Received: 13 October 2018 | Revised: 13 March 2019 | Accepted: 21 May 2019

DOI: 10.1002/nla.2256

RESEARCH ARTICLE

WILEY

Multigrid for second-order ADN elliptic systems

B. Lee

Department of Mathematics, Southern Methodist University, Dallas, Texas

Correspondence
B. Lee, Department of Mathematics, Southern Methodist University, Dallas, TX 75275.
Email: barryl@smu.edu

Summary

This paper theoretically examines a multigrid strategy for solving systems of elliptic partial differential equations (PDEs) introduced in the work of Lee. Unlike most multigrid solvers that are constructed directly from the whole system operator, this strategy builds the solver using a factorization of the system operator. This factorization is composed of an algebraic coupling term and a diagonal (decoupled) differential operator. Exploiting the factorization, this approach can produce decoupled systems on the coarse levels. The corresponding coarse-grid operators are in fact the Galerkin variational coarsening of the diagonal differential operator. Thus, rather than performing delicate coarse-grid selection and interpolation weight procedures on the original strongly coupled system as often done, these procedures are isolated to the diagonal differential operator. To establish the theoretical results, however, we assume that these systems of PDEs are elliptic in the Agmon–Douglis–Nirenberg (ADN) sense and apply the factorization and multigrid only to the principal part of the system of PDEs. Two-grid error bounds are established for the iteration applied to the complete system of PDEs. Numerical results are presented to illustrate the effectiveness of this strategy and to expose factors that affect the convergence of the methods derived from this strategy.

KEYWORDS

ADN elliptic, elliptic partial differential equations, multigrid method

AMS CLASSIFICATION

65M55; 65M70; 65N55; 65R20; 65Z05

1 | INTRODUCTION

We are interested in the d-dimensional ($d = 2, 3$) system of second-order partial differential equations (PDEs)

$$\mathcal{L}\mathbf{u} = \begin{bmatrix} \mathcal{L}_{11} & \mathcal{L}_{12} & \dots & \mathcal{L}_{1n} \\ \mathcal{L}_{21} & \mathcal{L}_{22} & \dots & \mathcal{L}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{L}_{n1} & \dots & \dots & \mathcal{L}_{nn} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}, \tag{1}$$

where each diagonal element $\mathcal{L}_{ii}$ is a second-order elliptic operator and each off-diagonal element $\mathcal{L}_{ij}$ is at most a second-order elliptic operator, with the full system being elliptic in the Agmon–Douglis–Nirenberg (ADN) sense (see Section 8.4.2 of the work of Renardy et al.[1]). We will assume that this system is completed with a system boundary operator, which with system (1) satisfies the Lopatinskii compatibility condition (see Section 8.4.2 of the work of

Renardy et al.[1]). Assuming that the boundary is smooth, the resulting boundary-value problem is well posed in the vector $H2$-norm, that is, the $H2$-norm on each component of the **vector function** $(v_1, v_2, \ldots, v_n)^t$:

$$\|v_i\|_2 := \left\{ \int \left((v_i)^2 + \sum_{j=1}^{d} \left(\frac{\partial v_i}{\partial x_j} \right)^2 + \sum_{j,k=1}^{d} \left(\frac{\partial^2 v_i}{\partial x_j \partial x_k} \right)^2 \right) d\mathbf{x} \right\}^{\frac{1}{2}}.$$

Two applications that lead to such systems are higher order implicit Runge–Kutta (IRK) discretizations of parabolic equations particularly with time-dependent diffusion coefficients (i.e., the system of PDEs cannot be expressed as a Kronecker product) and linear thin shell models.

In a sequence of papers,[2,3] we developed and analyzed a preconditioner and two-level solver for (1). These were based on approximating (1) as

$$\mathbf{f} = \mathcal{L}\mathbf{u} \approx \mathbf{M}\, \mathrm{diag}(\mathcal{L}_{ii})\mathbf{u} \quad \text{or} \quad \mathbf{f} = \mathcal{L}\mathbf{u} \approx \mathrm{diag}(\mathcal{L}_{ii})\, \mathbf{M}\mathbf{u}, \tag{2}$$

where $\mathbf{M}$ describes the cross-variable coupling in the differential system and which we assumed to be nonsingular. For example, consider a high-order IRK method for time integrating the scalar parabolic equation

$$\frac{\partial u}{\partial t} = \nabla \cdot D\nabla u.$$

For an s-stage IRK,[4] at each time-step Δt, a system of elliptic equations of the form

$$\begin{bmatrix} (\Delta t)^{-1} I + a_{11}\nabla \cdot D\nabla & a_{12}\nabla \cdot D\nabla & \cdots & a_{1s}\nabla \cdot D\nabla \\ a_{21}\nabla \cdot D\nabla & (\Delta t)^{-1} I + a_{22}\nabla \cdot D\nabla & \cdots & a_{2s}\nabla \cdot D\nabla \\ \vdots & \vdots & \vdots & \vdots \\ a_{s1}\nabla \cdot D\nabla & \cdots & \cdots & (\Delta t)^{-1} I + a_{ss}\nabla \cdot D\nabla \end{bmatrix} \begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_s \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{pmatrix}$$

must be solved for the stage increments $\hat{u}_i$'s. Because the scheme is implicit, for appropriately large Δt's, the system is of the form (2) with

$$\mathbf{M} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ a_{s1} & \cdots & \cdots & a_{ss} \end{bmatrix}$$

and $\mathrm{diag}(\nabla \cdot D\nabla)$. More complex and larger systems will arise when an s-stage IRK is applied to a system of parabolic equations (see Section 6).

Equation (2) is a factorization involving an algebraic coupling term and a diagonal differential term. Factorization (2) then leads to an effective preconditioner. In fact, one can consider (2) as an extension to a block-diagonal preconditioner for operators that are equivalent to their diagonal operators. Moreover, explicitly using the algebraic coupling matrix $\mathbf{M}$ in the interpolation operator for coarsening (1), we are able to obtain a two-level scheme that allows reuse of existing highly parallelized and efficient software packages. In this paper, we examine this two-level method and some modified algorithms using the ADN elliptic PDE theory. Specifically, this paper theoretically establishes the two-level methods of Lee[3] but applied only to the principal part of the system of PDEs.

The goal of this paper is to develop multigrid methods for systems of elliptic equations that permit an algebraic-differential factorization/separation using norm equivalence between the component operators. This factorization produces a system operator that approximates the original system operator and which can give an easier handle on the cross-variable coupling on the coarser grids. In particular, this new system operator permits the removal of the cross-variable coupling in the coarse-grid selection procedure and in the construction of the interpolation operators. However, to achieve good approximation to original system operator, a natural requirement is that the factorized operator has the same elliptic structure as the original operator. Thus, we consider second-order ADN elliptic systems that insure this, and generate the factorized operator from the principal part of the original elliptic operator. This means that only the principal part of the elliptic system is handled on the coarse levels. In this paper, we develop such algorithms and examine the convergence (i.e., bounds on the two-grid matrix) of them.

This paper is organized as follow. In Section 2, we review ADN elliptic systems and describe how to efficiently construct a factorized system of the form (2) for the principal part of a class of second-order ADN elliptic systems. In Section 3, we review multigrid principles for scalar elliptic equations and describe the challenges in extending these principles to

systems of elliptic equations. We also describe some of the common multigrid approaches used on these systems and discuss their difficulties. In Section 4, we develop several two-level methods where the coarse level uses the factorized operator. Because only the principal component of the system operator is used, the coarse-grid operators, with respect to the full operator, are generated in a non-Galerkin fashion. However, we show that the structure of the system of PDEs is approximated well. Furthermore, we develop an approximate Galerkin coarsening procedure for the principal part that leads to an algorithm that can be implemented using kernels of existing robust and efficient software packages for solving scalar PDEs. In Section 5, we develop bounds on the two-grid matrices of the develop methods. Finally, in Section 6, we conduct some numerical results to demonstrate the performance of these algorithms.

2 | ADN ELLIPTIC SYSTEMS

We consider systems of PDEs that are elliptic in the ADN sense. Recall that system (1) is ADN elliptic if

1. there exist sets of integer indices $\{s_i\}$, $s_i \leq 0$, and $\{t_j\}$, $t_j \geq 0$, such that

$$\text{order}(\mathcal{L}_{ij}) \leq s_i + t_j, \text{ and}$$

2. with the principal part of the system operator defined to be

$$\mathcal{L}^p = \left[\mathcal{L}^p_{ij}\right] \text{ where } \mathcal{L}^p_{ij} \text{ satisfies order} \left(\mathcal{L}^p_{ij}\right) = s_i + t_j,$$

and with the symbol of $\mathcal{L}^p$ evaluated at a nonzero vector η denoted by $\mathcal{L}^p(x,\eta)_{\text{symb}}$, then the matrix $\mathcal{L}^p(x,\eta)_{\text{symb}}$ is nonsingular, that is,

$$\det\left(\mathcal{L}^p(x,\eta)_{\text{symb}}\right) \neq 0.$$

Furthermore, the operator is said to be uniformly elliptic if there exist positive constants c_1, c_2 such that

$$c_1|\eta|^{2m} \leq |\det(\mathcal{L}^p(x,\eta)_{\text{symb}})| \leq c_2|\eta|^{2m} \tag{3}$$

for all (x,η) and where $2m = \sum_{i,j}(s_i + t_j)$. We will assume that (1) is uniformly elliptic, and consider ADN-elliptic systems described by indices $s_i = 0$ and $t_j = 2$ for all $i, j \in \{1, 2, \ldots, n\}$, and with each diagonal component of $\mathcal{L}^p$ being second-order scalar elliptic with its symbol satisfying $|\mathcal{L}^p_{ii}(x,\eta)_{\text{symb}}| = \sum_{l=1}^{d} \alpha_{i,l}|\eta_l|^2$, $\alpha_{i,l} > 0$, for all x.

