examples with various boundary conditions validated our theoretical analysis and demonstrated the effectiveness of our proposed SAI smoothers.

It is interesting to extend such SAI smoothers to general second-order elliptic PDEs with variable coefficients, see e.g. [28,42] for the potential idea of preconditioning by inverting Laplacian with our proposed multigrid solvers. We highlight that more sophisticated multigrid methods (e.g. operator-based interpolation) are required for effectively handling variable coefficients with drastic variations and our current used LFA techniques can not be directly applied. One promising approach is to use the generalized locally Toeplitz theory, which was exploited in [7,11] for approximately optimizing the multigrid Runge-Kutta smoothers in solving unsteady variable-coefficient convection-diffusion equations. It is also possible to apply our proposed SAI smoothers to elliptic optimal control problem [33] involving Laplacian. The MATLAB codes for implementing our proposed algorithms are publicly available online at the link: https://github.com/junliu2050/SAI-MG-Laplacian.

Declaration of competing interest

All authors declare that they have no conflicts of interest.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors appreciate the editor and anonymous referees for their constructive comments that have greatly improved the quality of this paper.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.laa.2022.10.004.

References

[1] M. Adams, M. Brezina, J. Hu, R. Tuminaro, Parallel multigrid smoothing: polynomial versus Gauss–Seidel, J. Comput. Phys. 188 (2003) 593–610.
[2] A.H. Baker, R.D. Falgout, T.V. Kolev, U.M. Yang, Multigrid smoothers for ultraparallel computing, SIAM J. Sci. Comput. 33 (2011) 2864–2887.
[3] N. Bell, S. Dalton, L.N. Olson, Exposing fine-grained parallelism in algebraic multigrid methods, SIAM J. Sci. Comput. 34 (2012) C123–C152.

[4] M. Benzi, C.D. Meyer, M. Tûma, A sparse approximate inverse preconditioner for the conjugate gradient method, SIAM J. Sci. Comput. 17 (1996) 1135–1149.
[5] M. Benzi, M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, SIAM J. Sci. Comput. 19 (1998) 968–994.
[6] M. Benzi, M. Tûma, A comparative study of sparse approximate inverse preconditioners, Appl. Numer. Math. 30 (1999) 305–340.
[7] D. Bertaccini, M. Donatelli, F. Durastante, S. Serra-Capizzano, Optimizing a multigrid Runge–Kutta smoother for variable-coefficient convection–diffusion equations, Linear Algebra Appl. 533 (2017) 507–535.
[8] D. Bertaccini, F. Durastante, Iterative Methods and Preconditioning for Large and Sparse Linear Systems with Applications, Chapman and Hall/CRC, 2018.
[9] D. Bertaccini, S. Filippone, Sparse approximate inverse preconditioners on high performance GPU platforms, Comput. Math. Appl. 71 (2016) 693–711.
[10] P. Birken, Designing optimal smoothers for multigrid methods for unsteady flow problems, in: 20th AIAA Computational Fluid Dynamics Conference, 2011, p. 3233.
[11] P. Birken, Optimizing Runge-Kutta smoothers for unsteady flow problems, Electron. Trans. Numer. Anal. 39 (2012) 298–312.
[12] M. Bolten, T.K. Huckle, C.D. Kravvaritis, Sparse matrix approximations for multigrid methods, Linear Algebra Appl. 502 (2016) 58–76.
[13] D. Braess, The contraction number of a multigrid method for solving the Poisson equation, Numer. Math. 37 (1981) 387–404.
[14] D. Braess, The convergence rate of a multigrid method with Gauss-Seidel relaxation for the Poisson equation, in: Multigrid Methods, Springer, 1982, pp. 368–386.
[15] D. Braess, The convergence rate of a multigrid method with Gauss-Seidel relaxation for the Poisson equation, Math. Comput. 42 (1984) 505–519.
[16] H. Brandén, Grid independent convergence using multilevel circulant preconditioning: Poisson's equation, BIT Numer. Math. 62 (2022) 409–429.
[17] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comput. 31 (1977) 333–390.
[18] J. Brannick, X. Hu, C. Rodrigo, L. Zikatanov, Local Fourier analysis of multigrid methods with polynomial smoothers and aggressive coarsening, Numer. Math., Theory Methods Appl. 8 (2015) 1–21.
[19] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, SIAM, Philadelphia, PA, 2000.
[20] O. Bröker, M.J. Grote, Sparse approximate inverse smoothers for geometric and algebraic multigrid, Appl. Numer. Math. 41 (2002) 61–80.
[21] O. Bröker, M.J. Grote, C. Mayer, A. Reusken, Robust parallel smoothing for multigrid via sparse approximate inverses, SIAM J. Sci. Comput. 23 (2001) 1396–1417.
[22] J. Brown, Y. He, S. MacLachlan, M. Menickelly, S.M. Wild, Tuning multigrid methods with robust optimization and local Fourier analysis, SIAM J. Sci. Comput. 43 (2021) A109–A138.
[23] E. Chow, R.D. Falgout, J.J. Hu, R.S. Tuminaro, U.M. Yang, A survey of parallelization techniques for multigrid solvers, in: Parallel Processing for Scientific Computing, 2006, pp. 179–201.
[24] J. Cosgrove, J. Diaz, A. Griewank, Approximate inverse preconditionings for sparse linear systems, Int. J. Comput. Math. 44 (1992) 91–110.
[25] Á.P. de la Riva, C. Rodrigo, F.J. Gaspar, A two-level method for isogeometric discretizations based on multiplicative Schwarz iterations, Comput. Math. Appl. 100 (2021) 41–50.
[26] F.J. Gaspar, J.L. Gracia, F.J. Lisbona, Fourier analysis for multigrid methods on triangular grids, SIAM J. Sci. Comput. 31 (2009) 2081–2102.
[27] F.J. Gaspar, J.L. Gracia, F.J. Lisbona, C. Rodrigo, On geometric multigrid methods for triangular grids using three-coarsening strategy, Appl. Numer. Math. 59 (2009) 1693–1708.
[28] T. Gergelits, K.-A. Mardal, B.F. Nielsen, Z. Strakos, Laplacian preconditioning of elliptic PDEs: localization of the eigenvalues of the discretized operator, SIAM J. Numer. Anal. 57 (2019) 1369–1394.
[29] N.I. Gould, J.A. Scott, Sparse approximate-inverse preconditioners using norm-minimization techniques, SIAM J. Sci. Comput. 19 (1998) 605–625.
[30] C. Greif, Y. He, A closed-form multigrid smoothing factor for an additive Vanka-type smoother applied to the Poisson equation, arXiv preprint, arXiv:2111.03190, 2021.
[31] M.J. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses, SIAM J. Sci. Comput. 18 (1997) 838–853.
[32] T. Guillet, R. Teyssier, A simple multigrid scheme for solving the Poisson equation with arbitrary domain boundaries, J. Comput. Phys. 230 (2011) 4756–4771.