Given a uniformly elliptic system, we assume that the boundary operator, denoted $\mathcal{B}$, satisfies the complementing condition (see Section 8.4.2 of the work of Renardy et al.[1]), which will imply that the system boundary-value problem is well posed in the appropriate Sobolev space, for example,

$$\sum_{j=1}^{n}\|u_j\|_2 \leq C\left[\sum_{j=1}^{n}\|f_j\|_0 + \sum_{j=1}^{r}\|g_j\|_{-1/2} + \sum_{j=1}^{n}\|u_j\|_0\right], \tag{4}$$

where the g_j's are the boundary sources for the r boundary conditions.

Given our assumption that system (1) is uniformly elliptic with $s_i = 0$ and $t_j = 2$, then $\det(\mathcal{L}^p(x,\eta)_{\text{symb}})$ is a polynomial of degree $2m = \sum_{i,j}(s_i + t_j) = 2n$. If we have

$$\mathcal{L}^p \approx \mathbf{M}\text{diag}\left(\mathcal{L}^p_{ii}\right) \tag{5}$$

with $\mathbf{M}$ nonsingular, then

$$\det\left[\mathbf{M}\text{diag}\left(\mathcal{L}^p_{ii}(x,\eta)\right)_{\text{symb}}\right] = \det(\mathbf{M})\det\left[\text{diag}\left(\mathcal{L}^p_{ii}(x,\eta)\right)_{\text{symb}}\right]. \tag{6}$$

Moreover, given the assumptions on the diagonal components, the degree of $\det[\text{diag}(\mathcal{L}^p_{ii}(x,\eta))_{\text{symb}}]$ is $2n$ and

$$\left(\Pi_{i=1}^{n}\underline{\alpha}_i\right)|\eta|^{2n} \leq \left|\det\left[\text{diag}\left(\mathcal{L}^p_{ii}(x,\eta)\right)_{\text{symb}}\right]\right| \leq \left(\Pi_{i=1}^{n}\bar{\alpha}_i\right)|\eta|^{2n}, \tag{7}$$

where $\underline{\alpha}_i = \min\{\alpha_{i,1}, \ldots, \alpha_{i,d}\}$ and $\bar{\alpha}_i = \max\{\alpha_{i,1}, \ldots, \alpha_{i,d}\}$. With the assumption that $\mathbf{M}$ is nonsingular, we see that $\mathbf{M}\text{diag}\left(\mathcal{L}_{ii}^p\right)$ is uniformly elliptic. We will assume that $\mathcal{B}$ satisfies the complementing condition with this factorized operator.

2.1 | Constructing M

To construct (5), one can obtain an $\mathbf{M}$ using operator equivalence. Consider component operators $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{ji}^p, j \neq i$. If $\mathcal{L}_{ji}^p$ is of order lower than 2, we can take $\mathbf{M}_{ji} = 0$. If $\mathcal{L}_{ji}^p$ is also of second order, then there exist positive constants c_{ji} and C_{ji} such that

$$c_{ji}\mathcal{L}_{ji}^p \leq \mathcal{L}_{ii}^p \leq C_{ji}\mathcal{L}_{ji}^p. \tag{8}$$

Because the elements of $\mathbf{M}$ reflect the equivalence between $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{ji}^p$, if $\frac{C_{ji}}{c_{ji}} = O(1)$, then component $\mathbf{M}_{ji}$ can be approximated with

$$\mathbf{M}_{ji}^h = \frac{\zeta_i^t \mathcal{L}_{ji}^{p,h} \zeta_i}{\zeta_i^t \mathcal{L}_{ii}^{p,h} \zeta_i}, \quad \zeta_i \text{ random}, \tag{9}$$

where $\mathcal{L}_{ii}^{p,h}, \mathcal{L}_{ji}^{p,h}$ are discretizations of these operators using a sufficiently small meshsize h. We can also take it to be the average of several $\mathbf{M}_{ji}^h$'s obtained from several different ζ_i's. Or, we can take ζ_i to be the resulting vector of applying a few V-cycles on $\mathcal{L}_{ii}^{p,h}$ with a random initial guess and zero right-hand side. This vector is the near-nullspace of $\mathcal{L}_{ii}^{p,h}$;[5] hence, we are considering the extreme case of the equivalence between $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{ji}^p$.

This choice of approximating $\mathbf{M}$ will insure that the sparsity structure of the second-order terms for $\mathbf{M}^h\text{diag}\left(\mathcal{L}_{ii}^p\right)$ and $\mathcal{L}^p$ are the same. In the work of Lee,[3] we considered the $\mathbf{M}$ factorization for the full operator $\mathcal{L}$. The components of $\mathbf{M}$ were approximated using (9) with $\mathcal{L}_{ii}^h$ and $\mathcal{L}_{ji}^h$. Approximating $\mathbf{M}$ using the full operator will generally lead to a more dense matrix, and thus, the ellipticity of $\mathbf{M}\text{diag}\left(\mathcal{L}_{ii}\right)$ generally can be quite different from that of $\mathcal{L}$ because their principal parts can be quite different.

Remark 1. For the continuum operator $\mathbf{M}^h\text{diag}(\mathcal{L}_{ii}^p)$, $\mathbf{M}^h$ is an $(n \times n)$ matrix. However, for the discrete operator $\mathbf{M}^h\text{diag}(\mathcal{L}_{ii}^{p,h})$, with the operator $\mathcal{L}_{ii}^p$ discretized on a grid of N_i points, $\mathbf{M}^h$ is an $(n \times n)$ block matrix with the ji'th block being $\mathbf{M}_{ji}^h \mathbf{I}^{N_i \times N_i}$, where $\mathbf{M}_{ji}^h$ is the scalar computed using (9). However, we will denote $\mathbf{M}^h$ as an $(n \times n)$ matrix in this discrete case because the n^2 values $\mathbf{M}_{ji}^h$ are the only degrees of freedom.

3 | MULTIGRID FOR SYSTEMS OF PDES

Our aim is to apply multigrid to (1). However, the development of robust and scalable multigrid for general systems of elliptic PDEs faces many challenges that are not encountered in its development for scalar elliptic PDEs. To see some of this complexity, consider the necessary complementary smoothing and coarse-grid correction principle for achieving scalability: What cannot be eliminated by relaxation must be captured through the coarse-grid correction (see Chapter 5 of the work of Briggs et al.[6] for a more thorough discussion). This principle describes the interplay between the two key procedures in a multigrid algorithm, the relaxation and coarse-grid correction, which are required to eliminate different components of the error. An effective coarse-grid procedure is achieved by carefully designing the intergrid interpolation operators so that the error components poorly eliminated by relaxation are in the range of these interpolation operators. These problematic error components are commonly associated with the near-nullspace of the operator, that is, in the algebraic setting, components that are approximately in the span of the eigenvector corresponding to the smallest eigenvalue of the operator, and in the PDE setting, functions that in the nullspace of the PDE operator disregarding the boundary conditions (e.g., constants for the Laplace equation). Thus, interpolation operators are formed to capture error components that give small energy norm. For scalar elliptic equations, this has been well analyzed: For example, if $\mathcal{L}$ is a scalar self-adjoint operator, then the interpolation operator is designed so that the component u satisfying

$$\frac{|(\mathcal{L}u, u)|}{\|u\|^2} \ll 1$$

is approximated well, or equivalently, this component, the so-called near-nullspace component, is well approximated in the range of the designed interpolation operator. Unfortunately, applying this methodology to system (1) is not as trivial

because the near-nullspace components can intervariably couple. However, in the simplified diagonally dominant case, the near-nullspace components are the padded vectors

$$\mathbf{u}_i = (0, \ \dots \ , 0, u_i, 0, \ \dots \ , 0)$$

where u_i is a near nullspace component of $\mathcal{L}_{ii}$. An appropriate interpolation operator is

$$\mathbf{P} = \text{diag}(P_{11}, P_{22}, \ \dots \ , P_{nn}), \tag{10}$$

where P_{ii} is the interpolation operator based on scalar operator $\mathcal{L}_{ii}$.[7,8,9] With this block diagonal interpolation operator, the coarse-grid operator is formed using the Galerkin coarse-grid procedure

$$\mathcal{L}^H = \mathbf{P}^t \mathcal{L}^h \mathbf{P}$$

in the two-grid case. We can use (10) to construct the interpolation operator for system (1) but there is no guarantee that the resulting multigrid method will be robust.

Block-diagonal interpolation operator (10) can be applied to (1). This will imply that variables interpolate to like variables, that is,

$$\mathbf{P}\mathbf{v}^{\mathbf{H}} = \text{diag}(P_{11}, P_{22}, \ \dots \ , P_{nn})\mathbf{v}^H = \left(P_{11}v_1^H, P_{22}v_2^H, \ \dots \ , P_{nn}v_n^H\right)^t$$

so that v_i^H interpolates only to variable v_i^H. To permit cross-variable interpolation, a full block interpolation operator can be designed[7,8,9]:

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ P_{n1} & \dots & \dots & P_{nn} \end{bmatrix}. \tag{11}$$

However, the construction of this operator is very complex, and a systematic and practical procedure for constructing it that leads to a robust multigrid algorithm is an active area of research.

Clearly, the development of multigrid principles for systems of PDEs is difficult because of the complexity introduced by the cross-variable coupling. Nevertheless, there are essentially three multigrid approaches described in literature for systems of PDEs. The first approach, which is mainly used in the algebraic multigrid (AMG) setting, is the linear system method: naively apply an AMG method specifically designed for scalar PDEs to the matrix equations resulting from the discretization of (1) or (2). Because the coarsening, and thereby interpolation weights, are determined from the full linear system, strong cross-variable coupling can lead to an imbalanced number of coarse degrees of freedom for the different variable types. This, in turn, can lead to poor coarse-grid approximations.

The second approach, which essentially uses some of the system PDE structure, is the unknown or multiple-coarsening methods. In these methods, only the diagonal component operators are used in the coarsening procedure, and possibly in the interpolation weights (the off-diagonal component operator coefficients may contribute by folding them into the diagonal component operator coefficients). Interpolation is performed only within the same variable type; hence, the interpolation operator is block diagonal. Applying the Galerkin coarsening procedure with this block diagonal interpolation operator produces a multiple-coarsening tree with level branches connecting the variable trunks (see Figure 1, right diagram). Applying Galerkin coarsening to only the diagonal of the full system operator creates a tree that couples the variable trunks only on the finest level (see Figure 1, left diagram). This latter approach can be used on systems that are essentially $\mathbf{H}^1$ equivalent.

The third existing scheme is the nodal approach.[7,8,9] Here, assuming that the discretization on the finest grid is nonstaggered in the sense that the physical variables are defined on the same gridpoints, coarsening and interpolation weights are generated nodewise and possibly using the full system operator. That is, as all variables are defined at each grid point, coarsening is performed using a measure that involves a "cumulation" of the local component operators. Of course, as in the AMG setting, although a physical staggered grid discretization can be used on the finest grid (e.g., some variable types are defined on cell centers whereas others are defined on edge centers), a virtual nonstaggered grid can be used on the finest level by grouping all the variables to define a new degree of freedom, and virtual nonstaggered grids can be generated on the coarser levels by choosing each coarse degree of freedom collectively over the variable types. Because all the variable types are located on the same node, nodal relaxation can be used.

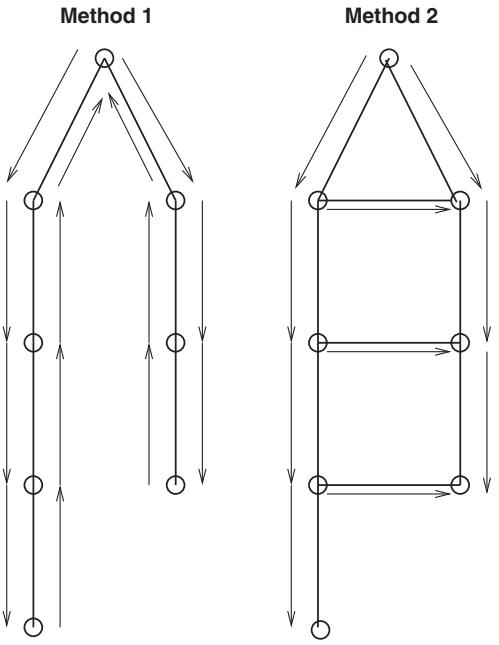

FIGURE 1 Multiple-coarsening schemes. The left diagram is a Gauss–Seidel version of the block diagonal scheme. V-cycles are performed over the variables separately, but the updated variable approximation is immediately used on the finest level to update the next variable V-cycle. The right diagram is a full coarsening scheme where all variables descend and ascend the tree together, but variable relaxation is done in a variable ordering, that is, on each level, all unknowns of Variable 1 are done, and then, all unknowns of Variable 2 are done using the updated Variable 1. Note that the different variable types may require different numbers of levels

A common feature of these multigrid approaches is that cross-variable couplings are not isolated out in the delicate coarse-grid selection procedure and interpolation formation. The cross-variable couplings complicate these procedures, and often can lead to bad coarse grids and poor interpolation operators. For the unknown or multiple-coarsening approaches, although the couplings are isolated, Galerkin coarsening on the full system operator (Figure 1, right diagram) produces coarse cross-variable operators that may not be accurate, as can be seen in that some PDE structures are not preserved. However, factorizing the operator into an algebraic coupling term and a diagonal operator can remove some of these complexities.

4 | MULTIGRID FOR SECOND-ORDER ADN ELLIPTIC SYSTEMS

Consider the decomposition

$$\mathcal{L} = \mathcal{L}^p + \mathcal{L}^o,$$

where $\mathcal{L}^o$ is the lower order system operator, and assume that

$$\mathcal{L}^p = (\mathbf{M} + \mathbf{T})\ \mathrm{diag}\left(\mathcal{L}^p_{ii}\right) \approx \mathbf{M}\ \mathrm{diag}\left(\mathcal{L}^p_{ii}\right), \tag{12}$$

where $\mathbf{T}$ reflects the discrepancy between $\mathcal{L}^p$ and $\mathbf{M}\mathrm{diag}(\mathcal{L}^p_{ii})$. An obvious way to solve (1) is to iterate

$$\mathcal{L}^p \mathbf{u}_j = \mathbf{f} - \mathcal{L}^o \mathbf{u}_{j-1}, \tag{13}$$

which has a convergence rate dependent on $\|[\mathcal{L}^p]^{-1}\mathcal{L}^o\|$. The challenge is then to develop an efficient algorithm for performing $[\mathcal{L}^p]^{-1}$. Some approaches are the two-level methods described in the work of Lee,[3] so that used with (13), we have a nested iteration. The convergence rate of this nested iteration will generally depend on the convergence rates of the inner and outer iterations. An alternative approach is to apply similar two-level methods directly to system (1). However, because an approximate $\mathbf{M}$-factorization can be generated only on $\mathcal{L}^p$, the coarse-grid operator must involve only this principal part. We develop these schemes and then establish bounds on their two-grid iteration matrices.

4.1 | Non-Galerkin approximate coarse-grid operator

Consider applying a Galerkin coarse-grid procedure on the discretized operator $\mathcal{L}^h$ using block-diagonal interpolation operator (10):

$$\mathbf{P}^t\mathcal{L}^h\mathbf{P} = \begin{bmatrix} P_{11}^t\mathcal{L}_{11}^hP_{11} & P_{11}^t\mathcal{L}_{12}^hP_{22} & \dots & P_{11}^t\mathcal{L}_{1n}^hP_{nn} \\ P_{22}^t\mathcal{L}_{21}^hP_{11} & P_{22}^t\mathcal{L}_{22}^hP_{22}^t & \dots & P_{22}^t\mathcal{L}_{2n}^hP_{nn} \\ \vdots & \vdots & \vdots & \vdots \\ P_{nn}^t\mathcal{L}_{n1}^hP_{11} & \dots & \dots & P_{nn}^t\mathcal{L}_{nn}^hP_{nn} \end{bmatrix}. \tag{14}$$

The ij'th element will be an accurate coarse-grid approximation to $\mathcal{L}_{ij}^h$ only if its structure (e.g., near-nullspace components) is similar to that of $\mathcal{L}_{ii}^h$ and $\mathcal{L}_{jj}^h$ because P_{ii} and P_{ji} are based on these operators. This generally will not be the case. However, consider the fine-grid error equation

$$\begin{aligned}[\mathbf{M}^h]^{-1}\mathbf{r}^h &= [\mathbf{M}^h]^{-1}\mathcal{L}^h\mathbf{e}^h \\ &= [\mathbf{M}^h]^{-1}(\mathcal{L}^{p,h} + \mathcal{L}^{o,h})\mathbf{e}^h \\ &\approx \operatorname{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{e}^h + [\mathbf{M}^h]^{-1}(\mathcal{L}^{o,h})\mathbf{e}^h. \end{aligned} \tag{15}$$

Assuming that $\mathcal{L}^{o,h}$ is majorized by $\mathcal{L}^{p,h}$, we take the coarse-grid problem to be

$$\mathbf{P}^t\operatorname{diag}(\mathcal{L}^{p,h})\mathbf{P}\mathbf{e}^{2h} = \mathbf{P}^t[\mathbf{M}^h]^{-1}\mathbf{r}^h, \tag{16}$$

which leads to the two-grid matrix

$$\left[\mathbf{I}^h - \mathbf{P}\left[\mathbf{P}^t\operatorname{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\right]^{-1}\mathbf{P}^t[\mathbf{M}^h]^{-1}\mathcal{L}^h\right]\mathbf{S}^\nu, \tag{17}$$

where $\mathbf{S}^\nu$ denotes ν sweeps of the smoother $\mathbf{S}$. The full operator $\mathcal{L}^h$ is used only in the relaxation and residual computation.

This coarsening produces decoupled coarse-grid problems, with the coupling incorporated only on the finest level. It may be desirable to incorporate the coupling on more levels, to capture the coupling on different scales. Assuming that the coarse-grid discretizations of $\mathcal{L}$ are not available, Equation (9) cannot be used to produce coarse approximations to $\mathbf{M}$. Nevertheless, we will use (9) with random vectors ζ_i of coarser scales to derive the coarse approximations to $\mathbf{M}$. With these coarse approximations, we can generate rediscretization-like coarse-grid operators. For example, consider grid $2h$. Rather than using the operator $\mathbf{P}^t[\mathbf{M}^h\operatorname{diag}(\mathcal{L}_{ii}^{p,h})]\mathbf{P}$ on grid $2h$, we use

$$\mathbf{M}^{2h}\mathbf{P}^t\operatorname{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}. \tag{18}$$