[33] Y. He, J. Liu, Smoothing analysis of two robust multigrid methods for elliptic optimal control problems, arXiv preprint, arXiv:2203.13066, 2022.
[34] Y. He, S. MacLachlan, Two-level Fourier analysis of multigrid for higher-order finite-element discretizations of the Laplacian, Numer. Linear Algebra Appl. 27 (2020) e2285.
[35] L.R. Hocking, C. Greif, Closed-form multigrid smoothing factors for lexicographic Gauss-Seidel, IMA J. Numer. Anal. 32 (2011) 795–812.
[36] W.H. Holter, A vectorized multigrid solver for the three-dimensional Poisson equation, Appl. Math. Comput. 19 (1986) 127–144.
[37] R. Huang, R. Li, Y. Xi, Learning optimal multigrid smoothers via neural networks, SIAM J. Sci. Comput. (2022), https://doi.org/10.1137/21M1430030, arXiv preprint, arXiv:2102.12071.
[38] F. Hülsemann, M. Kowarschik, M. Mohr, U. Rüde, Parallel geometric multigrid, in: Numerical Solution of Partial Differential Equations on Parallel Computers, Springer, 2006, pp. 165–208.
[39] K.S. Kang, Scalable implementation of the parallel multigrid method on massively parallel computers, Comput. Math. Appl. 70 (2015) 2701–2708.
[40] J. Kraus, P. Vassilevski, L. Zikatanov, Polynomial of best uniform approximation to and smoothing in two-level methods, Comput. Methods Appl. Math. 12 (2012) 448–468.
[41] J. Lottes, Optimal polynomial smoothers for multigrid V-cycles, arXiv preprint, arXiv:2202.08830, 2022.
[42] B.F. Nielsen, A. Tveito, W. Hackbusch, Preconditioning by inverting the Laplacian: an analysis of the eigenvalues, IMA J. Numer. Anal. 29 (2009) 24–42.
[43] Y. Notay, A. Napov, A massively parallel solver for discrete Poisson-like problems, J. Comput. Phys. 281 (2015) 237–250.
[44] K. Pan, D. He, H. Hu, An extrapolation cascadic multigrid method combined with a fourth-order compact scheme for 3D Poisson equation, J. Sci. Comput. 70 (2017) 1180–1203.
[45] C. Rodrigo, On the validity of the local Fourier analysis, J. Comput. Math. 37 (2019) 340–348.
[46] Y. Saad, Iterative Methods for Sparse Linear Systems, second edition, SIAM, Philadelphia, PA, 2003.
[47] W.-P. Tang, Toward an effective sparse approximate inverse preconditioner, SIAM J. Matrix Anal. Appl. 20 (1999) 970–986.
[48] W.-P. Tang, W.L. Wan, Sparse approximate inverse smoother for multigrid, SIAM J. Matrix Anal. Appl. 21 (2000) 1236–1252.
[49] U. Trottenberg, C.W. Oosterlee, A. Schuller, Multigrid, Academic Press, 2000.
[50] R. Wienands, W. Joppich, Practical Fourier Analysis for Multigrid Methods, CRC Press, 2004.
[51] X. Yang, R. Mittal, Efficient relaxed-Jacobi smoothers for multigrid on parallel computers, J. Comput. Phys. 332 (2017) 135–142.
[52] M. Yoon, G. Yoon, C. Min, On solving the singular system arisen from Poisson equation with Neumann boundary condition, J. Sci. Comput. 69 (2016) 391–405.
[53] J. Zhang, Acceleration of five-point red-black Gauss-Seidel in multigrid for Poisson equation, Appl. Math. Comput. 80 (1996) 73–93.
[54] J. Zhang, Fast and high accuracy multigrid solution of the three dimensional Poisson equation, J. Comput. Phys. 143 (1998) 449–461.