This choice allows us to preserve the $\mathcal{L}^p \approx \mathbf{M}\operatorname{diag}\left(\mathcal{L}_{ii}^p\right)$ structure on the coarser levels, which leads to a rediscretization-like coarsening: Because $\mathbf{M}^h$ has unit diagonal, we have

$$\begin{bmatrix} \mathcal{L}_{11}^{p,h} & \mathcal{L}_{12}^{p,h} & \dots & \mathcal{L}_{1n}^{p,h} \\ \mathcal{L}_{21}^{p,h} & \mathcal{L}_{22}^{p,h} & \dots & \mathcal{L}_{2n}^{p,h} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{L}_{n1}^{p,h} & \dots & \dots & \mathcal{L}_{nn}^{p,h} \end{bmatrix} \approx \mathbf{M}^h\operatorname{diag}\left(\mathcal{L}_{ii}^{p,h}\right) = \begin{bmatrix} \mathcal{L}_{11}^{p,h} & M_{12}^h\mathcal{L}_{22}^{p,h} & \dots & M_{1n}^h\mathcal{L}_{nn}^{p,h} \\ M_{21}^h\mathcal{L}_{11}^{p,h} & \mathcal{L}_{22}^{p,h} & \dots & M_{2n}^h\mathcal{L}_{2n}^{p,h} \\ \vdots & \vdots & \vdots & \vdots \\ M_{n1}^h\mathcal{L}_{11}^{p,h} & \dots & \dots & \mathcal{L}_{nn}^{p,h} \end{bmatrix}$$

and

$$\mathbf{M}^{2h}\mathbf{P}^t\operatorname{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P} = \begin{bmatrix} P_{11}^t\mathcal{L}_{11}^{p,h}P_{11} & M_{12}^{2h}\left(P_{22}^t\mathcal{L}_{22}^{p,h}P_{22}\right) & \dots & M_{1n}^{2h}\left(P_{nn}^t\mathcal{L}_{nn}^{p,h}P_{nn}\right) \\ M_{21}^{2h}\left(P_{11}^t\mathcal{L}_{11}^{p,h}P_{11}\right) & P_{22}^t\mathcal{L}_{22}^{p,h}P_{22} & \dots & M_{2n}^{2h}\left(P_{22}^t\mathcal{L}_{2n}^{p,h}P_{nn}\right) \\ \vdots & \vdots & \vdots & \vdots \\ M_{n1}^{2h}\left(P_{11}^t\mathcal{L}_{11}^{p,h}P_{11}\right) & \dots & \dots & P_{nn}^t\mathcal{L}_{nn}^{p,h}P_{nn} \end{bmatrix},$$

which, if the P_{ii}'s are good interpolation operators for the $\mathcal{L}_{ii}^{p,h}$'s, is approximately a rediscretization of the principal part of the system operator. Finally, the two-grid matrix corresponding to this method is

$$\left[\mathbf{I}^h - \mathbf{P}\left[\mathbf{M}^{2h}\mathbf{P}^t \text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\right]^{-1}\mathbf{P}^t\mathcal{L}^h\right]\mathbf{S}^\nu. \tag{19}$$

4.2 | Coarse-grid operator based on a distributive relaxation formulation

Because the principal part of $\mathcal{L}$ is assumed to have an $\mathbf{M}$-factorization, we can apply the technique of Lee,[3] where the full operator was assumed to have an $\mathbf{M}$-factorization. To keep the discussion self-contained, we present this technique again but in the context for solving the full ADN system. To this end, we note that a desired feature of the multigrid algorithm is the ability to re-use existing kernels from well-tested, high-performance software without having to dive into the internals of the software. To achieve this feature, a distributive relaxation (Section 3.4 of the work of Brandt et al.,[10] Section 8.7.3 of the work of Trottenberg et al.,[11] and the work of Yavneh[12]) formulation of the fine-grid system is considered. This relaxation operates on transformed variables.

To describe distributive relaxation for our systems, consider the fine-level error equation

$$\mathcal{L}^h\mathbf{e}^h = \mathbf{r}^h. \tag{20}$$

Because $\mathbf{M}^h$ is nonsingular, a change of variable gives the well-defined system

$$\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{v}^h = \mathbf{r}^h. \tag{21}$$

Fine-grid relaxation is applied to (21) to generate an approximation $\mathbf{v}_j^h$ and then the error correction is formed as $\mathbf{e}_j^h = (\mathbf{M}^h)^t\mathbf{v}_j^h$. This fine-grid relaxation can be performed pointwise but the resulting pointwise $\mathbf{v}_j^h$ is mapped to the nodal correction $\mathbf{e}_j^h$ by $(\mathbf{M}^h)^t$ that reflects the variable coupling of the principal part of $\mathcal{L}$. This overall procedure is a distributive relaxation that acts on the pointwise and nodal sets.

The multigrid process on (21) can also be viewed in a two-set frame. Consider the interpolation operator

$$\tilde{\mathbf{P}} = [(\mathbf{M}^h)^t]^{-1}\mathbf{P}, \tag{22}$$

where $\mathbf{P}$ is a block diagonal interpolation operator with the blocks being the interpolation generated using the diagonal operators $\mathcal{L}_{ii}^{p,h}$. Interpolation (22) maps a coarse set to a fine pointwise set. Using this operator to coarsen distributive system (21) leads to

$$\begin{aligned}\tilde{\mathbf{P}}^t\mathbf{r}^h &= \mathbf{P}^t[\mathbf{M}^h]^{-1}\mathbf{r}^h\\ &= \tilde{\mathbf{P}}^t\mathcal{L}^h(\mathbf{M}^h)^t\tilde{\mathbf{P}}\mathbf{v}^H\\ &= \tilde{\mathbf{P}}^t[\mathcal{L}^{p,h} + \mathcal{L}^{o,h}]\mathbf{P}\mathbf{v}^H\\ &\approx \mathbf{P}^t\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}^H + \mathbf{P}^t[\mathbf{M}^h]^{-1}\mathcal{L}^{o,h}\mathbf{v}^H.\end{aligned} \tag{23}$$

Dropping the second term in (23), which we assume to be dominated by the first term, we obtain (16).

In (23), the approximation arises because $\mathcal{L}^{p,h}$ is replaced with $\mathbf{M}^h\text{diag}(\mathcal{L}_{ii}^{p,h})$. A more accurate method can be derived by iteratively solving

$$\tilde{\mathbf{P}}^t\mathcal{L}^{p,h}\mathbf{P}\mathbf{v}^H = \tilde{\mathbf{P}}^t\mathbf{r}^h. \tag{24}$$

Consider the splitting

$$\mathcal{L}^{p,h} = (\mathbf{M}^h + \mathbf{T}^h)\ \text{diag}\left(\mathcal{L}_{ii}^{p,h}\right),$$

where $\mathbf{T}^h$ is a spatially dependent defect term and $\mathbf{T}^h\ \mathrm{diag}(\mathcal{L}_{ii}^{p,h})$ majorizes the lower order operator $\mathcal{L}^{o,h}$. This decomposition leads to the block Jacobi iteration

$$\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_j^H = \tilde{\mathbf{P}}^t\mathbf{r}^h - \tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H. \tag{25}$$

$[\mathbf{P}^t\mathrm{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}]^{-1}$ can be replaced with several multigrid V-cycles, and because $\mathbf{T}^h$ is not explicitly available, action $\tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}\mathbf{v}_{j-1}^H$ must be computed using

$$\tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H = \tilde{\mathbf{P}}^t\mathcal{L}^{p,h}\mathbf{P}\mathbf{v}_{j-1}^H - \mathbf{P}^t\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H. \tag{26}$$

This leads to the following two-grid method.

2Grid($\mathbf{u}^h, \mathbf{f}^h$):

1. presmooth: distributive relaxation $\mathbf{u}^h \leftarrow \mathbf{S}^h(\mathbf{u}^h, \mathbf{f}^h)$
2. coarse-grid problem: $\tilde{\mathbf{P}}^t\mathcal{L}^{p,h}\mathbf{P}\mathbf{v}^H = \tilde{\mathbf{P}}^t\mathbf{r}^h$.
   While not converged, iterate

$$\mathbf{P}^t\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_j^H = \tilde{\mathbf{P}}^t\mathbf{r}^h - \tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H := \mathbf{f}^H$$

   using V-cycles on $\mathbf{P}^t\mathrm{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}\mathbf{v}_j^H = \mathbf{f}^H$
3. coarse-grid correction: $\mathbf{u}^h \leftarrow \mathbf{u}^h + \mathbf{P}\mathbf{v}^H$
4. postsmooth: distributive relaxation $\mathbf{u}^h \leftarrow \mathbf{S}^h(\mathbf{u}^h, \mathbf{f}^h)$

To implement this two-grid algorithm, using functionality of existing robust software packages, such as the HYPRE library,[13] will generally require extracting internal data structures from the packages, and this requires extensive familiarity with the packages. For example, to solve the coarse-grid problem of either (16) or 2Grid, a separate solver cycle must be implemented. Because the coarse-grid operators $\mathbf{P}^t\mathrm{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}$ are generally extracted through the internal interfaces of the package, extracting them will require diving into the internal infrastructure of the package. This is indeed the case when using the HYPRE library. This issue can be seen more conspicuously in the formation of $\tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}\mathbf{v}_{j-1}^H$ in the above pseudo code. Because $\mathbf{T}^h$ is not explicitly available, this term must be computed as

$$\begin{aligned}\tilde{\mathbf{P}}^t\mathbf{T}^h\ \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H &= \mathbf{P}^t[\mathbf{M}^h]^{-1}\left\{\mathcal{L}^{p,h} - \mathbf{M}^h\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right\}\mathbf{P}\mathbf{v}_{j-1}^H \\ &= \mathbf{P}^t[\mathbf{M}^h]^{-1}\mathcal{L}^{p,h}\mathbf{P}\mathbf{v}_{j-1}^H - \mathbf{P}^t\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{P}\mathbf{v}_{j-1}^H.\end{aligned} \tag{27}$$

The second term can be computed after extracting the block-diagonal coarse-grid operator from the internals of HYPRE. However, to compute the first term, either its coarse-grid operator must be formed, or it can be computed as the action of this coarse-grid operator. For the former, forming the coarse-grid operator is nontrivial; for the latter, one needs to perform fine-grid computation when performing the coarse-grid iteration.

However, some of this complication can be eliminated by using a clever definition of the spatial hierarchy. Let the finest level be the finest level gridpoints viewed as a pointwise set, and the first coarse level be the finest gridpoints viewed as a nodal set. Then, $\mathbf{P} = \mathbf{I}$ and

$$\tilde{\mathbf{P}} = [(\mathbf{M}^h)^t]^{-1}.$$

Applying a Galerkin coarsening process to (21) leads to the coarse-grid

$$[\mathbf{M}^h]^{-1}\mathcal{L}^{p,h}\mathbf{v}^H = [\mathbf{M}^h]^{-1}\mathbf{r}^h, \tag{28}$$

and iteration (25) becomes

$$\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{v}_j^H = [\mathbf{M}^h]^{-1}\mathbf{r}^h - [\mathbf{M}^h]^{-1}\mathbf{T}^h\ \text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\mathbf{v}_{j-1}^H, \tag{29}$$

with $[\mathbf{M}^h]^{-1}\mathbf{T}^h\ \text{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{v}_{j-1}^H$ computed by

$$\left\{[\mathbf{M}^h]^{-1}\mathcal{L}^{p,h} - \text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right\}\mathbf{v}_{j-1}^H.$$

The structure of the resulting multigrid cycle is the same as **2Grid($\mathbf{u}^h, \mathbf{f}^h$)**, but now internal components do not have to be extracted. (For example, using HYPRE, all the operators in (29) are user defined and the multigrid solvers for $diag(\mathcal{L}_{ii}^{p,h})$ are available to the user.)

5 | TWO-GRID BOUNDS

We now establish bounds on the iteration matrices for the different two-level methods introduced in the previous section. Because the two-level method is essentially applied only to $\mathcal{L}^{p,h}$, the bounds for methods (17) and (29) are similar to the bounds given in the work of Lee,[3] but with addition terms involving $\left\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\right\|$. Again, to keep the discussion self-contained, the derivations of these bounds will be given in their entirety. A bound for method (19), which was not considered in the work of Lee,[3] will also be derived.

2-Grid Method (17)

Denoting $[\mathbf{P}^t\text{diag}(\mathcal{L}_{ii}^{p,h})\mathbf{P}]$ by $\mathbf{L}^{p,H}$, we have

$$\begin{aligned}
&\|[\mathbf{I}^h - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t[\mathbf{M}^h]^{-1}\mathcal{L}^h]\mathbf{S}^\nu\| \\
&= \|\{[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1} + [\mathcal{L}^{p,h}]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t[\mathbf{M}^h]^{-1}\}\mathcal{L}^h\mathbf{S}^\nu\| \\
&\le \{\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \|[\mathcal{L}^{p,h}]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t[\mathbf{M}^h]^{-1}\|\}\|\mathcal{L}^h\mathbf{S}^\nu\| \\
&= \left\{\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\|\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}[\mathbf{M}^h + \mathbf{T}^h]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t[\mathbf{M}^h]^{-1}\right\|\right\}\|\mathcal{L}^h\mathbf{S}^\nu\| \\
&\le \left\{\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\|\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t\right\|\|[\mathbf{M}^h + \mathbf{T}^h]^{-1}\|\right. \\
&\quad \left. + \|\mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t\|\|[\mathbf{M}^h + \mathbf{T}^h]^{-1} - [\mathbf{M}^h]^{-1}\|\right\}\|\mathcal{L}^h\mathbf{S}^\nu\| \\
&\le \left\{\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\|\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t\right\|\|[\mathbf{M}^h + \mathbf{T}^h]^{-1}\|\right. \\
&\quad \left. + \sum_{k=2}^{\infty}\left\|\mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t\right\|\|[\mathbf{M}^h]^{-1}\|\|\mathbf{T}^h[\mathbf{M}^h]^{-1}\|^{k-1}\right\}\|\mathcal{L}^h\mathbf{S}^\nu\|,
\end{aligned} \tag{30}$$

where the summation term is obtained from a Neumann series of $[\mathbf{M}^h + \mathbf{T}^h]^{-1}$. For the first term, for operators $\mathcal{L}$ with small nonprincipal components, by taking a sufficient number of iterations of a convergent relaxation scheme, this term can be made sufficiently small. For the second term, if the interpolation operator $\mathbf{P}$ is carefully constructed[10,11,14] so that $\|[\text{diag}(\mathcal{L}_{ii}^{p,h})]^{-1} - \mathbf{P}[\mathbf{L}^{p,H}]^{-1}\mathbf{P}^t\|$ is sufficiently small, then again a sufficient number of relaxation iterations will make this term sufficiently small. For the last term, provided that $\|\mathbf{T}^h[\mathbf{M}^h]^{-1}\|$ is sufficiently small, a sufficient number of relaxation steps is needed to make this term small. However, because $\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\|$ and $\mathbf{T}^h$ are generally spatially dependent, the overall bound can be h-dependent.

2-Grid Method (19)

Assume that interpolation operator $\mathbf{P}$ satisfies

$$\mathbf{P}^t\mathbf{P} = \mathbf{I}^{2h}, \tag{31}$$

and let the ij'th element of $[\mathbf{M}^h]^{-1}$ and $[\mathbf{M}^{2h}]^{-1}$ be denoted α_{ij} and β_{ij}, respectively. We have

$$
\begin{aligned}
&\left\| \left[\mathbf{I}^h - \mathbf{P}\left[\mathbf{M}^{2h}\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \mathbf{P}^t \mathcal{L}^h \right] \mathbf{S}^\nu \right\| \\
&\leq \left\{ \|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\| [\mathcal{L}^{p,h}]^{-1} - \mathbf{P}\left[\mathbf{M}^{2h}\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \mathbf{P}^t \right\| \right\} \|\mathcal{L}^h \mathbf{S}^\nu\| \\
&= \left\{ \|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\| [\mathcal{L}^{p,h}]^{-1} - \mathbf{P}\left[\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \left[\mathbf{M}^{2h} \right]^{-1} \mathbf{P}^t \right\| \right\} \|\mathcal{L}^h \mathbf{S}^\nu\| \\
&= \left\{ \|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\| [\mathcal{L}^{p,h}]^{-1} - \mathbf{P}\left[\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \mathbf{P}^t \mathbf{P} [\mathbf{M}^{2h}]^{-1} \mathbf{P}^t \right\| \right\} \|\mathcal{L}^h \mathbf{S}^\nu\| \\
&= \left\{ \left\| [\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1} \right\| + \left\| \left[\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \right]^{-1} [\mathbf{M}^h + \mathbf{T}^h]^{-1} - \mathbf{P}\left[\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \mathbf{P}^t \mathbf{P} [\mathbf{M}^{2h}]^{-1} \mathbf{P}^t \right\| \right\} \|\mathcal{L}^h \mathbf{S}^\nu\| \\
&\leq \left\{ \|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\| \left[\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \right]^{-1} \right\| \|[\mathbf{M}^h + \mathbf{T}^h]^{-1} - [\mathbf{M}^h]^{-1}\| \right. \\
&\quad + \left\| \left[\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \right]^{-1} \right\| \|[\mathbf{M}^h]^{-1} - \mathbf{P}[\mathbf{M}^{2h}]^{-1}\mathbf{P}^t\| \\
&\quad \left. + \left\| \left[\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \right]^{-1} - \mathbf{P}\left[\mathbf{P}^t \mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right) \mathbf{P} \right]^{-1} \mathbf{P}^t \right\| \|\mathbf{P}[\mathbf{M}^{2h}]^{-1}\mathbf{P}^t\| \right\} \|\mathcal{L}^h \mathbf{S}^\nu\|.
\end{aligned} \tag{32}
$$

Consider the term involving $\|[\mathbf{M}^h]^{-1} - \mathbf{P}[\mathbf{M}^{2h}]^{-1}\mathbf{P}^t\|$. For a fine-grid vector $\mathbf{v}^h$, we have

$$
\begin{aligned}
\left\{ [\mathbf{M}^h]^{-1} - \mathbf{P}[\mathbf{M}^{2h}]^{-1}\mathbf{P}^t \right\} \mathbf{v}^h &= \begin{pmatrix} \sum_{j=1}^n \left(\alpha_{1j} - \beta_{1j}\mathbf{P}_{11}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \\ \sum_{j=1}^n \left(\alpha_{2j} - \beta_{2j}\mathbf{P}_{22}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \\ \vdots \\ \sum_{j=1}^n \left(\alpha_{nj} - \beta_{nj}\mathbf{P}_{nn}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \end{pmatrix} \\
&= \begin{pmatrix} \sum_{j=1}^n \alpha_{1j} \left(1 - \frac{\beta_{1j}}{\alpha_{1j}}\mathbf{P}_{11}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \\ \sum_{j=1}^n \alpha_{2j} \left(1 - \frac{\beta_{2j}}{\alpha_{2j}}\mathbf{P}_{22}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \\ \vdots \\ \sum_{j=1}^n \alpha_{nj} \left(1 - \frac{\beta_{nj}}{\alpha_{nj}}\mathbf{P}_{nn}\mathbf{P}_{jj}^t \right) \mathbf{v}_j^h \end{pmatrix}.
\end{aligned} \tag{33}
$$

If $[\mathbf{M}^{2h}]^{-1}$ is a good approximation to $[\mathbf{M}^h]^{-1}$, then $\frac{\beta_{ij}}{\alpha_{ij}} \approx 1$ and $(1 - \frac{\beta_{ij}}{\alpha_{ij}}\mathbf{P}_{ii}\mathbf{P}_{jj}^t)$ is related to the interpolation/restriction errors of $\mathbf{P}_{ii}$ and $\mathbf{P}_{jj}^t$. Specifically, if $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{jj}^p$ have similar structures (e.g., similar near nullspace components), then $(1 - \frac{\beta_{ij}}{\alpha_{ij}}\mathbf{P}_{ii}\mathbf{P}_{jj}^t)$ is approximately the interpolation error in $\mathbf{P}_{ii}$. Note that requiring $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{jj}^p$ to be similar is less restrictive than in (14) where we required all the component operators to be similar.

The first and second terms of (32) are similar to the first and third terms of (30), and so controlling their bounds are similarly achieved. In addition, controlling the bound on the fourth term of (32) is similar to controlling the bound on the second term of (30). Lastly, as described above, the bound on the third term of (32) can be controlled if the $[\mathbf{M}^{2h}]^{-1}$ is a good approximation to $[\mathbf{M}^h]^{-1}$, $\mathcal{L}_{ii}^p$ and $\mathcal{L}_{jj}^p$ have similar structures, and the $\mathbf{P}_{ii}$'s are good interpolation operators.

2-Grid Method with (29)

Because only $\mathcal{L}^{p,h}$ is used on the coarse grid, the coarse-grid operator is

$$
[\mathbf{M}^h]^{-1}\mathcal{L}^{p,h} = [\mathbf{M}^h]^{-1}(\mathbf{M}^h + \mathbf{T}^h)\mathrm{diag}\left(\mathcal{L}_{ii}^{p,h} \right). \tag{34}
$$

Taking l iterations of (29) with initial approximation $\mathbf{v}_0 = \mathbf{0}$ corresponds to approximating $[\text{diag}(\mathcal{L}_{ii}^{p,h})]^{-1}\,[\mathbf{M}^h + \mathbf{T}^h]^{-1}\,\mathbf{M}^h$ with

$$\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}\left[\sum_{k=1}^{l}(-1)^{k-1}([\mathbf{M}^h]^{-1}\mathbf{T}^h)^{k-1}\right], \tag{35}$$

where the summated factor is an l-term Neumann expansion of $[\mathbf{M}^h + \mathbf{T}^h]^{-1}\mathbf{M}^h$. Assuming that the error in this truncated series expansion is bounded by ϵ, the two-grid matrix involving the distributive relaxation operator is

$$\begin{aligned}
&\left\|\left\{\mathbf{I}^h - [(\mathbf{M}^h)^t]^{-1}\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}\left[\sum_{k=1}^{l}(-1)^{k-1}([\mathbf{M}^h]^{-1}\mathbf{T}^h)^{k-1}\right][\mathbf{M}^h]^{-1}\mathcal{L}^h(\mathbf{M}^h)^t\right\}\mathbf{S}^\nu\right\| \\
&\le \left\|[(\mathbf{M}^h)^t]^{-1}[\mathcal{L}^h]^{-1} - [(\mathbf{M}^h)^t]^{-1}\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}\left[\sum_{k=1}^{l}(-1)^{k-1}([\mathbf{M}^h]^{-1}\mathbf{T}^h)^{k-1}\right][\mathbf{M}^h]^{-1}\right\|\|\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{S}^\nu\| \\
&\le \left\{\|[(\mathbf{M}^h)^t]^{-1}\|\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\|[(\mathbf{M}^h)^t]^{-1}[\mathcal{L}^{p,h}]^{-1} - [(\mathbf{M}^h)^t]^{-1}\right.\right. \\
&\quad\left.\left.\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}\left[[\mathbf{M}^h + \mathbf{T}^h]^{-1}\mathbf{M}^h - \sum_{k=l+1}^{\infty}(-1)^{k-1}([\mathbf{M}^h]^{-1}\mathbf{T}^h)^{k-1}\right][\mathbf{M}^h]^{-1}\right\|\right\}\|\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{S}^\nu\| \\
&= \left\{\|[(\mathbf{M}^h)^t]^{-1}\|\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \left\|[(\mathbf{M}^h)^t]^{-1}[\mathcal{L}^{p,h}]^{-1}\right.\right. \\
&\quad -[(\mathbf{M}^h)^t]^{-1}[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)]^{-1}[\mathbf{M}^h + \mathbf{T}^h]^{-1} + [(\mathbf{M}^h)^t]^{-1}\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1} \\
&\quad\left.\left.\left[\sum_{k=l+1}^{\infty}(-1)^{k-1}([\mathbf{M}^h]^{-1}\mathbf{T}^h)^{k-1}\right][\mathbf{M}^h]^{-1}\right\|\right\}\|\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{S}^\nu\| \\
&\le \left\{\|[(\mathbf{M}^h)^t]^{-1}\|\|[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\| + \epsilon\left\|[(\mathbf{M}^h)^t]^{-1}\left[\text{diag}\left(\mathcal{L}_{ii}^{p,h}\right)\right]^{-1}\right\|\|[\mathbf{M}^h]^{-1}\|\right\}\|\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{S}^\nu\|.
\end{aligned} \tag{36}$$

A bound on the first term is similar to the other methods. Furthermore, note that the number iterations of (29) on the coarse grid should lead to a balance between the two terms in the braces of (36).

Remark 2. The first terms in (30), (32), and (36) can be replaced with

$$\left\|\{[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\}\mathcal{L}^h\mathbf{S}^\nu\right\|\left\|[\mathcal{L}^{p,h}]^{-1}\mathcal{L}^{o,h}\mathbf{S}^\nu\right\| \le \left\|[\mathcal{L}^{p,h}]^{-1}\mathcal{L}^{o,h}\right\|\|\mathbf{S}^\nu\| \tag{37}$$

$$\begin{aligned}
\left\|\{[(\mathbf{M}^h)^t]^{-1}[\mathcal{L}^h]^{-1} - [\mathcal{L}^{p,h}]^{-1}\}\mathcal{L}^h(\mathbf{M}^h)^t\mathbf{S}^\nu\right\| &= \left\|[(\mathbf{M}^h)^t]^{-1}[\mathcal{L}^{p,h}]^{-1}\mathcal{L}^{o,h}(\mathbf{M}^h)^t\mathbf{S}^\nu\right\| \\
&\le \left\|[\mathcal{L}^{p,h}]^{-1}\mathcal{L}^{o,h}\right\|\left\|[(\mathbf{M}^h)^t]^{-1}\right\|\left\|(\mathbf{M}^h)^t\right\|\|\mathbf{S}^\nu\|.
\end{aligned} \tag{38}$$

Interestingly, the convergence rate of the outer iteration (13) arises in these two-grid bounds.

Remark 3. From the above analysis, the performance of the three methods will depend on different factors. For method (29), one can see from (36) and (38) that the condition number of $\mathbf{M}^h$ will affect this method's convergence. Furthermore, as mentioned earlier, the convergence of method (19) will depend on the similarity between the diagonal component operators $\mathcal{L}_{ii}^p$ through the interpolation operators $\mathbf{P}_{ii}$. In particular, for diagonal component operators that have different diffusion anisotropies, this method can even diverge. This problem can arise specifically in semicoarsening schemes: The different anisotropies will lead to different coarse grids, which would then lead to an additional interpolation error in $\mathbf{P}_{ii}\mathbf{P}_{jj}^t$, $i \ne j$, in (33) because an additional interpolation must be performed to define $\mathbf{P}_{ii}\mathbf{P}_{jj}^t$ on the appropriate grids. We will illustrate these issues in the numerical experiments.

6 | NUMERICAL EXPERIMENTS

We examine the performance of the different methods applied to several systems of PDEs. These systems include a two-dimensional linear thin-shell problem, a strongly coupled system of diffusion equations, which will be used to explore the different issues that can affect the convergence of method (19), and a system of parabolic equations with time-dependent diffusion coefficients that is time-marched with a high-order IRK scheme. We denote method (29) by ADN1, method (17) by ADN2, and method (19) by ADN3. We compare these methods as stand-alone solvers and as preconditioners for a restarted generalized minimal residual method (GMRES(10)), and to these methods using the full operator rather than its principal part. These latter methods will be denoted FULL1, FULL2, and FULL3. Unless otherwise specified, the approximate inversion of all diagonal operators is achieved with one $V(1,1)$ PFMG iteration, which is an operator-dependent scalar multigrid solver in the HYPRE package. Finally, although the experiments involve structured grid discretizations for implementation simplicity, these methods can also be applied to unstructured grid discretizations. Indeed, this is one of the values of these methods.

Experiment 1: linear thin shell

A simple thin shell model for shells with small curvature is described by the equations

$$\begin{aligned}\Delta^2 s + \Gamma^2 K(z,w) &= 0\\ \Delta^2 w - \Gamma^2 K(z,s) &= p,\end{aligned}$$

where

$$K(z,w) = z_{xx}w_{yy} - 2z_{xy}w_{xy} + z_{yy}w_{xx},$$

$s(x,y)$ is the stress, $w(x,y)$ is the displacement of the shell mean surface under a load, p, $z(x,y)$ is the functional description of the shell's geometry, and Γ^2 is a parameter proportional to the inverse of the thickness of the shell (see the work of Trottenberg et al.[11], Section 8.5). Introducing the variables $v = \Delta s$ and $u = \Delta w$, this fourth-order system is converted into the second-order system

$$\begin{aligned}\Delta v + \Gamma^2 K(z,w) &= 0\\ \Delta s - v &= 0\\ \Delta u - \Gamma^2 K(z,s) &= p\\ \Delta w - u &= 0,\end{aligned} \tag{39}$$

which is the system to be solved. We take homogeneous Dirichlet boundary conditions on the variables v, s, u, w and apply a finite-difference discretization with $h = \frac{1}{100}$ in the x and y directions on a unit square. We also take $p = 1$, a zero initial guess for all the variables, and thin shells with $\Gamma^2 = 80$ and geometries $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = -1\}$ and $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = 1\}$. The stopping criterion for the solvers is that the residual be less than 10^{-8}. This corresponds to a ten-order drop in the magnitude of the initial residual.

Tables 1 and 2 tabulate the results for shells $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = -1\}$ and $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = 1\}$, respectively. For shell $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = -1\}$, we see that ADN1 and ADN2 perform moderately as stand-alone solvers and as preconditioners. However, ADN3 performs badly. This is due to the poor approximation of $[\mathbf{M}^{l+1}]^{-1}$ to $[\mathbf{M}^{l}]^{-1}$, as tabulated in Table 3. For shell $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = 1\}$, all the methods perform poorly. This is due to the poor conditioning of $\mathbf{M}^h$, a (4×4) matrix with condition number 12804. The improved performance using the full operator in the preconditioner is an artifact of this poor conditioning (this improvement is not as dramatic when Γ^2 is reduced, which produces a better conditioned $\mathbf{M}^h$).

Experiment 2: factors affecting the performance of ADN3

As analyzed in Section 5, the convergence of ADN3 depends on both the accuracy of $[\mathbf{M}^{l+1}]^{-1}$ to $[\mathbf{M}^{l}]^{-1}$ and the similarity of the principal parts of the diagonal operators. In this set of experiments, we examine these conditions to see how they affect the convergence of ADN3. We consider a spatial three-dimensional problem with system operator

$$\begin{bmatrix} (-\nabla\cdot D_{11}\nabla + \mathbf{b}_{11}\cdot\nabla) & (-\nabla\cdot D_{12}\nabla + \mathbf{b}_{12}\cdot\nabla) & 0 \\ (-\nabla\cdot D_{21}\nabla + \mathbf{b}_{21}\cdot\nabla) & (-\nabla\cdot D_{22}\nabla + \mathbf{b}_{22}\cdot\nabla) & (-\nabla\cdot D_{23}\nabla + \mathbf{b}_{23}\cdot\nabla) \\ 0 & (-\nabla\cdot D_{32}\nabla + \mathbf{b}_{32}\cdot\nabla) & (-\nabla\cdot D_{33}\nabla + \mathbf{b}_{33}\cdot\nabla) \end{bmatrix} \tag{40}$$

TABLE 1 Convergence and timings for shell $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = -1\}$

Method	# Iterations	Setup time	Solve time
ADN1	92	0.01	2.0
ADN2	95	0.02	1.8
ADN3	Diverges	0.03	-
FULL1	133	0.01	2.9
FULL2	134	0.01	2.6
FULL3	Diverges	0.03	-
GMRES(10)-ADN1	34	0.02	1.1
GMRES(10)-ADN2	45	0.02	1.4
GMRES(10)-ADN3	No convergence after 2,000 its ($6.73 * 10^{-5}$)	0.03	69.6
GMRES(10)-FULL1	39	0.02	1.3
GMRES(10)-FULL2	54	0.02	1.7
GMRES(10)-FULL3	No convergence after 2,000 its ($6.02 * 10^{-5}$)	0.03	69.5

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

TABLE 2 Convergence and timings for shell $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = 1\}$. Issues arise from the poor conditioning of $\mathbf{M}^h$, condition number 12804

Method	# Iterations	Setup time	Solve time
ADN1	Diverges	0.02	-
ADN2	Diverges	0.02	-
ADN3	Diverges	0.02	-
FULL1	Diverges	0.02	-
FULL2	Diverges	0.02	-
FULL3	Diverges	0.02	-
GMRES(10)-ADN1	187	0.02	8.3
GMRES(10)-ADN2	329	0.03	13.4
GMRES(10)-ADN3	405	0.03	14.4
GMRES(10)-FULL1	94	0.02	4.1
GMRES(10)-FULL2	161	0.03	6.5
GMRES(10)-FULL3	399	0.03	14.2

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

TABLE 3 Problematic $\|[\mathbf{M}^l]^{-1} - [\mathbf{M}^{l+1}]^{-1}\|_F$ for shell $\{z_{xx} = 1, z_{xy} = 0, z_{yy} = -1\}$

Level l	$\|[\mathbf{M}^l]^{-1} - [\mathbf{M}^{l+1}]^{-1}\|_F$
0	$5.54 * 10^1$
1	$5.37 * 10^1$
2	$6.03 * 10^1$
3	$5.68 * 10^1$

and take $\mathbf{b}_{ij} = \frac{0.05}{h}\text{diag}(D_{ij})$, leading to relatively large lower order terms. For the first test, we take the diffusion tensors

$$D_{11} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad D_{22} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.2 & 0 \\ 0 & 0 & 1.1 \end{bmatrix} \quad D_{33} = \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 2.3 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} 1.6 & 0 & 0 \\ 0 & 3.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad D_{12} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 5.0 & 0 \\ 0 & 0 & 3.0 \end{bmatrix} \quad D_{32} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 2.4 & 0 \\ 0 & 0 & 3.0 \end{bmatrix}$$

$$D_{23} = \begin{bmatrix} 2.005 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 2.07 \end{bmatrix}.$$

Thus, the principal parts of the diagonal operators of system (40) are similar. Again, a finite difference discretization is applied with equally spaced meshing with $h = \frac{1}{50}$ for a unit cube, homogenous Dirichlet boundary conditions are applied to all of the variables, and the same stopping criterion is used. Table 4 shows the accuracy of $[\mathbf{M}^{l+1}]^{-1}$ to $[\mathbf{M}^l]^{-1}$. Furthermore, because the diagonal operators are similar, PFMG will construct interpolation operators (and restriction operators because the restriction operator is the transpose of the interpolation operator) $\mathbf{P}_{ii}$ that are similar. Table 5 displays the convergence of ADN3 as a stand-alone solver and as a preconditioner for this problem. We see good convergence for this

TABLE 4 ADN3: $\|[\mathbf{M}^l]^{-1} - [\mathbf{M}^{l+1}]^{-1}\|_F$ for system (40) with similar diagonal operators

Level l	$\|[\mathbf{M}^l]^{-1} - [\mathbf{M}^{l+1}]^{-1}\|_F$	
	ADN3	FULL3
0	$2.04 * 10^{-1}$	$2.08 * 10^{-1}$
1	$8.13 * 10^{-2}$	$8.48 * 10^{-2}$
2	$1.83 * 10^{-1}$	$1.86 * 10^{-1}$
3	$2.14 * 10^{-1}$	$2.18 * 10^{-1}$

Note. ADN = Agmon–Douglis–Nirenberg.

TABLE 5 Convergence and timings for system (40) with similar diagonal operators

Method	# Iterations	Setup time	Solve time
ADN3	35	0.37	7.7
GMRES(10)-ADN3	29	0.38	10.1
FULL3	47	0.37	10.4
GMRES(10)-FULL3	33	0.40	11.6

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

TABLE 6 Convergence and timings for system (40) with dissimilar diagonal operators

Method	# iterations	Setup time	Solve time
ADN3	Diverges	0.36	-
GMRES(10)-ADN3	582	0.37	261.4
FULL3	Diverges	0.36	-
GMRES(10)-FULL3	no convergence after 1,000 its ($4.22 * 10^{-6}$)	0.39	460.5

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

test. We also see that FULL3 does not perform well because, including the lower order terms, the interpolation operators $\mathbf{P}_{ii}$ generated by PFMG will be slightly different from each other.

The next test considers diagonal operators that are different. Consider system (40) with diffusion tensors

$$D_{11} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad D_{22} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 3.0 & 0 \\ 0 & 0 & 2.0 \end{bmatrix} \quad D_{33} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 3.0 \end{bmatrix}$$
$$D_{21} = \begin{bmatrix} 1.6 & 0 & 0 \\ 0 & 3.5 & 0 \\ 0 & 0 & 2.0 \end{bmatrix} \quad D_{12} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 5.0 & 0 \\ 0 & 0 & 3.0 \end{bmatrix} \quad D_{32} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 2.4 & 0 \\ 0 & 0 & 4.0 \end{bmatrix}$$
$$D_{23} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 5.07 \end{bmatrix}.$$

This minor difference in the diagonal operators will generate interpolation operators that are slightly different. Table 6 displays the convergence performance of ADN3 as a stand-alone solver and as a preconditioner. Because the accuracy of $[\mathbf{M}^{l+1}]^{-1}$ is not poor, as illustrated in Table 7, we can contribute the poor performance to difference in the interpolation operators. This is further corroborated by the results for FULL3. The addition of the lower order terms will lead to stronger differences in the interpolation operators.

Experiment 3: higher order IRK time discretizations

We now consider higher order IRK method[4] applied to time-dependent PDEs. Given an ordinary differential equation

$$\frac{du}{dt} = f(t, u), \tag{41}$$

an s-stage IRK method finds the approximation to u at time t_{i+1} using stage values $\{\hat{u}_j\}_{j=1,\dots,s}$. Let $\Delta t = t_{i+1} - t_i$. The IRK approximation to u_{i+1} is determined by the formulas

$$u_{i+1} = u_i + \Delta t \sum_{j=1}^{s} b_j f(t_i + c_j \Delta t, \hat{u}_j) \tag{42}$$

TABLE 7 ADN3: $\|[\mathbf{M}^{\boldsymbol{l}}]^{-\mathbf{1}} - [\mathbf{M}^{\boldsymbol{l+1}}]^{-\mathbf{1}}\|_{\boldsymbol{F}}$ for system (40) with dissimilar diagonal operators

Level l	$\|[\mathbf{M}^{l}]^{-1} - [\mathbf{M}^{l+1}]^{-1}\|_F$	
	ADN3	**FULL3**
0	$1.55 * 10^{-1}$	$1.58 * 10^{-1}$
1	$1.58 * 10^{-1}$	$1.62 * 10^{-1}$
2	$1.00 * 10^{-1}$	$1.05 * 10^{-1}$
3	$1.62 * 10^{-1}$	$1.66 * 10^{-1}$

Note. ADN = Agmon–Douglis–Nirenberg.

$$\hat{u}_i = u_i + \Delta t \sum_{j=1}^{s} a_{ij} f(t_i + c_j \Delta t, \hat{u}_j), \quad i = 1, \ldots, s, \tag{43}$$

where $\{c_j\}$, $\{b_j\}$, and $\{a_{ii}\}$ are chosen constants that guarantee high accuracy and good stability. From (42), u_{i+1} can be constructed once the $\hat{u}_j$'s are known. However, to determine the $\hat{u}_j$'s, one must solve Equation (43), which is an $(s \times s)$ system for the unknowns $\hat{u}_j$ because of the variable-coupling summation term. Specifically, for parabolic PDEs, (43) will require solving systems of PDEs. For example, for the linear parabolic equation

$$\frac{\partial u}{\partial t} - \nabla \cdot D(x,t)\nabla u + \mathbf{b}(t) \cdot \nabla u = f,$$

with time-dependent coefficients, an s-stage IRK time-discretization generates a strongly coupled system of PDEs of the form

$$\hat{u}_i - \Delta t \sum_{j=1}^{s} a_{ij} \left[\nabla \cdot D(x, t_i + c_j \Delta t)\nabla \hat{u}_j + \mathbf{b}(t_i + c_j \Delta t) \cdot \nabla \hat{u}_j\right] = g_i, \quad i = 1, \ldots, s. \tag{44}$$

The time-dependent coefficients generate a system that cannot be written as a standard Kronecker product operator. Hence, a Kronecker product preconditioner[15] cannot be directly applied.

We consider the parabolic systems generated by the operator

$$\begin{pmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial t} \\ \frac{\partial}{\partial t} \end{pmatrix} + \begin{bmatrix} (-\nabla \cdot D_{11}\nabla + \mathbf{b}_{11} \cdot \nabla) & (-\nabla \cdot D_{12}\nabla + \mathbf{b}_{12} \cdot \nabla) & 0 \\ (-\nabla \cdot D_{21}\nabla + \mathbf{b}_{21} \cdot \nabla) & (-\nabla \cdot D_{22}\nabla + \mathbf{b}_{22} \cdot \nabla) & (-\nabla \cdot D_{23}\nabla + \mathbf{b}_{23} \cdot \nabla) \\ 0 & (-\nabla \cdot D_{32}\nabla + \mathbf{b}_{32} \cdot \nabla) & (-\nabla \cdot D_{33}\nabla + \mathbf{b}_{33} \cdot \nabla) \end{bmatrix}. \tag{45}$$

Again, we take $\mathbf{b}_{ij} = \frac{0.05}{h} diag(D_{ij})$, discretize the spatial component of the operator using finite differencing with spatial meshsize $h = \frac{1}{50}$ for a unit cube, and apply homogeneous Dirichlet boundary conditions on all the variables. To balance the discretization error, assuming that the spatial and time discretizations are of order p and q, respectively, we take

$$\Delta t = h^{\frac{p}{q}}. \tag{46}$$

In particular, using the RadauIIA IRK method with

$$[a] = \begin{bmatrix} \frac{5}{12} & -\frac{1}{12} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix},$$

which has order $q = 3$ and produces a (6×6) system of PDEs, we take $\Delta t = h^{\frac{2}{3}}$. Finally, the diffusion coefficients are taken to be

$$D_{11} = e^{\alpha_{11}t}\begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad D_{22} = e^{\alpha_{22}t}\begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.1 \end{bmatrix} \quad D_{33} = e^{\alpha_{33}t}\begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.1 \end{bmatrix}$$

$$D_{21} = e^{\alpha_{21}t}\begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 3.0 \end{bmatrix} \quad D_{12} = e^{\alpha_{12}t}\begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad D_{32} = e^{\alpha_{32}t}\begin{bmatrix} 3.0 & 0 & 0 \\ 0 & 3.5 & 0 \\ 0 & 0 & 4.0 \end{bmatrix}$$

$$D_{23} = e^{\alpha_{23}t}\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}.$$

For the first test, we choose $\alpha_{11} = -0.1, \alpha_{12} = -0.1, \alpha_{21} = -0.15, \alpha_{22} = -0.1, \alpha_{23} = -0.05, \alpha_{32} = -0.1, \alpha_{33} = -0.1$, and consider solving the linear system arising at time $t = 1$ with a zero initial approximation for all the variables (we are interested in the performance of the solver). Table 8 tabulates the results. We see that with the exception of ADN3, the methods perform very well. Furthermore, we see that using the full operator gives generally the same results with the exception of the GMRES(10) preconditioned with ADN3. For ADN3 with the full operator, again the addition of the lower order terms lead to diagonal interpolation operators that are more different than when only the principal part of the operator is used.

The next test takes an oscillating set of α's: $\alpha_{11} = -0.1, \alpha_{12} = 0.2, \alpha_{21} = -0.15, \alpha_{22} = 0.5, \alpha_{23} = -0.05, \alpha_{32} = 0.2, \alpha_{33} = -0.1$. Table 9 gives the results, which are very similar to the first test.

TABLE 8 Convergence and timings for implicit Runge–Kutta (IRK) applied to system (45) with $\alpha_{11} = -0.1, \alpha_{12} = -0.1, \alpha_{21} = -0.15, \alpha_{22} = -0.1, \alpha_{23} = -0.05, \alpha_{32} = -0.1, \alpha_{33} = -0.1$

Method	# iterations	Setup time	Solve time
ADN1	39	0.47	55.2
ADN2	55	0.55	65.5
ADN3	Diverges	0.87	-
FULL1	34	0.47	48.3
FULL2	55	0.57	66.2
FULL3	Diverges	0.82	-
GMRES(10)-ADN1	19	0.56	34.2
GMRES(10)-ADN2	37	0.64	37.9
GMRES(10)-ADN3	145	0.85	246.6
GMRES(10)-FULL1	16	0.55	28.4
GMRES(10)-FULL2	35	0.64	36.1
GMRES(10)-FULL3	193	0.96	347.2

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

TABLE 9 Convergence and timings for implicit Runge–Kutta (IRK) applied to system (45) with $\alpha_{11} = -0.1, \alpha_{12} = 0.2, \alpha_{21} = -0.15, \alpha_{22} = 0.5, \alpha_{23} = -0.05, \alpha_{32} = 0.2, \alpha_{33} = -0.1$

Method	# Iterations	Setup time	Solve time
ADN1	39	0.48	55.3
ADN2	55	0.51	65.2
ADN3	Diverges	0.83	-
FULL1	34	0.59	48.2
FULL2	55	0.65	65.9
FULL3	Diverges	0.89	-
GMRES(10)-ADN1	19	0.62	33.9
GMRES(10)-ADN2	37	0.56	37.9
GMRES(10)-ADN3	142	0.87	242.5
GMRES(10)-FULL1	16	0.63	28.4
GMRES(10)-FULL2	35	0.55	36.1
GMRES(10)-FULL3	229	0.96	424.0

Note. ADN = Agmon–Douglis–Nirenberg; GMRES = generalized minimal residual method.

7 | CONCLUSION

In this paper, we theoretically examined the multigrid strategy for solving systems of elliptic PDEs introduced in the work of Lee.[3] This strategy is applied only to the principal part of the differential system, and the strategy has the advantage that it can be implemented using efficient kernels in existing high-performance software packages without much software intrusion. In particular, because the multigrid components are required only for the diagonal components of the principal part of the operator, construction of these components can be obtained using existing, well-tested multigrid software packages. Two-grid analyses were conducted to expose factors that can affect the convergence of the methods. Numerical experiments were conducted to validate some of the theory and illustrate the effectiveness of this strategy.

ACKNOWLEDGEMENT

The author would like to thank two anonymous referees for their comments and suggestions that improved this paper.

CONFLICT OF INTEREST

The author declares no conflict of interest.

ORCID

B. Lee https://orcid.org/0000-0003-1948-7384

REFERENCES

1. Renardy M, Rogers RC. An introduction to partial differential equations. New York, NY: Springer-Verlag; 1993.
2. Lee B. Guidance for choosing multigrid preconditioners for systems of elliptic partial differential equations. SIAM J Sci Comput. 2009;31(4):2803–2831.
3. Lee B. Parallel preconditioners and multigrid solvers for stochastic polynomial chaos discretizations of the diffusion equation at the large scale. Numer Linear Algebra Appl. 2016;23(1):5–36.
4. Butcher JC. The numerical analysis of ordinary differential equations: Runge–Kutta and general linear methods. Hoboken, NJ: John Wiley & Sons; 1987.
5. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation (SA) multigrid. SIAM Review. 2005;47:317–346.
6. Briggs WL, Henson VE, McCormick SF. A multigrid tutorial. 2nd ed. Philadelphia, PA: SIAM; 2000.
7. Dendy JE Jr. Black box multigrid for systems. Appl Math Comput. 1986;19:57–74.
8. Dendy JE Jr. Semicoarsening multigrid for systems. Electron Trans Numer Anal. 1997;6:97–105.
9. Clees T, Stüben K. Algebraic multigrid for industrial semiconductor device simulation. In: Challenges in scientific computing - CISC 2002: Proceedings of the Conference Challenges in Scientific Computing Berlin, October 2–5, 2002. Berlin, Germany: Springer; 2002.
10. Brandt A, Livne OE. Multigrid techniques: 1984 guide, with applications to fluid dynamics. Philadelphia, PA: SIAM; 1984.
11. Trottenberg U, Oosterlee CW, Schüller A. Multigrid. San Diego, CA: Academic Press; 2001.
12. Yavneh I. A method for devising efficient multigrid smoothers for complicated PDE systems. SIAM J Sci Comput. 1993;14:1437–1463.
13. Hypre Software. Available from: https://computation.llnl.gov/casc/hypre/software.html
14. Hackbusch W. Multi-grid methods and applications. Berlin, Germany: Springer-Verlag; 1985.
15. Ullmann E. A Kronecker product preconditioner for stochastic Galerkin finite element discretizations. SIAM J Sci Comput. 2010;32(2):923–946.

How to cite this article: Lee B. Multigrid for second-order ADN elliptic systems. *Numer Linear Algebra Appl*. 2019;26:e2256. https://doi.org/10.1002/nla.2256